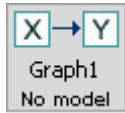


The graph box in the main workspace looks like this:



### **Possible Parent Boxes of the Graph Box:**

- Another graph box
- A graph manipulation box
- A search box
- A parametric model box
- An instantiated model box
- An updater box

A graph box that is the child of any other box will contain a direct copy of the graph its parent box contains.

### **Possible Child Boxes of the Graph Box:**

- Another graph box
- A graph manipulation box
- A search box
- A parametric model box
- A comparison box
- A data box
- A knowledge box

### **Using the Graph Box:**

When you open the graph box for the first time, you will be presented with several options for which kind of graph you would like to create. The options are: a directed acyclic graph, a structural equation model graph, a general graph, or a time lag graph.

### **Directed Acyclic Graphs**

A directed acyclic graph, or DAG, is a directed graph containing no cycles. DAGs can be used to create any causal model, and are the only kind of graph accepted by a Bayes parametric model.

If you choose to create a DAG, the following window will open:

**Graph Structure Editor**

Make new graph:

- An empty graph (to be constructed manually).
- A random DAG.

Parameters for Random DAG:

Number of measured nodes: 6

Number of latent nodes: 0

Maximum number of edges: 6

Maximum indegree: 3

Maximum outdegree: 3

Maximum degree: 6

Connected: No

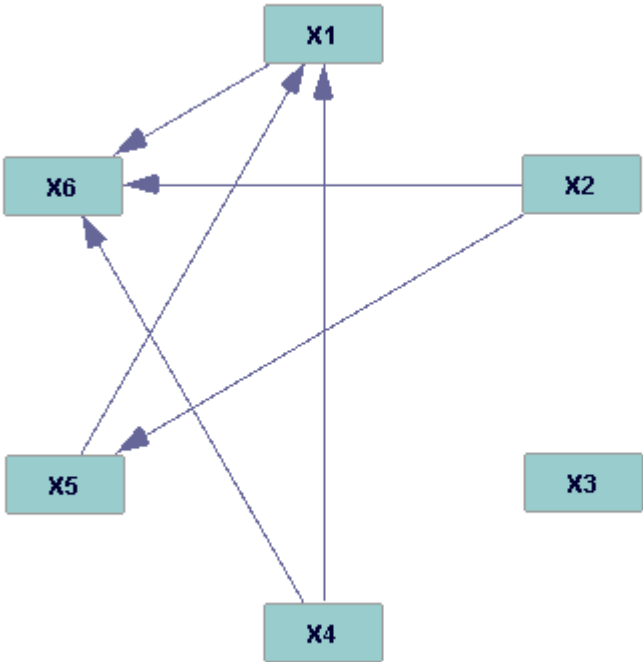
Draw uniformly from all such DAGs

Guarantee maximum number of edges

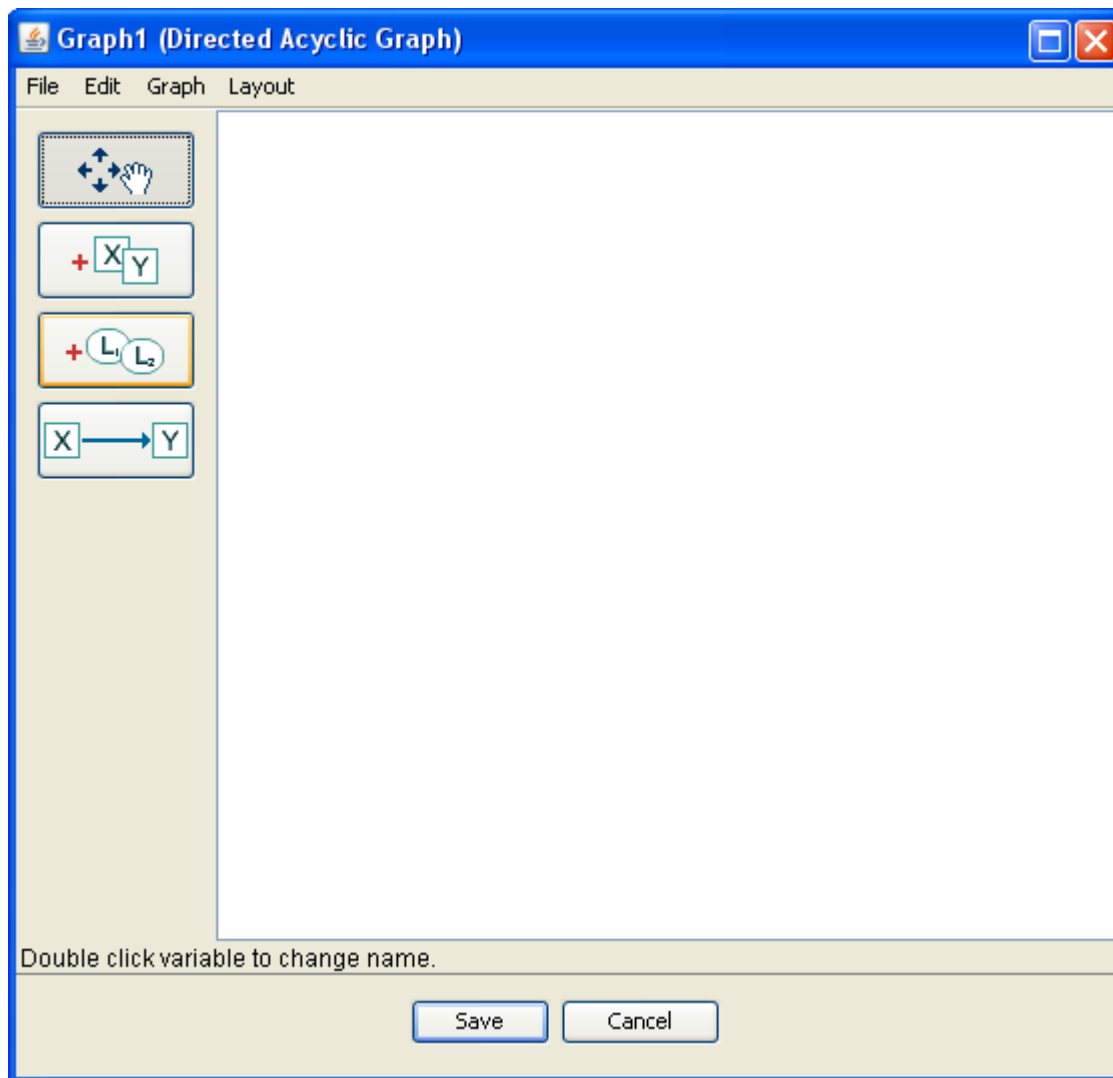
OK Cancel

If you choose the random DAG option, Tetrad will create a graph for you according to your specifications. You can specify the number of measured and latent nodes in the graph, the maximum number of total edges in the graph, the maximum indegree, outdegree, and degree of each node, and whether you would like the graph to be connected. Additionally, you can specify whether you would like Tetrad to guarantee that your graph contains the maximum number of edges possible. (Depending on your specifications for maximum degree, this may be less than the maximum number of edges you specified.) After Tetrad generates your graph, you can manually edit it.

Here is an example random DAG, generated using the default settings:



If you choose to manually create a DAG, you will see the following window:



You can create measured variables by clicking on the +XY button, and latent variables by clicking on the +L<sub>1</sub>L<sub>2</sub> button. You can draw edges between variables by clicking the arrow button. Just as in the workspace, you can move, highlight, and delete items by clicking on the hand button. In DAG mode, the graph box will not allow you to create cycles in your graph.

### Structural Equation Model Graphs

A structural equation model graph, or SEM graph, specifies the graphical structure of a SEM model. In a SEM graph, the causal structure is represented by directed arrows, and correlated errors are represented by bidirected arrows. Cycles are permitted in a SEM graph.

If you choose to create a SEM graph, the following window will appear:

**Graph Structure Editor** ✕

Make new graph:

An empty graph (to be constructed manually).

A random DAG.

Parameters for Random DAG:

Number of measured nodes:

Number of latent nodes:

Maximum number of edges:

Maximum indegree:

Maximum outdegree:

Maximum degree:

Connected:  ▼

Draw uniformly from all such DAGs

Guarantee maximum number of edges

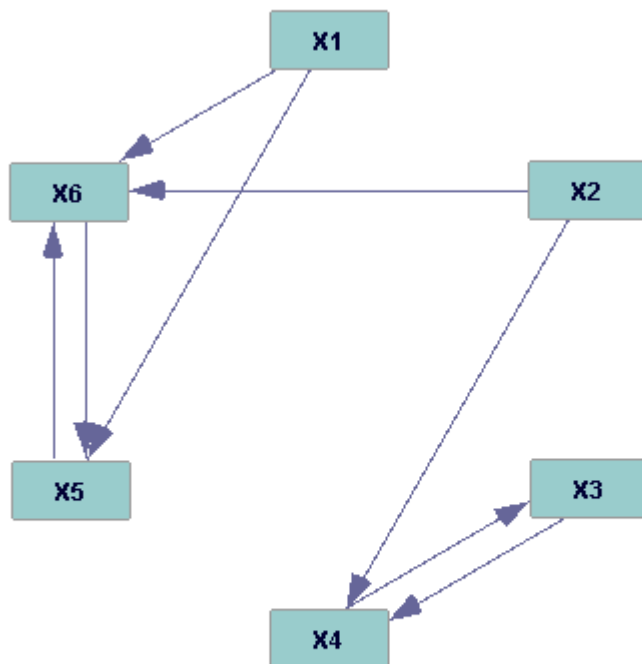
Add cycles? (Adds edges)  ▼

Minimum number of cycles

Minimum cycle length

The specifications for a random SEM graph are the same as those for a random DAG, with the exception that you may instruct Tetrad to put cycles in your graph. If you do so, there may be more edges in the graph than the maximum number you specified. Additionally, you can specify the minimum number of cycles necessary, and the minimum number of nodes in each cycle.

Here is an example of a random SEM graph, with cycles, using the default settings:



Note that this graph contains no bidirected edges. Tetrad will not randomly generate a graph with bidirected edges, but you can manually add them to a randomly generated graph.

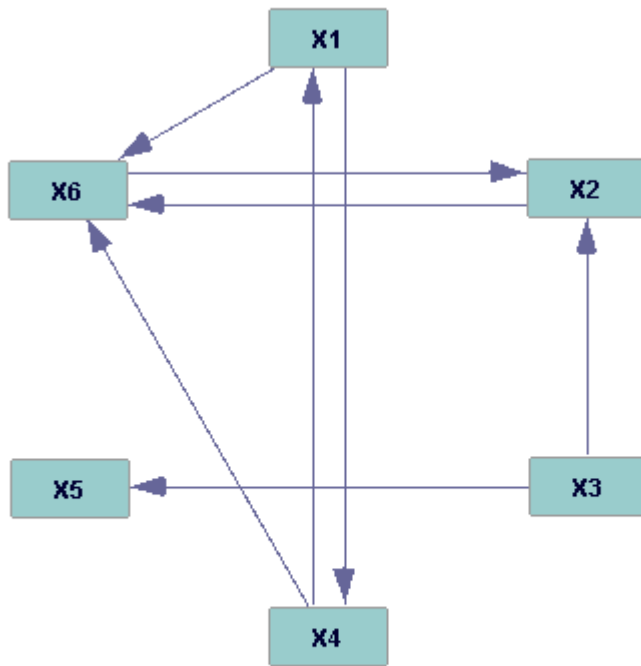
If you decide to manually create a SEM graph, a window will open up which functions exactly like that which opens for a manually created DAG, except that there is a button which allows you to create bidirected edges, and the graph box will allow you to create cycles.

### General Graphs

A general graph is the least restricted type of graph. Like a SEM graph, a general graph represents causal relationships with directed arrows and correlated errors with bidirected arrows. A general graph has one other type of edge, however. In a general graph, an edge with a small circle at its end signifies uncertainty as to whether or not that end should contain an arrow or not. General graphs with uncertain edges will be used later to specify latent models; generally, if one is creating a model, one should not create a graph with uncertain edges. The significance of uncertain edges is explained in more detail in the search box section, in connection with the FCI search algorithm. Cycles are permitted in general graphs.

Like DAGs and SEM graphs, if you choose to create a general graph, you may either create one yourself or instruct Tetrad to generate a random graph according to your specifications. The specifications for a general graph are the same as those for a SEM graph.

Here is an example of a randomly created general graph with cycles, using the default specifications:



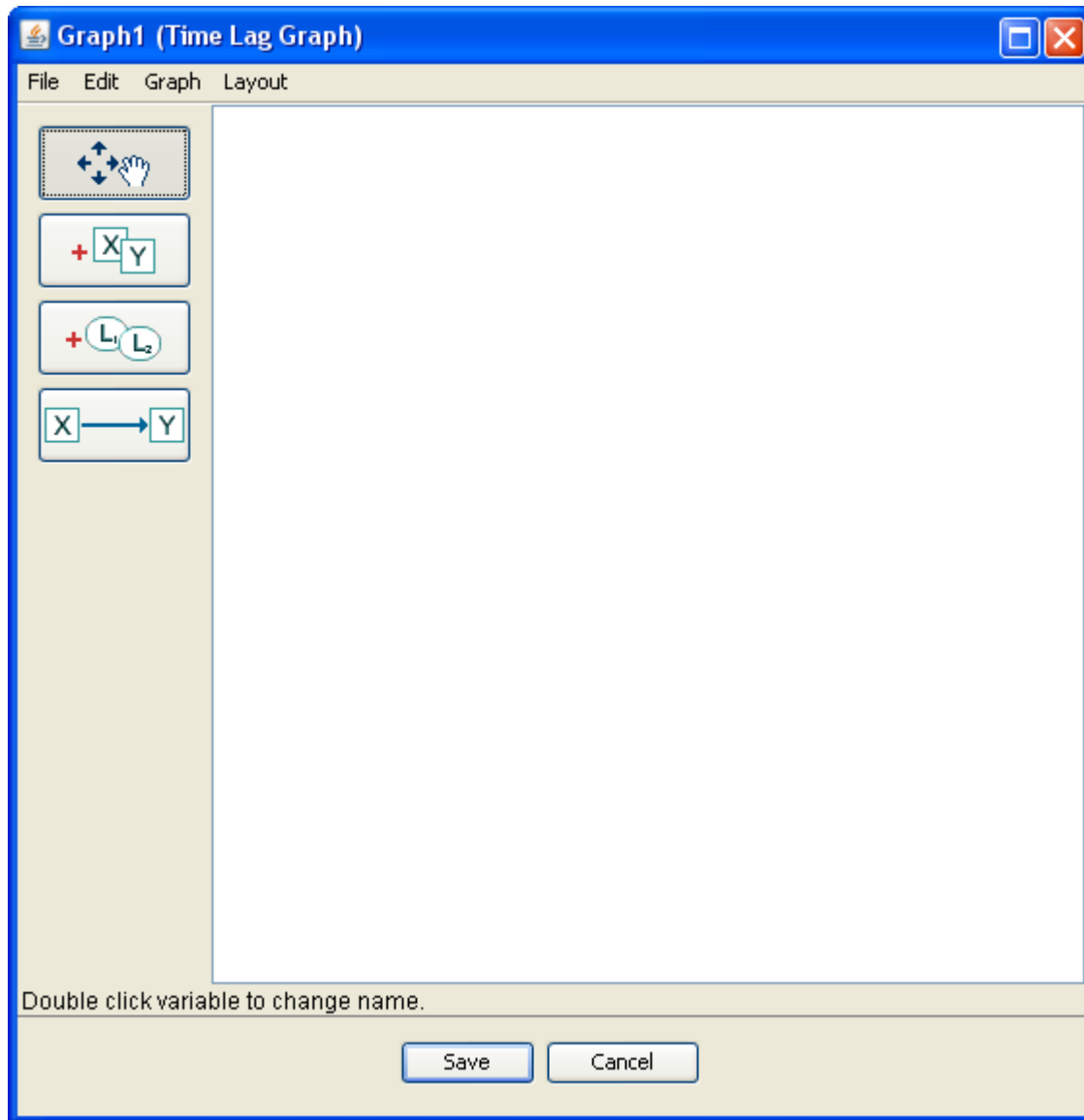
Note that this is a valid SEM graph. Tetrad will not randomly generate a general graph with uncertain or bidirected edges. You can manually add such edges to the graph.

If you decide to manually create a general graph, a window will open up which functions exactly like that which opens for a manually created DAG, except that there are buttons which allow you to create bidirected and uncertain edges, and the graph box will allow you to create cycles.

### Time Lag Graphs

A time lag graph represents the graphical structure of a time series. Time lag graphs have only one kind of permissible edge (directed) and they do not permit cycles.

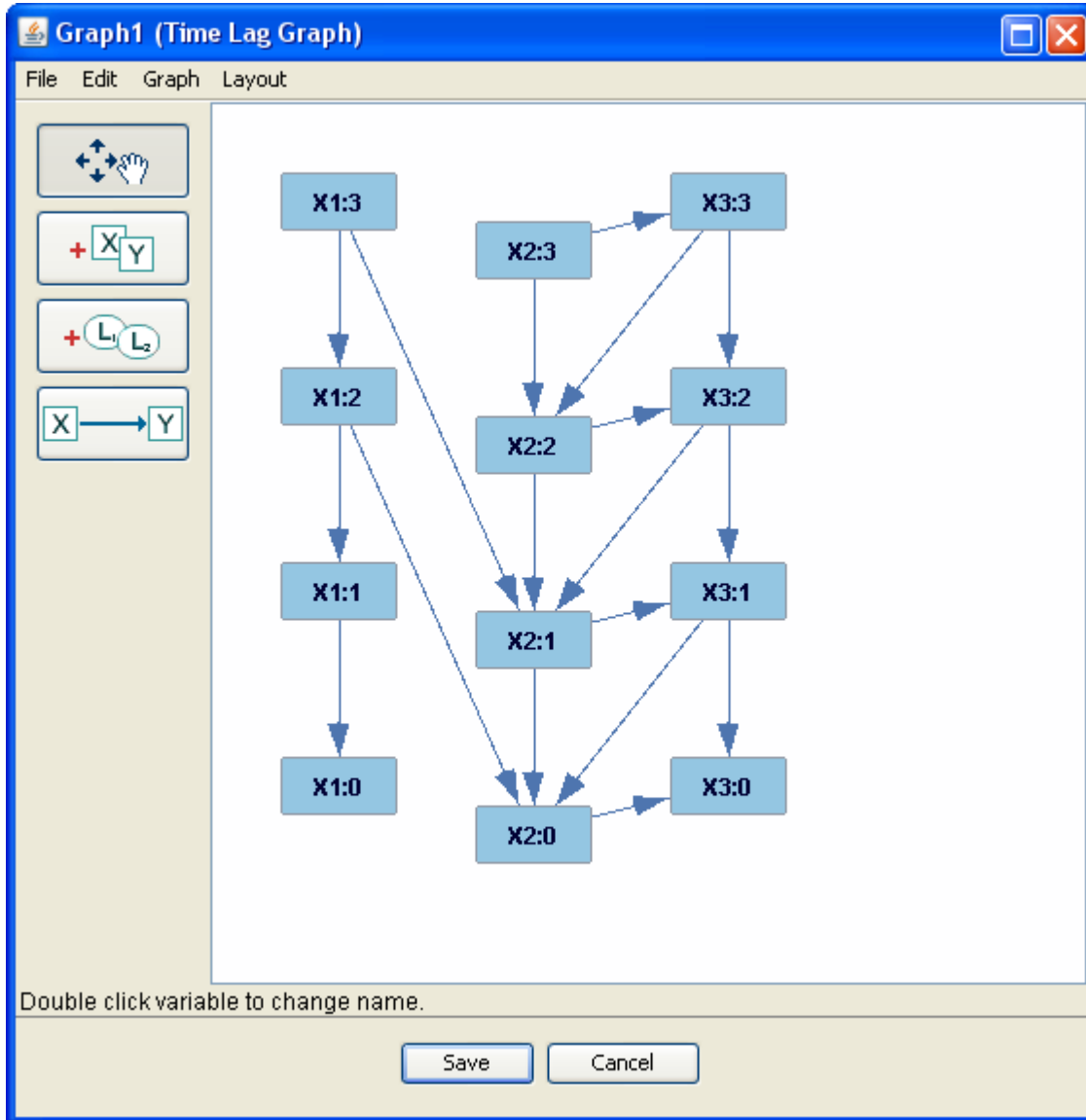
Currently, Tetrad will not randomly generate a time lag graph; you must create one manually. If you choose to create a time lag graph, the following window will open:



Just as in a DAG graph, you can create variables by clicking on the +XY box or +L<sub>1</sub>L<sub>2</sub> box, and edges by clicking on the variable box. By default, the graph box assumes that you only wish to work with one lag. You can change this by clicking Edit: Configurations... and increasing the maximum number of lags. Because the graph represents a time series, the graph box will not allow you to create edges from a variable in one stage to a variable in a previous stage. The graph box will automatically reproduce edges created between and within lags where they are implied. To change the flow of the graph from top to bottom (which is the default setting), click Layout and choose your preferred flow. If you would like some of the variables in every lag to be offset from the others by a certain amount (to, for example, increase the visibility of the edges) manually create the offset in lag 0 using the hand tool. Then, click on Layout: Copy lag 0. This will reproduce the offset in all lags in the graph.



Here is an example of a manually created time lag graph:



To create the edges between the X1s in all lags, one can simply create the edge between X1:1 and X1:0; the graph box automatically fills in the rest. Creating an edge from X2:0 to X3:0 causes the graph box to automatically create an edge from X2 to X3 in all lags, and creating an edge from X1:2 to X2:0 causes the graph box to automatically create an edge from every X1 to the X2 two lags later. Deleting any one of these edges would cause the graph box to automatically delete the others. The offset of the X2 variable in each lag is created using Copy lag 0.

### **Loading or Saving a Graph**

If you wish, you can save a graph as an XML document, text document, or R graph. You can also save just the image of a graph. These options are under the File tab in the graph box tool bar.

You can also load a graph into the graph box instead of manually creating one. The graph must be in an XML document, and formatted the same way that Tetrad formats graphs when it saves them to XML documents. If you are going to load a graph, it is therefore best to load one that you have saved from a previous Tetrad box.

Another way to use the same graph for multiple graph boxes is to copy it. First, use the hand tool to highlight the entire graph. Then, click Edit: Copy. Open up a graph box of the same type, and click Edit: Paste. A copy of the graph should appear in the window. This process does not work for time lag graphs.

Additionally, you can ensure that two graphs containing the same variables are laid out in the same way by copying the layout. First, use the hand tool to highlight the entirety of the graph whose layout you want to copy. Then, click Layout: Copy Layout. Open up the other graph box and click Layout: Paste Layout. The second graph should conform to the layout of the first. Both graphs must be of the same type for this procedure to work.

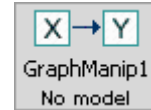
## **Graph Information**

The Graph tab in the graph box toolbar provides information about the graph you have created (as well as a few other functions).

Clicking Graph: Graph Properties will open a window that lists how many variables and edges the graph contains of each type, as well as the maximum degree and whether or not the graph is cyclic.

Graph: Paths contains information about directed paths, treks, and adjacencies. When you click this tab, a window will open, and you will be able to choose a “from” variable, a “to” variable, and what information you would like to see about their relationship. If you choose “directed paths” as the relationship, Tetrad will list every directed path from the first variable to the second. If you choose “select all” as the second variable, Tetrad will list every possible directed path from the “from” variable. If you choose “treks” as the relationship, Tetrad will list every trek between the first variable and the second. If you choose “select all” for the second variable, then Tetrad will list every trek containing the “from” variable. If you choose “adjacencies” as the relationship, Tetrad will list every node adjacent to the “from” variable, and whether that node is a parent, child, or of ambiguous relation to it. For large or heavily connected graphs, hesitate before choosing “select all” as the second variable, as this may give more information than is easily navigable.

Graph: Random Graph (Random DAG in a directed acyclic graph) allows you to have Tetrad generate a random graph according to your specifications (exactly as it does when you first created the model). This will destroy the current graph, if one exists. Choosing this option in a time lag graph will create a random graph which is *not* a time lag graph.



The graph manipulation box in the main workspace looks like this:

### **Possible Parent Boxes of the Graph Manipulation Box:**

- A graph box
- Another graph manipulation box
- A search box
- An instantiated model box

A graph manipulation box, unlike a graph box, *must* have a parent.

### **Possible Child Boxes of the Graph Manipulation Box:**

- A graph box
- Another graph manipulation box
- A search box
- A parametric model box
- A comparison box
- A data box

### **Definitions**

An unshielded collider occurs when two non-adjacent variables both contain an edge into the same variable (for example,  $X \rightarrow Y \leftarrow Z$ , when  $X$  and  $Z$  are non-adjacent).

A Markov equivalence class is a set of DAGs which are causally equivalent. The DAGs in a Markov equivalence class always have the same variables, adjacencies, and unshielded colliders. For example,  $X \rightarrow Y \rightarrow Z$ ,  $X \leftarrow Y \leftarrow Z$  and  $X \leftarrow Y \rightarrow Z$  make up one Markov equivalence class.

A pattern is a graph containing directed and/or undirected edges which represents a Markov equivalence class. In a pattern, no matter how the undirected edges are oriented, unless they create an unshielded collider, the resulting DAG will always be in the same Markov equivalence class. For example, the Markov equivalence class containing  $X \rightarrow Y \rightarrow Z \leftarrow A$  and  $X \leftarrow Y \rightarrow Z \leftarrow A$  is represented by the pattern  $X - Y \rightarrow Z \leftarrow A$ .

The Markov blanket of a variable is that variable's parents, its children, and the other parents of its children. All variables in a graph not in the Markov blanket of a variable  $X$  are independent of  $X$  conditional on the Markov blanket. The Markov blanket of  $X$  is the smallest set of variables (not including  $X$ ) with this property.

### **Using the Graph Manipulation Box:**

A graph manipulation box takes as its input a graph of some kind, either directly from a graph box, or from the model of a search or instantiated model box. You can then specify

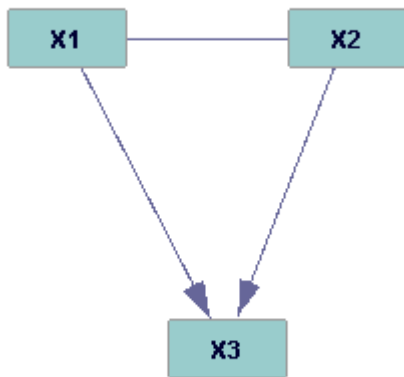
changes for Tetrads to make to that graph, and the graph manipulation box will output a graph which meets those specifications.

When you double click on the graph manipulation box for the first time, you will be presented with several options for the changes you would like to make to the graph. The options are: choose DAG in pattern, show DAGs in pattern, generate pattern from DAG, make bidirected edges undirected, and extract Markov blanket.

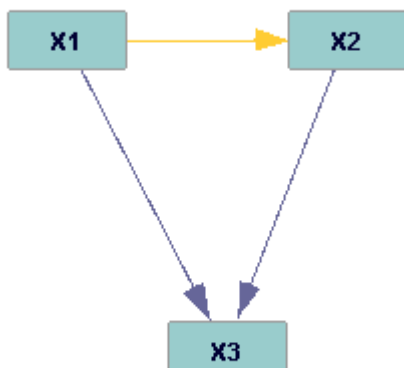
### Choose DAG in Pattern

If you choose this option, then for every undirected edge in your input graph, Tetrads will assign a direction to the edge such that the output graph is a directed acyclic graph (DAG). If the input graph is a pattern, then Tetrads will output a DAG in the Markov equivalence class represented by the pattern. However, the input graph need not be a pattern in order for “Choose DAG in Pattern” to run correctly; the requirements are that it contain only directed and undirected edges, and that there be some assignment of direction to the undirected edges which will make it a DAG.

For instance, suppose the graph manipulation box is run with a parent box containing the following graph:

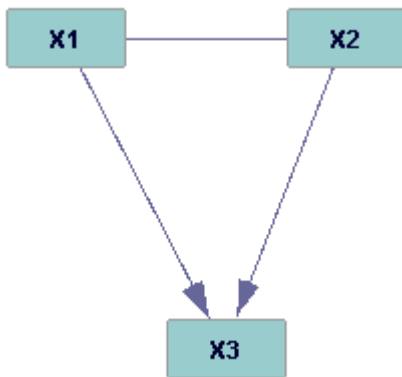


Tetrads will output the following DAG:

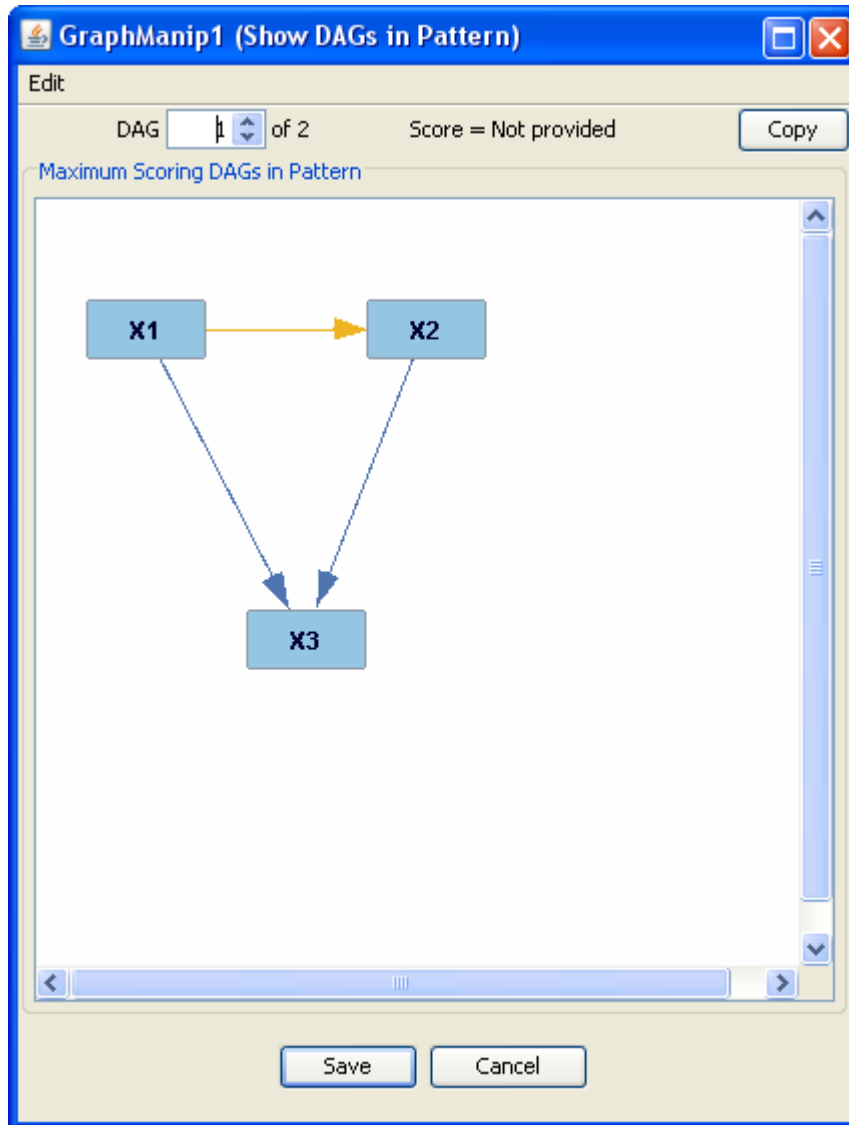


### Show DAGs in Pattern

If you choose this option, Tetrad will find all possible configurations of assignments of directions to undirected edges which result in DAGs. The input need not be a pattern in order for “Show DAGs in Pattern” to run correctly; it is sufficient that it contain only directed and undirected edges. If the input is a pattern, this option outputs the Markov equivalence class which it represents. For example, suppose the following graph is given as input:



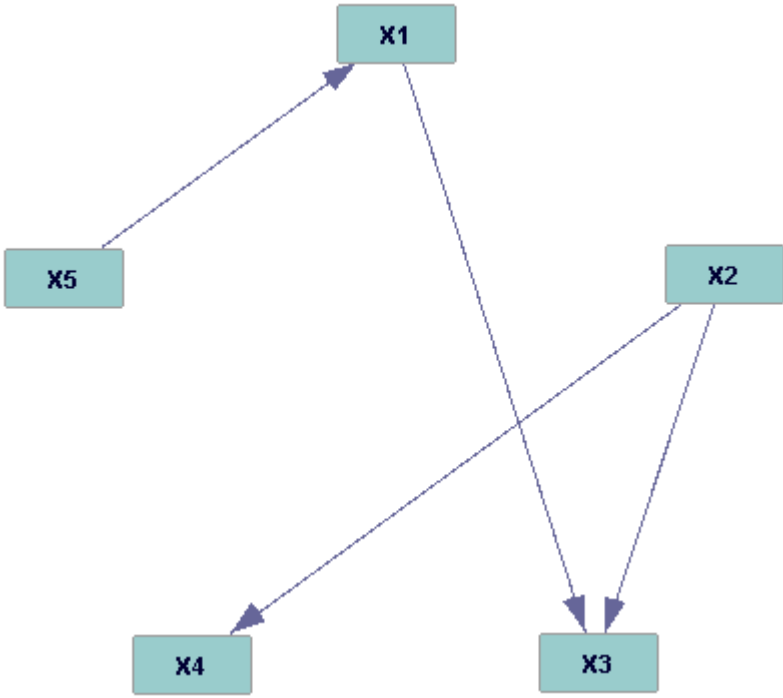
The graph manipulation box will open the following window:



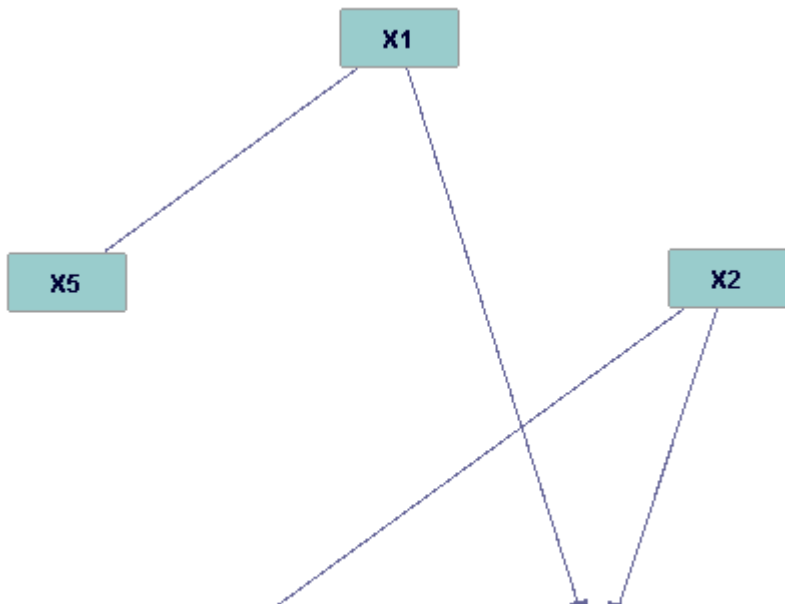
If you click on the up arrow in the DAG box at the top of the window, you will see a graph which is identical, except that the orange edge is directed into X1 instead of X2. There are only two possible DAGs in this pattern. If the input graph had been larger, with more undirected edges, there might be more possible DAGs, and you would be able to click through the DAG box to view any one of them.

### Generate Pattern from DAG

If you choose this option, Tetrad will output a pattern which represents the Markov equivalence class of the input graph. The input graph for "Generate Pattern from DAG" *must* be a DAG. For example, suppose the following is given as an input graph:



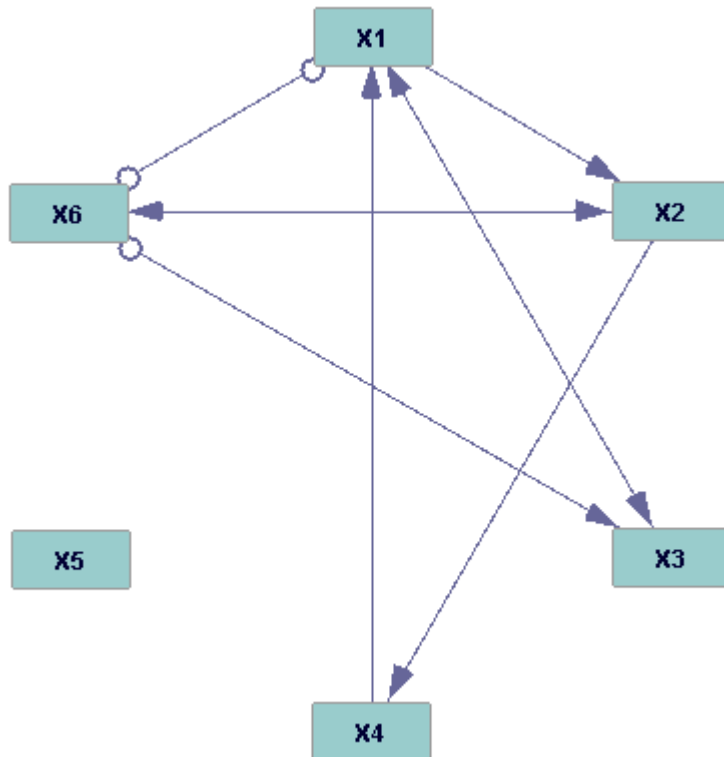
Tetrad will output the following pattern:



### Make Bidirected Edges Undirected

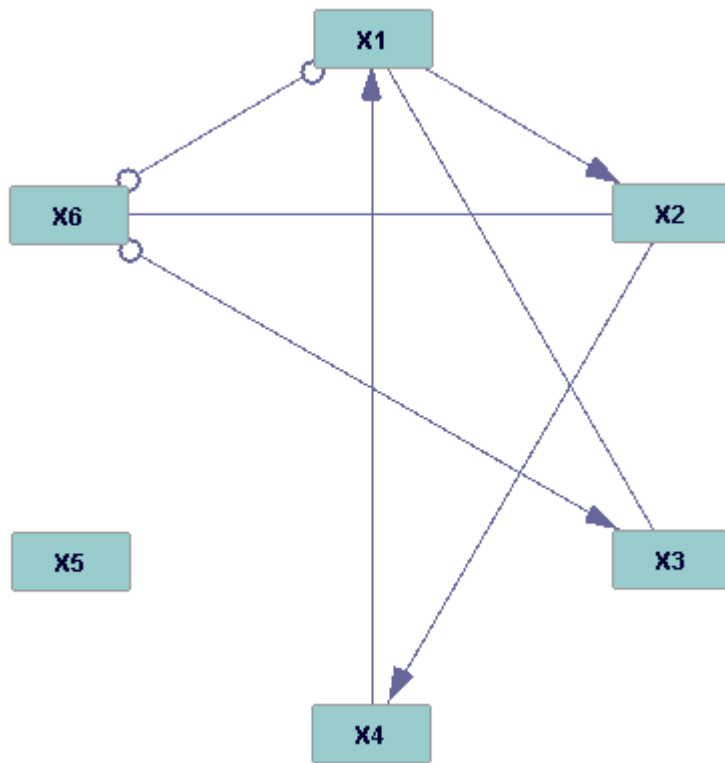
The function of this option is very simple: it makes undirected any bidirected edges present in the input graph. In this mode, the graph manipulation box will accept as input graphs with any kind of edge, with or without cycles. If there are no bidirected edges in the input graph, the graph manipulation box will simply output a copy of the original graph.

For example, suppose the following graph is given as input:



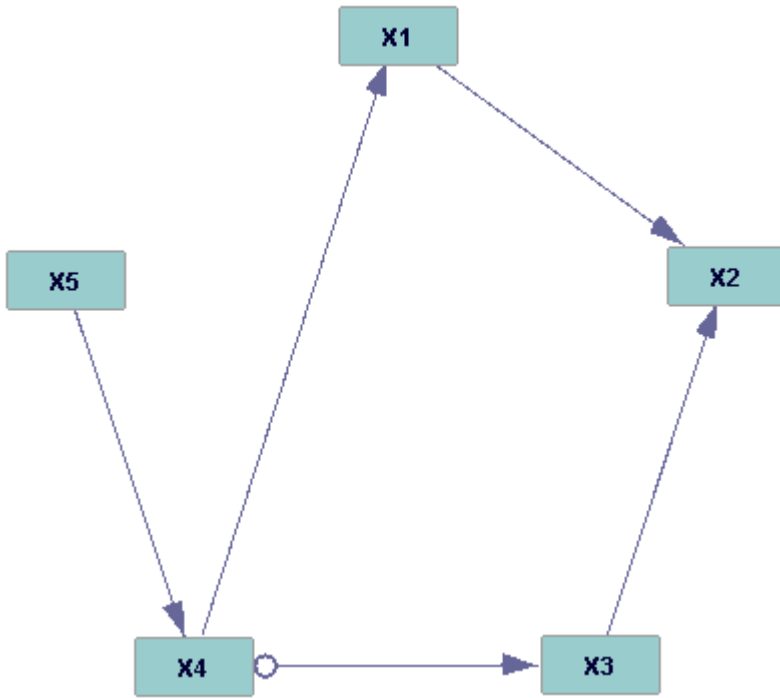


The graph manipulation box will output:

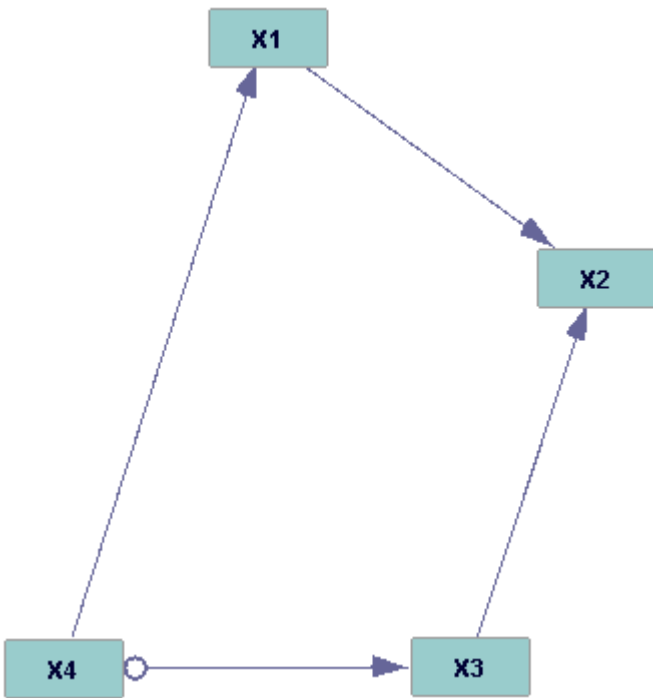


#### Extract Markov Blanket

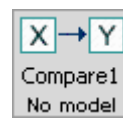
If you choose this option, a window will appear asking you to choose a target variable from a drop-down list. Tetrad will then output a window with the Markov blanket of that variable in your graph. In this mode, the graph manipulator box will accept graphs with any kind of edge, but only directed edges will be considered in determining whether a variable is in the extraction. However, once two variables are in the extraction via direct edges, any edge they have between them, whatever the type, will be shown in the output graph. For example, suppose the following graph is given as input, and X1 is chosen as the target variable:



The graph manipulation box will output the following:



The comparison box in the main workspace looks like this:



### **Possible Parent Boxes of the Comparison Box:**

- A graph box
- A graph manipulation box
- Another comparison box
- A parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- An estimator box
- An updater box
- A classify box
- A knowledge box
- A search box
- A regression box

### **Possible Child Boxes of the Comparison Box:**

- Another comparison box

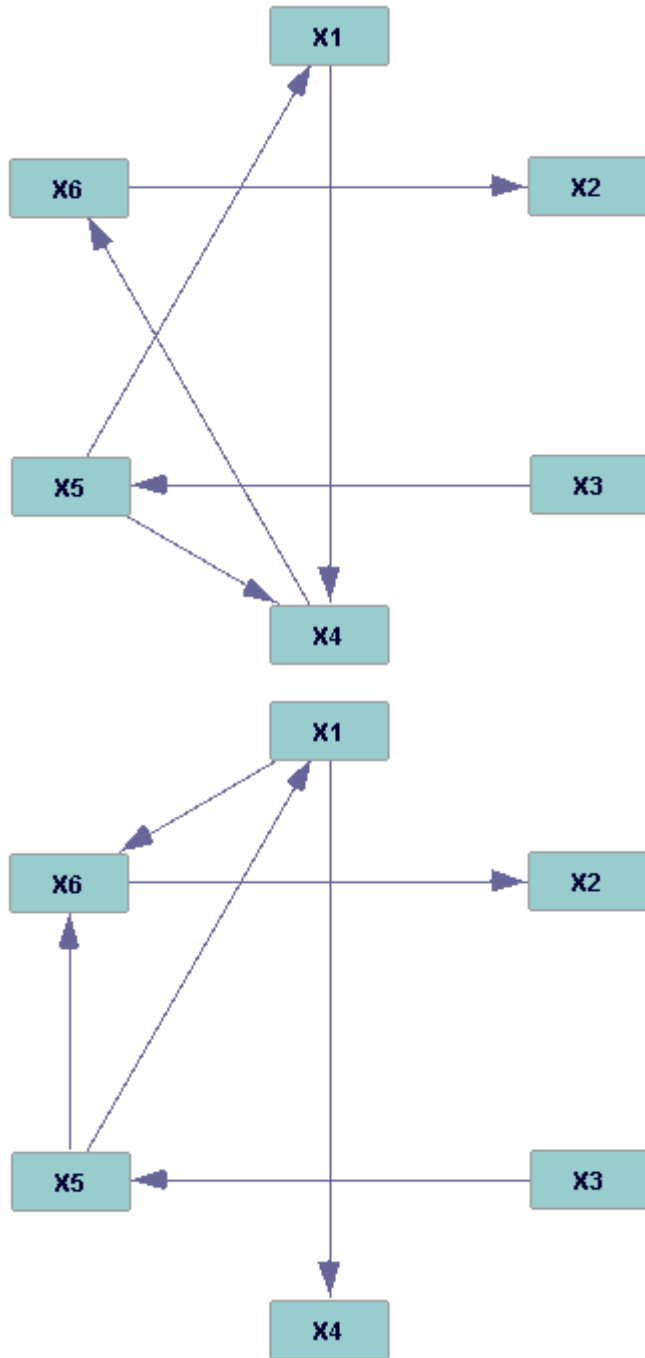
### **Using the Comparison Box:**

The comparison box compares two graphs or instantiated models, one of them identified as a reference graph, and determines which edges must be added to or taken away from the other graph to make it identical to the reference graph. The comparison box can perform three types of comparison: graph comparisons, tabular graph comparisons, and edge weight similarity comparisons.

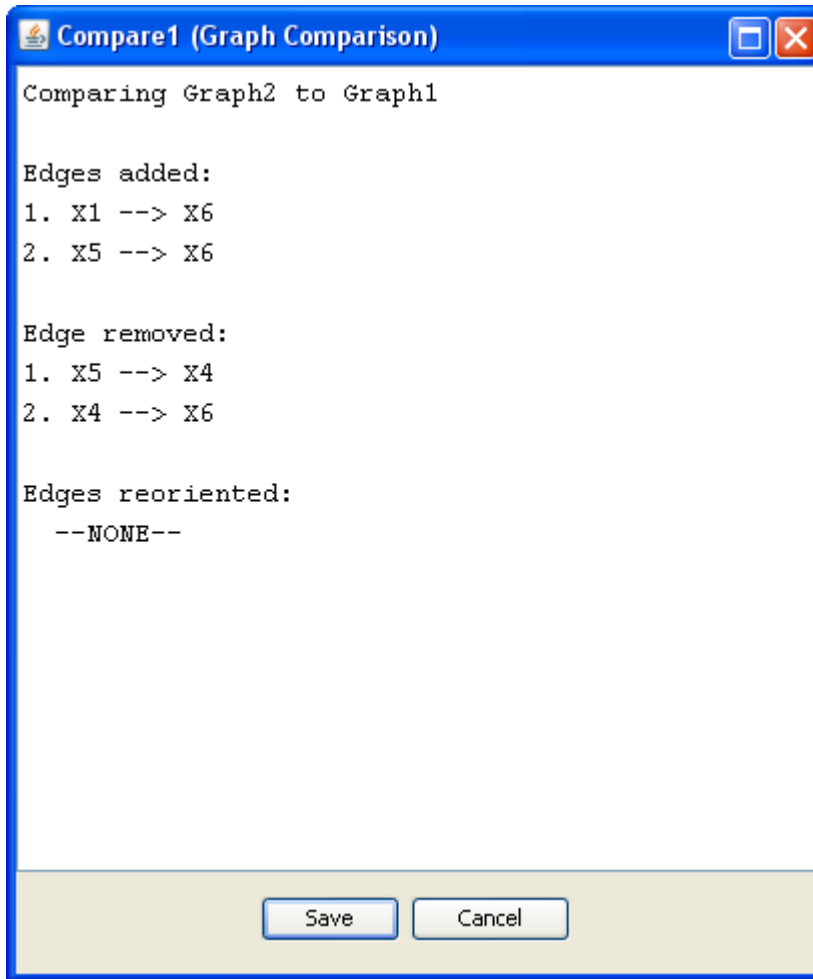
#### **Graph Comparisons**

A graph comparison compares two graphs, and gives a textual list of the edges which must be added to or taken away from one to make it identical to the other.

Take, for example, the following two graphs. The first is the reference graph, the second is the graph to be compared to it.



When these two graphs are input into the graph comparison box, a window appears which allows you to specify which of the two graphs is the reference graph, and whether it has latent variables. When the algorithm runs, the following window results:



When the listed changes have been made to the second graph, it will be identical to the first graph.

### Tabular Graph Comparisons

A tabular graph comparison tallies up and presents counts of the differences and similarities between a true graph and a reference graph. Consider the example used in the above section; once again, we'll let graph one be the true graph. Just as above, when the graphs are input to the tabular graph comparison box, we must specify which of the graphs is the reference graph, and whether it contains latent variables. We must also specify whether, when running a simulation with repeated use of the box (as in repeated simulation runs), the table should retain information about all runs, or overwrite the old information with every run. When the algorithm has run, the following window results:

|    |      | C1-T    | C2-T   | C3-T   | C4-T    | C5-T   | C6-T   | C7 |
|----|------|---------|--------|--------|---------|--------|--------|----|
|    | MULT | ADJ_COR | ADJ_FN | ADJ_FP | APT_COR | APT_FN | APT_FP |    |
| 1  | 1    | 4       | 2      | 2      | 4       | 2      | 2      |    |
| 2  |      |         |        |        |         |        |        |    |
| 3  |      |         |        |        |         |        |        |    |
| 4  |      |         |        |        |         |        |        |    |
| 5  |      |         |        |        |         |        |        |    |
| 6  |      |         |        |        |         |        |        |    |
| 7  |      |         |        |        |         |        |        |    |
| 8  |      |         |        |        |         |        |        |    |
| 9  |      |         |        |        |         |        |        |    |
| 10 |      |         |        |        |         |        |        |    |
| 11 |      |         |        |        |         |        |        |    |
| 12 |      |         |        |        |         |        |        |    |
| 13 |      |         |        |        |         |        |        |    |
| 14 |      |         |        |        |         |        |        |    |
| 15 |      |         |        |        |         |        |        |    |
| 16 |      |         |        |        |         |        |        |    |
| 17 |      |         |        |        |         |        |        |    |

The first column lists the number of adjacencies in the reference graph that are also in the true graph. The second column lists the number of adjacencies in the reference graph which are not in the compared graph. The third column lists the number of adjacencies in the comparison graph which are not in the reference graph. The next three columns list analogous information for arrowpoints (orientations of edges).

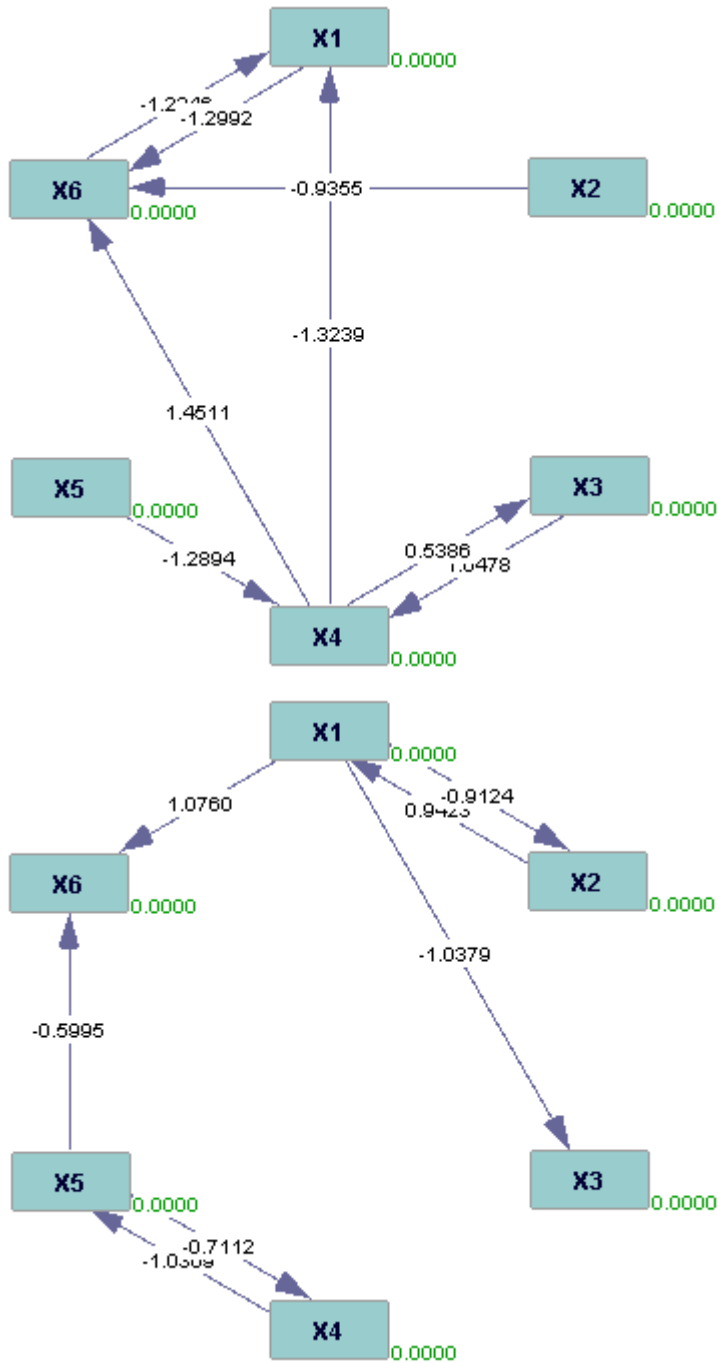
Had the algorithm been repeated three times (and had we instructed Tetrad to append rather than reset information) there would be three filled rows in the table: one containing the information for each run.

### Edge Weight Similarity Comparisons

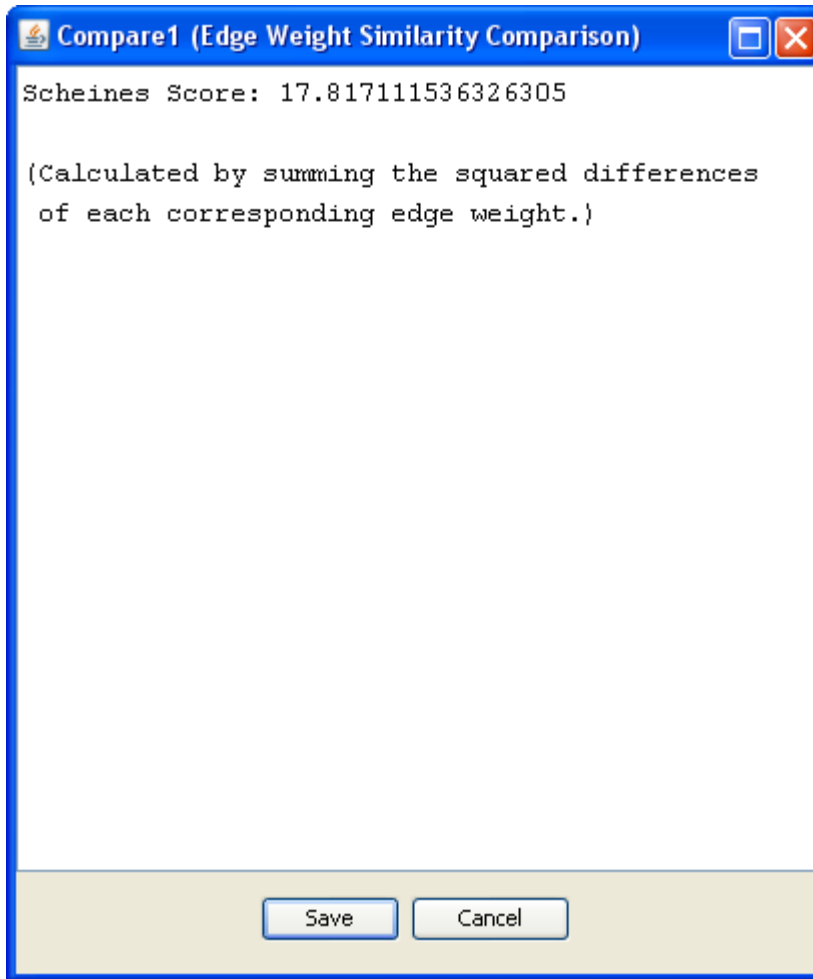
Edge weight (linear coefficient) similarity comparisons compare two linear SEM instantiated models. The output is a score equal to the sum of the squares of the differences between each corresponding edge weight in each model. Therefore, the lower the score, the more similar the two graphs are. The score has peculiarities: it does not take account of the variances of the variables, and may therefore best be used with standardized models; the

complete absence of an edge is scored as 0—so a negative coefficient compares less well with a positive coefficient than does no edge at all.

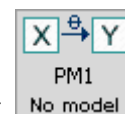
Consider, for example, an edge weight similarity comparison between the following two SEM IMs:



When they are input into an edge weight similarity comparison, the following window results:



This is, unsurprisingly, a high score; the input models have few adjacencies in common, let alone similar parameters.



The parametric model box in the main workspace looks like this:

### **Possible Parent Boxes of the Parametric Model Box:**

- A graph box
- A graph manipulation box
- Another parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- An estimator box
- A search box
- A regression box



### **Possible Child Boxes of the Parametric Model Box:**

- A graph box
- A graph manipulation box
- A comparison box
- Another parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- An estimator box
- A knowledge box
- A search box

### **Using the Parametric Model Box:**

The parametric model box takes input (usually a graph) and creates a model of the causal relationships within it.

When you open the parametric model box for the first time, you will be presented with several options for the type of model you would like to create. The options are: Bayes parametric model, SEM parametric model, and generalized SEM parametric model.

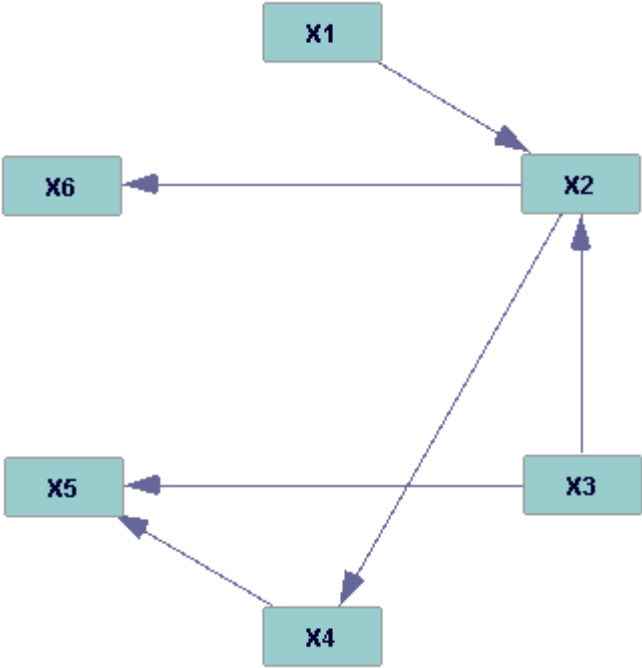
### **Bayes Parametric Models**

A Bayes parametric model takes as input a directed acyclic graph (DAG). Bayes PMs represent causal structures in which all of the variables are categorical, or discrete. This means that, for each variable, there is a finite set of values which that variable can assume. Consider a variable representing the state of a light switch. The switch is either “on” or “off”; no other values are possible.

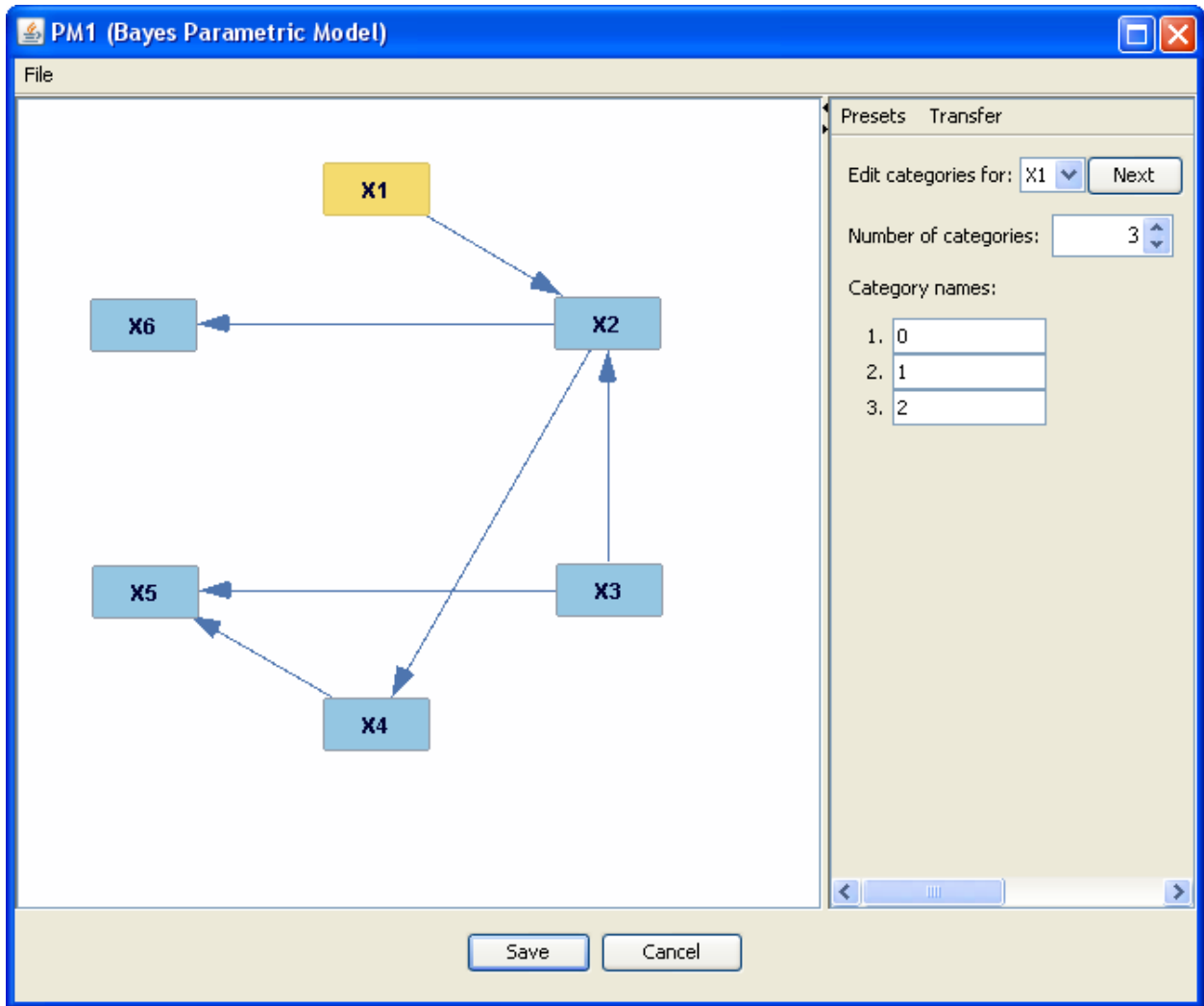
Bayes PMs consist of three components: the graphical representation of the causal structure of the model; for each named variable, the number of categories which that variable can assume; and the names of the categories associated with each variable. In the light switch example, the variable representing the light switch would have two categories, named “on” and “off.”

If you choose to create a Bayes PM, a window will open, allowing you to choose between manually assigning categories to the variables and having Tetrad randomly assign them. If you choose to have Tetrad assign the categories, you can specify a minimum and maximum number of categories possible for any given variable. You can then manually edit the number of categories and category names.

Take, for example, the following DAG:



Here is an example of a randomly created Bayes PM of this DAG, using the default variable settings:



To view the number and names of the categories associated with each variable, you can click on that variable in the graph, or choose it from the drop-down menu on the right. In this graph, X1 and X2 each have three categories, and the rest of the variables have four categories. The categories are named numerically by default.

The number of categories associated with a particular variable can be changed by clicking up or down in the drop-down menu on the right. Names of categories can be changed by overwriting the text already present. Additionally, several commonly-used preset variable names are provided under the “Presets” tab on the right. If you choose one of these configurations, the number of categories associated with the current variable will automatically be changed to agree with the configuration you have chosen. If you want all of the categories associated with a variable to have the same name with a number appended (e.g., x1, x2, x3), choose the “x1, x2, x3...” option under Presets.

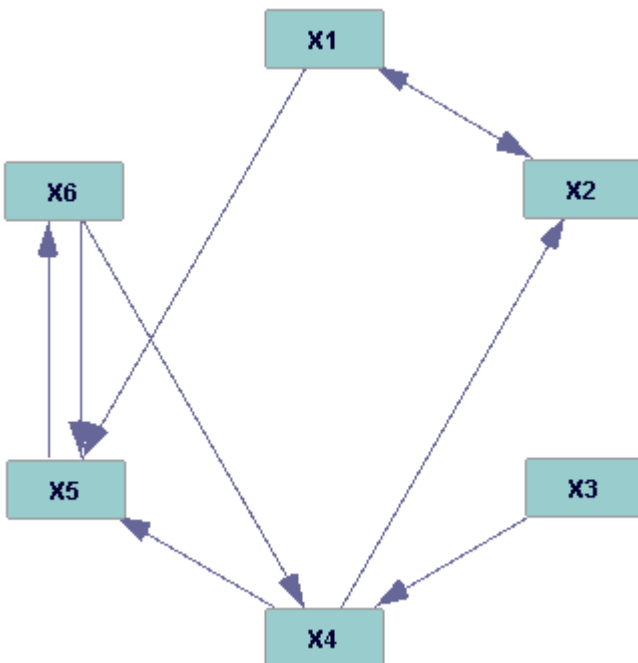
If you choose to manually create a Bayes PM, each variable will initially be assigned two categories, named numerically.

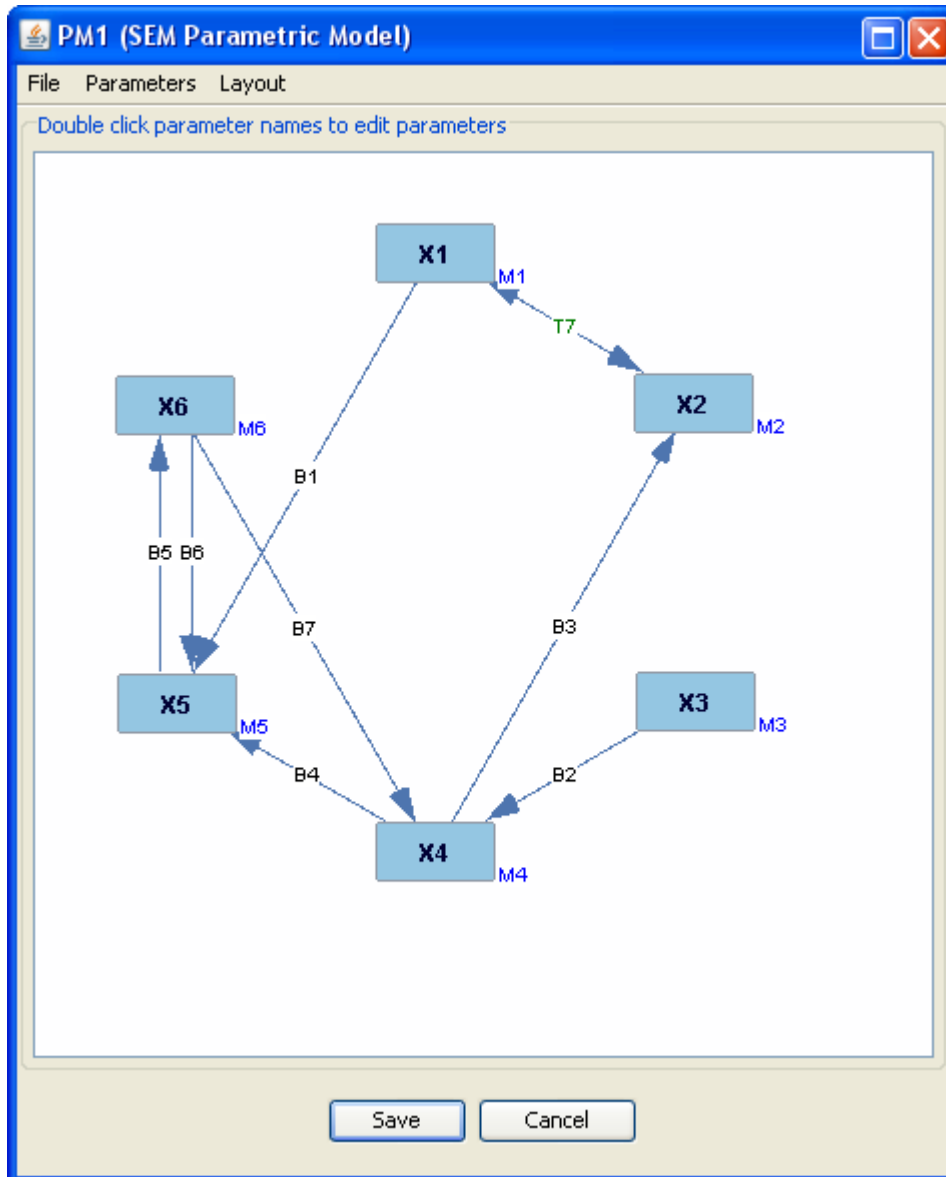
## SEM Parametric Models

The parametric model of a structural equation model (SEM) takes as input a SEM graph. SEM PMs represent causal structures in which all variables are continuous. Consider, for example, a variable which represents the length of time in seconds it takes for a ball thrown from point A to reach point B. It might take three seconds, or ten, or 5,320,192. It might even take a fractional number of seconds. There are an infinite number of possible lengths of time.

A SEM PM contains two components: the graphical causal structure of the model, and a list of parameters used in a set of linear equations representing the causal structure of the model. Each variable in a SEM PM is a linear function of a subset of the other variables and of an error term drawn from a Normal distribution.

Here is an example of a SEM graph and the SEM PM that Tetrad creates from it:





You can see the error terms in the model by clicking Parameters: Show Error Terms. In a SEM model, a bidirected edge indicates that error terms are correlated, so when this option is turned on, the edge between X1 and X2 will instead run between their error terms.

If you double click on the name of a parameter, a window will open allowing you to change the parameter's name and set its starting value for estimation.

The screenshot shows a dialog box titled "Parameter Properties" with a close button (X) in the top right corner. The dialog contains the following information:
 

- Parameter Type: Linear Coefficient
- Parameter Name:
- Fixed for Estimation?
- Starting Value for Estimation:
  - Drawn randomly
  - Set to:

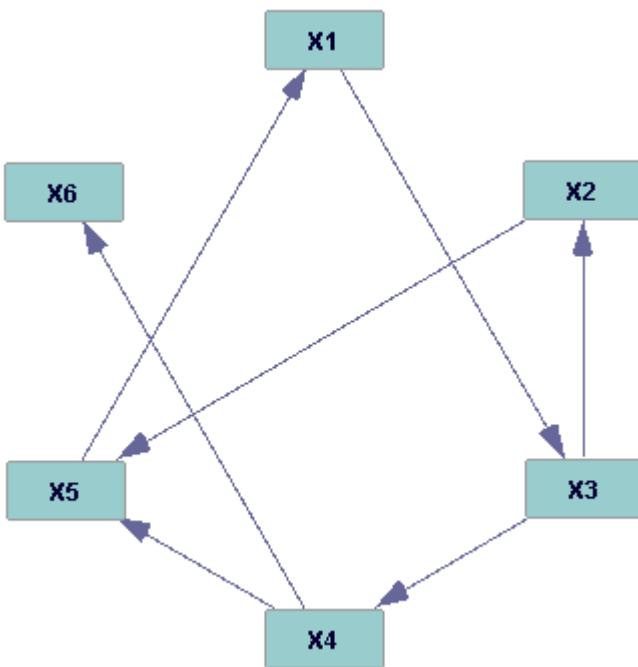
To change the parameter's name, overwrite the text box. If you are going to use the SEM PM as input to an estimator box, you can set the starting value for the estimation of this parameter by clicking "set to" and overwriting the text box. You can also instruct

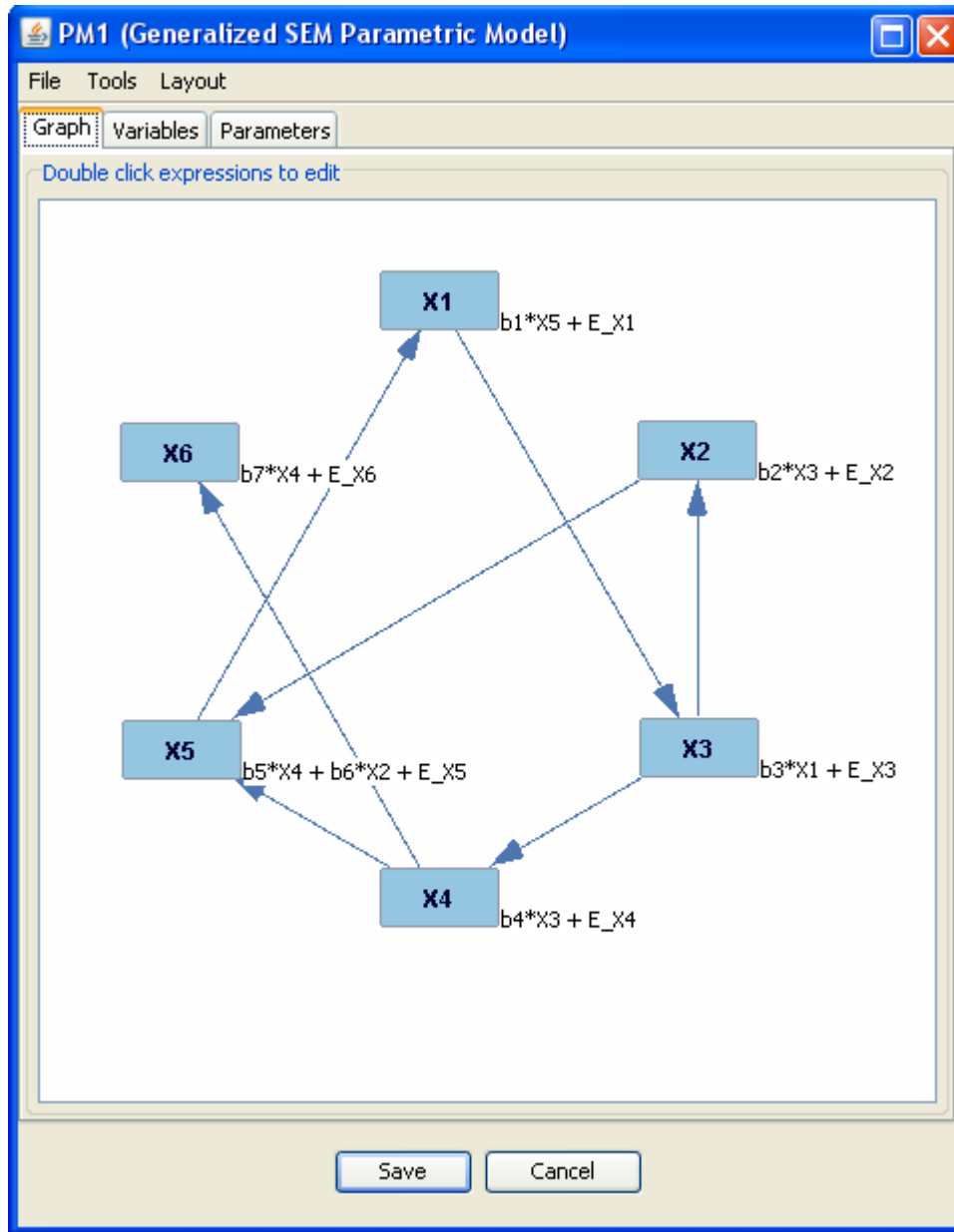
the estimator box to use a specific value for this parameter by checking “fixed for estimation.”

### Generalized SEM Parametric Models

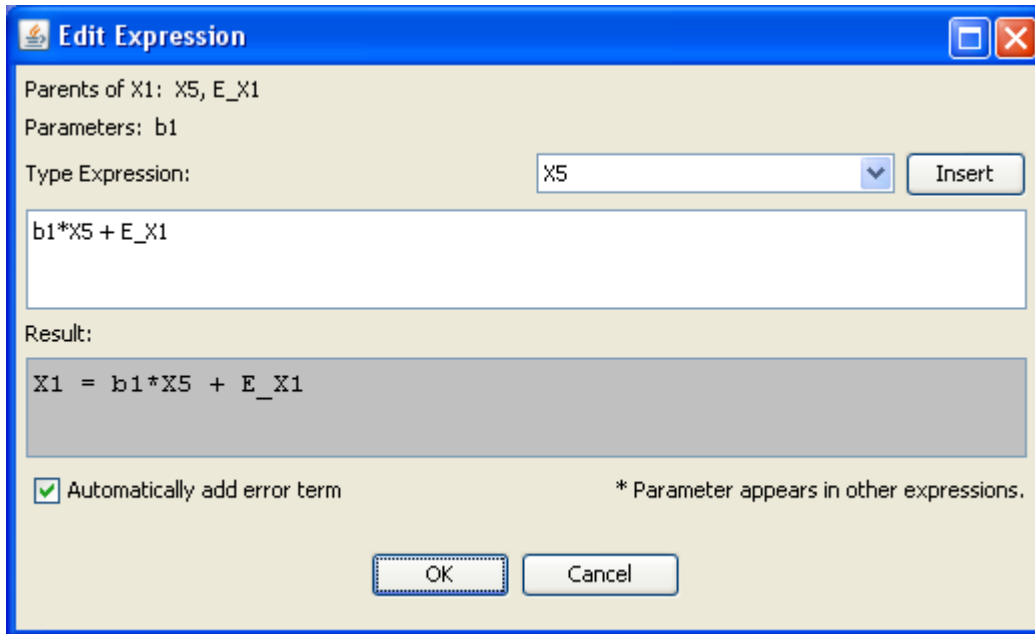
A generalized SEM parametric model takes as input a SEM graph. Like a SEM PM, it represents causal structures in which all variables are continuous (see the SEM Parametric Model section for an explanation of continuous variables). Also like a SEM PM, a generalized SEM PM contains two components: the graphical causal structure of the model, and a set of equations representing the causal structure of the model. Each variable in a generalized SEM PM is a function of a subset of the other variables and an error term. By default, the functions are linear and the error terms are drawn from a Normal distribution, but the purpose of a generalized SEM PM is to allow editing of these features. The generalized SEM PM cannot currently interpret bidirected edges.

Here is an example of a general graph and the default generalized SEM PM Tetrad creates using it:





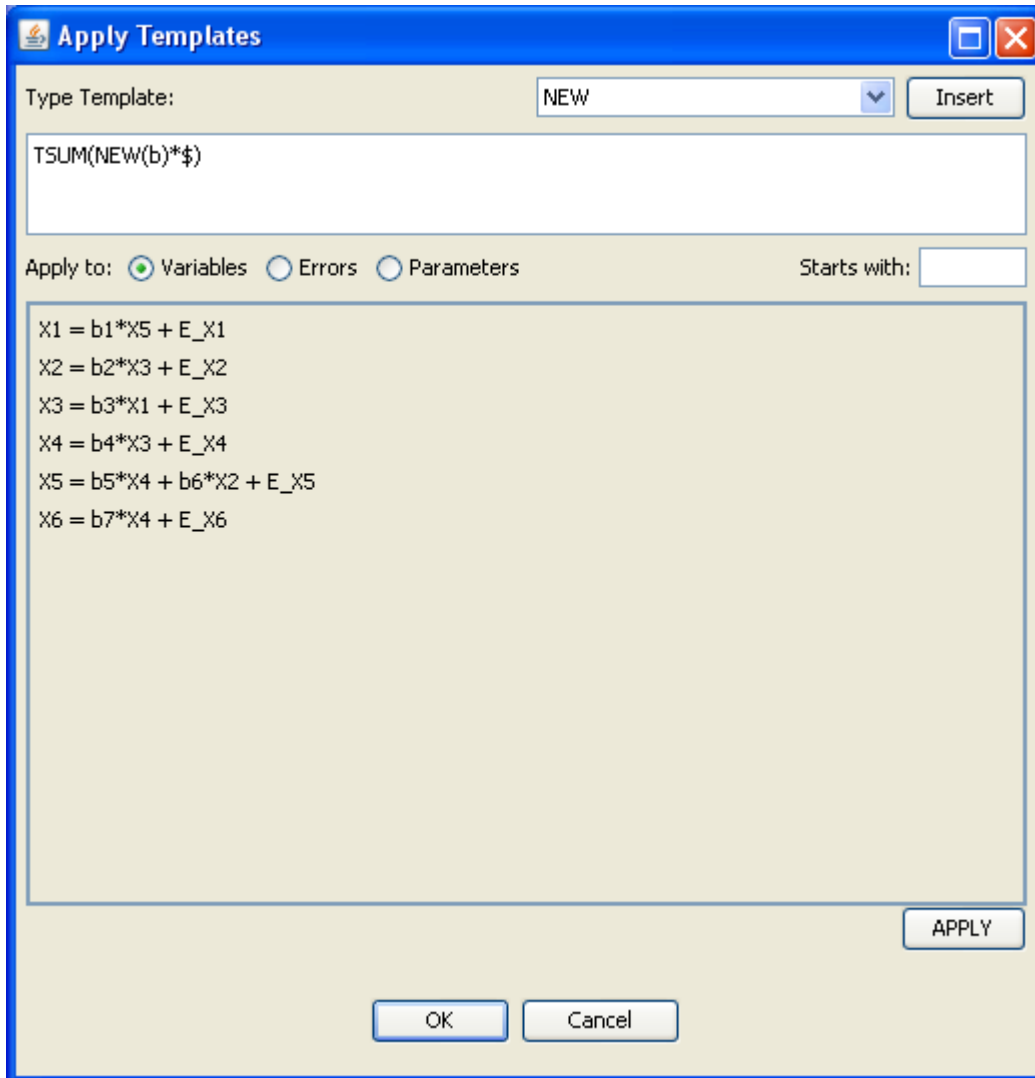
You can view the error terms by clicking Tools: Show Error Terms. If you click on the Variables tab, you will see a list of the variables and the functions which define them, and a list of the error terms, and the distributions from which their values will be drawn. Values will be drawn independently for each case if the model is instantiated (see IM box) and used to simulate data (see data box). If you click the Parameters tab, you will see a list of the parameters and the distributions from which they are drawn. When the model is instantiated in the IM box, a fixed value of each parameter will be selected according to the specified distribution. Clicking on an expression or parameter anywhere one is listed—in the graph or in one of the other tabs—will open up a window allowing you to edit the function or distribution. For instance, if you double click on the expression next to X1,  $b1 * X5 + E\_X1$ , the following window opens:



You can use this window to change the formula which defines each variable. The drop-down menu at the top of the window lists valid operators and expressions. You could, for example, change the expression from linear to quadratic by replacing  $b1*X5+E\_X1$  with  $b1*X5^2+E\_X1$ . You can also form more complicated functions, such as exponential or sine functions. If the expression you type is well-formed, it will appear in black text; if it is invalid, it will appear in red text. Tetrad will not accept any invalid changes. Individual parameters can be edited in the same way.

If you wish for several expressions or parameters to follow the same non-linear model, you may wish to use the Apply Templates tool. This allows you to edit at one time the formulae or parameters associated with several variables. To use the Apply Templates tool, click Tools: Apply Templates.... This will open the following window:

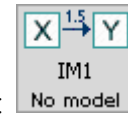




You can choose to edit variables, error terms, or parameters by clicking through the “apply to” radio buttons. If you type a letter or expression into the “starts with” box, the template you create will apply only to variables, error terms, or parameters which begin with that letter for expression. For example, in the given generalized PM, there are two types of parameters: s1-s6 and b1-b7. If you click on the “Parameters” radio button and type “b” into the “Starts with” box, only parameters b1-b7 will be affected by the changes you make.

The “Type Template” box itself works in the same way that the “Type Expression” box works in the “Edit Expression” window, with a few additions. If you scroll through the drop-down menu at the top of the window, you will see the options NEW, TSUM, and TPROD. Adding NEW to a template creates a new parameter for every variable the template is applied to. TSUM means “sum the values of this variable’s parents,” and TPROD means “multiply the values of this variable’s parents.” The contents of the parentheses following TSUM and TPROD indicate any operations which should be performed upon each variable in the sum or product, with the dollar sign (\$) functioning as a wild card. For example, in the image above, TSUM(NEW(b)\*\$) means that, for each parent variable of the variable in question, a new

parameter starting with “b” should be created and multiplied by the parent variable, and then all of the products should be added together.



The instantiated model box in the main workspace looks like this:

### **Possible Parent Boxes of the Instantiated Model Box:**

- A parametric model box
- Another instantiated model box

### **Possible Child Boxes of the Instantiated Model Box:**

- A graph box
- A graph manipulation box
- A comparison box
- A parametric model box
- Another instantiated model box
- A data box
- A data manipulation box
- An estimator box
- An updater box
- A classify box
- A knowledge box
- A search box

### **Using the Instantiated Model Box:**

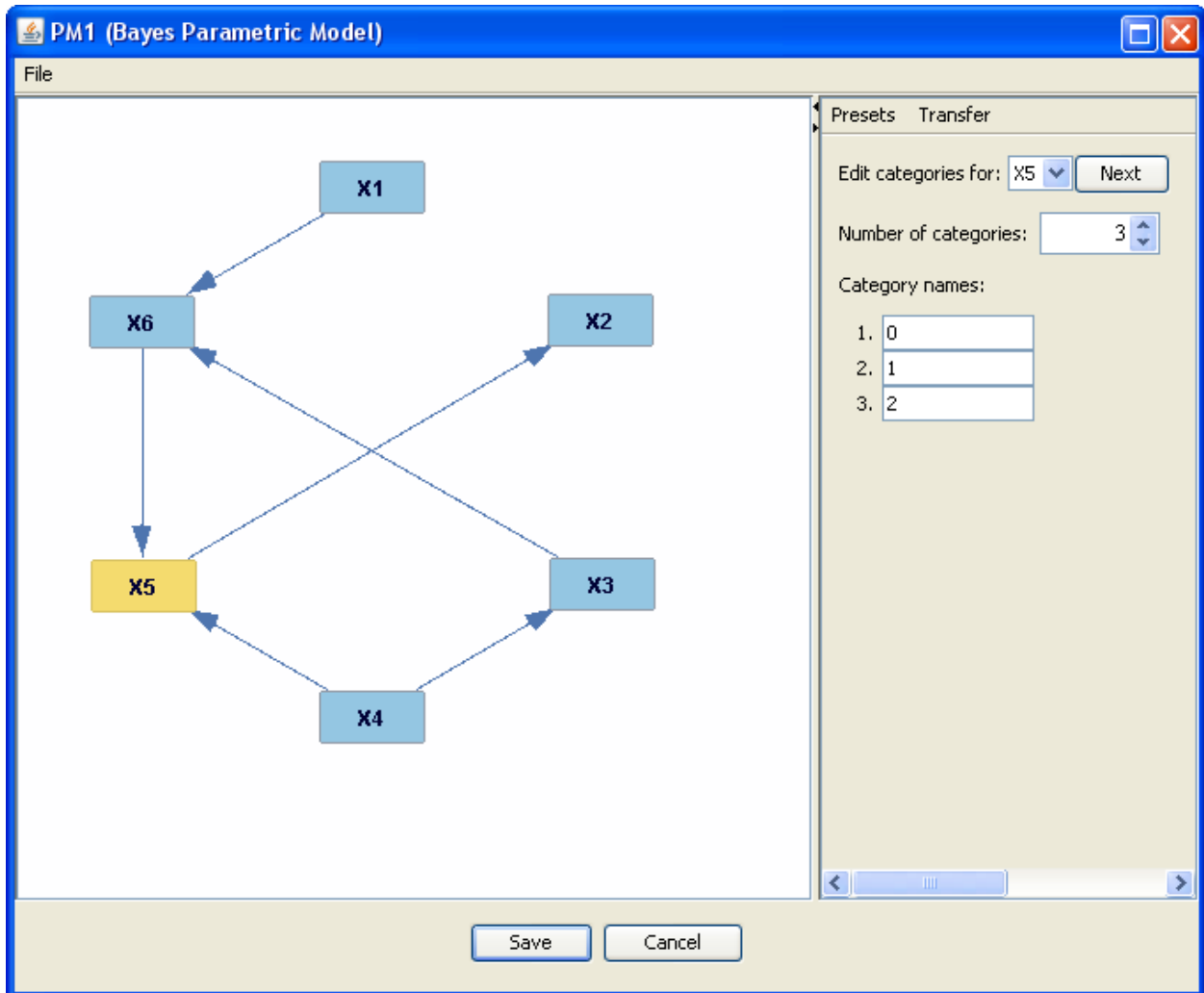
The instantiated model (IM) box takes a parametric model and assigns the variables and parameters values. The type of IM you create depends on the type of PM you use as input. A Bayes PM becomes a Bayes IM or Dirichlet IM, a SEM PM becomes a SEM IM or a standardized SEM IM, and a generalized SEM PM becomes a generalized SEM IM.

### **Bayes Instantiated Models:**

A Bayes IM consists of a Bayes parametric model with defined probability values for all variables. This means that, conditional on the values of each of its parent variables, there is a defined probability that a variable will take on each of its possible values. For each assignment of a value to each of the parents of a variable X, the probabilities of the several values of X must sum to 1.

If you choose to create a Bayes IM, a window will open allowing you to either manually set the probability values of the model or have Tetrad assign them randomly. If you choose to have Tetrad assign probability values, you can manually edit them later. If you uncheck “Pick new random values every time this Bayes IM is re-initialized,” then if you destroy the model in the box and create a new one, the box will remember the random values it created.

Here is an example of a Bayes PM and its randomly created instantiated model:



1. Choose the next variable to edit: X5

2. Scroll to a row (that is, combination of parent values) in the table below.

3. Click in the appropriate box and assign a probability to each value of the chosen variable in that row.

| X4 | X6 | X5=0   | X5=1   | X5=2   |
|----|----|--------|--------|--------|
| 0  | 0  | 0.0346 | 0.4425 | 0.5229 |
| 0  | 1  | 0.3469 | 0.2891 | 0.3640 |
| 0  | 2  | 0.0505 | 0.4195 | 0.5300 |
| 1  | 0  | 0.1756 | 0.4766 | 0.3478 |
| 1  | 1  | 0.3573 | 0.2953 | 0.3474 |
| 1  | 2  | 0.1182 | 0.1350 | 0.7469 |
| 2  | 0  | 0.2829 | 0.3403 | 0.3768 |
| 2  | 1  | 0.6757 | 0.2820 | 0.0423 |
| 2  | 2  | 0.3529 | 0.5635 | 0.0837 |

Right click in table to randomize.

In the model above, when X4 and X6 are both 0, the probability that X5 is 0 is 0.0346, that X5 is 1 is 0.4425, and that X5 is 2 is 0.5229. Since X5 *must* be 0, 1, or 2, those three values must add up to one, as must the values in every row.

To view the probability values of a variable, either double click on the variable in the graph or choose it from the drop-down menu on the right. You can manually set a given probability value by overwriting the text box. Be warned that changing the value in one cell will delete the values in all of the other cells in the row. Since the values in any row must sum to one, if all of the cells in a row but one are set, Tetrad will automatically change the value in the last cell to make the sum correct. For instance, in the above model, if you change the first row such that the probability that X5 = 0 is 0.5000 and the probability that X5 = 1 is 0.4000, the probability that X5 = 2 will automatically be set to 0.1000.

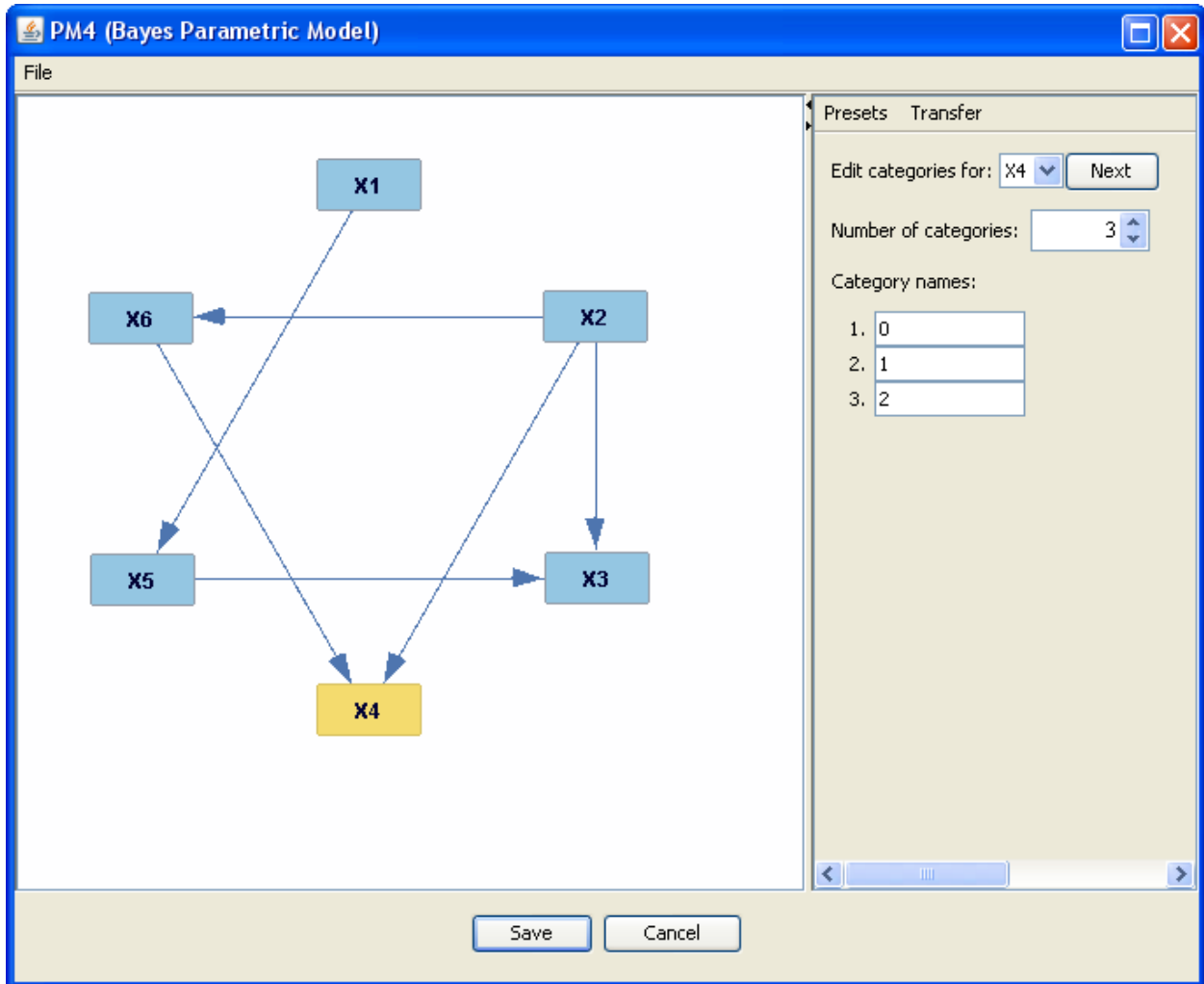
If you right click on a cell in the table, you can choose to randomize the probabilities in the row containing that cell, randomize the values in all incomplete rows in the table, randomize the entire table, or randomize the table of every variable in the model. You can also choose to clear the row or table.

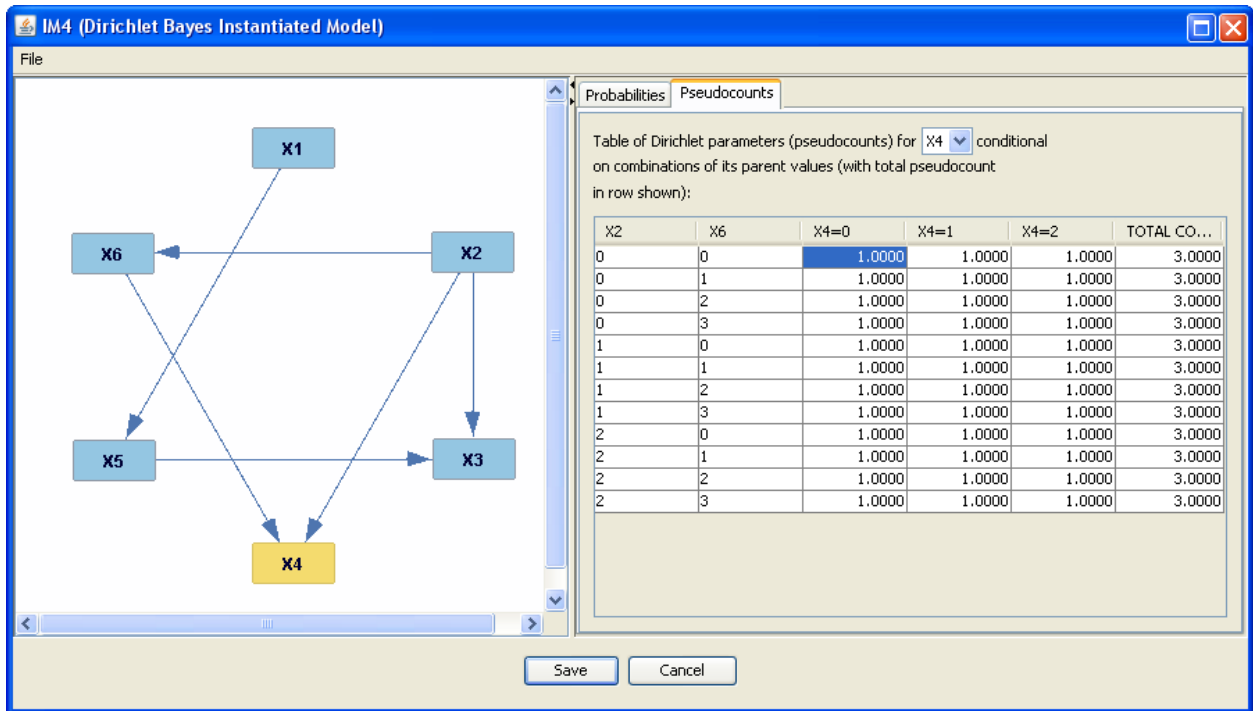
### Dirichlet Instantiated Models:

A Dirichlet instantiated model is a specialized form of a Bayes instantiated model. Like a Bayes IM, a Dirichlet IM consists of a Bayes parametric model with defined probability values. Unlike a Bayes IM, these probability values are not manually set or assigned randomly. Instead, the pseudocount is manually set or assigned uniformly, and the probability values are derived

from it. The pseudocount of a given value of a variable is the number of data points for which the variable takes on that value, conditional on the values of the variable's parents, where these numbers are permitted to take on non-negative real values. Since we are creating models without data, we can set the pseudocount to be any number we want. If you choose to create a Dirichlet IM, a window will open allowing you to either manually set the pseudocounts, or have Tetrad set all the pseudocounts in the model to one number, which you specify.

Here is an example of a Bayes PM and the Dirichlet IM which Tetrad creates from it when all pseudocounts are set to one:

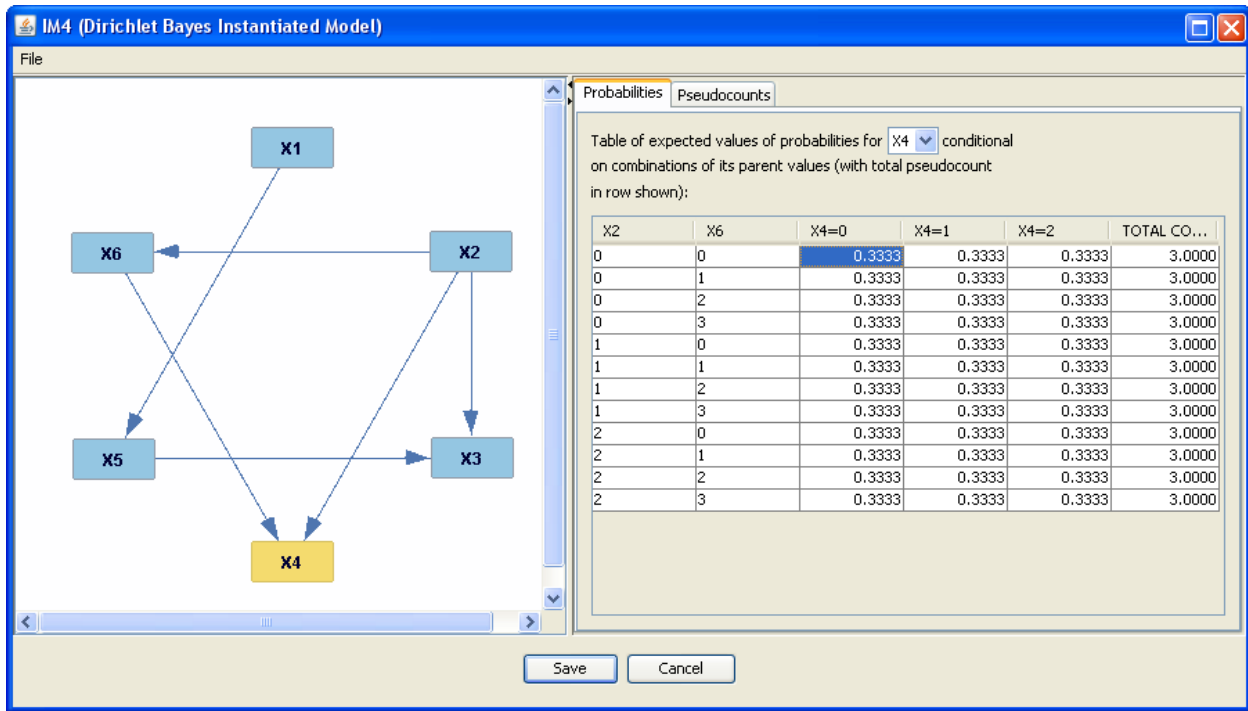




In the above model, when  $X2=0$  and  $X6=0$ , there is one (pseudo) data point at which  $X4=0$ , one at which  $X4=1$ , and one at which  $X4=2$ . There are three total (pseudo) data points in which  $X2=0$  and  $X6=0$ . You can view the pseudocounts of any variable by clicking on it in the graph or choosing it from the drop-down menu at the top of the window. To edit the value of a pseudocount, double click on it and overwrite it. The total count of a row cannot be directly edited.

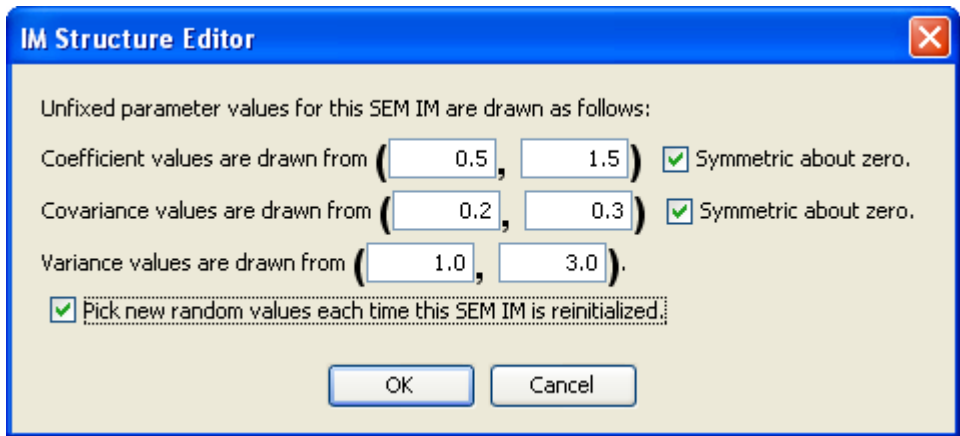
From the pseudocounts, Tetrad determines the conditional probability of a category. This estimation is done by taking the pseudocount of a category and dividing it by the total count for its row. For instance, the total count of  $X4$  when  $X2=0$  and  $X6=0$  is 3. So the conditional probability of  $X4=0$  given that  $X2=0$  and  $X6=0$  is  $1/3$ . The reasoning behind this is clear: in a third of the data points in which  $X2$  and  $X6$  are both 0,  $X4$  is also 0, so the probability that  $X4=0$  given that  $X2$  and  $X6$  also equal 0 is probably one third. This also guarantees that the conditional probabilities for any configuration of parent variables add up to one, which is necessary.

To view the table of conditional probabilities for a variable, click the Probabilities tab. In the above model, the Probabilities tab looks like this:



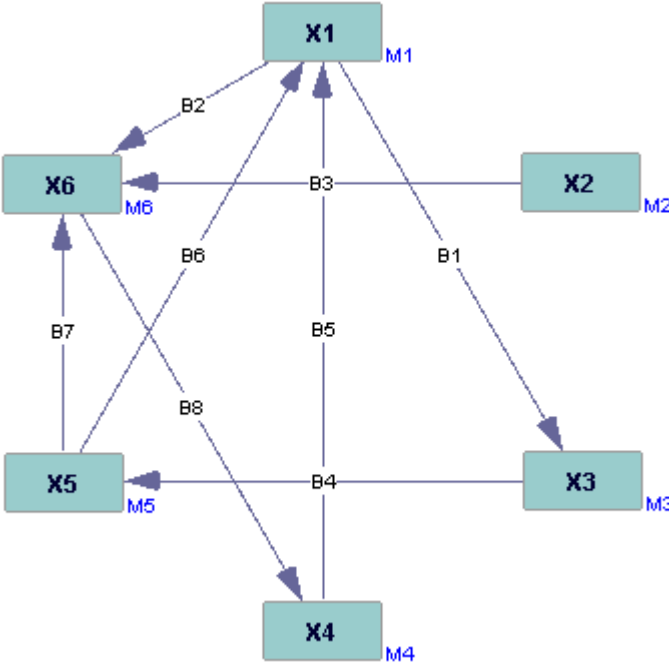
SEM Instantiated Models:

A SEM instantiated model is a SEM parametric model in which the parameters and error terms have defined values. If you choose to create a SEM IM, the following window will open:

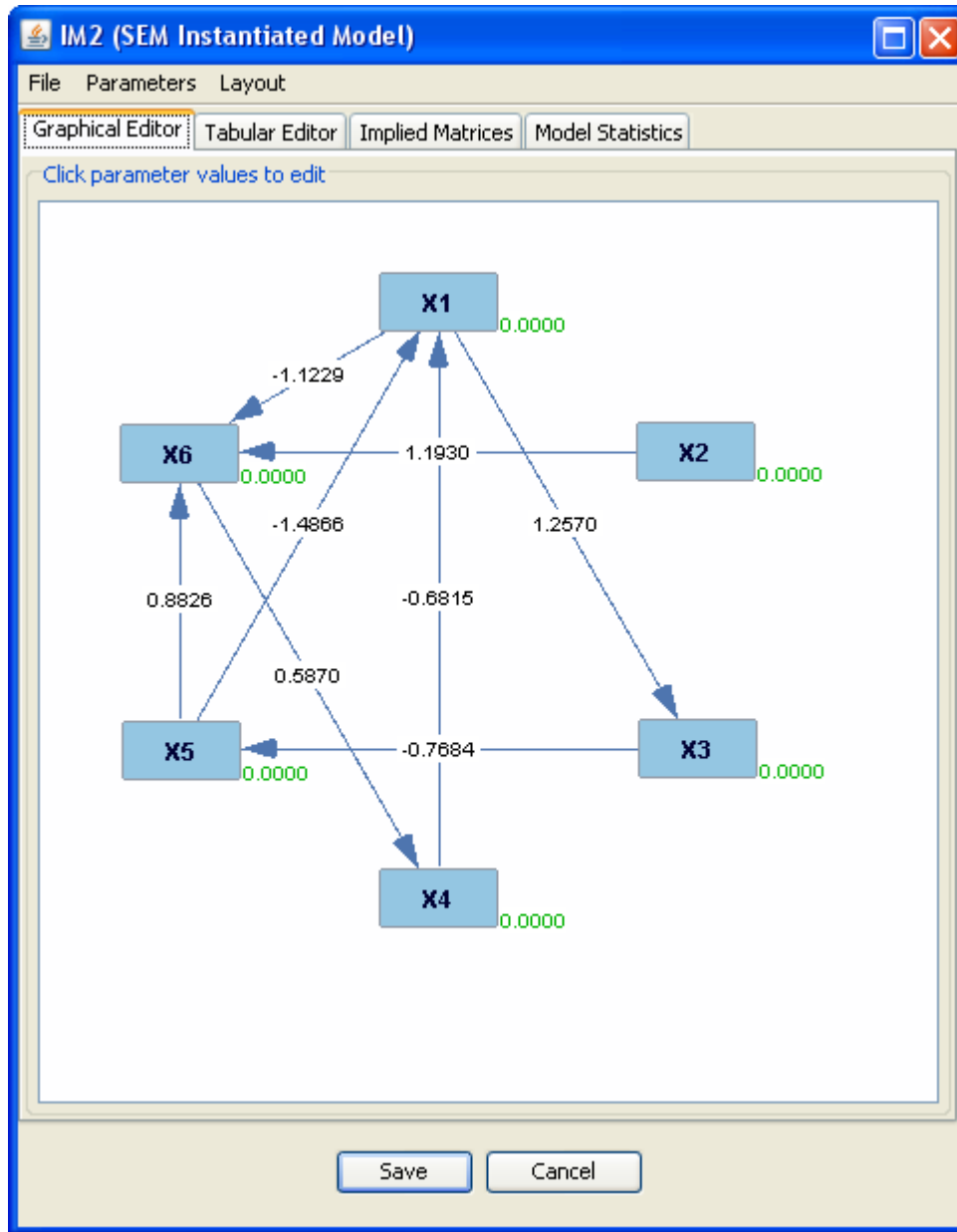


Using this box, you can specify the ranges of values from which you want coefficients, covariances, and variances to be drawn for the parameters in the model. In the above box, for example, all linear coefficients will be between -1.5 and -0.5 or 0.5 and 1.5. If you uncheck “symmetric about zero,” they will only be between 0.5 and 1.5.

Here is an example of a SEM PM and a SEM IM generated from it using the default settings:







You can now manually edit the values of parameters in one of two ways. Double clicking on the parameter in the graph will open up a small text box for you to overwrite. Or you can click on the Tabular Editor tab, which will show all of the parameters in a table which you can edit. The Tabular Editor tab of our SEM IM looks like this:

Click parameter values to edit

| From | To | Type       | Value   | SE | T | P |
|------|----|------------|---------|----|---|---|
| X1   | X3 | Edge Coef. | 1.2570  | *  | * | * |
| X1   | X6 | Edge Coef. | -1.1229 | *  | * | * |
| X2   | X6 | Edge Coef. | 1.1930  | *  | * | * |
| X3   | X5 | Edge Coef. | -0.7684 | *  | * | * |
| X4   | X1 | Edge Coef. | -0.6814 | *  | * | * |
| X5   | X1 | Edge Coef. | -1.4866 | *  | * | * |
| X5   | X6 | Edge Coef. | 0.8826  | *  | * | * |
| X6   | X4 | Edge Coef. | 0.5870  | *  | * | * |
| X1   | X1 | Std. Dev.  | 1.0618  | *  | * | * |
| X2   | X2 | Std. Dev.  | 1.1771  | *  | * | * |
| X3   | X3 | Std. Dev.  | 1.6402  | *  | * | * |
| X4   | X4 | Std. Dev.  | 1.1122  | *  | * | * |
| X5   | X5 | Std. Dev.  | 1.5470  | *  | * | * |
| X6   | X6 | Std. Dev.  | 1.4599  | *  | * | * |
| X1   | X1 | Mean       | 0.0000  | *  | * | * |
| X2   | X2 | Mean       | 0.0000  | *  | * | * |
| X3   | X3 | Mean       | 0.0000  | *  | * | * |
| X4   | X4 | Mean       | 0.0000  | *  | * | * |
| X5   | X5 | Mean       | 0.0000  | *  | * | * |
| X6   | X6 | Mean       | 0.0000  | *  | * | * |

Save Cancel

In an estimator box, the Tabular Editor tab provides statistics showing how robust the estimation of each parameter is. This is the function of the SE, T, and P columns. Our SEM IM, however, is in an instantiated model box, so these columns are empty.

The Implied Matrices tab shows matrices of different kinds of relationships between variables in the model. In the Implied Matrices tab, you can view the covariance or correlation matrix for all variables (including latents) or just measured variables. In our SEM IM, the Implied Matrices tab looks like this:

The screenshot shows a software window titled "IM2 (SEM Instantiated Model)" with a menu bar (File, Parameters, Layout) and four tabs: Graphical Editor, Tabular Editor, Implied Matrices (selected), and Model Statistics. A dropdown menu is open, showing "Implied covariance matrix (all variables)". Below the menu is a table with 6 rows and 6 columns. The columns are labeled X1 through X5, and the rows are labeled X1 through X6. The table contains numerical values representing the implied covariance matrix. At the bottom of the window are "Save" and "Cancel" buttons.

|    | X1       | X2      | X3       | X4      | X5     |
|----|----------|---------|----------|---------|--------|
| X1 | 10.5353  |         |          |         |        |
| X2 | 0.5393   | 1.3855  |          |         |        |
| X3 | 10.1405  | 0.6779  | 11.5373  |         |        |
| X4 | -7.6475  | 0.3449  | -7.0866  | 7.2893  |        |
| X5 | -4.2003  | -0.5209 | -4.3508  | 2.5206  | 2.2676 |
| X6 | -14.1987 | 0.5875  | -13.5442 | 11.6683 | 5.4250 |

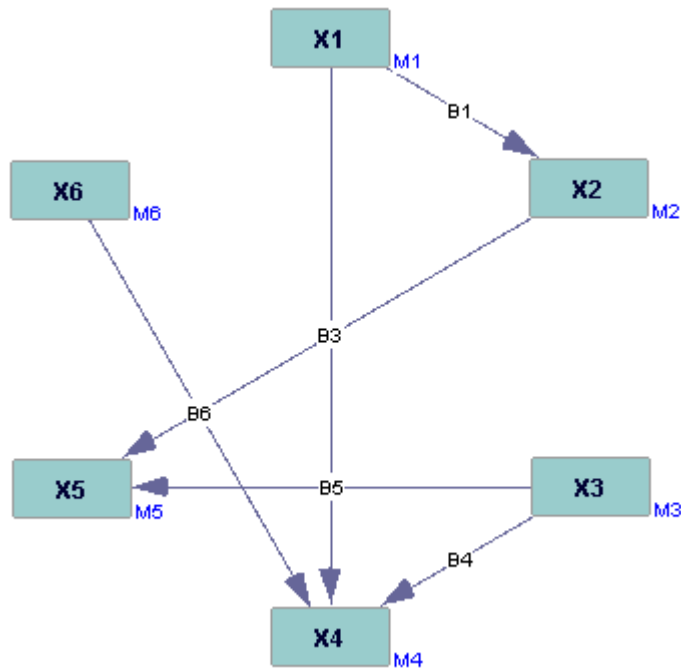
You can choose the matrix you wish to view from the drop-down menu at the top of the window. Only half of any matrix is shown, because in a well-formed acyclic model, both halves should be identical. The cells in the Implied Matrices tab cannot be edited.

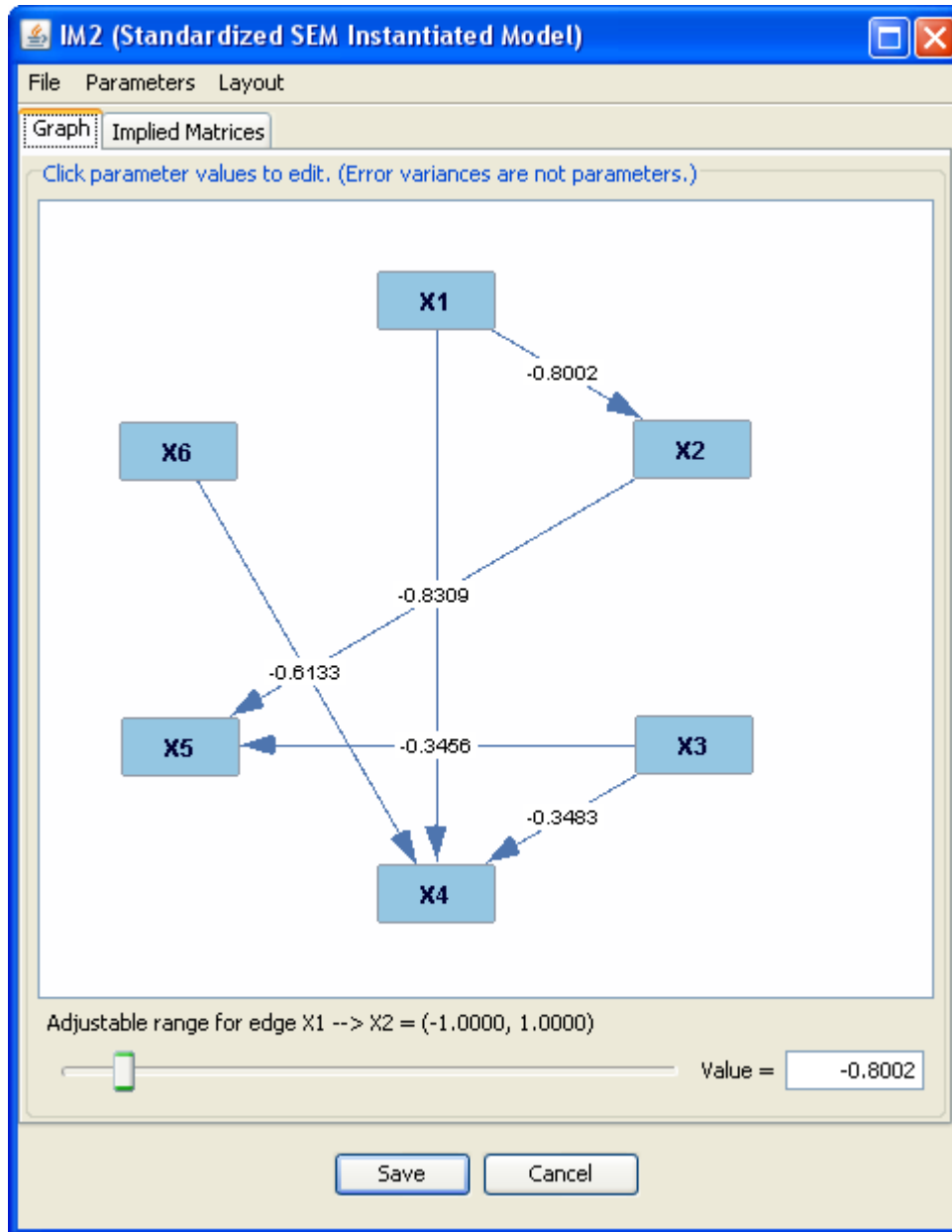
In an estimator box, the Model Statistics tab provides goodness of fit statistics for the SEM IM which has been estimated. Our SEM IM, however, is in an instantiated model box, so no estimation has occurred, and the Model Statistics tab is empty.

Standardized SEM Instantiated Models:

A standardized SEM instantiated model consists of a SEM parametric model with defined values for its parameters. In a standardized SEM IM, each variable (not error terms) has a Normal distribution with 0 mean and unit variance. The input PM to a standardized SEM IM must be acyclic.

Here is an example of an acyclic SEM PM and the standardized SEM IM which Tetrad creates from it:



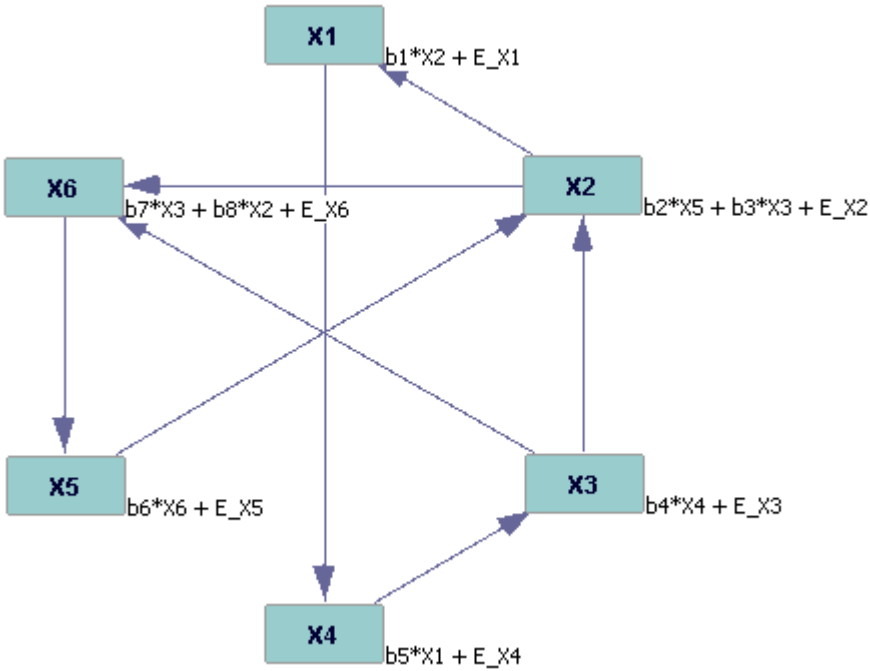


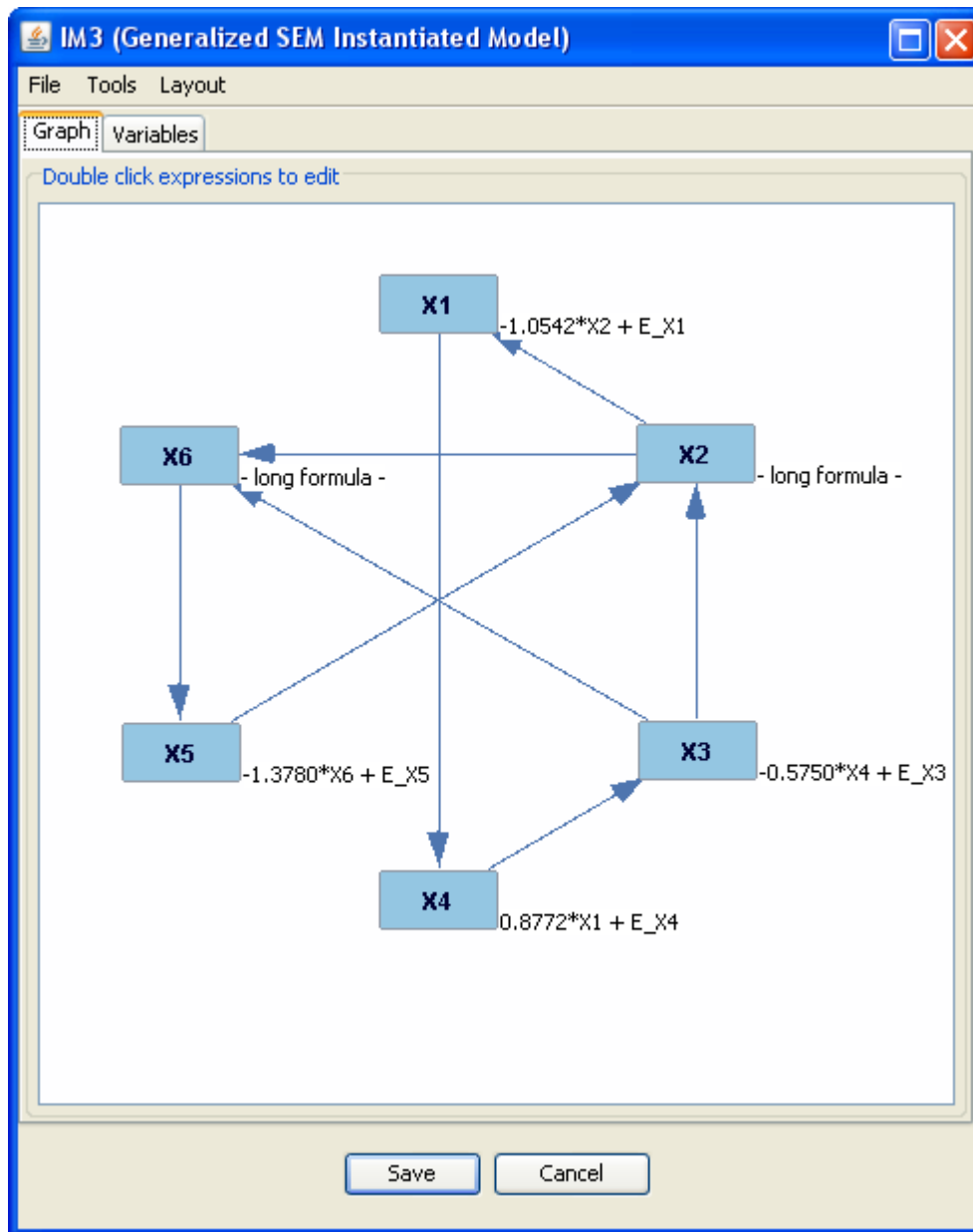
To edit a parameter, double click on it. A slider will open at the bottom of the window (shown above for the parameter between X1 and X2). Click and drag the slider to change the value of the parameter, or enter the value you wish into the box. The value must stay within a certain range in order for the Normal distribution to stay standardized, so if you attempt to overwrite the text box on the bottom right with a value outside the listed range, Tetrad will not allow it. In a standardized SEM IM, error terms are not considered parameters and cannot be edited, but you can view them by clicking Parameters: Show Error Terms. The Implied Matrices tab works in the same way that it does in a normal SEM IM.

Generalized SEM Instantiated Models:

A generalized SEM instantiated model is a generalized SEM parametric model with defined values for its parameters. Since the distributions of the parameters were specified in the SEM PM, Tetrad does not give you the option of specifying these before it creates the instantiated model.

Here is an example of a generalized SEM PM and its generalized SEM IM:





Note that the expressions for X6 and X2 are not shown, having been replaced with the words “long formula.” Formulae over a certain length—the default setting is 25 characters—are hidden to improve visibility. Long formulae can be viewed in the Variables tab, which lists all variables and their formulae. You can change the cutoff point for long formulae by clicking Tools: Formula Cutoff.

If you double click on a formula in either the graph or the Variables tab, you can change the value of the parameters in that formula.

| X  | Y  |
|----|----|
| 12 | 17 |
| 12 | 17 |

Data1  
No model

The data box in the main workspace looks like this:

### **Possible Parent Boxes of the Data Box:**

- A graph box
- A graph manipulation box
- A parametric model box
- An instantiated model box
- Another data box
- A data manipulation box
- An estimator box
- An updater box

### **Possible Child Boxes of the Data Box:**

- A graph box
- A comparison box
- A parametric model box
- An instantiated model box
- Another data box
- A data manipulation box
- An estimator box
- A classify box
- A knowledge box
- A search box
- A regression box

### **Using the Data Box:**

The data box stores the actual data sets from which causal structures are determined. Data can be loaded into the data box from a preexisting source, manually filled in Tetrad, or simulated from an instantiated model.

### **Data Files**

Correlation or covariance matrices loaded into Tetrad should be ascii text files, with a sample size row, followed by the names of variables on the next row, followed by a lower triangular matrix. For example:

```
1000
X1    X2    X3    X4    X5    X6
1.0000
0.0312    1.0000
-0.5746    0.4168    1.0000
-0.5996    0.4261    0.9544    1.0000
```



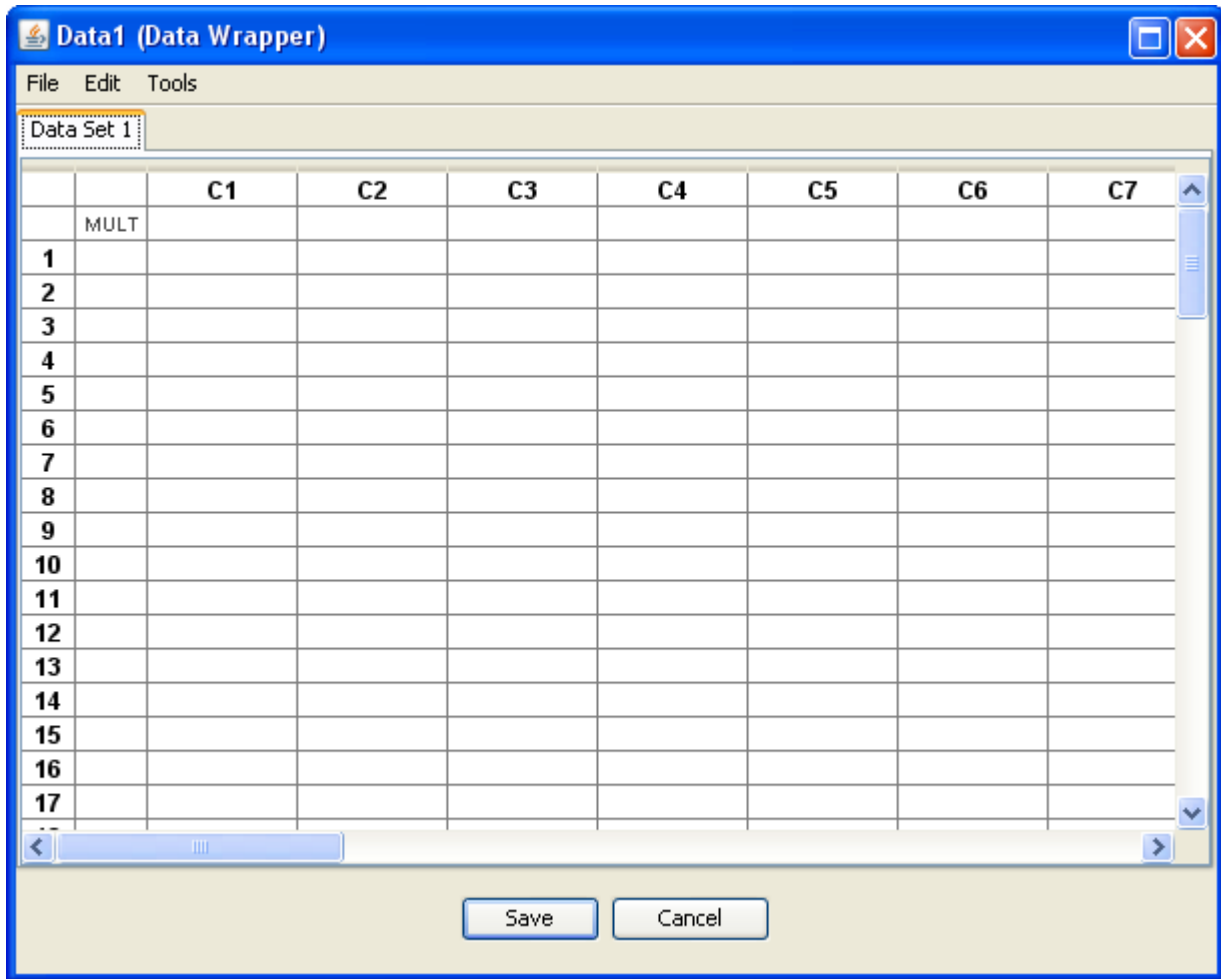
|        |        |         |         |        |        |
|--------|--------|---------|---------|--------|--------|
| 0.8691 | 0.0414 | -0.4372 | -0.4487 | 1.0000 |        |
| 0.6188 | 0.0427 | -0.1023 | -0.0913 | 0.7172 | 1.0000 |

Categorical data should also be an ascii text file, with columns representing the values of each variable in each case. Beyond that, there is a great deal of flexibility in the layout: delimiters may be tabs, white space, or commas, comments and missing data may be marked by any symbol you like, there may be a row of variable names or not, and case IDs may be present or not. There should be no sample size. For example:

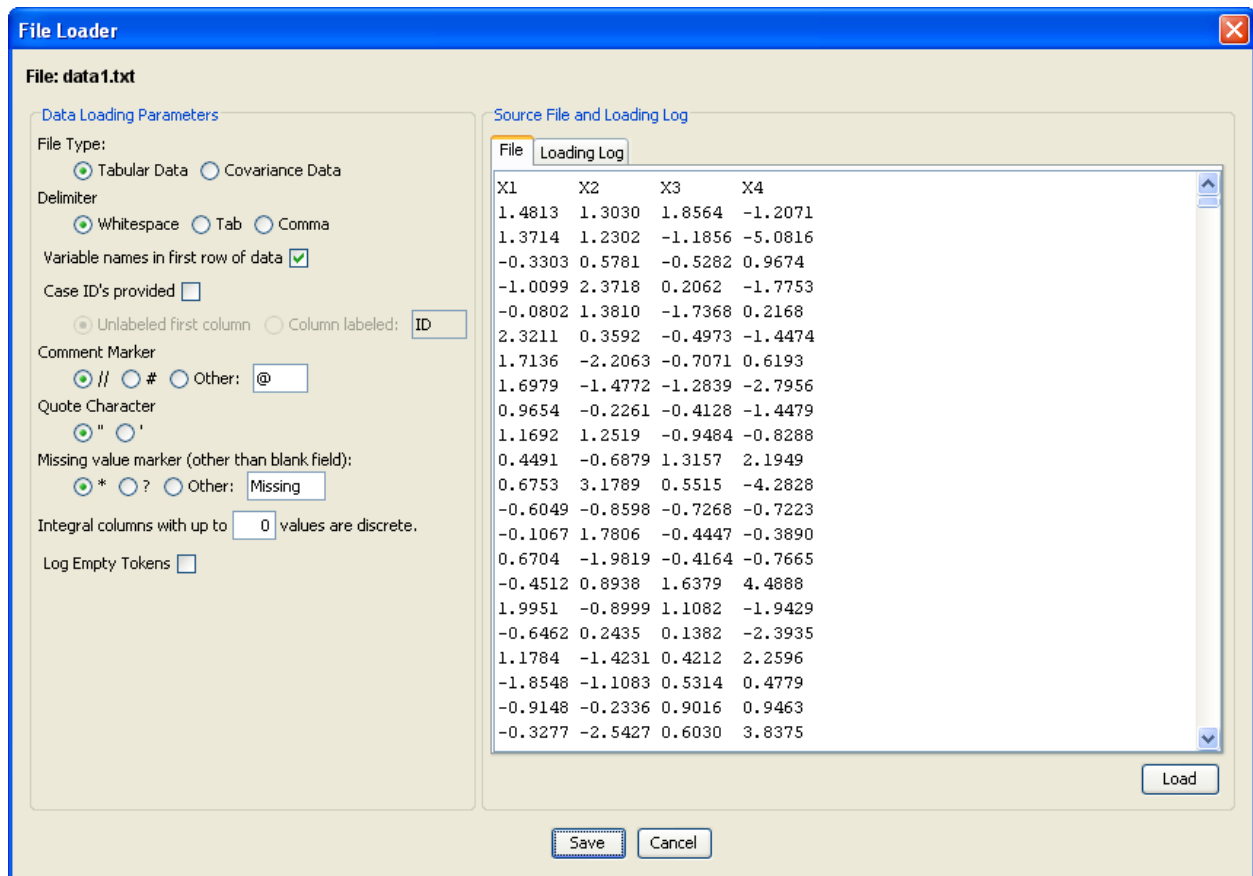
| X1      | X2      | X3      | X4      | X5      |
|---------|---------|---------|---------|---------|
| -3.0133 | 1.0361  | 0.2329  | 2.7829  | -0.2878 |
| 0.5542  | 0.3661  | 0.2480  | 1.6881  | 0.0775  |
| 3.5579  | -0.7431 | -0.5960 | -2.5502 | 1.5641  |
| -0.0858 | 1.0400  | -0.8255 | 0.3021  | 0.2654  |
| -0.9666 | -0.5873 | -0.6350 | -0.1248 | 1.1684  |
| -1.7821 | 1.8063  | -0.9814 | 1.8505  | -0.7537 |
| -0.8162 | -0.6715 | 0.3339  | 2.6631  | 0.9014  |
| -0.3150 | -0.5103 | -2.2830 | -1.2462 | -1.2765 |
| -4.1204 | 2.9980  | -0.3609 | 4.8079  | 0.6005  |
| 1.4658  | -1.4069 | 1.7234  | -1.7129 | -3.8298 |

#### Loading Data:

If you have preexisting data sets that you would like to manipulate or run searches on, you can load them into a data box. In this case, your data box should have no input. When you double click on it, a window will appear asking which type of data object you would like to instantiate. Choose "Data Wrapper." An empty data window will appear.



Click File: Load Data... and select the text file that contains your data. The following window will appear:



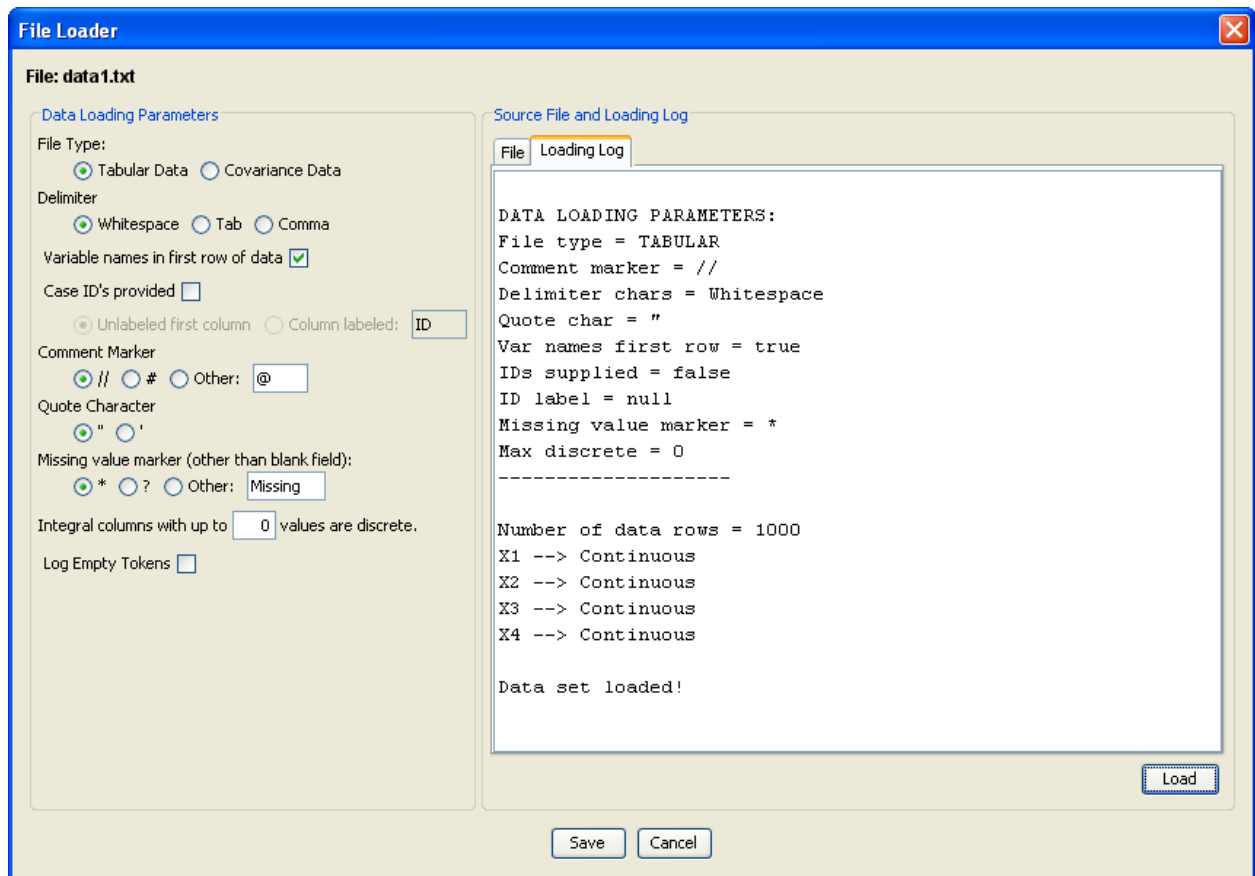
The text of the source file appears on the right. On the left, you can tell Tetrad what your file looks like, so that it can correctly load data. If you are loading tabular data values, select the “Tabular Data” button. If you are loading a covariance matrix, select “Covariance Data.” Note that if you are loading a covariance matrix, your text file should contain only the lower half of the matrix, as Tetrad will not accept an entire matrix.

Below the file type, you can specify the delimiter between data values. If you do not list the variable names in the file, you should uncheck “Variable names in first row of data.” If you provide case IDs, check the “Case IDs provided” box. If the case ID column is labeled, provide the name of the label; otherwise, the case ID column should be the first column, and you should check “unlabeled first column.”

Below this, you can specify your comment markers, quote characters, and the character which marks missing data values. You can also discretize columns. If you put a number in “Integral columns with up to \_\_\_ data values are discrete,” then Tetrad will treat variables which take that number or fewer values as discrete variables.

If you check “Log Empty Tokens,” then when you manipulate data, if you have logging turned on, Tetrad will alert you to the placement of missing data values.

Before you click “Save,” you must click “Load,” below the window which shows your source file. This will provide you with statistics on your data:



You can now click “Save,” and your data set will appear in the window.

The screenshot shows a window titled "Data1 (Data Wrapper)" with a menu bar containing "File", "Edit", and "Tools". Below the menu bar is a tab labeled "data1.txt". The main area contains a data table with the following structure:

|    |      | C1      | C2      | C3      | C4      | C5 | C6 | C7 |
|----|------|---------|---------|---------|---------|----|----|----|
|    | MULT | X1      | X2      | X3      | X4      |    |    |    |
| 1  | 1    | 1.4813  | 1.3030  | 1.8564  | -1.2071 |    |    |    |
| 2  | 1    | 1.3714  | 1.2302  | -1.1856 | -5.0816 |    |    |    |
| 3  | 1    | -0.3303 | 0.5781  | -0.5282 | 0.9674  |    |    |    |
| 4  | 1    | -1.0099 | 2.3718  | 0.2062  | -1.7753 |    |    |    |
| 5  | 1    | -0.0802 | 1.3810  | -1.7368 | 0.2168  |    |    |    |
| 6  | 1    | 2.3211  | 0.3592  | -0.4973 | -1.4474 |    |    |    |
| 7  | 1    | 1.7136  | -2.2063 | -0.7071 | 0.6193  |    |    |    |
| 8  | 1    | 1.6979  | -1.4772 | -1.2839 | -2.7956 |    |    |    |
| 9  | 1    | 0.9654  | -0.2261 | -0.4128 | -1.4479 |    |    |    |
| 10 | 1    | 1.1692  | 1.2519  | -0.9484 | -0.8288 |    |    |    |
| 11 | 1    | 0.4491  | -0.6879 | 1.3157  | 2.1949  |    |    |    |
| 12 | 1    | 0.6753  | 3.1789  | 0.5515  | -4.2828 |    |    |    |
| 13 | 1    | -0.6049 | -0.8598 | -0.7268 | -0.7223 |    |    |    |
| 14 | 1    | -0.1067 | 1.7806  | -0.4447 | -0.3890 |    |    |    |
| 15 | 1    | 0.6704  | -1.9819 | -0.4164 | -0.7665 |    |    |    |
| 16 | 1    | -0.4512 | 0.8938  | 1.6379  | 4.4888  |    |    |    |
| 17 | 1    | 1.9951  | -0.8999 | 1.1082  | -1.9429 |    |    |    |

At the bottom of the window, there are "Save" and "Cancel" buttons.

You can now save this data set to a text file by clicking File: Save Data.

It is possible to load multiple data sets into the same data box. In order to do so, you must load all of the data sets at the same time. When the window opens asking you for the name of the file with your data in it, select all of the data sets you would like to load, by holding down the shift key. (Your data sets must all be in the same directory in order for you to do this.) When the file loader window opens, you will be able to set specifications for each data set individually using the “Previous” and “Next” buttons at the bottom of the window, and you will be able to load each set individually or all at the same time. If you save the window before loading every data set, any data set you did not load will not be included in the data box.

In addition to loading data from a file, you can manually enter data values and variable names by overwriting cells in the data table.

### Manipulating Data:

Under the Edit tab, there are several options to manipulate data. If you select a number of cells and click “Clear Cells,” Tetrad will replace the data values in the selected cells with a missing data marker. If you select an entire row or column and click “Delete selected rows or

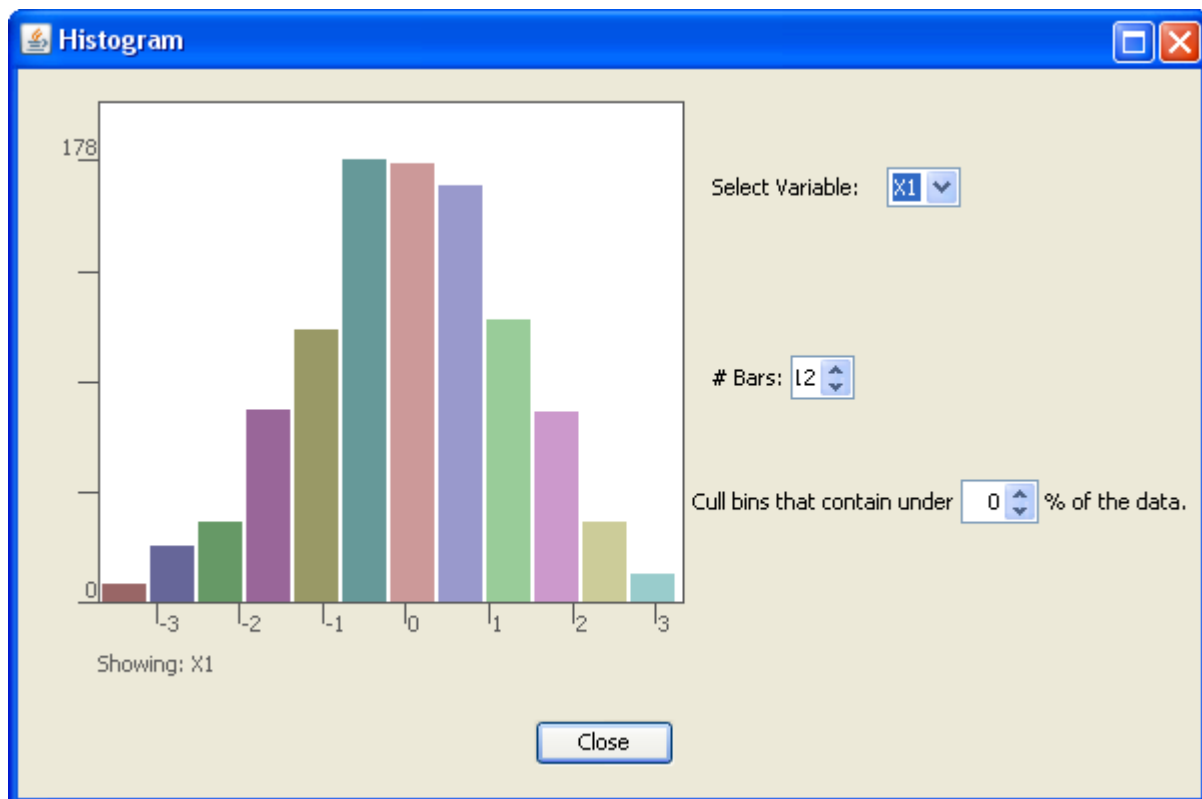
columns,” Tetrad will delete all data values in the row or column, and the name of the row or column. (To select an entire column, click on the category number above it, labeled C1, C2, C3, and so on. To select an entire row, click on the row number to the left of it, labeled 1, 2, 3, and so on.) You can also copy, cut, and paste data values to and from selected cells. You can choose to show or hide category names, and if you click on “Set Constants Col to Missing,” then in any column in which the variable takes on only one value (for example, a column in which every cell contains the number 2) Tetrad will set every cell to the missing data marker.

Under the Tools tab, the Calculator tool allows you add and edit relationships between variables in the graph. For more information on how the Calculator tool works, see the section on the data manipulation box.

### Data Information:

Under the Tools tab, there are options to view information about your data in several different formats.

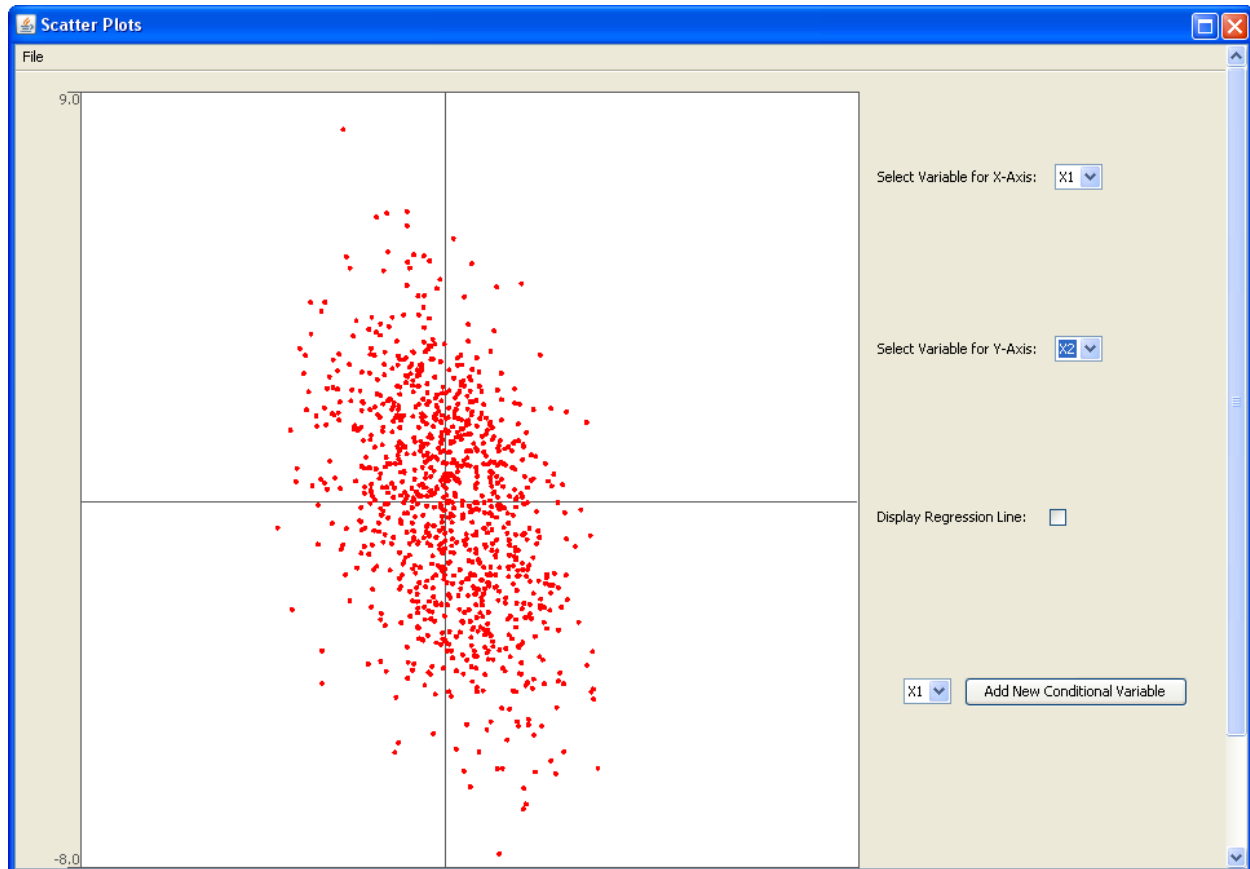
The Histograms tool shows histograms of the variables in the data set.



These show the distribution of data for each variable, with the width of each bar representing a range of values, and height of each bar representing how many data points fall into that range. Using histograms, you can determine whether each variable has a distribution that is approximately Normal. To select a variable to view, choose it from the drop-down menu on the right. You can increase or decrease the number of bars in the histogram (and therefore

decrease or increase the range of each bar, and increase or decrease the accuracy of the histogram) using the menu on the right. You can also view only ranges with a certain amount of the data using the “cull bins” menu.

The Scatter Plots tool allows you to view scatter plots of two variables plotted against each other.

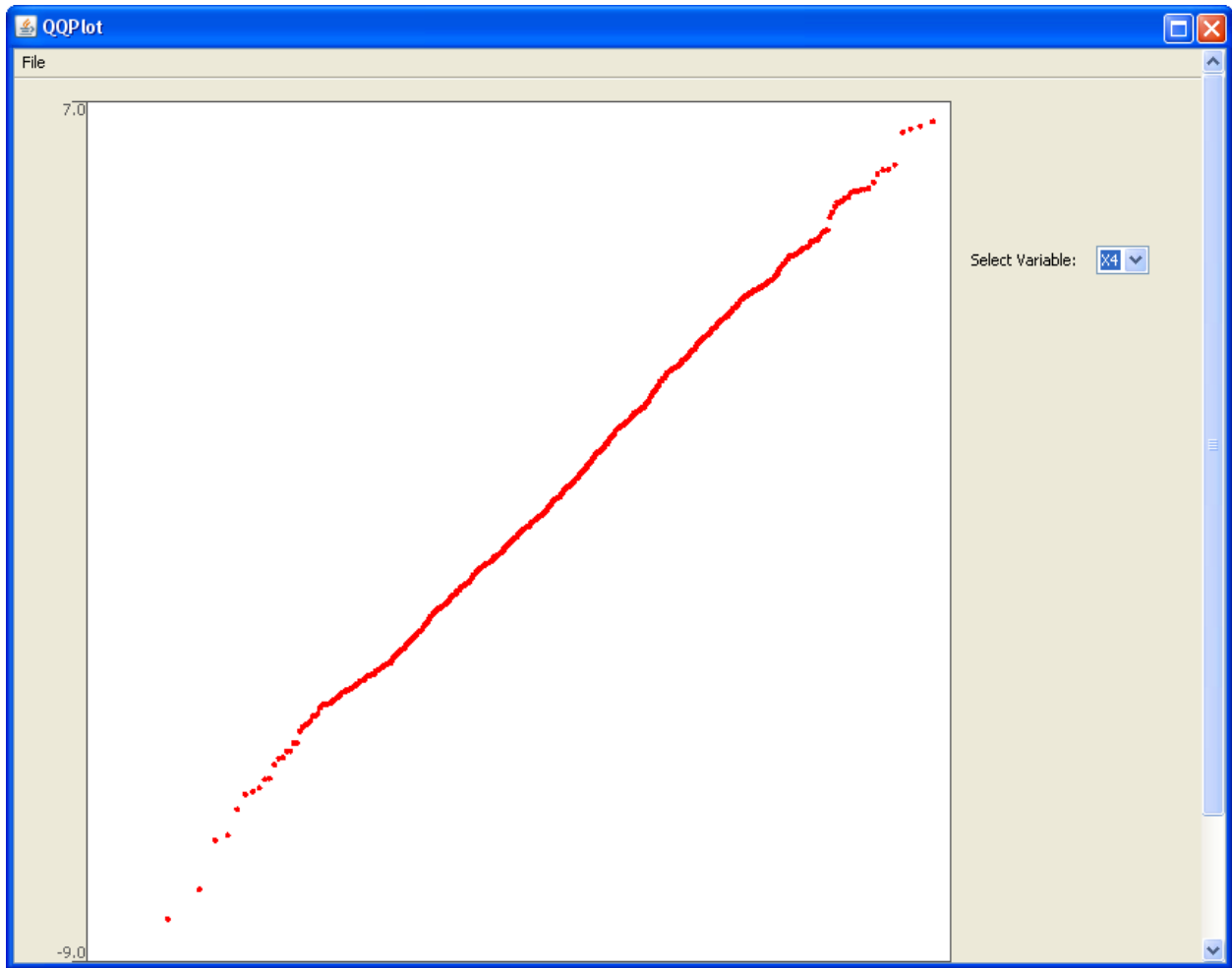


To view a variable as the x- or y-axis of the graph, select it from one of the drop-down menus to the right. To view the regression line of the graph, check the box on the right.

You can see the correlation of two variables conditional on a third variable by using the Add New Conditional Variable button at the bottom of the window. This will open up a slider and a box in which you can set the granularity of the slider. By moving the slider to the left or right, you can change the range of values of the conditional variable for which the scatter plot shows the correlation of the variables on the x- and y- axes. You can increase and decrease the width of the ranges by changing the granularity of the slider. A slider with granularity 1 will break the values of the conditional variable into sections one unit long, etc. The granularity cannot be set lower than one.

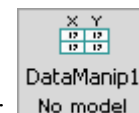
In a well-formed model, the scatter plot of a variable plotted against itself should appear as a straight line along the line  $y = x$ .

The Q-Q Plot tool is a test for normality of distribution.



If a variable has a distribution which is approximately Normal, its Q-Q plot should appear as a straight line with a positive slope. You can select the variable whose Q-Q plot you wish to view from the drop-down menu on the right.

The Normality Tests tool gives a text box with the results of the Kolmogorov and Anderson Darling Tests for normality for each variable. The Descriptive Statistics tool gives a text box with statistical information such as the mean, median, and variance of each variable.



The data manipulation box in the main workspace looks like this:

**Possible Parent Boxes of the Data Manipulation Box:**

- A graph box
- A graph manipulation box
- A parametric model box



- An instantiated model box
- A data box
- Another data manipulation box
- An estimator box
- A search box
- A regression box

### **Possible Child Boxes of the Data Manipulation Box:**

- A graph box
- A comparison box
- A parametric model box
- An instantiated model box
- A data box
- Another data manipulation box
- An estimator box
- A classify box
- A knowledge box
- A search box
- A regression box

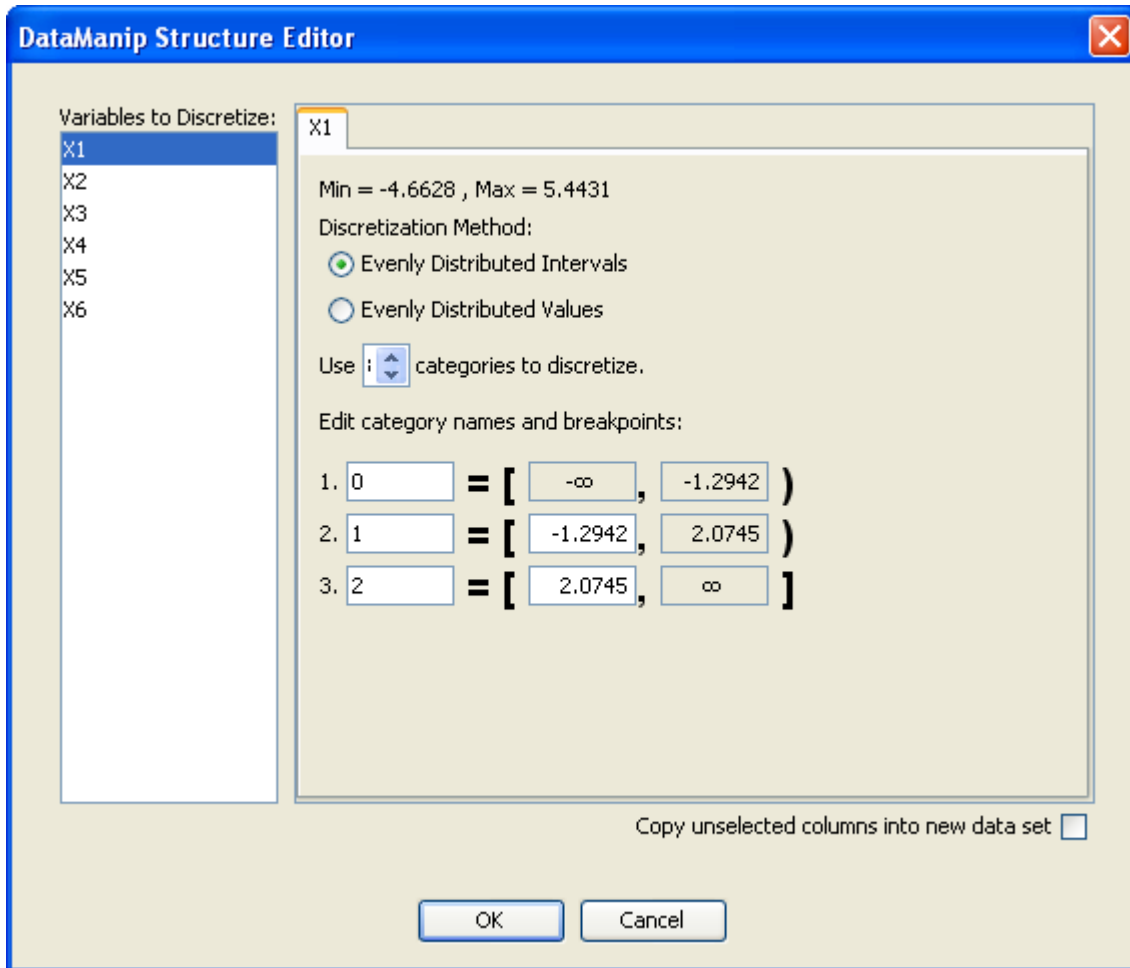
### **Using the Data Manipulation Box:**

The data manipulation box takes as input a data box, and allows you to easily modify its data sets in various ways. When you double click on the data manipulation box for the first time, a window opens up allowing you to choose how you would like to manipulate your data. The possible operations are split into several categories: general operations, conversion operations, missing value operations, row operations, and column operations. You can only perform one operation in each data manipulation box unless you destroy the contents of the current box (see the General Procedures chapter).

### **General Operations:**

#### **Discretize Dataset**

This operation allows you to make some or all variables in a data set discrete. If you choose it, a window will open.



When the window first opens, no variables are selected, and the right side of the window appears blank; in this case, we have already selected X1 ourselves. In order to discretize a variable, Tetrad assigns all data points within a certain range to a category. You can tell Tetrad to break the range of the dataset into approximately even sections (Evenly Distributed Intervals) or to break the data points themselves into approximately even chunks (Evenly Distributed Values). Use the scrolling menu to increase or decrease the number of categories to create. You can also rename categories by overwriting the text boxes on the left, or change the ranges of the categories by overwriting the text boxes on the right. To discretize another variable, simply select it from the left. If you want your new data set to include the variables you did not discretize, check the box at the bottom of the window.

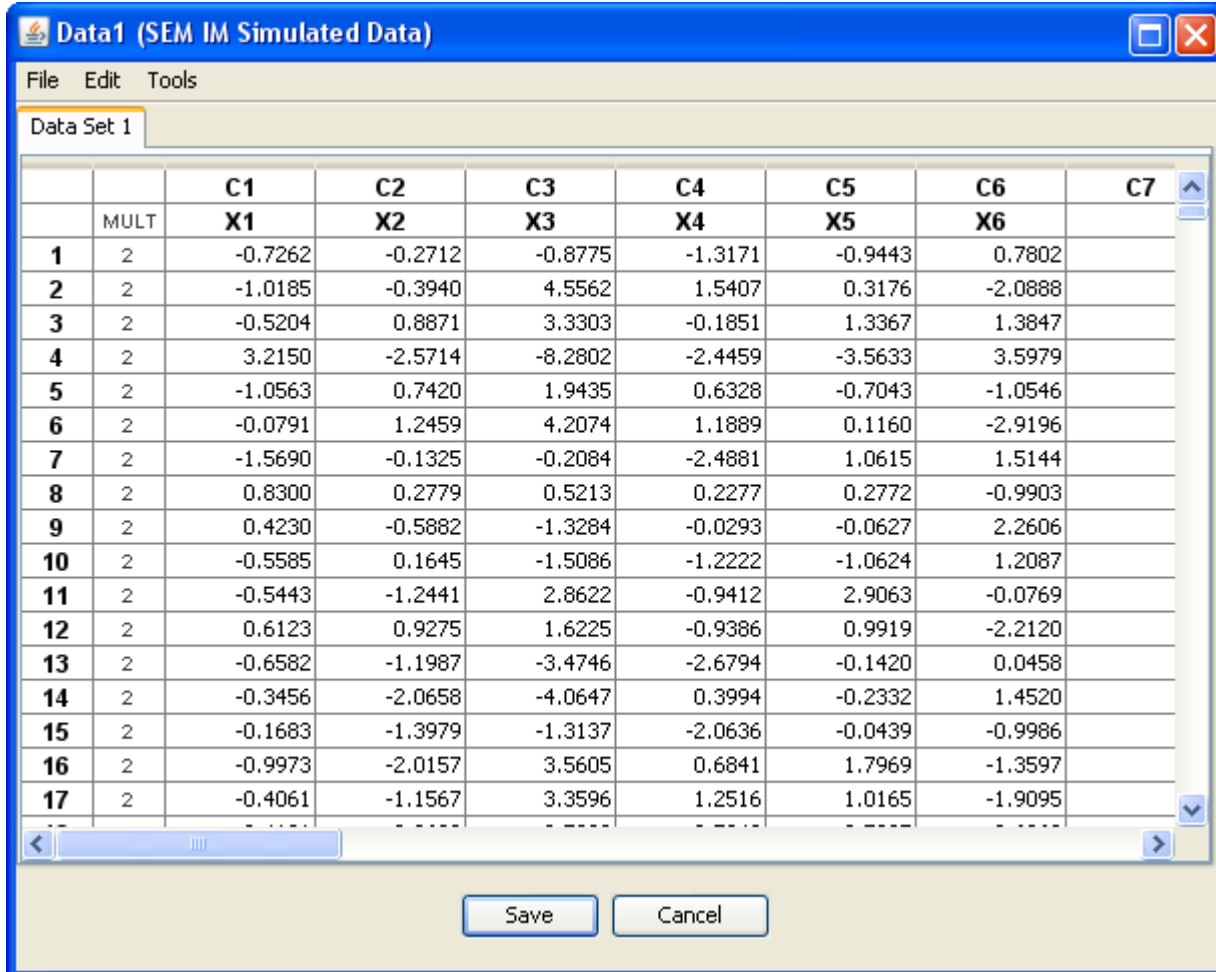
You may discretize multiple variables at once by selecting multiple variables. In this case, the ranges are not shown, as they will be different from variable to variable.

### Convert Numerical Discrete to Continuous

If you choose this option, any discrete variables with numerical category values will be treated as continuous variables with real values. For example, “1” will be converted to “1.0.”

## Expand Case Multipliers

A case multiplier is a number to the left of a row which indicates how many instances of that particular configuration of variables occur.



|    |      | C1      | C2      | C3      | C4      | C5      | C6      | C7 |
|----|------|---------|---------|---------|---------|---------|---------|----|
|    | MULT | X1      | X2      | X3      | X4      | X5      | X6      |    |
| 1  | 2    | -0.7262 | -0.2712 | -0.8775 | -1.3171 | -0.9443 | 0.7802  |    |
| 2  | 2    | -1.0185 | -0.3940 | 4.5562  | 1.5407  | 0.3176  | -2.0888 |    |
| 3  | 2    | -0.5204 | 0.8871  | 3.3303  | -0.1851 | 1.3367  | 1.3847  |    |
| 4  | 2    | 3.2150  | -2.5714 | -8.2802 | -2.4459 | -3.5633 | 3.5979  |    |
| 5  | 2    | -1.0563 | 0.7420  | 1.9435  | 0.6328  | -0.7043 | -1.0546 |    |
| 6  | 2    | -0.0791 | 1.2459  | 4.2074  | 1.1889  | 0.1160  | -2.9196 |    |
| 7  | 2    | -1.5690 | -0.1325 | -0.2084 | -2.4881 | 1.0615  | 1.5144  |    |
| 8  | 2    | 0.8300  | 0.2779  | 0.5213  | 0.2277  | 0.2772  | -0.9903 |    |
| 9  | 2    | 0.4230  | -0.5882 | -1.3284 | -0.0293 | -0.0627 | 2.2606  |    |
| 10 | 2    | -0.5585 | 0.1645  | -1.5086 | -1.2222 | -1.0624 | 1.2087  |    |
| 11 | 2    | -0.5443 | -1.2441 | 2.8622  | -0.9412 | 2.9063  | -0.0769 |    |
| 12 | 2    | 0.6123  | 0.9275  | 1.6225  | -0.9386 | 0.9919  | -2.2120 |    |
| 13 | 2    | -0.6582 | -1.1987 | -3.4746 | -2.6794 | -0.1420 | 0.0458  |    |
| 14 | 2    | -0.3456 | -2.0658 | -4.0647 | 0.3994  | -0.2332 | 1.4520  |    |
| 15 | 2    | -0.1683 | -1.3979 | -1.3137 | -2.0636 | -0.0439 | -0.9986 |    |
| 16 | 2    | -0.9973 | -2.0157 | 3.5605  | 0.6841  | 1.7969  | -1.3597 |    |
| 17 | 2    | -0.4061 | -1.1567 | 3.3596  | 1.2516  | 1.0165  | -1.9095 |    |

In the above example, the case multipliers are listed in the column labeled “mult.” The first row has a case multiplier of 2, so there are two instances of that configuration of variables, but only one case is shown. The Expand Case Multipliers option lists each case as its own row, with a case multiplier of 1. The expanded version of the above data set looks like this:

**DataManip1 (Expand Case Multipliers)**

File Edit Tools

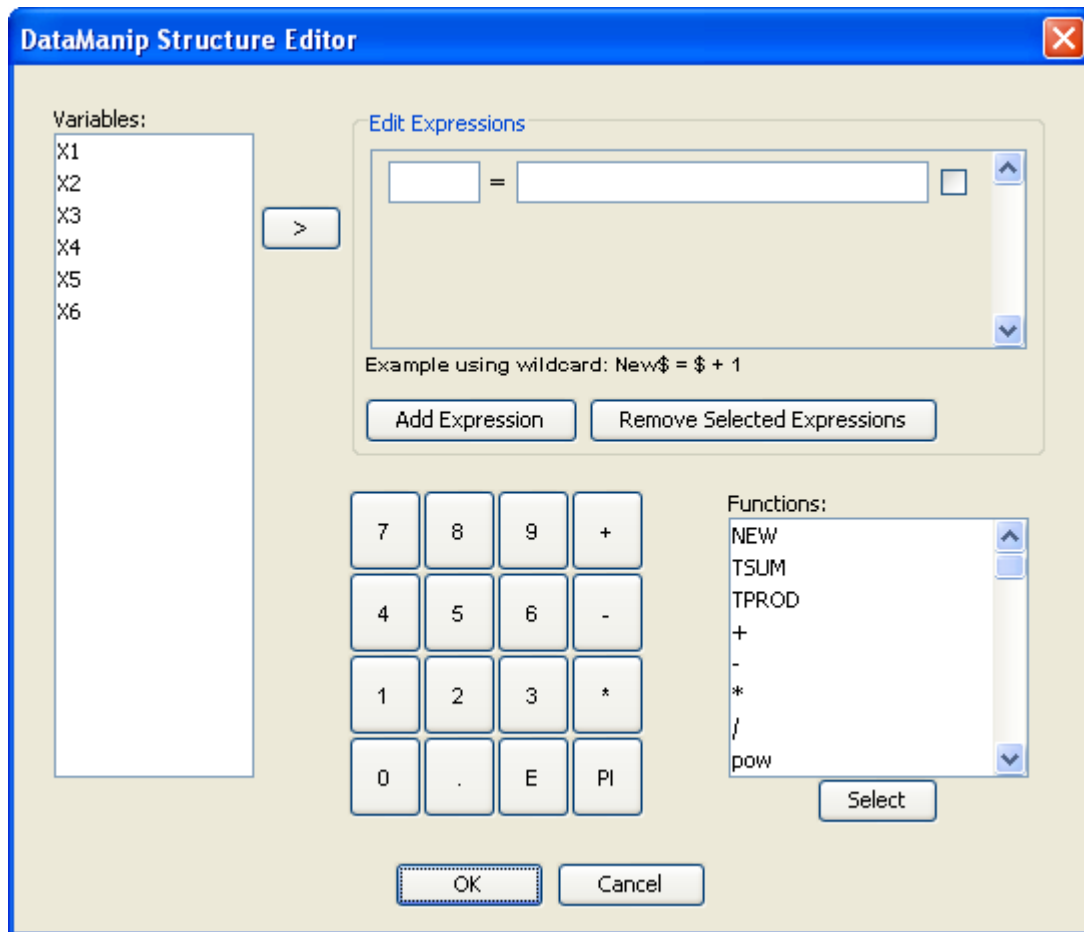
Data Set 1

|    |      | C1      | C2      | C3      | C4      | C5      | C6      | C7 |
|----|------|---------|---------|---------|---------|---------|---------|----|
|    | MULT | X1      | X2      | X3      | X4      | X5      | X6      |    |
| 1  | 1    | -0.7262 | -0.2712 | -0.8775 | -1.3171 | -0.9443 | 0.7802  |    |
| 2  | 1    | -0.7262 | -0.2712 | -0.8775 | -1.3171 | -0.9443 | 0.7802  |    |
| 3  | 1    | -1.0185 | -0.3940 | 4.5562  | 1.5407  | 0.3176  | -2.0888 |    |
| 4  | 1    | -1.0185 | -0.3940 | 4.5562  | 1.5407  | 0.3176  | -2.0888 |    |
| 5  | 1    | -0.5204 | 0.8871  | 3.3303  | -0.1851 | 1.3367  | 1.3847  |    |
| 6  | 1    | -0.5204 | 0.8871  | 3.3303  | -0.1851 | 1.3367  | 1.3847  |    |
| 7  | 1    | 3.2150  | -2.5714 | -8.2802 | -2.4459 | -3.5633 | 3.5979  |    |
| 8  | 1    | 3.2150  | -2.5714 | -8.2802 | -2.4459 | -3.5633 | 3.5979  |    |
| 9  | 1    | -1.0563 | 0.7420  | 1.9435  | 0.6328  | -0.7043 | -1.0546 |    |
| 10 | 1    | -1.0563 | 0.7420  | 1.9435  | 0.6328  | -0.7043 | -1.0546 |    |
| 11 | 1    | -0.0791 | 1.2459  | 4.2074  | 1.1889  | 0.1160  | -2.9196 |    |
| 12 | 1    | -0.0791 | 1.2459  | 4.2074  | 1.1889  | 0.1160  | -2.9196 |    |
| 13 | 1    | -1.5690 | -0.1325 | -0.2084 | -2.4881 | 1.0615  | 1.5144  |    |
| 14 | 1    | -1.5690 | -0.1325 | -0.2084 | -2.4881 | 1.0615  | 1.5144  |    |
| 15 | 1    | 0.8300  | 0.2779  | 0.5213  | 0.2277  | 0.2772  | -0.9903 |    |
| 16 | 1    | 0.8300  | 0.2779  | 0.5213  | 0.2277  | 0.2772  | -0.9903 |    |
| 17 | 1    | 0.4230  | -0.5882 | -1.3284 | -0.0293 | -0.0627 | 2.2606  |    |

Save Cancel

### Calculate

The Calculate option allows you to add and edit relationships between variables in your data set, and to add new variables to the data set.



In many ways, this tool works like the Edit Expression window in a generalized SEM parametric model. To edit the formula that defines a variable (which will change that variable's values in the table) type that variable name into the text box to the left of the equals sign. To create a new variable, type a name for that variable into the text box to the left of the equals sign. Then, in the box on the right, write the formula by which you wish to define a new variable in place of, or in addition to, the old variable. You can select functions from the scrolling menu below. For an explanation of the meaning of some the functions, see the section on generalized SEM models in the Parametric Model Box chapter. To edit or create several formulae at once, click the "Add Expression" button, and another blank formula will appear. To delete a formula, check the box next to it and click the "Remove Selected Expressions" button.

When you click "Save" a table will appear listing the data. Values of variables whose formulae you changed will be changed, and any new variables you created will appear with defined values.

### Shift Data

This operation shifts the placement of columns in a (usually time lagged) data set up or down a number of rows. You specify which columns are shifted how much in which direction. This operation may cause you to lose values which would make the data set uneven.

In the window which allows you to choose which columns to shift, there is also a Search tab. You can use this tab on a data set which has already been shifted to search for ways to unshift it.

## **Conversion Operations**

### Convert to Correlation Matrix

This operation takes a tabular data set and outputs the lower half of the correlation matrix of that data set.

### Convert to Covariance Matrix

This operation takes a tabular data set and outputs the lower half of the covariance matrix of that data set.

### Convert to Time Lag Data

This operation takes a tabular data set and outputs a time lag data set, in which each variable is recorded several times over the course of an experiment. You can specify the number of lags in the data. Each contains the same data, shifted by one “time unit.”. For instance, if the original data set had 1000 cases, and you specify that the time lag data set should contain two lags, then the third stage variable values will be those of cases 1 to 998, the second stage variable values will be those of cases 2 to 999, and the first stage variable values will be those of cases 3 to 1000.

### Convert to AR Residuals

This operation is performed on a time lag data set. Tetrad performs a linear regression on each variable in each lag with respect to each of the variables in the previous lag, and derives the error terms. The output data set contains only the error terms.

### Convert to Residuals

The input for this operation is a directed acyclic graph (DAG) and a data set. Tetrad performs a linear regression on each variable in the data set with respect to all of the variables which the graph shows to be its parents, and derives the error terms. The output data set contains only the error terms.

### Convert to Standardized Form

This operation manipulates the data in your data set such that each variable has 0 mean and unit variance. The values of your data will change if you choose this option.

## **Missing Value Operations**

### Remove Cases with Missing Values

If you choose this operation, Tetrad will remove any row in which one or more of the values is missing.

#### Replace Missing Values with Column Mode

If you choose this operation, Tetrad will replace any missing value markers with the most commonly used value in the column.

#### Replace Missing Values with Column Mean

If you choose this operation, Tetrad will replace any missing value markers with the average of all of the values in the column.

#### Replace Missing Values with Regression Predictions

If you choose this operation, Tetrad will perform a linear regression on the data in order to estimate the most likely value of any missing value.

#### Replace Missing Values by Extra Category

This operation takes as input a discrete data set. For every variable which has missing values, Tetrad will create an extra category for that variable (named by default “Missing”) and replace any missing data markers with that category.

#### Inject Missing Data Randomly

If you choose this operation, Tetrad will replace randomly selected data values with a missing data marker. You can set the probability with which any particular value will be replaced (that is, approximately the percentage of values for each variable which will be replaced with missing data markers).

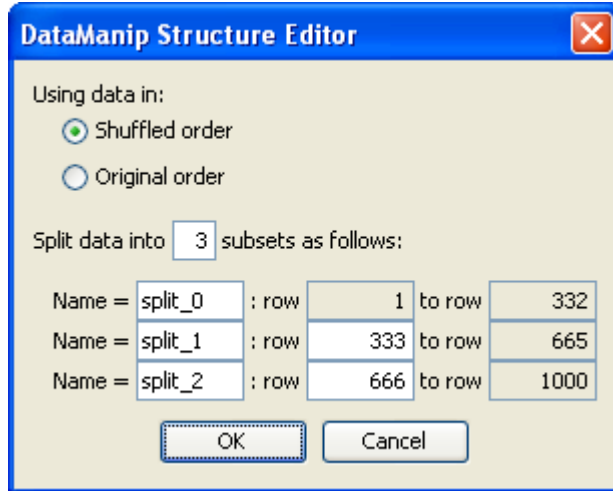
### **Row Operations**

#### Bootstrap Sample

This operation draws a random subset of the input data set (you specify the size of the subset) with replacement (that is, cases which appear once in the original data set can appear multiple times in the subset). The resulting data set can be used along with similar subsets to achieve more accurate estimates of parameters.

#### Split by Cases

This operation allows you to split a data set into several smaller data sets. When you choose it, a window opens.



If you would like the subsets to retain the ordering they had in the original set, click “Original Order.” Otherwise, the ordering of the subsets will be assigned at random. You can also increase and decrease the number of subsets created, and specify the range of each subset.

### Permute Rows

This operation randomly reassigns the ordering of a data set’s cases.

## **Column Operations**

### Copy Continuous Variables

This operation takes as input a data set and creates a new data set containing only the continuous variables present in the original.

### Copy Discrete Variables

This operation takes as input a data set and creates a new data set containing only the discrete variables present in the original.

### Copy Selected Variables

As explained above, you can select an entire column in a data set by clicking on the C1, C2, C3, etc... cell above the column. To select multiple columns, press and hold the “control” key while clicking on the cells. Once you have done so, you can use the Copy Selected Variables tool to create a data set in which only those columns appear.

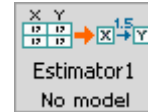
### Remove Constant Columns

This operation takes a data set as input, and creates a data set which contains all columns in the original data set except for those with constant values (such as, for example, a column containing nothing but 2’s).

### Restrict to Graph Nodes

This operation takes as input a data set and a graph. It outputs a data set containing only those variables which appear in both the data set and the graph.





The estimator box in the main workspace looks like this:

### **Possible Parent Boxes of the Estimator Box:**

- A parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- Another estimator box

### **Possible Child Boxes of the Estimator Box:**

- A graph box
- A graph manipulation box
- A comparison box
- A parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- Another estimator box
- An updater box
- A search box

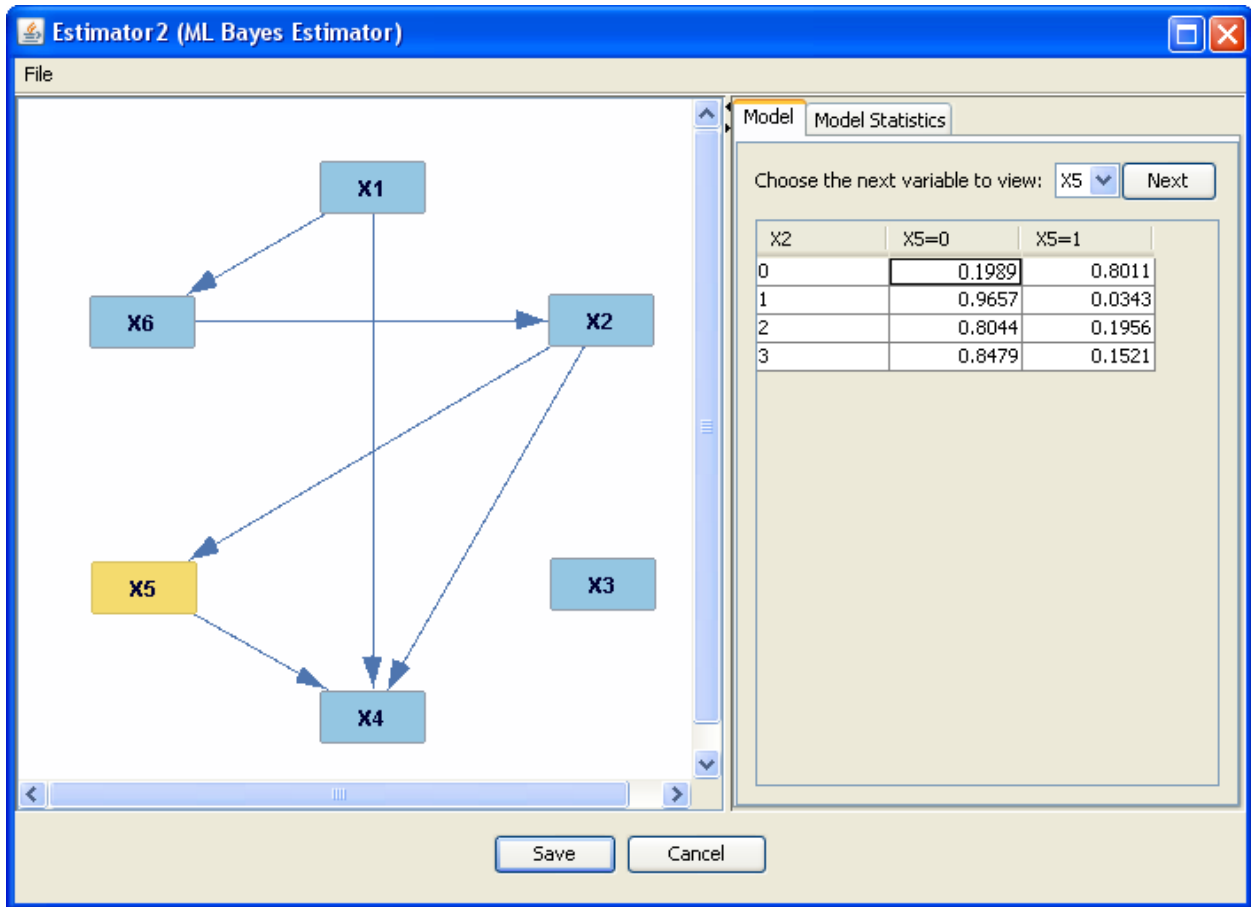
### **Using the Estimator Box:**

The estimator box takes as input a data box and a parametric model box. Using these inputs, Tetrad estimates, tests, and outputs an instantiated model for the data. In general, the estimator box works much like the instantiated model box, with a few additional functions. Tetrad is capable of performing Maximum Likelihood (ML) Bayes estimations, Dirichlet estimations, SEM estimations, and Expectation Maximization (EM) Bayes estimations. Tetrad estimators, other than the EM Bayes estimator, are designed not to accept missing values. If your data set contains missing values, the missing values can be interpolated or removed using the data manipulation box. (Note that missing values are allowed in various Tetrad search procedures; see the section on the search box.)

### **ML Bayes Estimations:**

Bayes nets are acyclic graphical models parameterized by the conditional probability distribution of each variable on its parents' values, as in the instantiated model box. When the model contains no unrecorded variables, the ML estimate of each model parameter is just the corresponding conditional frequency from the data.

The ML Bayes estimator, because it estimates Bayes IMs, works only on models with discrete variables. The model estimated must not include latent variables, and the input data set must not include missing data values. A sample estimate looks like this:



The Model tab works exactly as it does in a Bayes instantiated model. The Model Statistics tab provides the p-value for a chi square test of the model, degrees of freedom, the chi square value, and the Bayes Information Criterion (BIC) score of the model.

### Dirichlet Estimations:

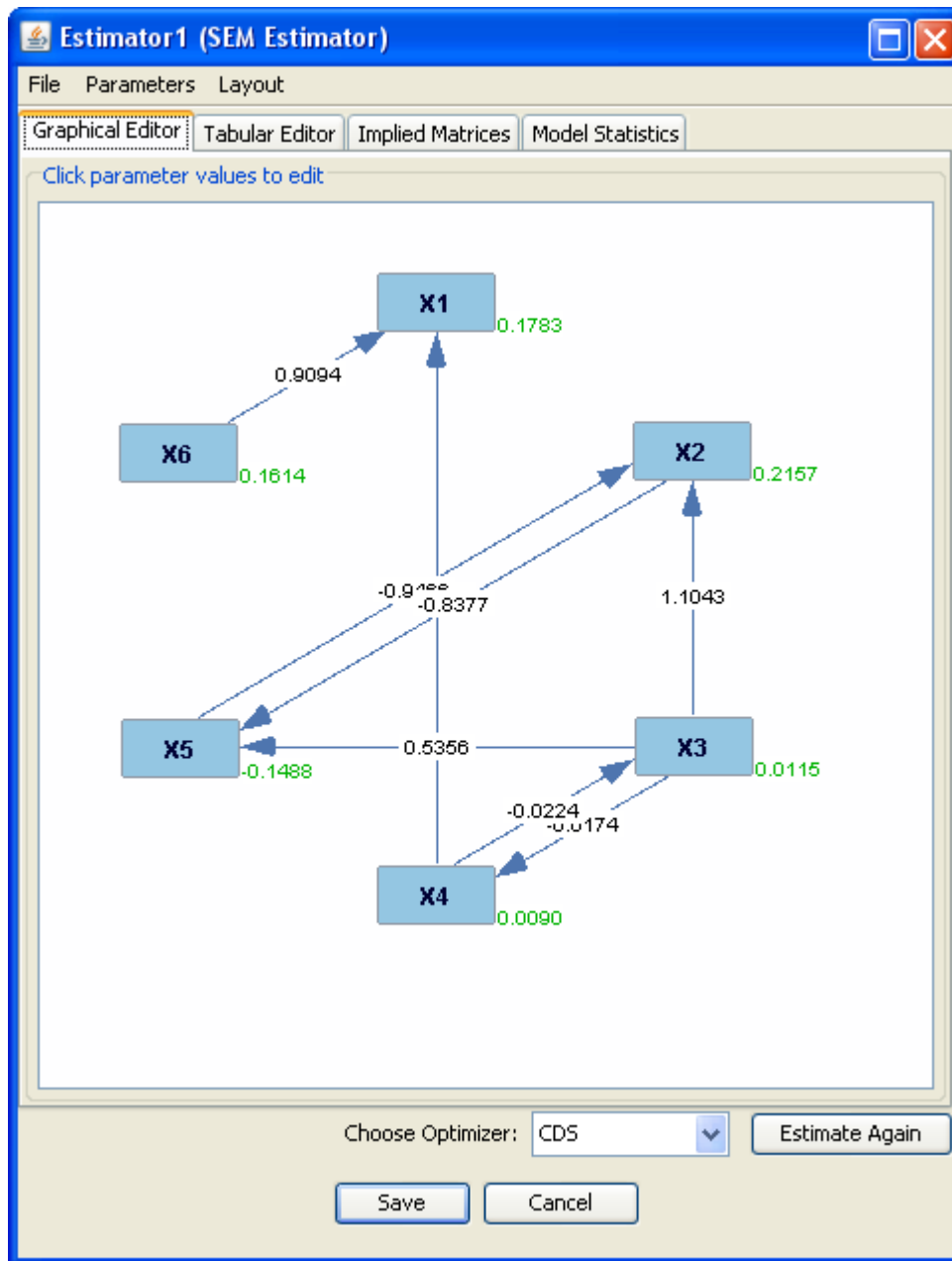
A Dirichlet estimate estimates a Bayes instantiated model using a Dirichlet distribution for each category. In a Dirichlet estimate, the probability of each value of a variable (conditional on the values of the variable's parents) is estimated by adding to a prior pseudo count (which is 1, by default) of cases, for each configuration, the number of cases in which the variable takes that value and then dividing by the total number of cases in the prior and in the data with that configuration of parent variables. The default prior pseudo-count can be changed inside the box. (For a full explanation of pseudocounts and Dirichlet estimate, see the section on Dirichlet instantiated models.)

Because of the way Dirichlet estimates work, a Dirichlet estimate can be used as input to another Dirichlet estimate of the same parameters (along with more data) to gain more accurate results.

Dirichlet estimates do not work if the input data set contains missing data values.

### SEM Estimates:

A SEM estimator estimates the values of parameters for a SEM parametric model. SEM estimates do not work if the input data set contains missing data values. The output of a SEM estimation looks like this:



Tetrad has estimated the values of the parameters. Tetrad uses four optimizers to search the configurations of parameters: CDS, EM, regression, and random search. Accurate regression estimates presuppose that the input parametric model be a DAG, and its associated statistics are based on a linear, Gaussian model. The EM optimizer has the same input constraints as regression, but can handle latent variables. The CDS optimizer can handle cycles, bidirected edges, and latent variables. You can change which optimizer Tetrad uses by choosing it from the drop-down menu at the bottom of the window and clicking “Estimate Again.”

If the graph for the SEM is a DAG, and we may assume that the SEM is linear with Gaussian error terms, we use multilinear regression to estimate coefficients and residual variances. Otherwise, we use a standard maximum likelihood fitting function (see Bollen,

*Structural Equations with Latent Variables*, Wiley, 1989, pg. 107) to minimize the distance between (a) the covariance over the variables as implied by the coefficient and error covariance parameter values of the model and (b) the sample covariance matrix. Following Bollen, we denote this function Fml; it maps points in parameter values space to real numbers, and, when minimized, yields the maximum likelihood estimation point in parameter space.

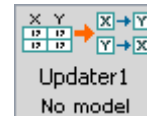
In either case, an Fml value may be obtained for the maximum likelihood point in parameter space, either by regression or by direct minimization of the Fml function itself. The value of Fml at this minimum (maximum likelihood) point, multiplied by  $N - 1$  (where  $N$  is the sample size), yields a chi square statistics ( $ch^2$ ) for the model, which when referred to the chi square table with appropriate degrees of freedom, yields a model p value. The degrees of freedom (dof) in this case is equal to the  $m(m-1)/2 - f$ , where  $m$  is the number of measured variables, and  $f$  is the number of free parameters, equal to the number of coefficient parameters plus the number of covariance parameters. (Note that the degrees of freedom may be negative, in which case estimation should not be done.) The BIC score is calculated as  $ch^2 - dof * \log(N)$ .

The Tabular Editor and Implied Matrices tabs function exactly as they do in the instantiated model box, but in the estimator box, the last three columns of the table in the Tabular Editor tab are filled in. The SE, T, and P columns provide the standard errors, t statistics, and p values of the estimation.

The Model Statistics tab provides the degrees of freedom, chi square, p value, and BIC score of a test of the model. It should be noted that while these test statistics are standard, they are not in general correct. See Mathias Drton, 2009, Likelihood ratio tests and singularities. *Annals of Statistics* 37(2):979-1012. arXiv:math.ST/0703360.

EM Bayes Estimations:

The EM Bayes estimator takes the same input and gives the same output as the ML Bayes estimator, but is designed to handle data sets with missing data values, and the input model can contain latent variables.



The updater box in the main workspace looks like this:

**Possible Parent Boxes of the Updater Box:**

- An instantiated model box
- An estimator box

### **Possible Child Boxes of the Updater Box:**

- A graph box
- A comparison box
- An instantiated model box
- A data box

### **Using the Updater Box:**

The updater box takes an instantiated model as input, and, given information about the values of parameters in that model, updates the information about the values and relationships of other parameters. There are four available updater algorithms in Tetrad: the approximate updater, the row summing exact updater, the CPT invariant updater, and the SEM updater. All except for the SEM updater function only when given Bayes instantiated models as input; the SEM updater functions when given a SEM instantiated model as input. None of the updaters work on cyclic models.

### **Approximate Updater**

The approximated updater is a fast but inexact algorithm. It randomly draws a sample data set of [size] from the instantiated model and calculates the conditional frequency of the variable to be estimated.

Take, for example, the following instantiated model:

IM1 (Bayes Instantiated Model)

File

```

    graph TD
      X6 --> X2
      X5 --> X2
      X4 --> X2
      X4 --> X3
      X2 --> X1
      X2 --> X3
  
```

1. Choose the next variable to edit: X1

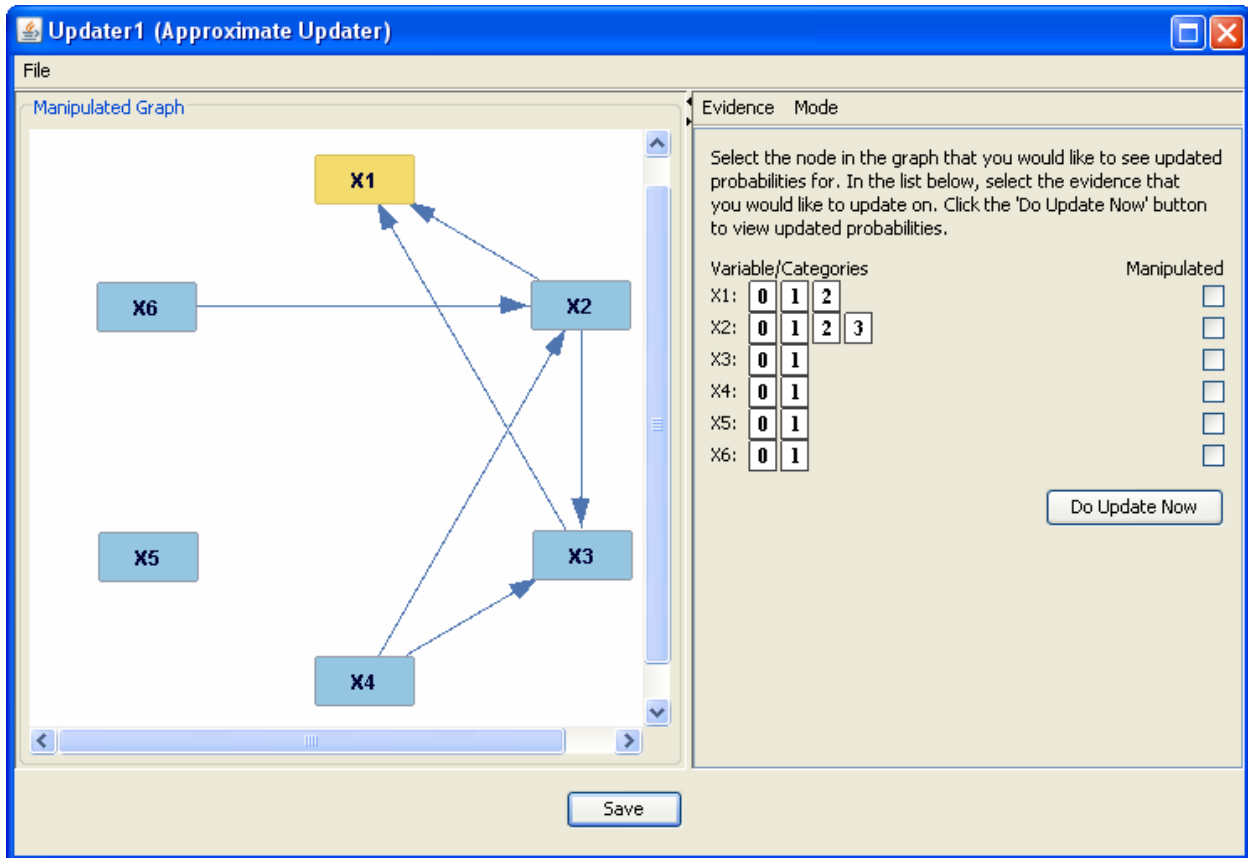
2. Scroll to a row (that is, combination of parent values) in the table below.

3. Click in the appropriate box and assign a probability to each value of the chosen variable in that row.

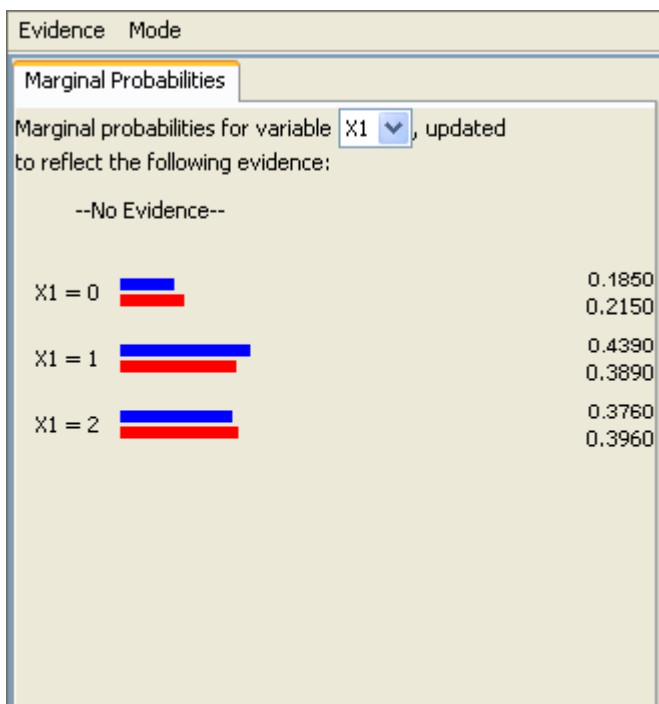
| X2 | X3 | X1=0   | X1=1   | X1=2   |
|----|----|--------|--------|--------|
| 0  | 0  | 0.3187 | 0.4018 | 0.2795 |
| 0  | 1  | 0.1860 | 0.7768 | 0.0372 |
| 1  | 0  | 0.2570 | 0.4871 | 0.2559 |
| 1  | 1  | 0.3398 | 0.1282 | 0.5320 |
| 2  | 0  | 0.2936 | 0.3976 | 0.3088 |
| 2  | 1  | 0.2713 | 0.3521 | 0.3766 |
| 3  | 0  | 0.4887 | 0.2668 | 0.2445 |
| 3  | 1  | 0.0361 | 0.4090 | 0.5548 |

Right click in table to randomize.

When it is input into the approximate updater, the following window results:



If we click “Do Update Now” now, without giving the updater any evidence, the right side of the screen changes to show us the marginal probabilities of the variables. Marginal probabilities are the overall probabilities that a variable will take on a given value; they do not take into account the values of the variable’s parents.

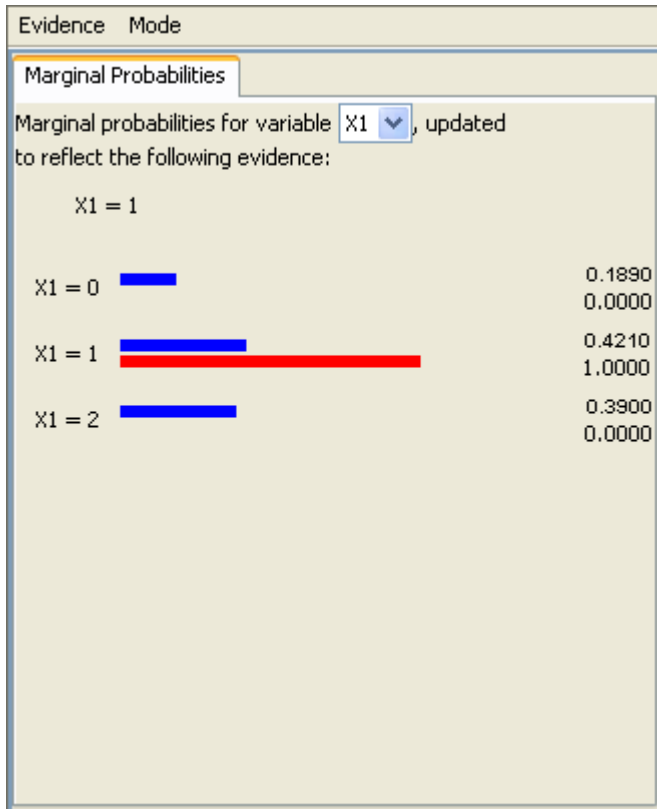


The blue lines, and the values listed across from them, indicate the probability that the variable takes on the given value in the input instantiated model. The red lines indicate the probability that the variable takes on the given value, given the evidence we’ve added to the updater.



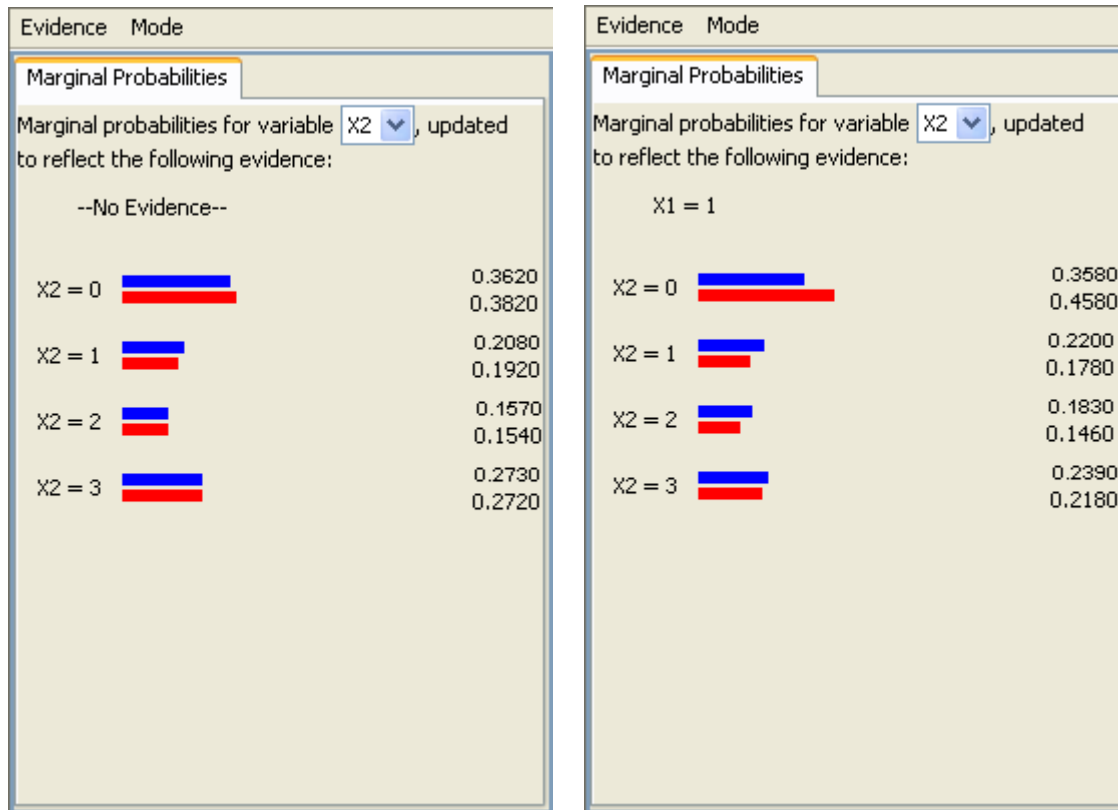
Since we have added no evidence to the updater, the red and blue lines are very similar in length. To view the marginal probabilities for a variable, either click on the variable in the graph to the left, or choose it from the scrolling menu at the top of the window. At the moment, they should all be very close to the marginal probabilities taken from the instantiated model.

Now, we'll return to the original window. We can do so by clicking "Edit Evidence" under the Evidence tab. Suppose we know that X1 takes on the value 1 in our model, or suppose we merely want to see how X1 taking that value affects the values of the other variables. We can click on the box that says "1" next to X1. When we click "Do Update Now," we again get a list of the marginal probabilities for X1.



Now that we have added evidence, the "red line" marginal probabilities have changed; for X1, the probability that X1=1 is 1, because we've told Tetrad that that is the case. Likewise, the probabilities that X1=0 and X1=2 are both 0.

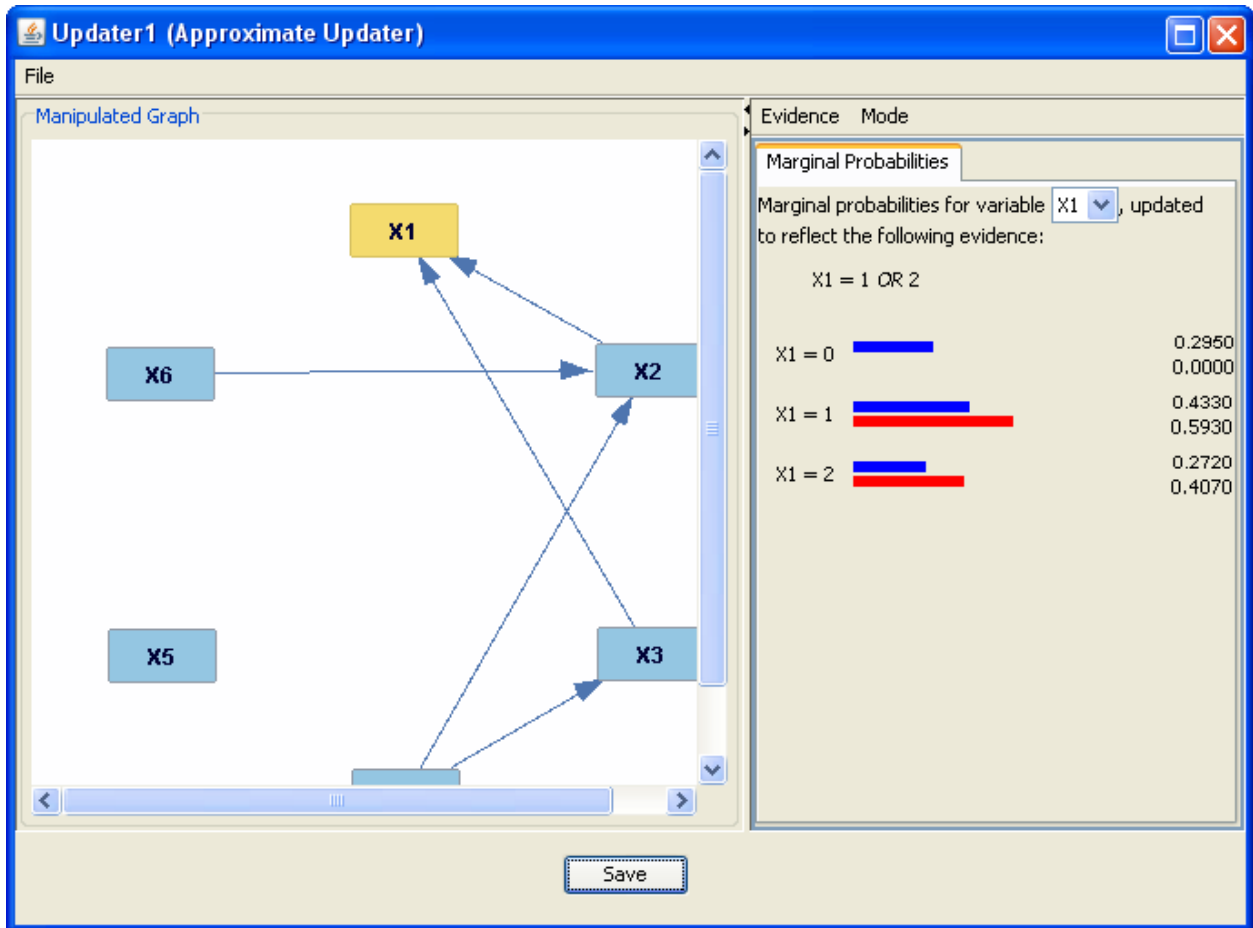
Now, let's look at the updated marginal probabilities for X2, a parent of X1.



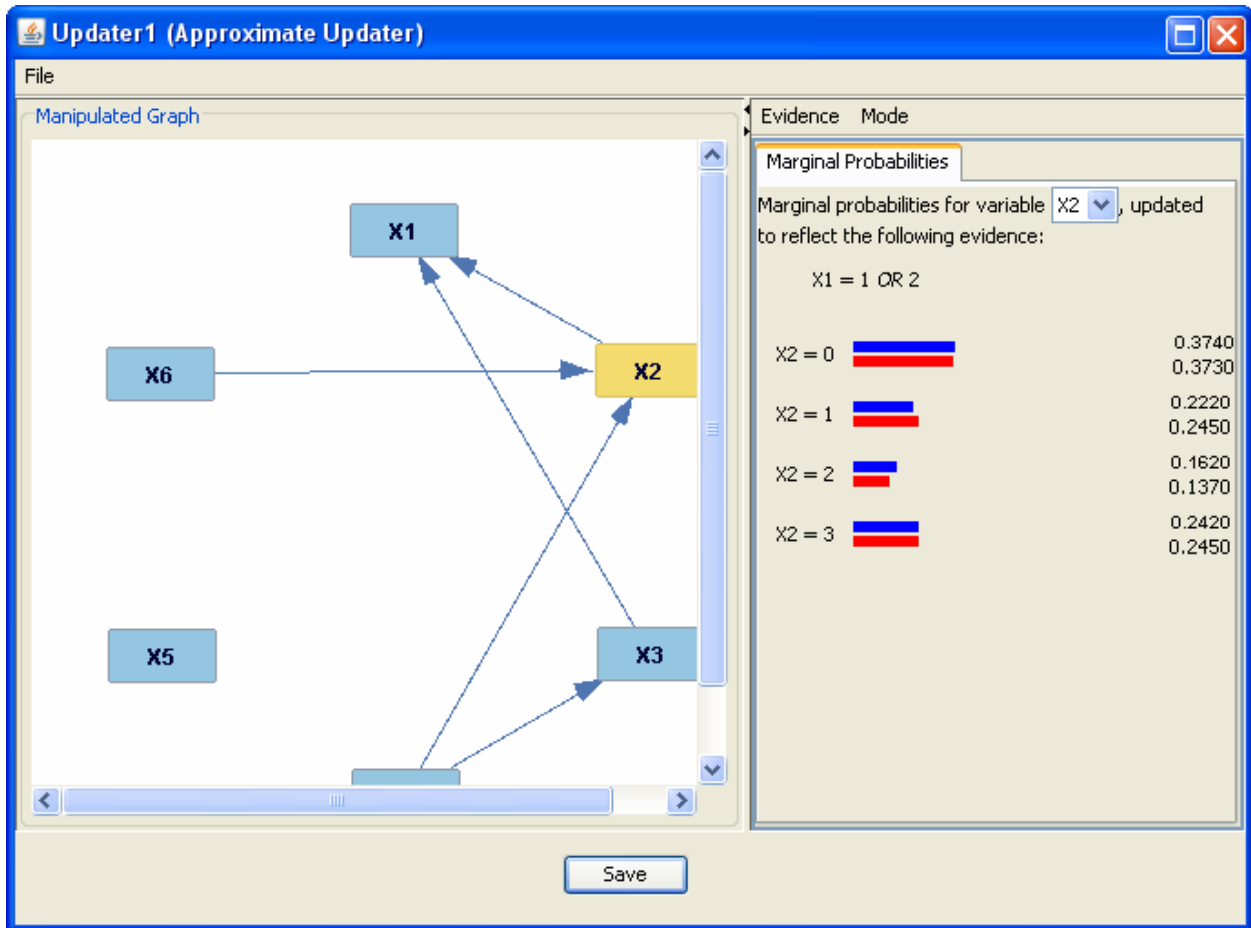
The image on the left is the marginal probabilities before we added the evidence that  $X_1=1$ . The image on the right is the updated marginal probabilities. They have changed; in particular, it has become much more likely that  $X_2=0$ .

Under the Mode tab, we can change the type of information that the updater box gives us. The mode we have been using so far is "Marginals Only (Multiple Variables)." We can switch the mode to "In-Depth Information (Single Variable)." Under this mode, when we perform the update, we receive more information (such as log odds and joints, when supported), but only about the variable which was selected in the graph when we performed the update. To view information about a different variable, we must re-edit the evidence with that variable selected. Joints are supported not supported for the approximate updater.

If the variable can take one of several values, or if we know the values of more than one variable, we can select multiple values by pressing and holding the Shift key and then making our selections. For instance, in the model above, suppose that we know that  $X_1$  can be 1 or 2, but not 0. We can hold the Shift key and select the boxes for 1 and 2, and when we click "Do Update Now," the marginal probabilities for  $X_2$  look like this:



Since X1 must be 1 or 2, the updated probability that it is 0 is now 0. The marginal probabilities of X2 also change:



The updated marginal probabilities are much closer to their original values than they were when we knew that X1 was 1.

Finally, if we are arbitrarily setting the value of a variable—that is, the values of its parents have no effect on its value—we can check the “Manipulated” box next to it while we are editing evidence, and the update will reflect this information.

For instance, suppose we set the variable X2 to the value 1, but click the “Manipulated” box. When we do the update,

### Row Summing Exact Updater

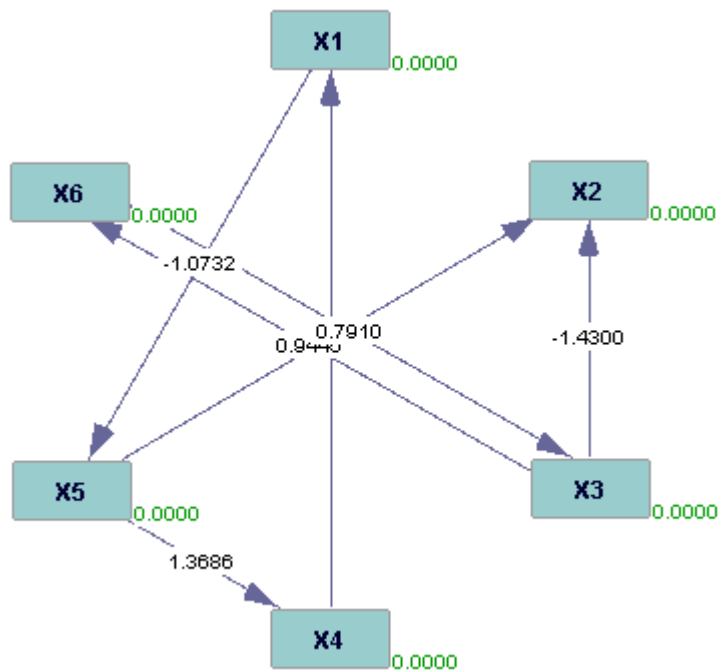
The row summing exact updater is a slower but more accurate updater than the approximate updater. The complexity of the algorithm depends on the number of variables and the number of categories each variable has. It creates a full exact conditional probability table and updates from that. Its window functions exactly as the approximate updater does, with two exceptions: in “Multiple Variables” mode, you can see conditional as well as marginal probabilities, and in “Single Variable” mode, you can see joint values.

## CPT Invariant Exact Updater

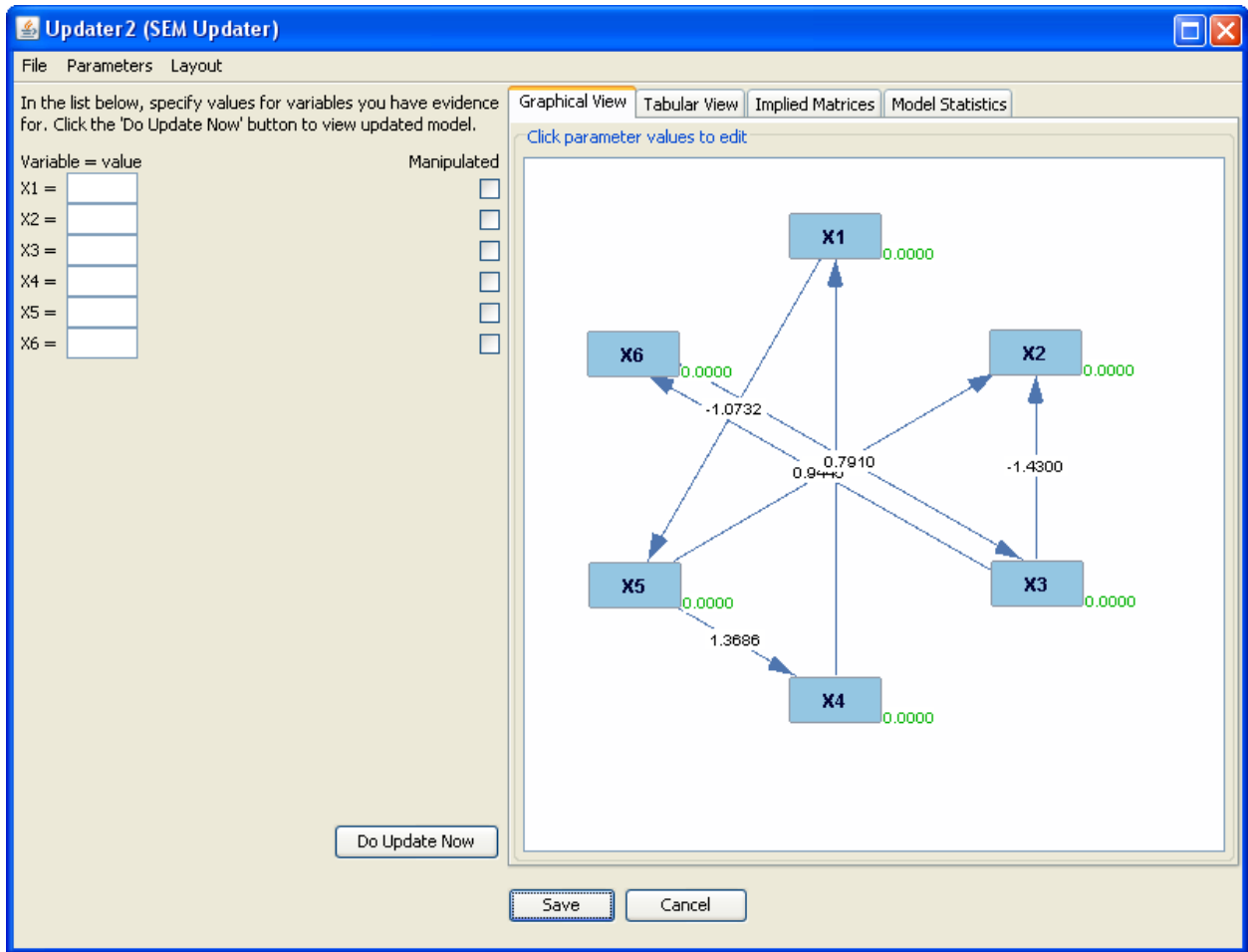
The CPT invariant exact updater is more accurate than the approximate updater, but slightly faster than the row summing exact updater. Its window functions exactly as the approximate updater down, with one exception: in “Multiple Variables” mode, you can see conditional as well as marginal probabilities.

## SEM Updater

The SEM updater does not deal with marginal probabilities; instead, it deals with means.



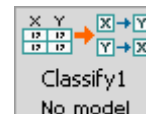
When it is input to the SEM updater, the following window results:



Suppose we know that the mean of X1 is .5. When we enter that value into the text box on the left and click “Do Update Now,” the model on the right updates to reflect that mean, changing the means of both X1 and several other variables. The new model looks like this:

The means of X2, X4, and X5 have all changed. If we click the “Manipulated” check box as well, it means that we have arbitrarily set the mean of X1 to .5, and that the value of its parent variable, X4, has no effect on it. The graph, as well as the updated means, changes to reflect this.

The rest of the window has the same functionality as a SEM instantiated model window.



The classify box in the main workspace looks like this:

**Possible Parents Boxes of the Classify Box:**

- An instantiated model box
- A data box
- A data manipulation box

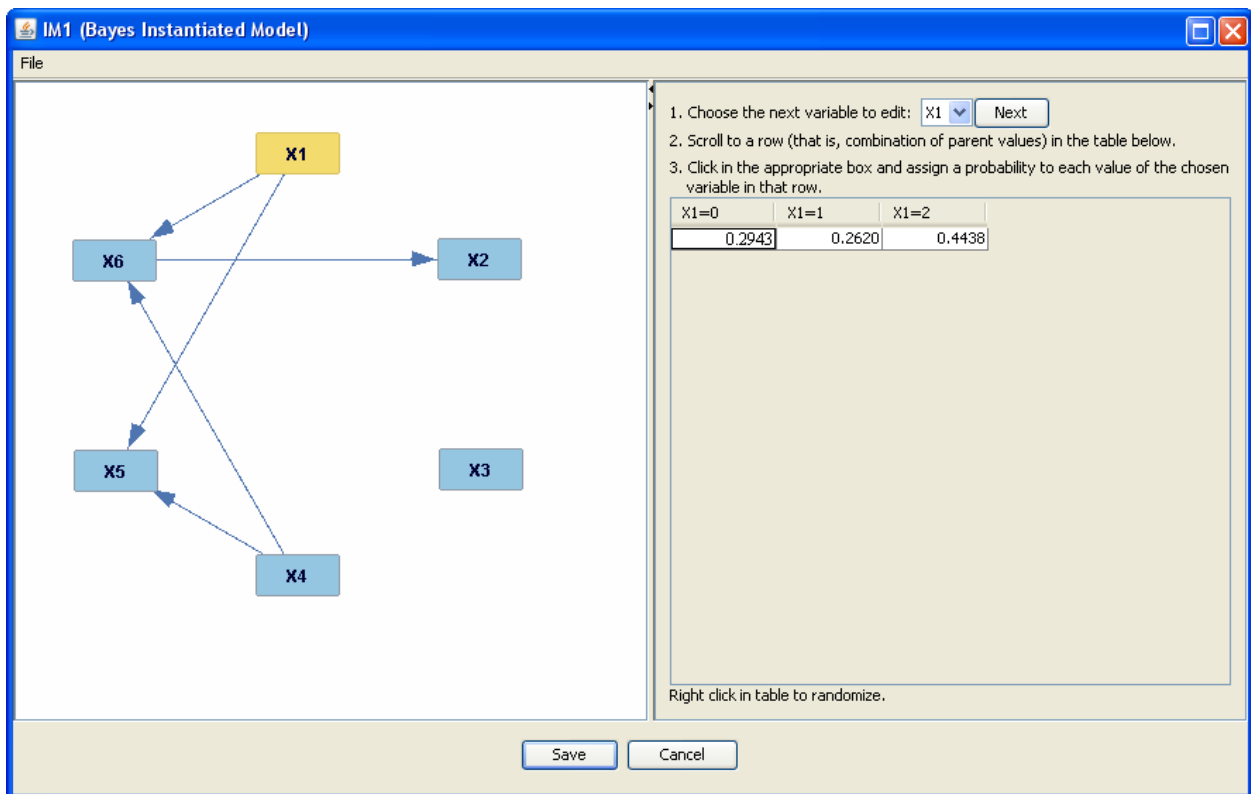
**Possible Child Boxes of the Classify Box:**

- A comparison box

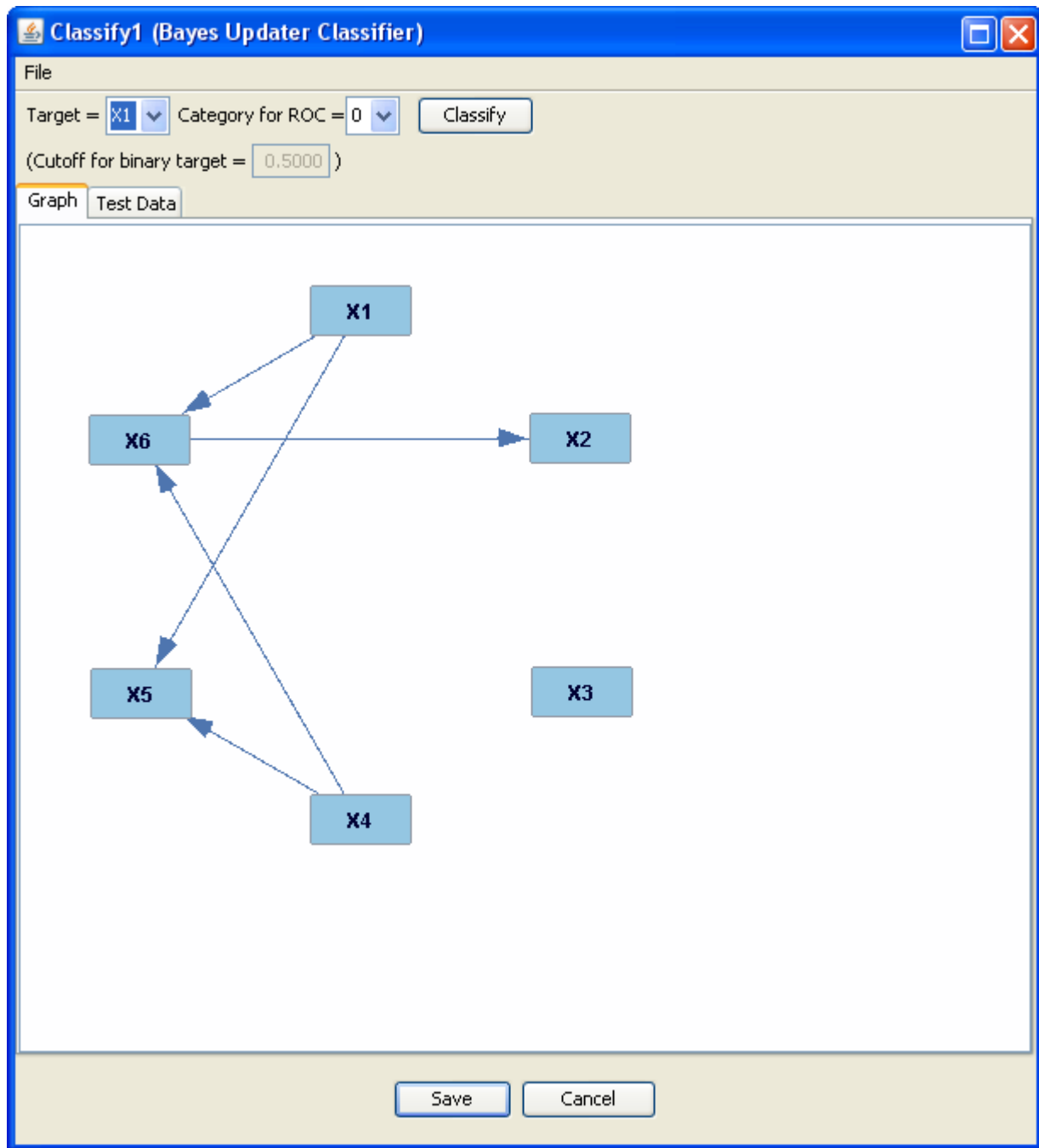
**Using the Classify Box:**

The classify box takes as input a categorical data set and a Bayes instantiated model and, for a given variable, estimates that variable's value in each case based on the other variables' values in that case.

Take, for example, the following instantiated model, and a data set derived from it:



When they are input to the classify box, the following window results:



By clicking on the “Test Data” tab, we can see the data set which we input. Now, suppose we want to estimate the values of X1 in each case. We select X1 from the scrolling menu at the top and click “Classify.” Three new tabs now appear: “Classification,” “ROC Plot,” and “Confusion Matrix.” The classification tab, in this case, looks like this:



**Classify1 (Bayes Updater Classifier)**

File

Target = X1 Category for ROC = 0

(Cutoff for binary target = 0.5000)

Graph Test Data **Classification** ROC Plot Confusion Matrix

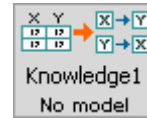
|    |      | C1-T   | C2      | C3      | C4      | C5 | C6 | C7 |
|----|------|--------|---------|---------|---------|----|----|----|
|    | MULT | Result | P(X1=0) | P(X1=1) | P(X1=2) |    |    |    |
| 1  | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |
| 2  | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |
| 3  | 1    | 2      | 0.0134  | 0.2016  | 0.7850  |    |    |    |
| 4  | 1    | 0      | 0.3785  | 0.3442  | 0.2773  |    |    |    |
| 5  | 1    | 0      | 0.7870  | 0.0164  | 0.1965  |    |    |    |
| 6  | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |
| 7  | 1    | 0      | 0.7870  | 0.0164  | 0.1965  |    |    |    |
| 8  | 1    | 0      | 0.6335  | 0.1634  | 0.2032  |    |    |    |
| 9  | 1    | 1      | 0.1255  | 0.6584  | 0.2161  |    |    |    |
| 10 | 1    | 2      | 0.3163  | 0.0489  | 0.6348  |    |    |    |
| 11 | 1    | 0      | 0.5479  | 0.1236  | 0.3284  |    |    |    |
| 12 | 1    | 1      | 0.1135  | 0.5211  | 0.3654  |    |    |    |
| 13 | 1    | 0      | 0.7937  | 0.0318  | 0.1745  |    |    |    |
| 14 | 1    | 2      | 0.0791  | 0.0585  | 0.8624  |    |    |    |
| 15 | 1    | 0      | 0.5540  | 0.3846  | 0.0614  |    |    |    |
| 16 | 1    | 0      | 0.7870  | 0.0164  | 0.1965  |    |    |    |
| 17 | 1    | 0      | 0.7937  | 0.0318  | 0.1745  |    |    |    |
| 18 | 1    | 1      | 0.1255  | 0.6584  | 0.2161  |    |    |    |
| 19 | 1    | 1      | 0.1891  | 0.4301  | 0.3808  |    |    |    |
| 20 | 1    | 0      | 0.5540  | 0.3846  | 0.0614  |    |    |    |
| 21 | 1    | 1      | 0.2950  | 0.5146  | 0.1904  |    |    |    |
| 22 | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |
| 23 | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |
| 24 | 1    | 2      | 0.0791  | 0.0585  | 0.8624  |    |    |    |
| 25 | 1    | 2      | 0.1012  | 0.1337  | 0.7651  |    |    |    |

There is a column for each of the possible values which X1 can take, and for each case, Tetrad has computed the probability that X1 will take each value, based on the configuration of other variables in that case. It then chooses the category with the highest conditional probability, and assigns X1 that value for that case. Comparison to the test data will show that the values are reasonably (though not completely) similar.

The ROC plot tab [receiver operating characteristic].

The confusion matrix tab provides information on how similar the estimated values of the target variable are to the true values (if the true values are available). Because classification is a

form of estimation, the variable which is classified does not have to be in the input data set; it merely has to be in the input instantiated model. In this case, the ROC plot and confusion matrix tabs will not appear.



The knowledge box in the main workspace looks like this:

### **Possible Parent Boxes of the Knowledge Box:**

- A graph box
- A graph manipulation box
- A parametric model box
- An instantiated model box
- A data box
- A data manipulation box

### **Possible Child Boxes of the Knowledge Box:**

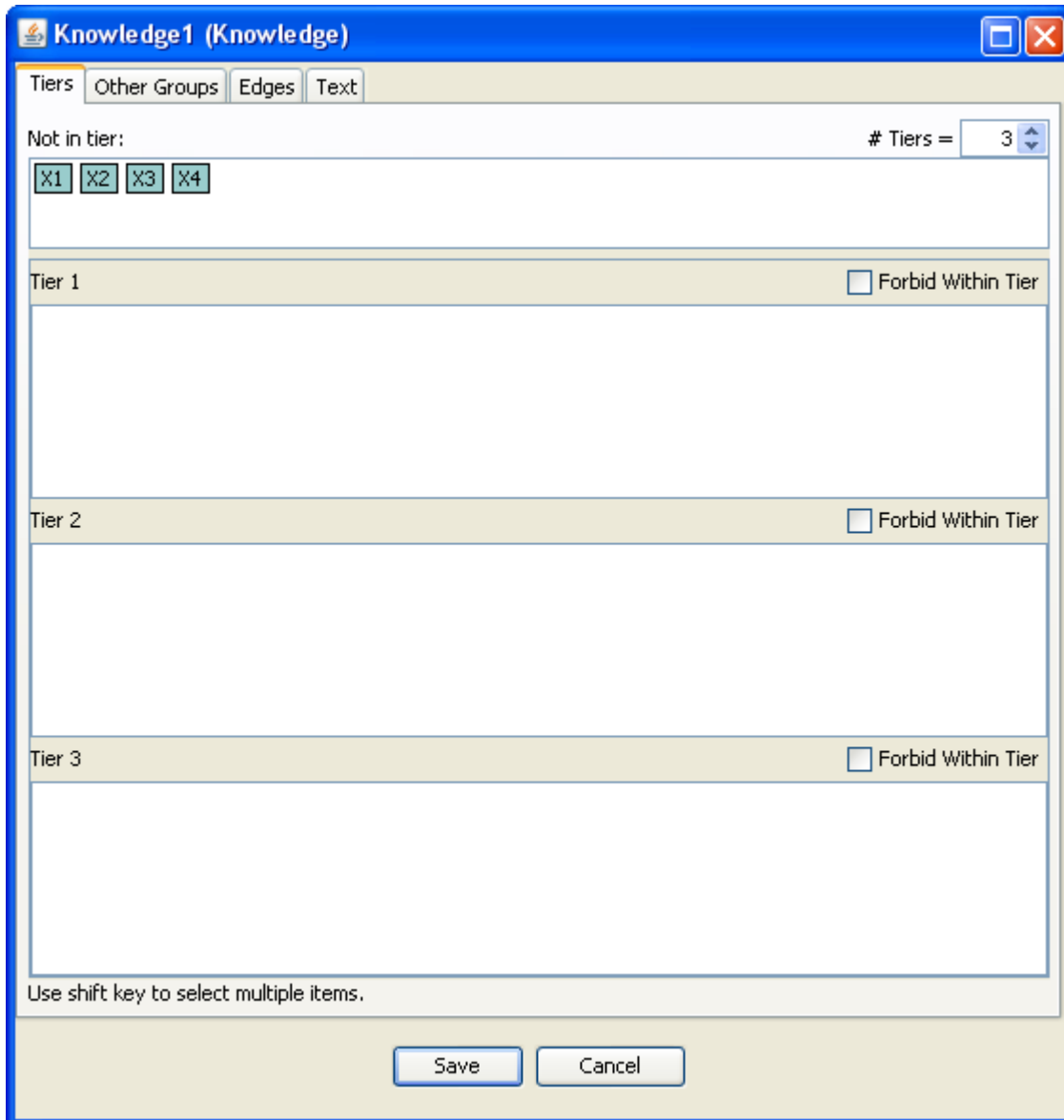
- A comparison box
- A search box

### **Using the Knowledge Box:**

The knowledge box takes as input a graph or a data set and imposes additional constraints onto it, generally to make search algorithms easier. There are three types of constraints you can add using the knowledge box: tiers of occurrence, forbidden or required groups, and forbidden or required edges.

#### Tiers

The tiers tab for a graph with four variables looks like this:

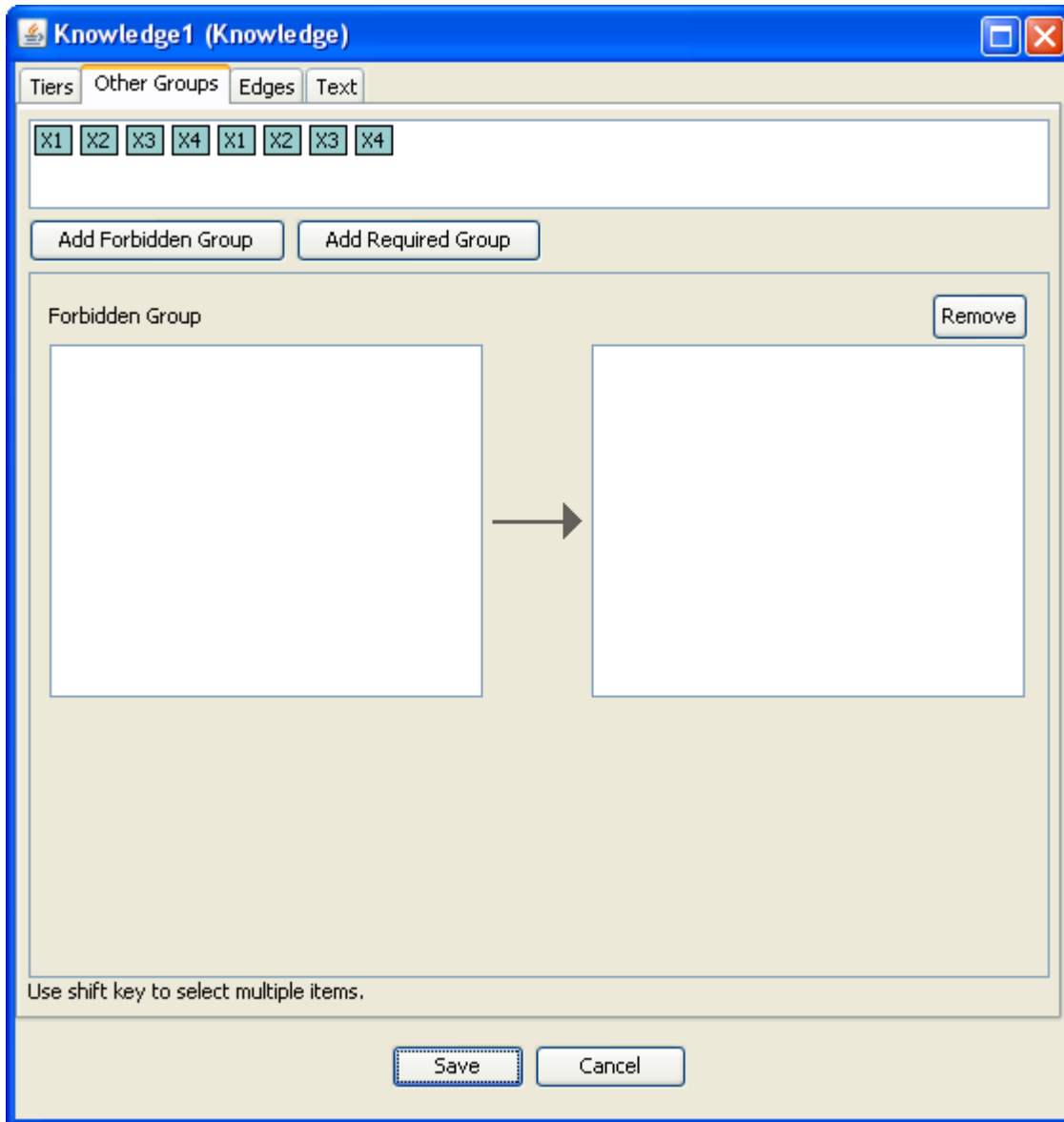


Tiers separate your variables into a time line. Variables in higher-numbered tiers occur later than variables in lower-numbered tiers, which gives Tetrad information about causation. For example, a variable in Tier 3 could not possibly be a cause of a variable in Tier 1.

To place a variable in a tier, click on the variable in the “Not in tier” box, and then click on the box of the tier. If you check the “Forbid Within Tier” box for a tier, variables in that tier will not be allowed to be causes of each other. To increase or decrease the number of tiers, use the scrolling box in the upper right corner of the window.

## Groups

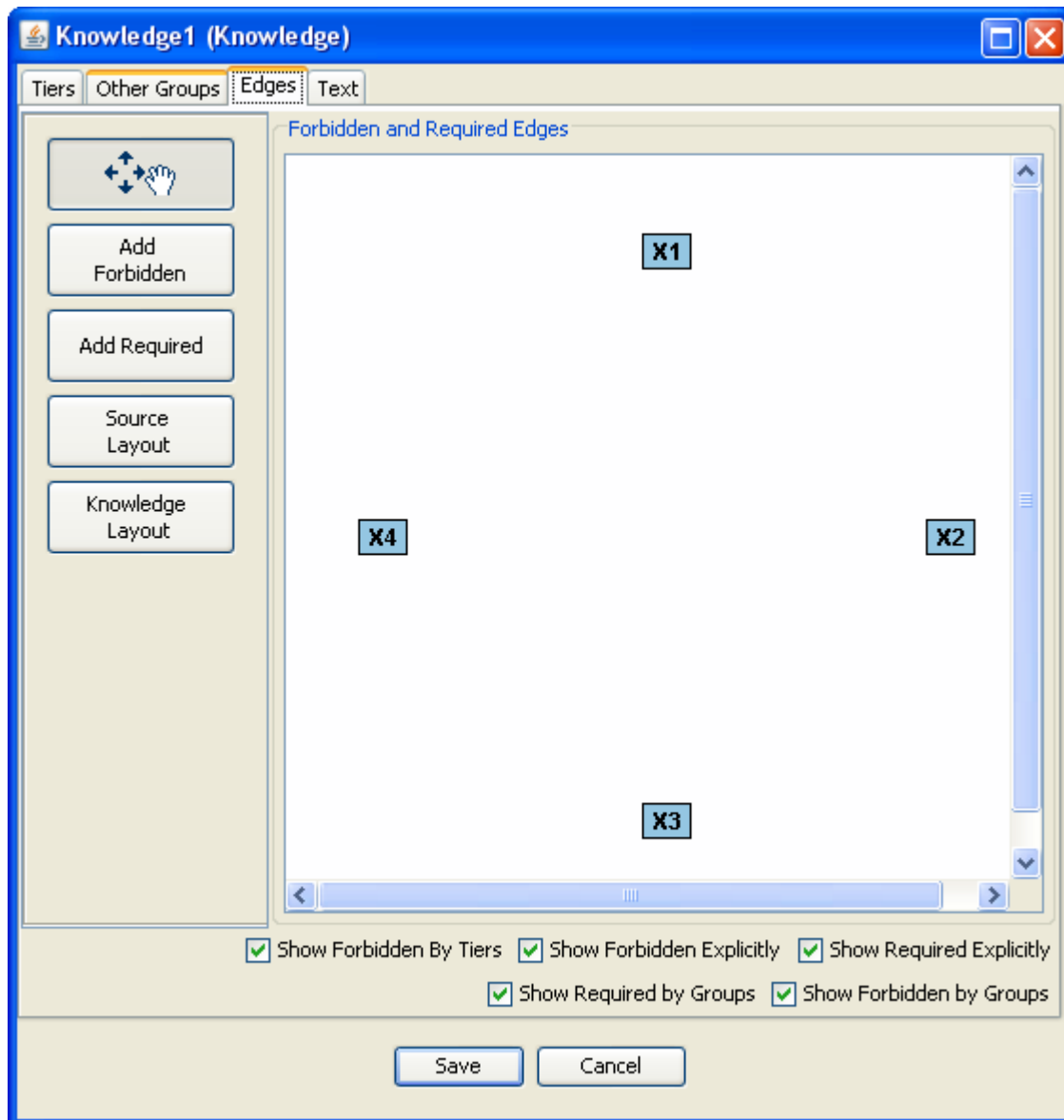
The groups tab for a graph with four variables looks like this:



In the groups tab, you can specify certain groups of variables which are forbidden or required to cause other groups of variables. To add a variable to the “cause” section of a group, click on the variable in the box at the top, and then click on the box to the left of the group’s arrow. To add a variable to the “effect” section of a group, click on the variable in the box at the top, and then click on the box to the right of the group’s arrow. You can add a group by clicking on one of the buttons at the top of the window, and remove one by clicking the “remove” button above the group’s boxes.

### Edges

The edges tab for a graph with four variables looks like this:

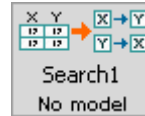


In the edges tab, you can require or forbid individual causal edges between variables. To add an edge, click the type of edge you'd like to create, and then click and drag from the “cause” variable to the “effect” variable.

You can also use this tab to see the effects of the knowledge you created in the other tabs by checking and unchecking the boxes at the bottom of the window. You can adjust the layout to mimic the layout of the source (by clicking “source layout”) or to see the variables in their timeline tiers (by clicking “knowledge layout”).

## Text

The text tab contains a textual representation of the knowledge you have created in the other three tabs. Knowledge cannot be edited in the text tab.



The search box in the main workspace looks like this:

### **Possible Parent Boxes of the Search Box:**

- A graph box
- A graph manipulation box
- A parametric model box
- An instantiated model box
- A data box
- A data manipulation box
- An estimator box
- A knowledge box
- Another search box
- A regression box

### **Possible Child Boxes of the Search Box:**

- A graph box
- A graph manipulation box
- A comparison box
- A parametric model box
- A data manipulation box
- Another search box

### **Using the Search Box:**

The search box searches for causal explanations represented by directed graphs. The result of a search is not necessarily—and not usually—a unique graph, but an object such as a pattern that represents a set of graphs, usually a Markov Equivalence class. More alternatives can be found by varying the parameters of search algorithms. The search box can perform nineteen possible functions (the number may have increased since this manual was posted), which fall into seven categories: pattern searches, DAG searches, DG searches, PAG searches, multiple indicator model searches, feature selection searches, and factor analysis. Some of these search procedures can be used in combination.

### **Pattern Searches**

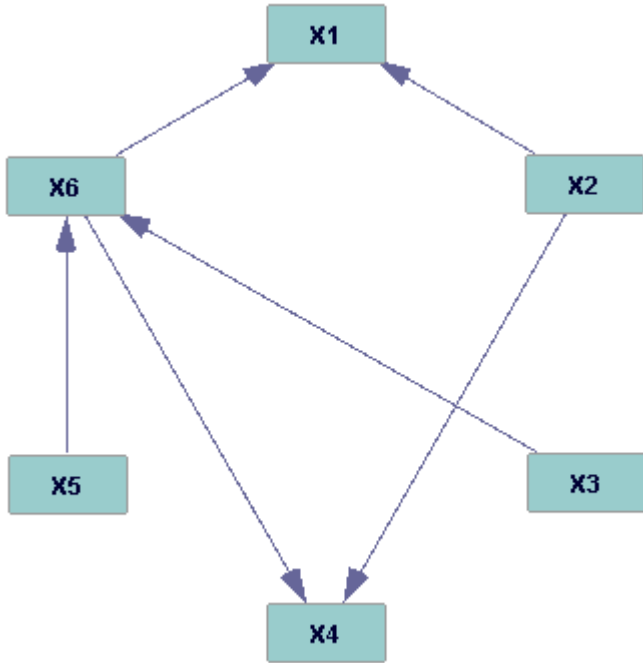
Pattern searches take as input data sets and output a pattern (or at least, a graph resembling a pattern) to represent the underlying causal structure generating the data. For more information on patterns, see the section on the graph manipulation box.

### The PC Algorithm

The PC algorithm (Spirtes and Glymour, *Social Science Computer Review*, 1991) is a pattern search which assumes that the underlying causal structure of the input data is acyclic, and that no two variables are caused by the same latent (unmeasured) variable. In addition, it is assumed that the input data set is either entirely continuous or entirely discrete; if the data set is continuous, it is assumed that the causal relation between any two variables is linear, and that the distribution of each variable is Normal. Finally, the sample should ideally be i.i.d.. Simulations show that PC and several of the other algorithms described here often succeed when these assumptions, needed to prove their correctness, do not strictly hold. The PC algorithm will sometimes output double headed edges. In the large sample limit, double headed edges in the output indicate that the adjacent variables have an unrecorded common cause, but PC tends to produce false positive double headed edges on small samples.

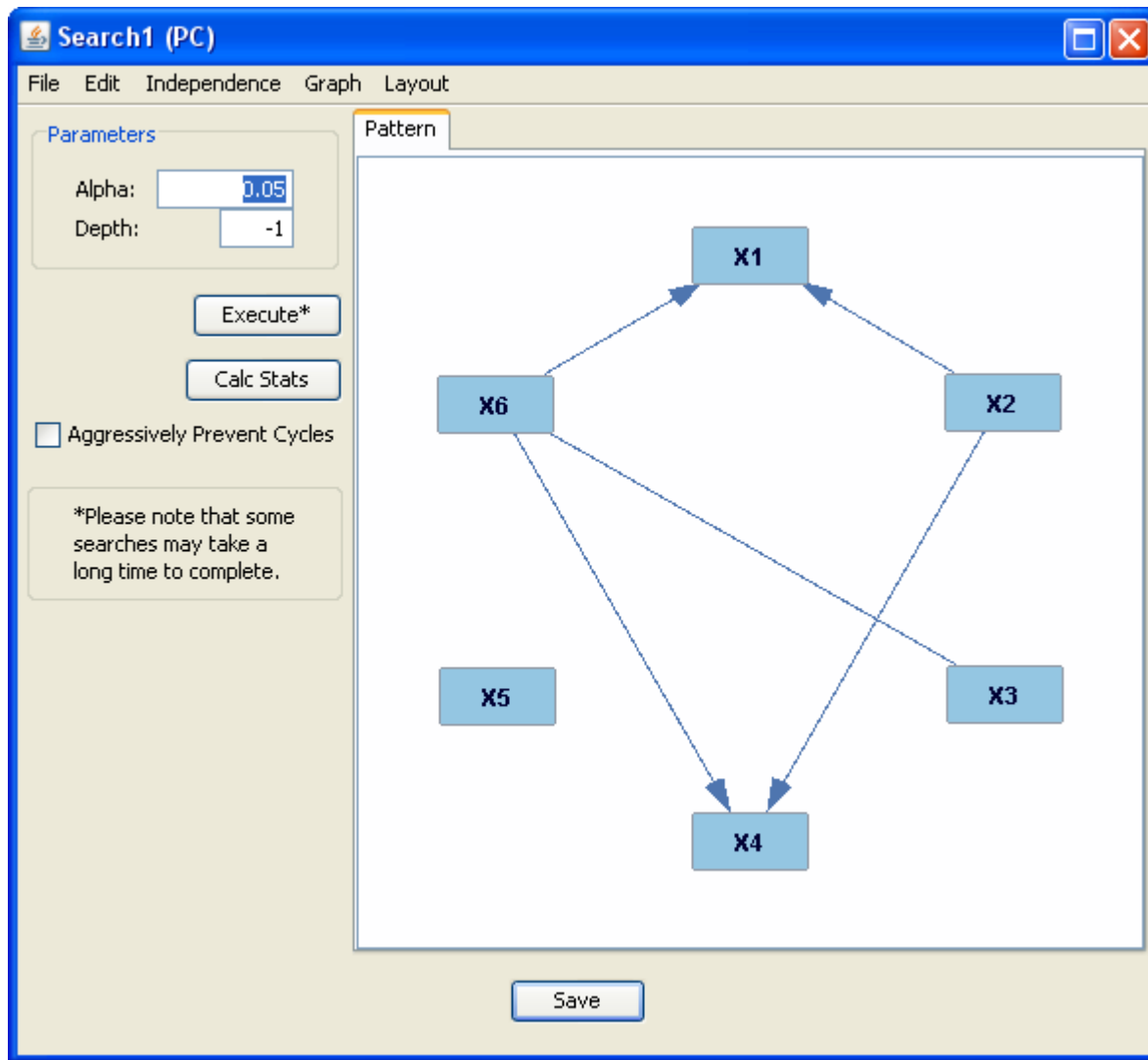
The PC algorithm is correct whenever decision procedures for independence and conditional independence are available. The procedure conducts a sequence of independence and conditional independence tests, and efficiently builds a pattern from the results of those tests. As implemented in TETRAD, PC is intended for multinomial and approximately Normal distributions with i.i.d. data. The tests have an alpha value for rejecting the null hypothesis, which is always a hypothesis of independence or conditional independence. For continuous variables, PC uses tests of zero correlation or zero partial correlation for independence or conditional independence respectively. For discrete or categorical variables, PC uses either a chi square or a g square test of independence or conditional independence (see *Causation, Prediction, and Search* for details on tests). In either case, the tests require an alpha value for rejecting the null hypothesis, which can be adjusted by the user. The procedures make no adjustment for multiple testing. (For PC, CPC, JPC, JCPC, FCI, all testing searches.)

Consider a discrete data set with the following underlying causal structure:



When input into the PC algorithm, the following window results:





The output is a pattern in this case, though it isn't always. The PC algorithm sometimes outputs graphs with cycles and bidirected edges; under ideal conditions, a bidirected edge between two variables indicates a latent common cause. The algorithm is deterministic, so given the same input and parameters, the output pattern will always be the same. You can, however change the parameters, and thereby possibly change the output. The choice of appropriate alpha values requires some experience and depends on the sample size and the number of variables. For small models, such as the one illustrated, and small samples ( $< 500$ ), the default 0.05 alpha value is commonly used. Alpha values should be decreased as the sample size increases. In general, smaller alpha values will give sparser graphs. If, for example, the user is concerned to avoid false positive edges, the sample size should be lowered. Again, because of multiple hypothesis testing in the search, to reduce false positives the alpha value should be lowered (we do NOT however recommend using the Bonferroni adjustment). For example, in simulations with 5,000 variables and sample sizes of 250, an alpha value of  $10^{-7}$  on data from a sparse model (5,000 edges) finds the preponderance of edges and a 7 to 8% false positive rate.

It is wise to test PC, or any other search on the best simulations you can produce that are similar to your actual data in distribution family, sample size, and number of variables. See the section on Simulation.

The two parameters on the left side of the window, “Alpha” and “Depth,” determine the way in which Tetrad performs independence tests between variables. The alpha value is a threshold for independence; the higher it is set, the less discerning Tetrad is when determining the independence of two variables. The depth value specifies the maximum size of subsets of variables which Tetrad can condition on when testing for independence.

If you check the “Aggressively Prevent Cycles” box, and then click “Execute,” the search will rerun, and the output graph will be acyclic.

If you click the Calc Stats button, the algorithm will run again, and two tabs will appear next to the “Pattern” tab: “DAG in Pattern” and “DAG Model Statistics.” The “DAG in Pattern” tab provides the same function as the option of that name in the graph manipulation box. The “DAG Model Statistics” tab provides information on the p-value, degrees of freedom, chi square, and BIC score of the search, for that chosen DAG.

Under the Independence tab, you can specify whether Tetrad uses the chi square or the g square test for independence of variables, for discrete data, or whether Tetrad uses the Fisher Z test, the Cramer T test, or linear regression for independence of variables, for continuous data.

In general, the Graph tab in the search box functions in the same way that it does in the graph box. There are, however, a few additional functionalities in the search box. In particular, the “Meek Orientation” option orients the edges of the graph according to the Meek Rules. [C. Meek, Causal inference and causal explanation with background knowledge. In *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence*. pages 403-410. 1995.]. The “Global Score-based Reorientation” option runs the GES edge orientation algorithm on only the edges present in the graph. (See the GES section for further details.)

### The PCPattern Algorithm

The PCPattern algorithm runs under the same assumptions and with the same input as the PC algorithm. The sole difference between the algorithms is that the output of a PCPattern search is *always* a pattern.

### The PCD Algorithm

The PC deterministic (PCD) algorithm runs under the same assumptions, with similar input and the same output as the PC algorithm. However, the PCD algorithm allows for continuous variables in the model to have deterministic relationships. The procedure is heuristic.

For more information on the PCD algorithm, see C. Glymour, “Learning the Structure of Deterministic Systems” in A. Gopnik and L. Schultz in *Causal Learning: Psychology, Philosophy, and Computation*, Oxford University Press, 2007.

### The CPC Algorithm

The conservative PC (CPC) algorithm runs under the same assumptions, with the same input and similar output to the PC algorithm. The CPC algorithm has a more sensitive test for orientation of colliders than PC, however, and will occasionally output ambiguous triples of undirected edges.

For more information on the CPC algorithm, see J. Ramsey, P. Spirtes, and J. Zhang (2006). Adjacency-faithfulness and conservative causal inference. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 401-408, Oregon, AUAI Press.

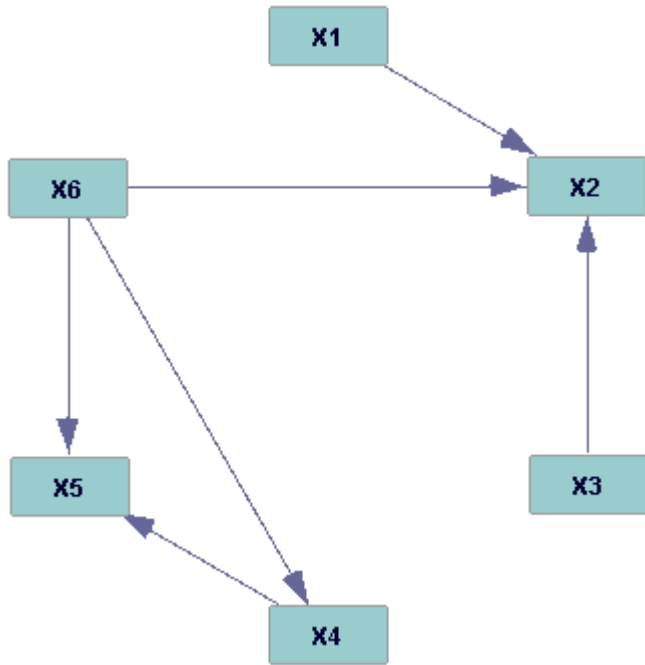
### The JPC Algorithm

The JPC algorithm is a sort of “metasearch” of the PC algorithm. It runs under the same assumptions, with the same input and the same output as the PC algorithm. JPC first runs a PC search on the input data. Then, in order to determine edge additions, removals and further orientations, for each pair of edges in the output graph, it checks for independence of the edges conditional on larger subsets of other variables. It then repeats the process until the output of the algorithm converges.

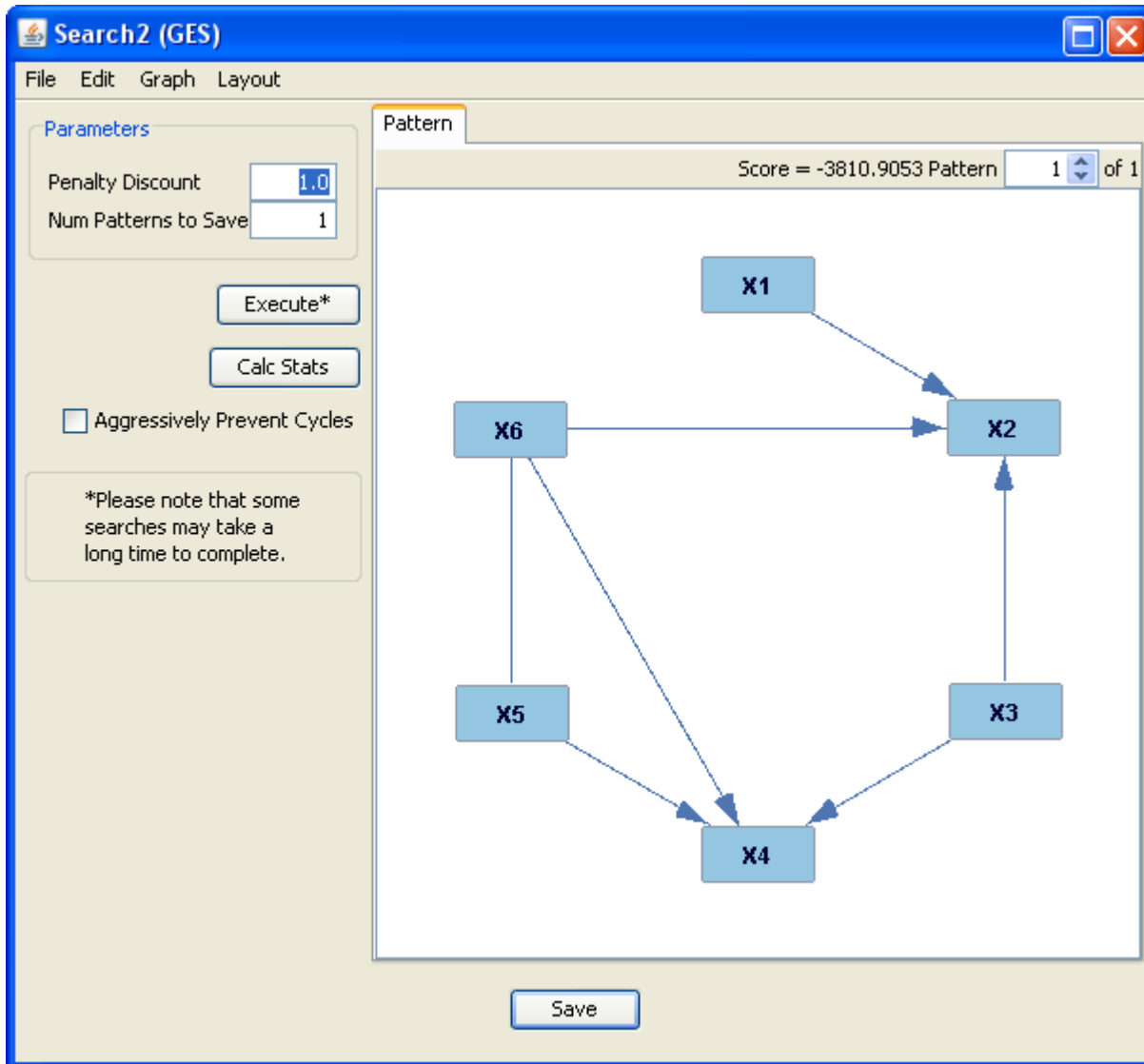
For more information on the JPC algorithm, see Ramsey, Joseph (2010). Bootstrapping the PC and CPC Algorithms to Improve Search Accuracy. Technical report 187, Department of Philosophy, Carnegie Mellon University.

### The GES Algorithm

The GES (greedy equivalency search) algorithm runs under the same input assumptions as the PC algorithm. Consider a continuous data set with the following underlying causal structure:



When input into the GES algorithm, the following window results:



The GES algorithm is a stable algorithm, so given the same input and parameters, the output patterns will always be the same.

The “Penalty Discount” and “Num Patterns to Save” parameters on the left side of the window determine how many and which of the highest scoring patterns the algorithm outputs. GES is a score-based orientation algorithm; it scores possible orientations of edges between variables, and the higher the score, the better the approximation should be. The number in the “Num Patterns to Save” box determines how many of the patterns will appear in the scrolling menu at the top right of the window; if the number is 5, then the five highest-scoring patterns will be saved. The number in the “Penalty Discount” box affects which edges are discarded; the higher the penalty discount, the more robust (i.e. certain) an edge must be to remain in the graph.

For more information on the GES algorithm, see Chickering, D. (1996). Learning equivalence classes of Bayesian-network structures. *Proceedings of the Twelfth Conference on Uncertainty in AI*, Portland, Ore.: Morgan Kaufmann, 150-157.

## The iMAGES Algorithm

The iMAGES algorithm runs the GES algorithm on multiple data sets. It runs the algorithm on all data sets multiple times, with increasing penalty discounts, until there are no three-variable cliques left in the graph (this requirement can be adjusted). iMAGES scores each model separately on each data set and averages the scores, choosing at each step in its procedure, the Markov equivalence class, or pattern, with the best average score.

In order to use iMAGES on multiple data sets, you will have to be able to load several data sets into a data box. For information on how to do this, see the section on the data box. In all other ways, the iMAGES search window functions like the GES search window.

For more information on the iMAGES algorithm, see Ramsey, J.D *et al.* Six problems for Causal Inference from fMRI. *NeuroImage* . Vol. 48, Issue 2. 15 Jan 2010, pages 1545-1558.

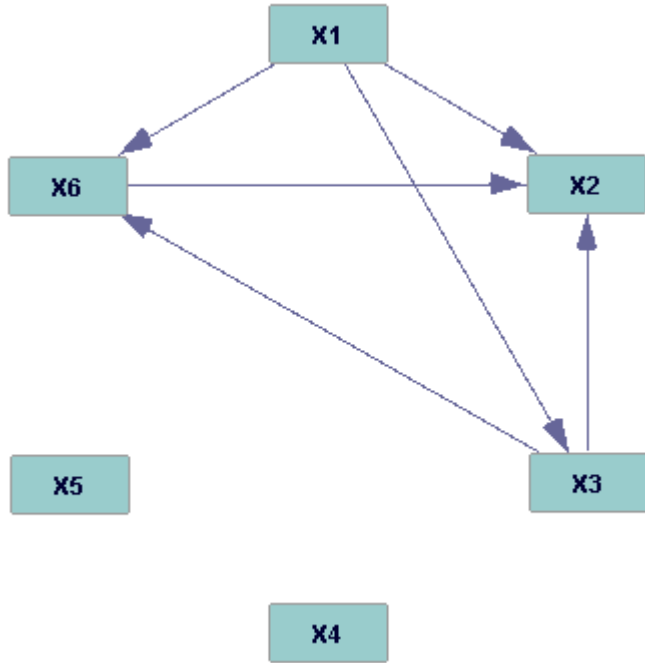
## **DAG Searches**

DAG searches take as input a data set and output a DAG.

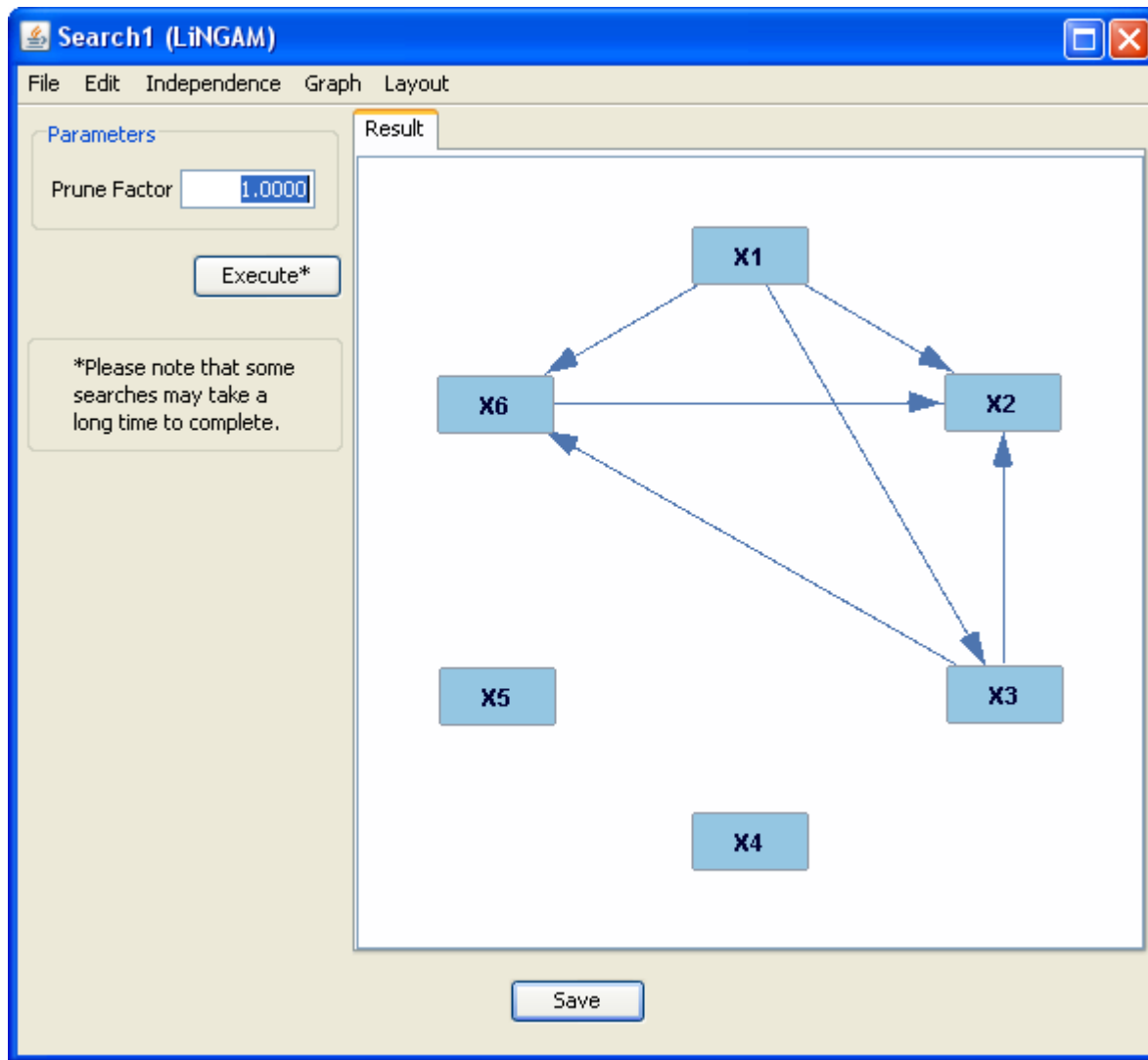
## The LiNGAM Search

The LiNGAM algorithm takes as input a data set (assumed to have an underlying causal structure representable by an acyclic linear SEM with non-Normal error terms) and outputs a DAG. The algorithm is correct if the generating process is linear, the recorded variables are not deterministically related to one another, at most one disturbance or error term is Normal, there are no unobserved common causes of recorded variables, and the sampling is i.i.d. To assess non-Normality of error terms for data, one may first calculate the residuals of the data (in the Data Manipulation box—“Convert to Residuals”, which requires a DAG as input in addition to the data) and then in the Data Box, under “Tools”, select Normality Tests. The Tetrad suite does not provide tests of linearity.

For instance, consider a data set with the following underlying structure, in which all of the error terms have Poisson distributions (Note: to simulate such a structure, you will need to know how to use generalized SEM models; for more information, see the section on parametric model box.):



When such a dataset is input into the LiNGAM search box, the following window results:



In this case, LiNGAM outputs a graph identical to the true underlying structure; obviously, this will not always be true. LiNGAM will always output the same graph, given the same input and parameters.

You can change the output by adjusting the “Prune Factor” parameter on the left side of the window. LiNGAM begins with a completely dependent graph (every variable is the child of every other variable); it then prunes inaccurate dependencies one by one. The prune factor is the threshold for pruning edges.

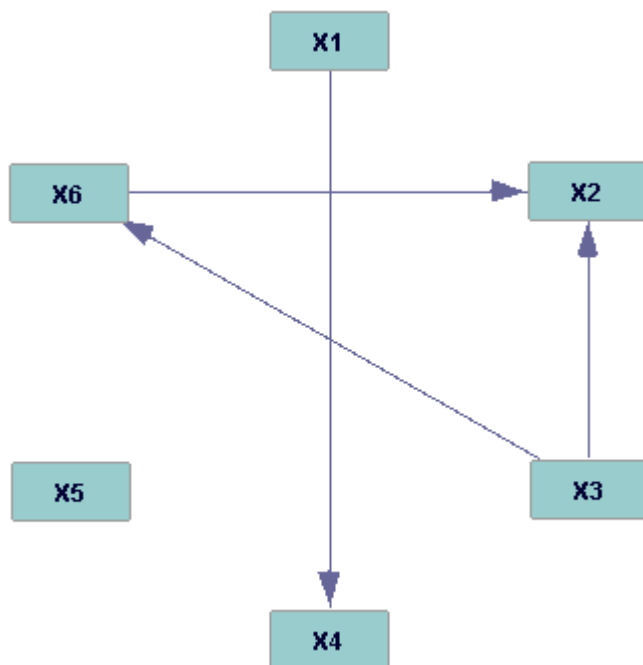
Under the Independence tab, you can select one of several tests for independence of variables which LiNGAM can use. The Graph tab contains all of the functions that it does in a graph box, with a few additional options: “Meek Orientation” orients the edges in the graph according to the Meek rules, and “Global Score-based Reorientation” runs the GES algorithm on only the edges present in the box (see the GES section under “Pattern Searches” for more information).



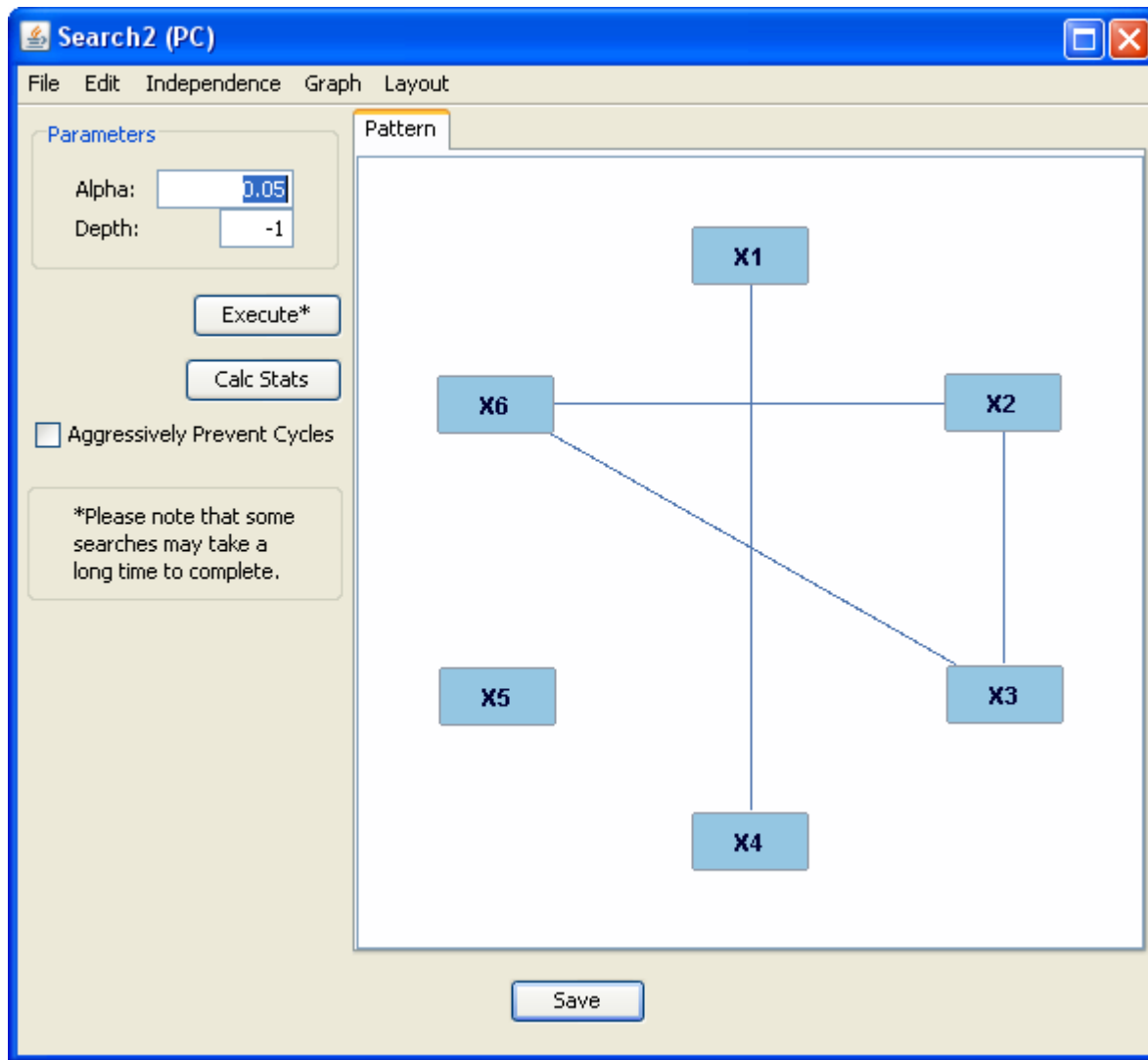
For more information on the LiNGAM algorithm, see P. O. Hoyer and A. Hyttinen. “Bayesian discovery of linear acyclic causal models”. Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI-2009), Montreal, Canada, 2009. The version of LiNGAM implemented in Tetrad is an exact translation of the R version provided by the authors of the algorithm and gives the same output.

### The LiNGAM Pattern Search

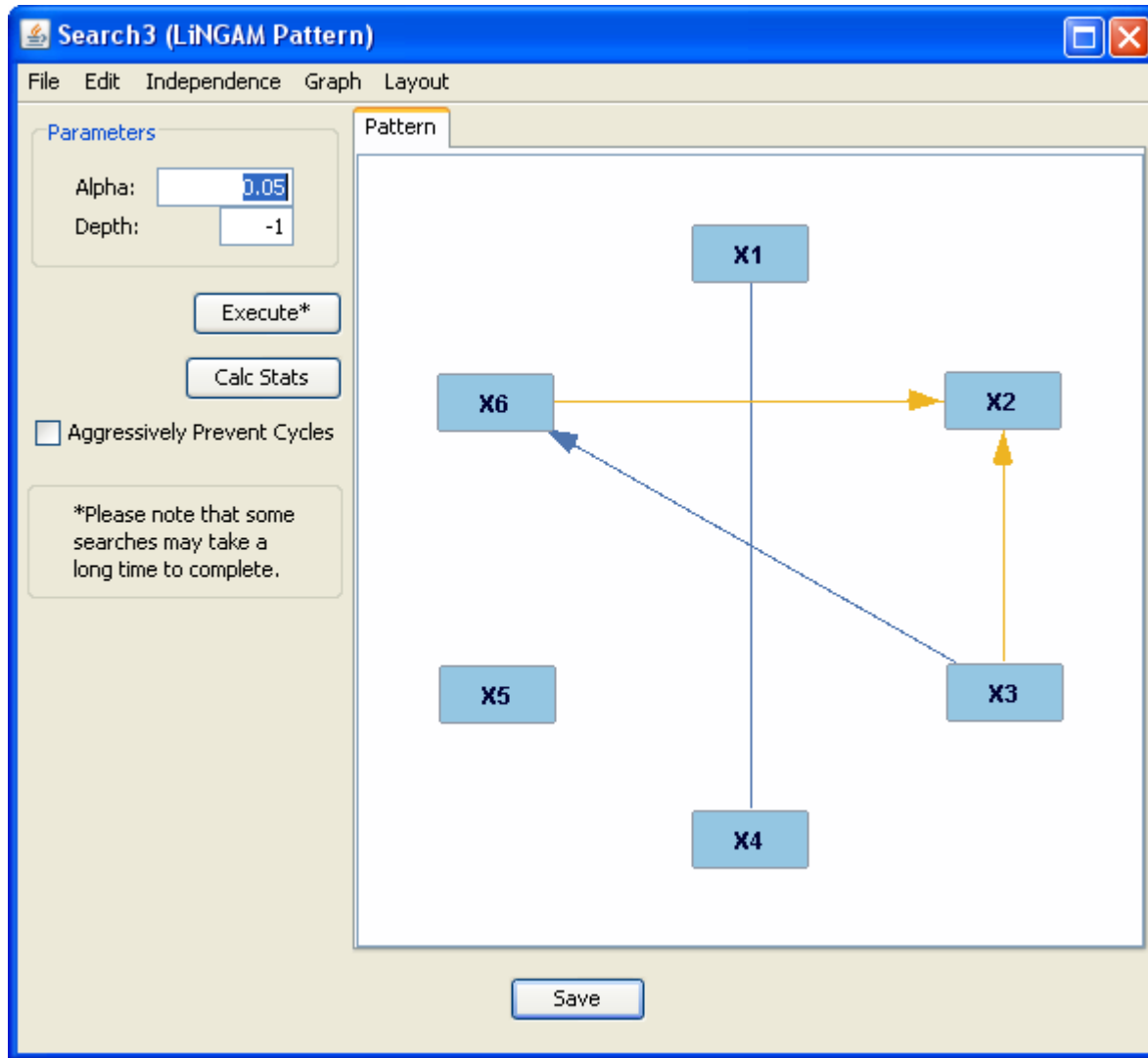
The LiNGAM Pattern search takes as input a pattern search and the data set which the pattern search was run on, and attempts to orient any unoriented edges. Like LiNGAM, it assumes that the underlying causal structure of the data set is an acyclic linear SEM. Consider a data set with the following underlying causal structure, in which all of the variables have error terms with Normal distributions:



When this data set is input into the PC search, Tetrad produces the following output:



When this search box and the data set are input into a LiNGAM Pattern search box, the following window results:



It is identical to the true causal graph, with the exception of one unoriented edge. Obviously, such results do not occur in every case.

The yellow edges in the output graph are edges which were unoriented in the PC output, and are now oriented. The parameters on the left side of the window can be adjusted, but as this is not the PC algorithm, doing so will have no effect.

In all other ways, the LiNGAM Pattern window functions like the LiNGAM window.

For more information on the LiNGAM Pattern algorithm, see P. O. Hoyer, A. Hyvärinen, R. Scheines, P. Spirtes, J. Ramsey, G. Lacerda, and S. Shimizu. Causal discovery of linear acyclic models with arbitrary distributions. Submitted, *Uncertainty in Artificial Intelligence*.

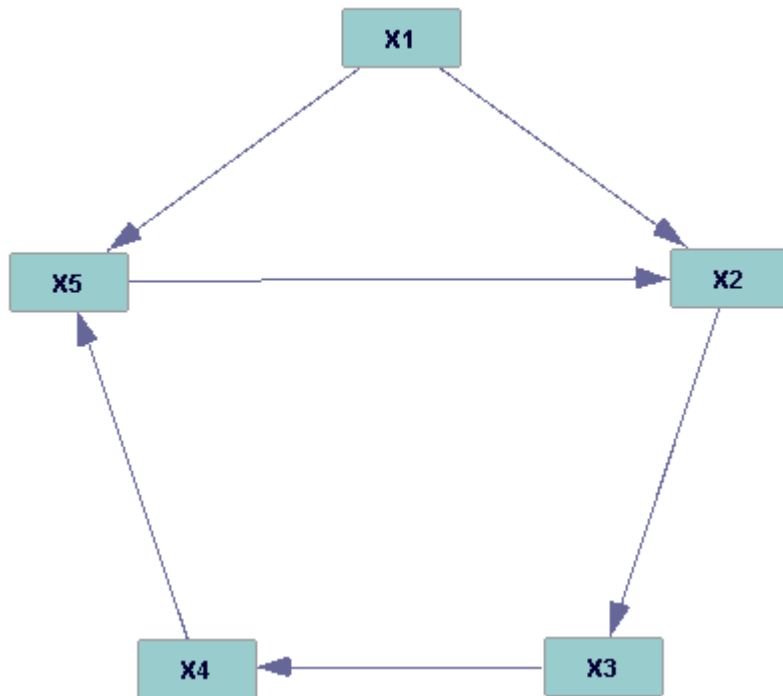
## DG Searches

Directed Graph

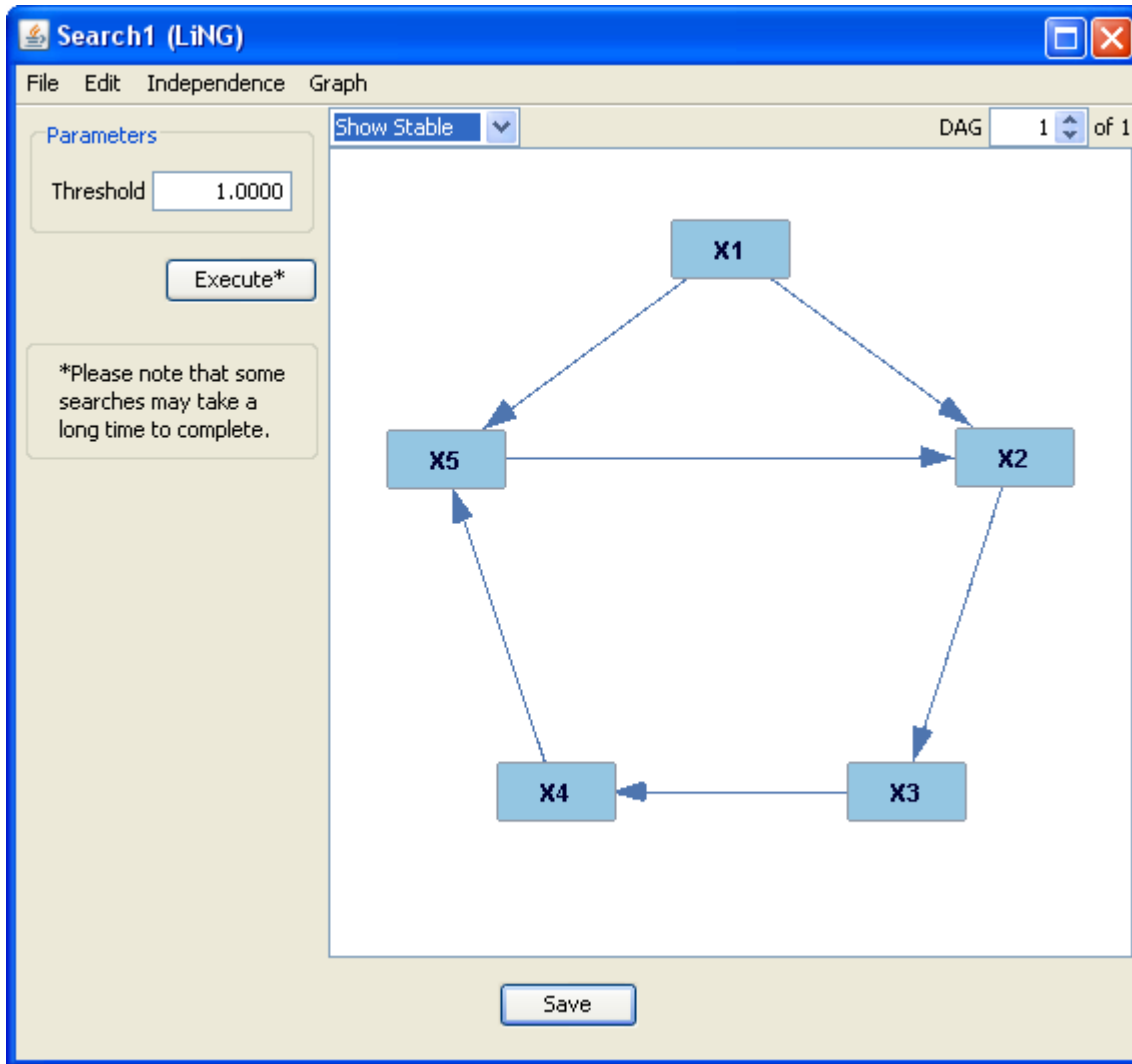
## The LiNG Search

The LiNG algorithm takes as input a data set (assumed to have an underlying causal structure representable by a linear SEM with non-Normal error terms) and outputs a directed graph. Unlike LiNGAM, LiNG can handle data sets in which the underlying causal structure is cyclic.

Take, for example, a data set with the following underlying causal structure:



When this data set is input into the LiNG search, the following window results:



Like LiNGAM, LiNG begins with a completely dependent graph, then prunes dependencies one by one. By changing the value of the “Threshold” parameter, you can specify the threshold of certainty beyond which LiNG will not prune a dependency.

LiNG will search for and record every model which fits the data and parameters it is given. These models fall into two types: stable and unstable. A stable model contains only stable cycles; an unstable model contains at least one unstable cycle. A stable cycle is one in which the absolute value of the product of the coefficients of the variables in the cycle is less than or equal to one. You can choose to see only stable models, only unstable models, or all models using the drop-down menu at the top left of the window. To scroll through the returned models, use the scrolling menu at the top right.

### PAG Searches

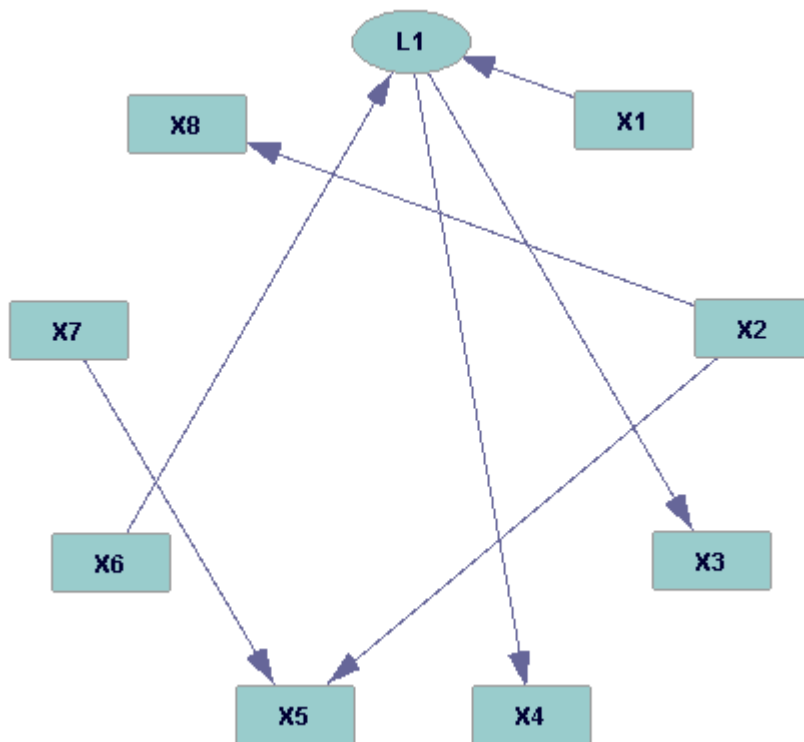
Partial ancestral graph (PAG) searches take as input data sets and return PAGs as output.

## The FCI Search

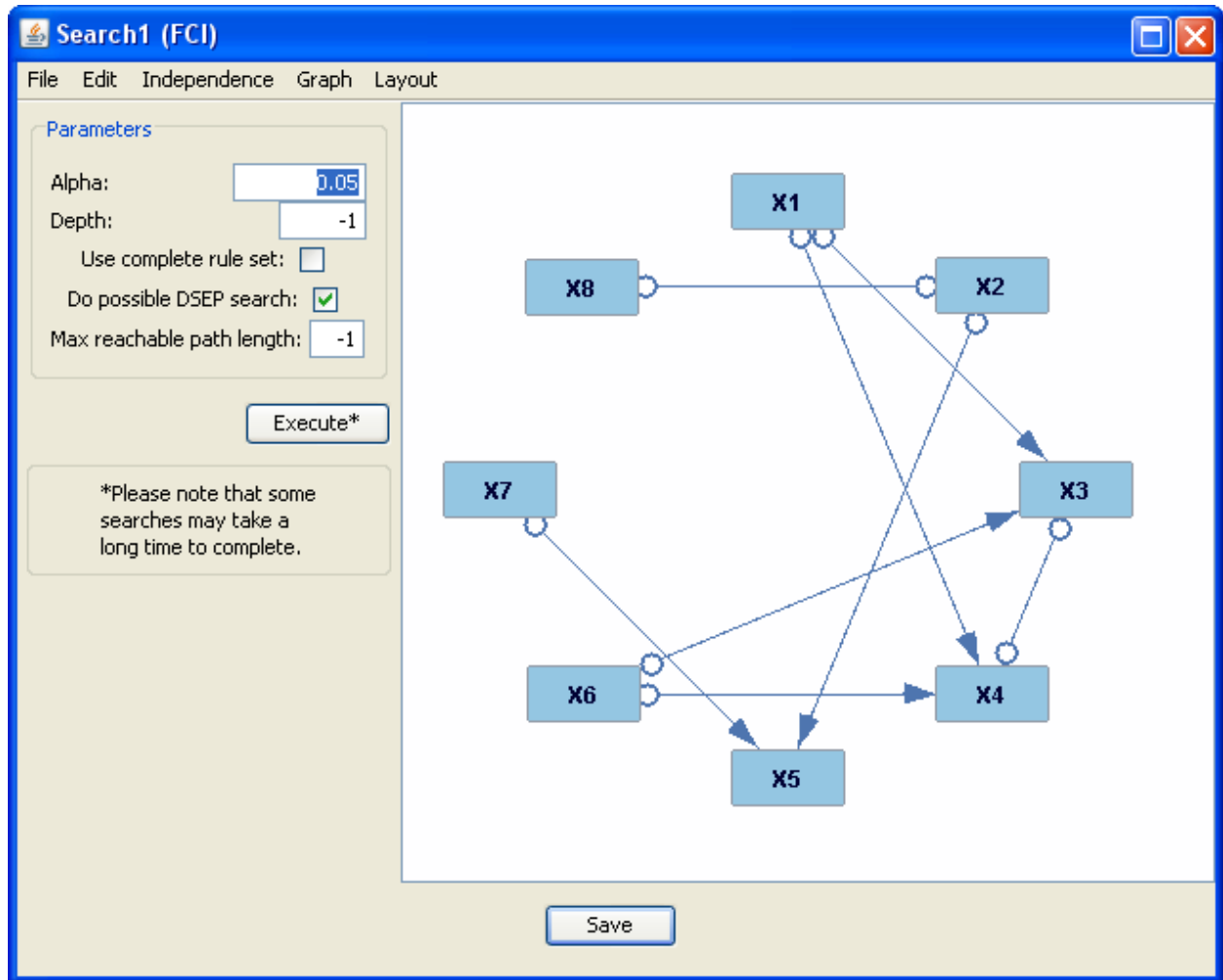
Like the PC search, the FCI search takes as input a data set whose underlying causal structure is assumed to be represented by a DAG. However, the causal structure of the data set input into the FCI algorithm may include unknown latent variables (variables not present in the data set), or sample selection bias. It is assumed that no relationship between variables is deterministic. For continuous data, the input is assumed to be a linear SEM with Normal error terms.

As in PC output, a bidirected edge between two variables indicates the presence of a latent common cause of the two. The FCI algorithm, however, also has an undetermined edge: the presence of a small circle at the end of an edge indicates that Tetrad cannot determine whether or not that edge should contain an arrow.

Consider, for example, a data set with the following underlying causal structure, where the variable represented by an oval is unrecorded:



When the data set is input into the FCI algorithm, the following window results:



This graph is considerably different from the true graph. For example, the collider at the latent variable L1 is not present; instead, Tetrad reads this as X1 and X6 directly causing X3 and X4. There are also several ambiguous edges, particularly the one between X3 and X4 and X8 and X2; Tetrad knows that these variables are causally connected, but is uncertain which causes the other, or if they are both caused by the same latent variable.

The functionality of the FCI window is much the same as that of the PC window, with the exception of the three check boxes underneath the alpha and depth parameters on the left. If the "Use complete rule set" box is left unchecked, the original FCI algorithm is run; if it is checked, the algorithm runs using the complete (and more extensive) rule set for orienting edges. The "Do possible DSEP search" option is checked by default. This operation can take a lot of time, so you may wish to uncheck this option for further executions of the algorithm. The "Max reachable path length" number indicates how long the path from a collider to a conditioning variable may be in d-separation; when it is -1, the path length is unbounded.

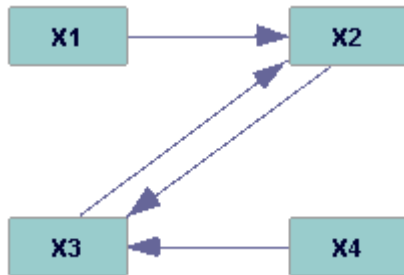
### The CFCI Search

The conservative FCI (CFCI) algorithm is analogous to the CPC algorithm, in that it runs the FCI search with a more conservative orientation algorithm for colliders. Therefore, CFCI will occasionally return ambiguous triples where FCI does not. The CFCI output window has the same functionality as the FCI output window.

### The CCD Search

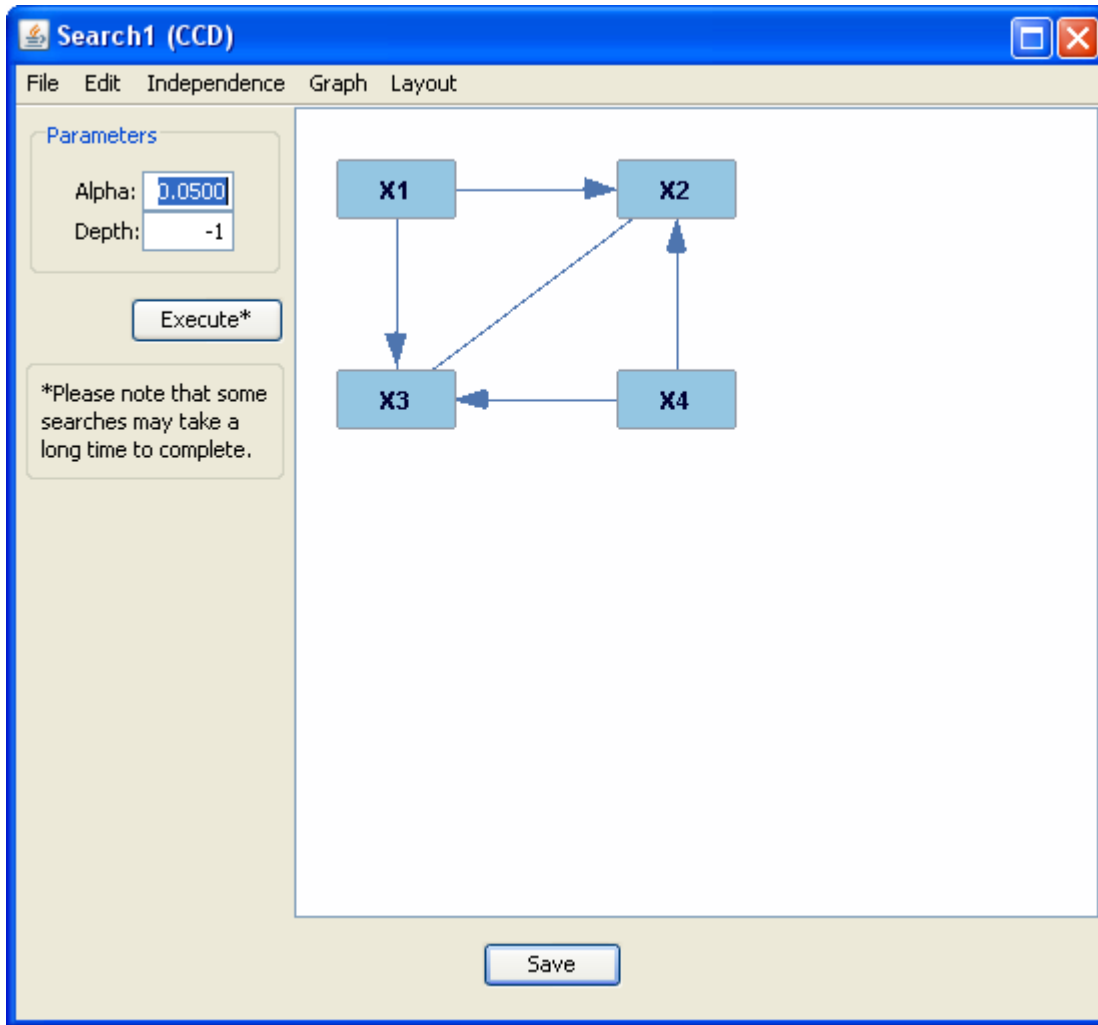
The cyclic causal discovery (CCD) algorithm runs with similar input and output to the PC algorithm, but data sets input to the CCD algorithm can have cyclic underlying causal structures. In addition, the CCD algorithm, like the FCI algorithm, sometimes outputs ambiguous edges. The output of the CCD algorithm is a d-separation equivalence class, not, in general, a Markov equivalence class. In a pattern, representing a Markov Equivalence class, every edge in the pattern, whether directed or not, occurs in every graph in the Markov Equivalence class. That is not true of CCD output because two cyclic graphs that are d-separation equivalent can have different pairs of variables that are adjacent. Every edge that is in any graph in the d-separation equivalence class will be represented in the CCD output.

For example, consider a data set with the following underlying causal structure:



When input to the CCD search, the following window results:





The output contains only one edge between X2 and X3, and refuses to orient it. The window functions much like the PC window; in particular, the “Alpha” and “Depth” parameters on the left side of the window have the same meaning they have in the PC output window.

Under the “Graph” tab, there is an option not present in the PC output window (though it is present in the CPC window) called “Underlinings.” This option presents a textual representation of all ambiguous triples in the output graph.

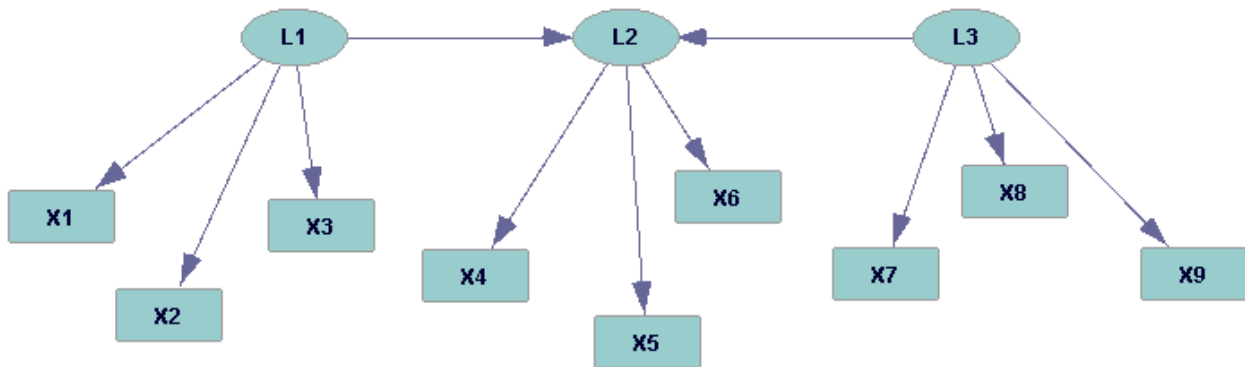
### Multiple Indicator Model Searches

The three multiple indicator model (MIM) searches, when used in conjunction, determine the presence, clusters, and causal relations of latent variables for certain classes of models. They share assumptions: relations should be approximately linear, variables should be approximately Normal, or if categorical variables are used, they should be projects of Normal variables, sampling should be i.i.d., and, as with all the searches discussed here except PCD, the recorded variables should not be deterministically related.

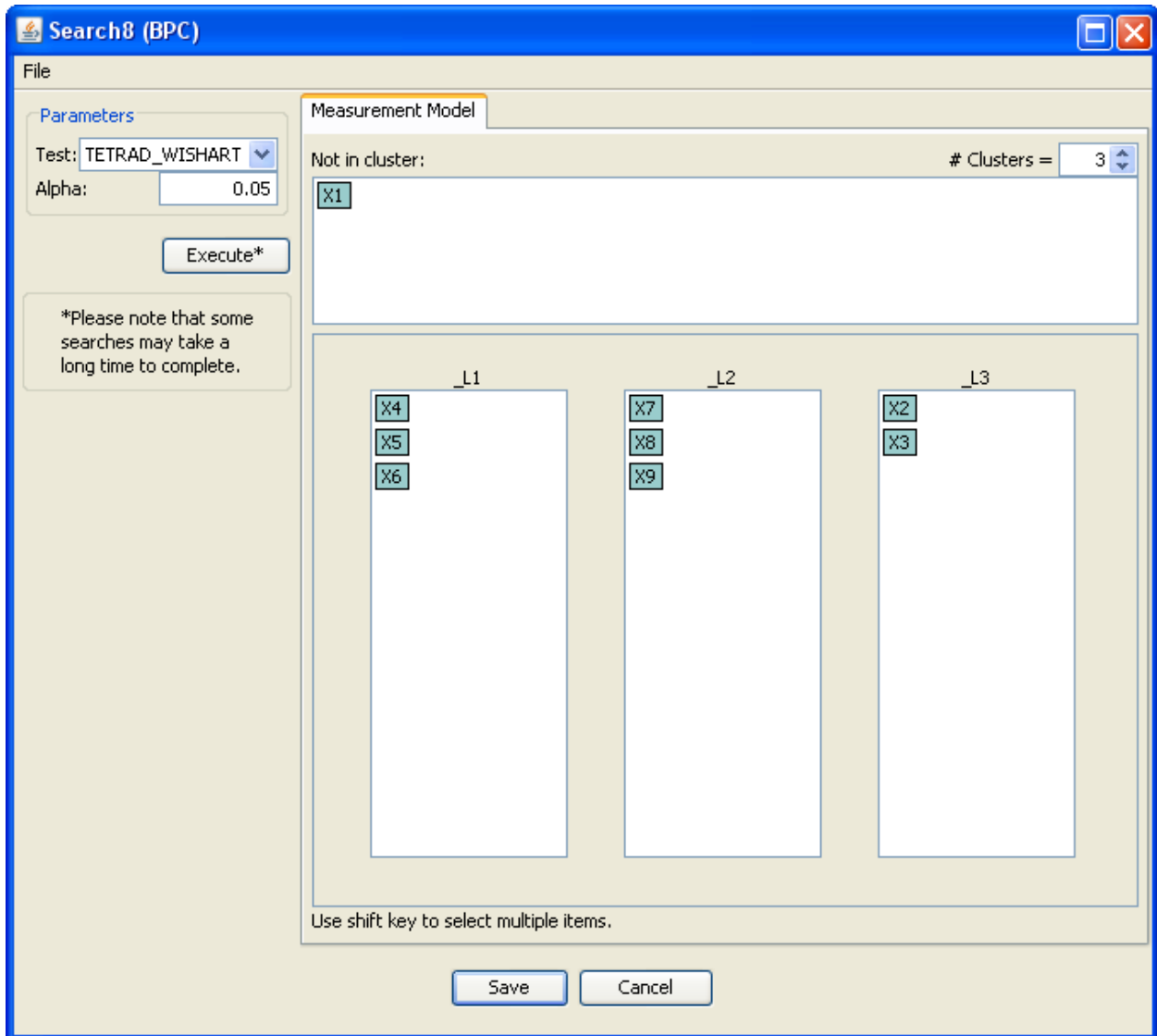
## The BPC Search

The build pure clusters (BPC) algorithm takes as input a data set in which there are known to be latent common causes between large clusters of variables, and returns a graph showing which measured variables are in which clusters. BPC uses tests for conditional independence and vanishing tetrads (see Silva, Scheines, Glymour, and Spirtes, *Journal of Machine Learning Research*, February, 2006).

Take, for example, a data set with the following underlying causal structure:



When input into the BPC search box, the following window results:



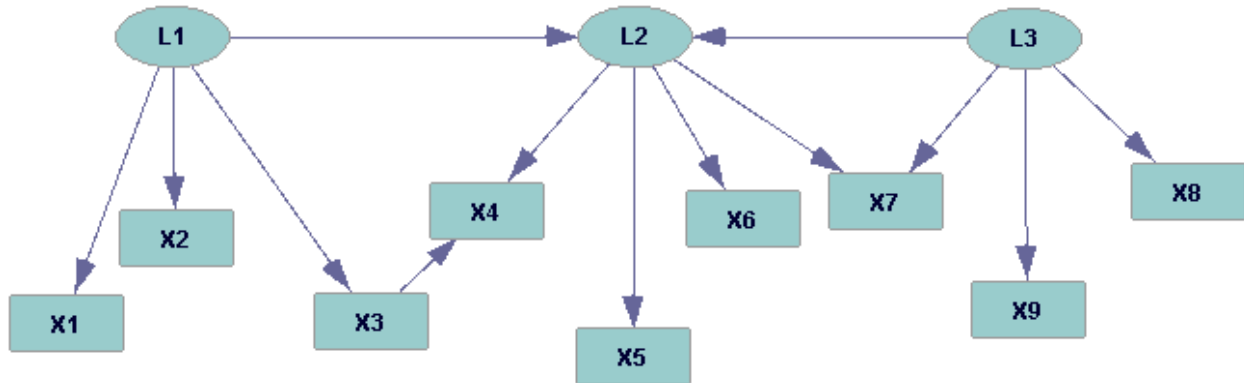
The horizontal box at the top of the window contains all variables that BPC has not placed in a cluster. The three vertical boxes beneath it represent the three latent variables that BPC has found, and the clusters they correspond to. In this case, BPC has found the cluster of all variables except for X1; by changing the Alpha parameter on the left and rerunning the search, you can occasionally improve the performance of BPC, and in this case, setting the Alpha level to 0.1 allows BPC to place X1 in the proper cluster. If you know a particular clustering to be correct, you can also manually place a variable in a cluster by clicking and dragging it to the correct box.

This box can now be used as input to a Purify or MIMBuild search.

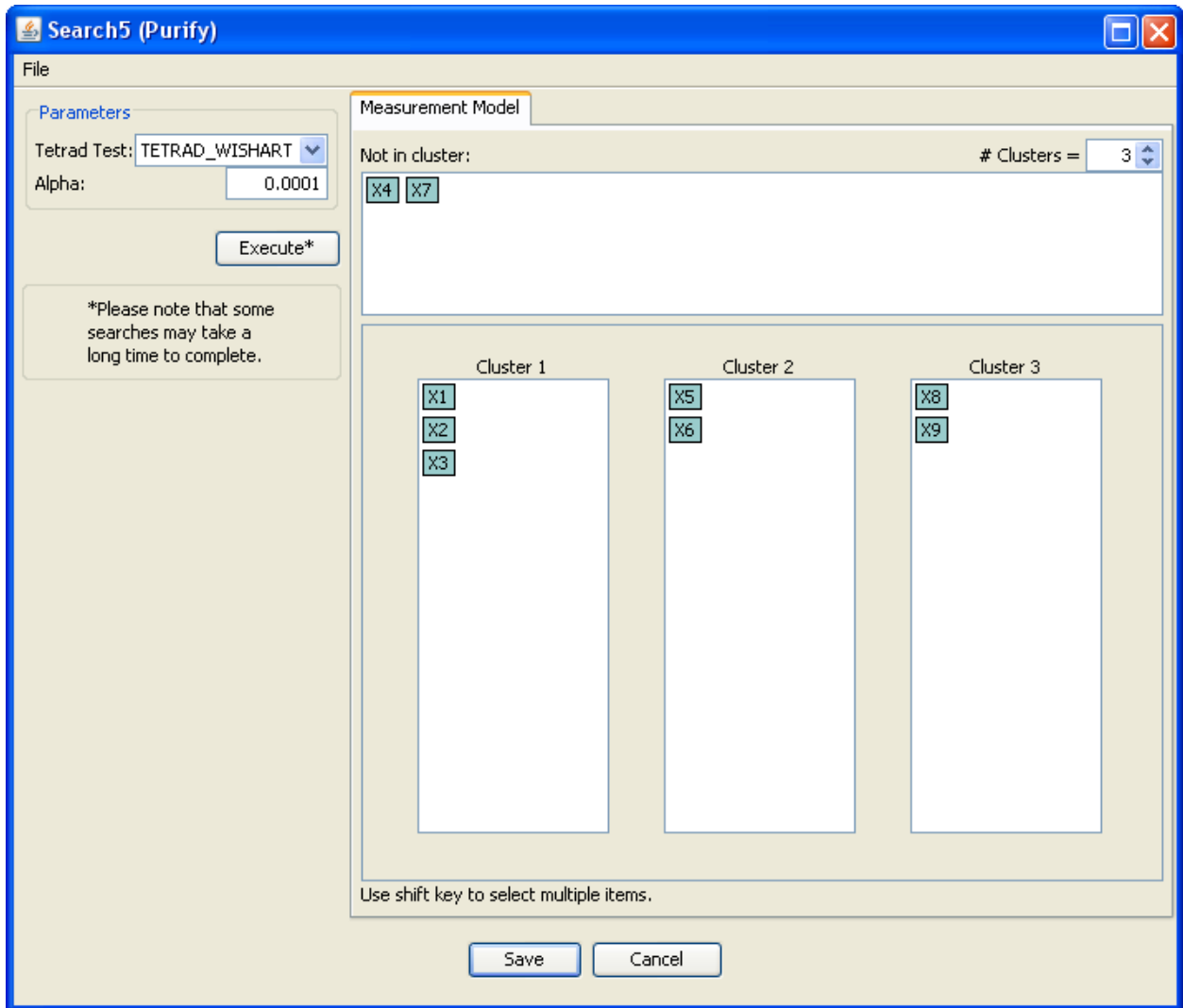
### The Purify Search

The Purify algorithm takes as input a data set in which latent variables are common causes between large clusters of measured variables, and returns a graph in which all “impurities” in the clusters have been removed: measured variables which are the effects of two or more latent variables, measured variables in one cluster which cause measured variables in another cluster, etc. An impurity is any measured variable that is causally linked to two or more clusters. Purify can be run on two kinds of input: a data set and the general graph representing its underlying causal structure, or the output of the BPC search.

Take, for example, a data set with the following underlying causal structure:



When it and the above graph are input into the Purify search, the following window results:



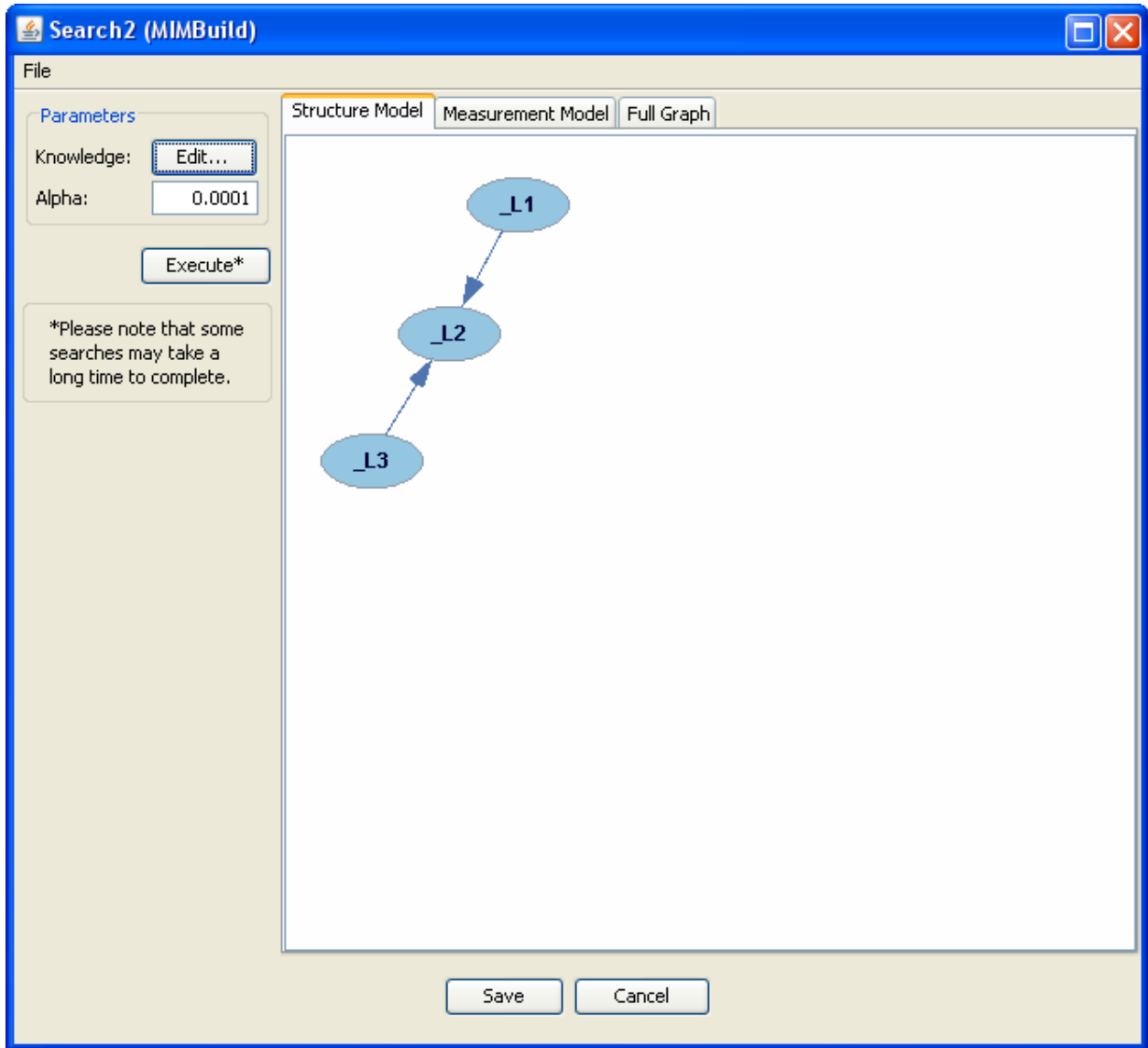
The output window for Purify works in the same way as the output window for BPC. Notice that Purify has failed to place two variables (X4 and X7) in clusters; X4 had an edge with a variable in another cluster (X3), and X7 was a member of two clusters (Cluster 2 and Cluster 3).

This box can now be used as input to a MIMBuild search.

### The MIMBuild Search

When you have a data set in which the presence and descendents of latent variables are known, the MIMBuild algorithm can be used to determine the causal relationships between the latent variables. When run on cyclic models, MIMBuild generally correctly determine adjacencies; however, it may incorrectly determine the orientation of edges. Generally, MIMBuild takes as input the output of the BPC or Purify search.

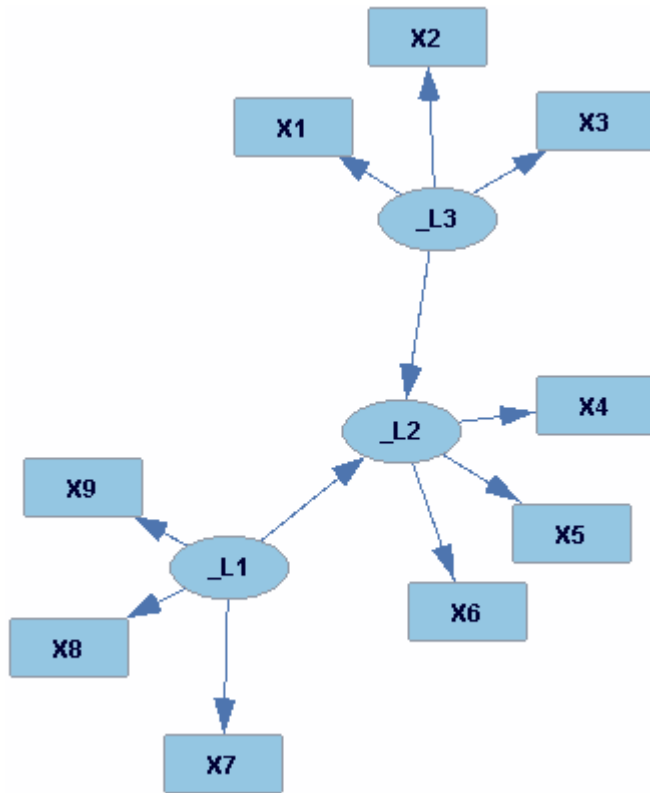
Take the output of the example used in the BPC section. When input to MIMBuild, the following window results:



The graph shown in the “Structure Model” tab shows the causal structure of the latent variables *only*. When we compare this to the true causal graph (see the BPC section) we see that this is indeed the relationship between the latent variables.

The “Measurement Model” tab shows the clusters of the graph in the same format used in the Purify output window; for more information, see the Purify and BPC sections.

The “Full Graph” tab shows the entire causal structure of the model as MIMBuild has found it, including latent and measured variables. In this case, the graph shown looks like this:



Comparing this to the true causal structure, we see that it is correct.

As in the other MIM searches, the Alpha level of this search can be changed using the text box on the left side of the window. In addition, if you have additional knowledge about the causal structure of the latent variables (such as required or forbidden edges between them) you can input that knowledge to MIMBuild using the “Edit...” button on the left side of the graph. The resulting window functions exactly as the knowledge box does (see the knowledge box section for more details). As knowledge about the measured variables should have been input before reaching MIMBuild, only knowledge about latent variables can be input here.

## Feature Selection Searches

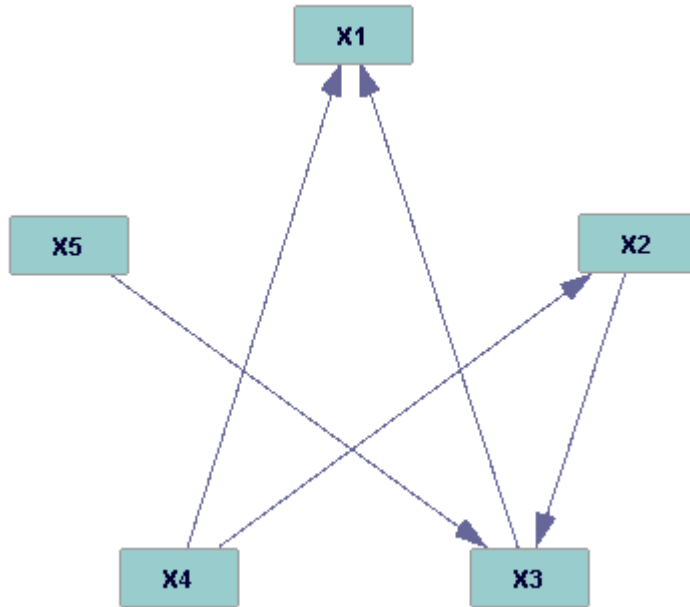
Feature selection searches extract small sections of the causal structure of a larger data set.

### The MBFS Search

The Markov Blanket Fan Search (MBFS) searches for and displays the graph of the Markov blanket of a variable in a data set. The Markov blanket of a variable X in a DAG is the smallest set of variables conditional on which X is independent of all other variables in graph

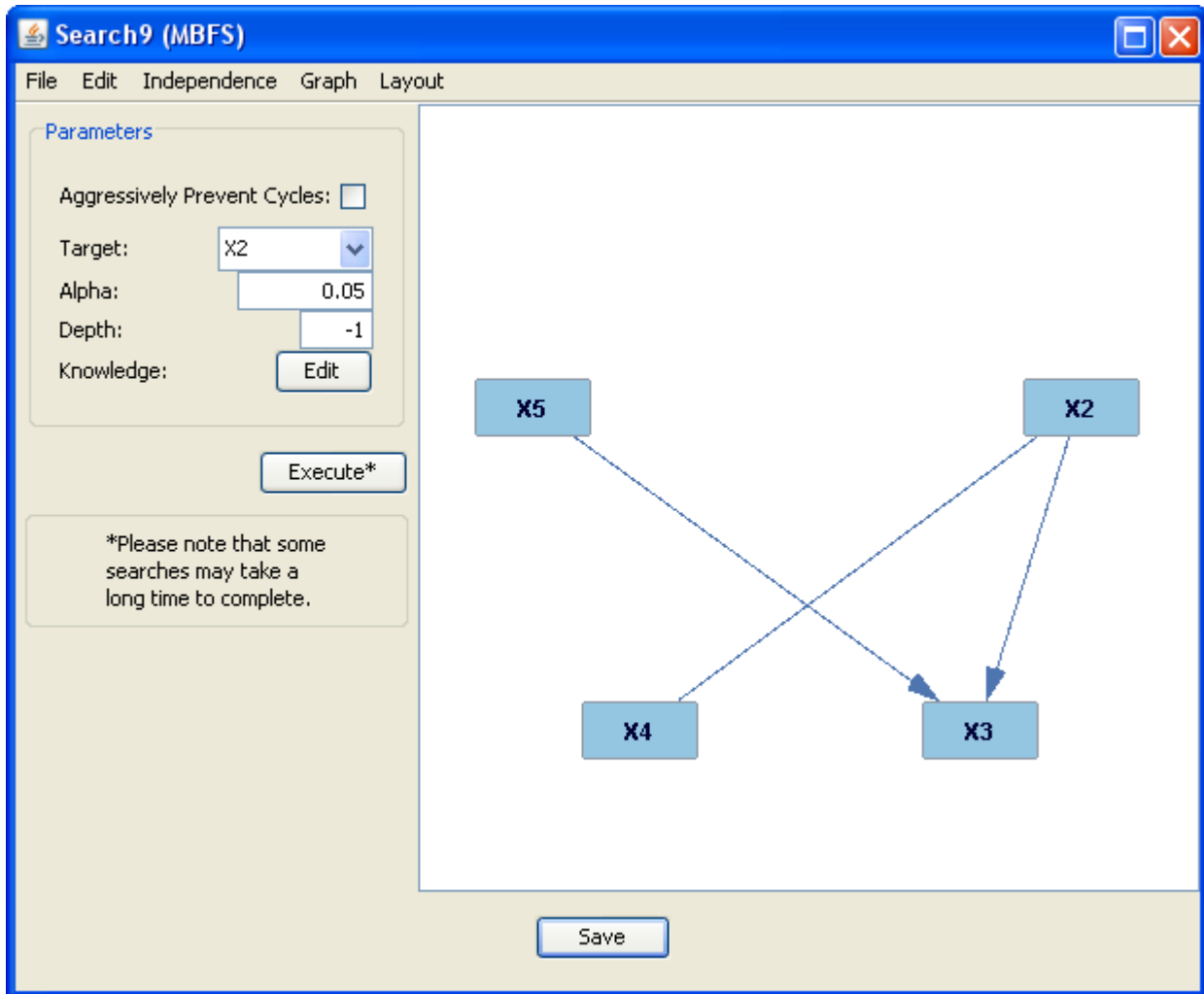
(except X itself, of course). Markov blankets are therefore an ideal subset of variables to use in classification or prediction of a target variable when the value of the target is unknown but values of other variables have been recorded for a new case or collection of cases.

Take, for example, a data set with the following underlying causal structure:



When this data set is input into MBFS with X2 as the target variable, the following window results:





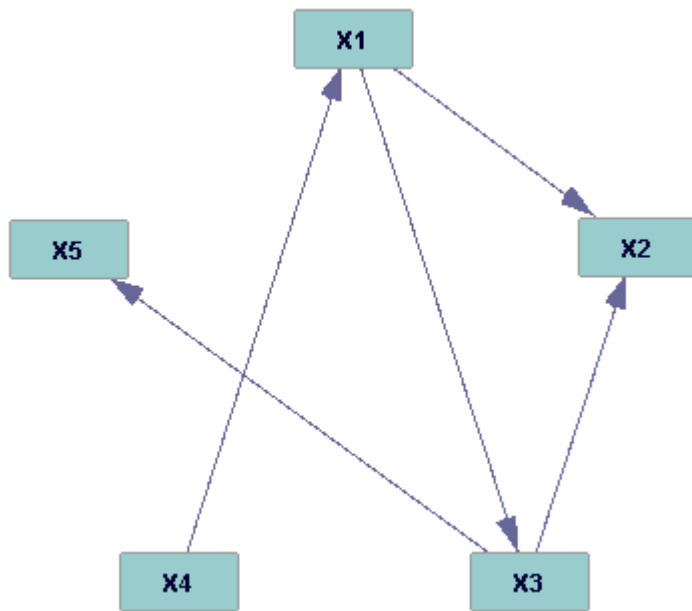
Like other searches, MBFS is not always certain of the orientation of the edges in its output; in this case, it cannot determine whether X4 causes X2 or vice versa, but either way, X4 belongs in the Markov blanket.

As in the PC output window, you can change the Alpha and Depth values of the search in the text boxes on the left. You can also rerun the search with a different target variable, or add knowledge. When you press the “Edit” button, a window appears which functions exactly like the knowledge box. All of the other functionality of the window is the same as that of the PC or FCI output windows.

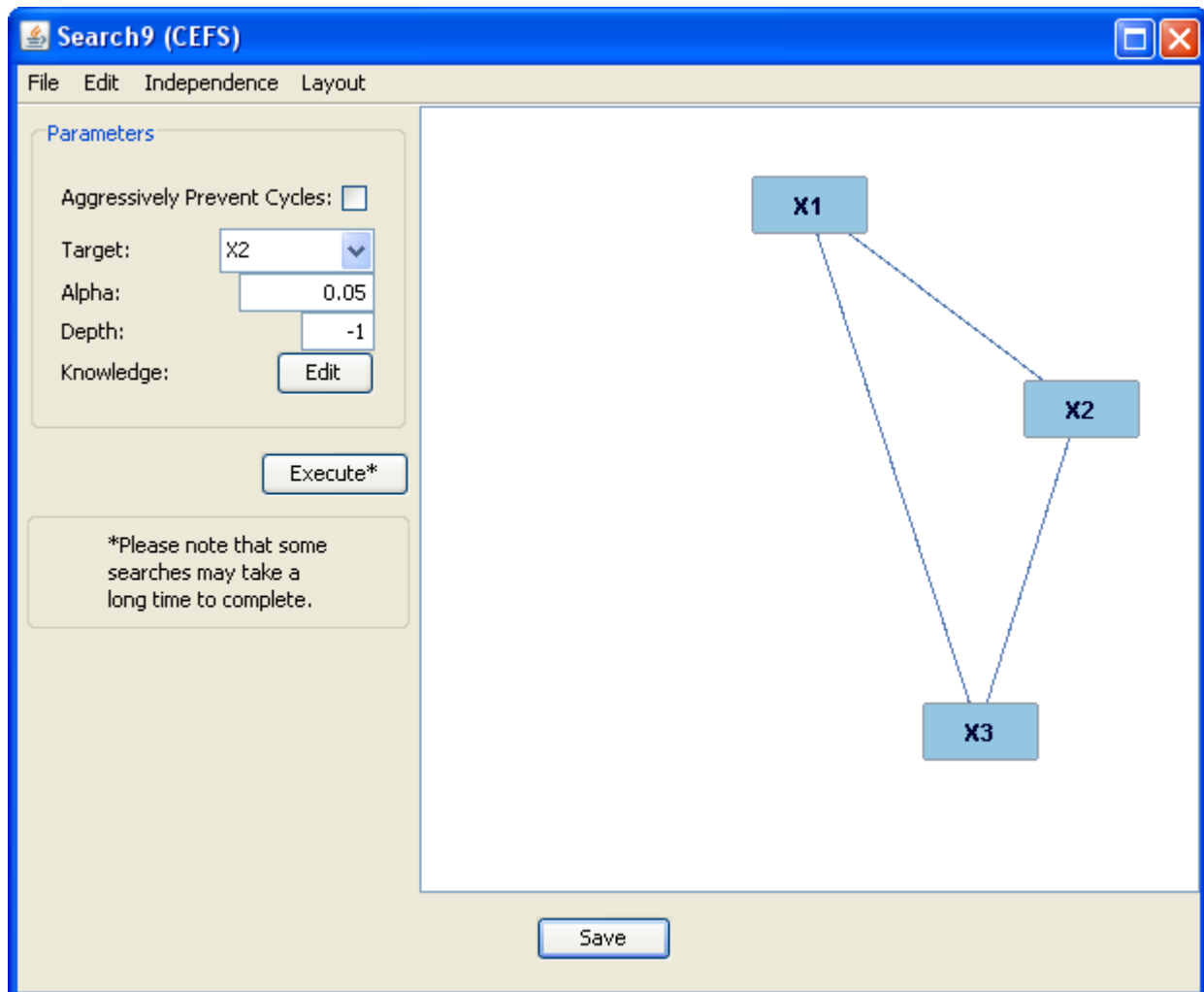
### The CEFS Search

The Causal Environment Fan Search (CBFS) searches for and displays the graph of the “causal environment” of a variable in a data set. (The causal environment of a variable is that variable’s parents and its children.)

Take, for example, a data set with the following underlying causal structure:



When input into CEFS with X2 as the target variable, the following window results:



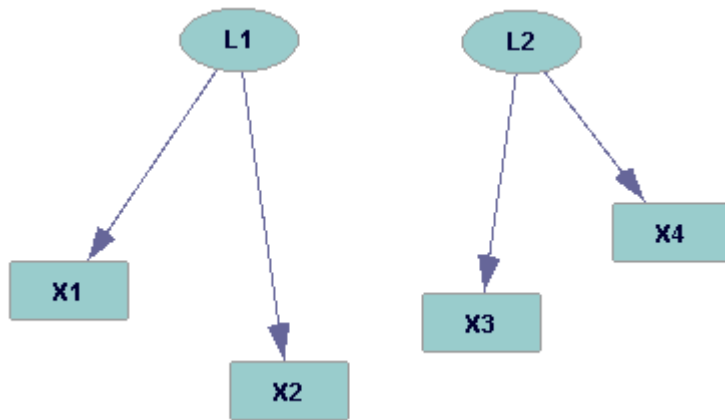
As you can see, CEFS is not always certain of the orientation of edges; however, regardless of orientation, X1 and X3 should be in the graph. The CEFS output window parameters and tabs function just as the MBFS parameters and tabs do.

## Factor Analysis

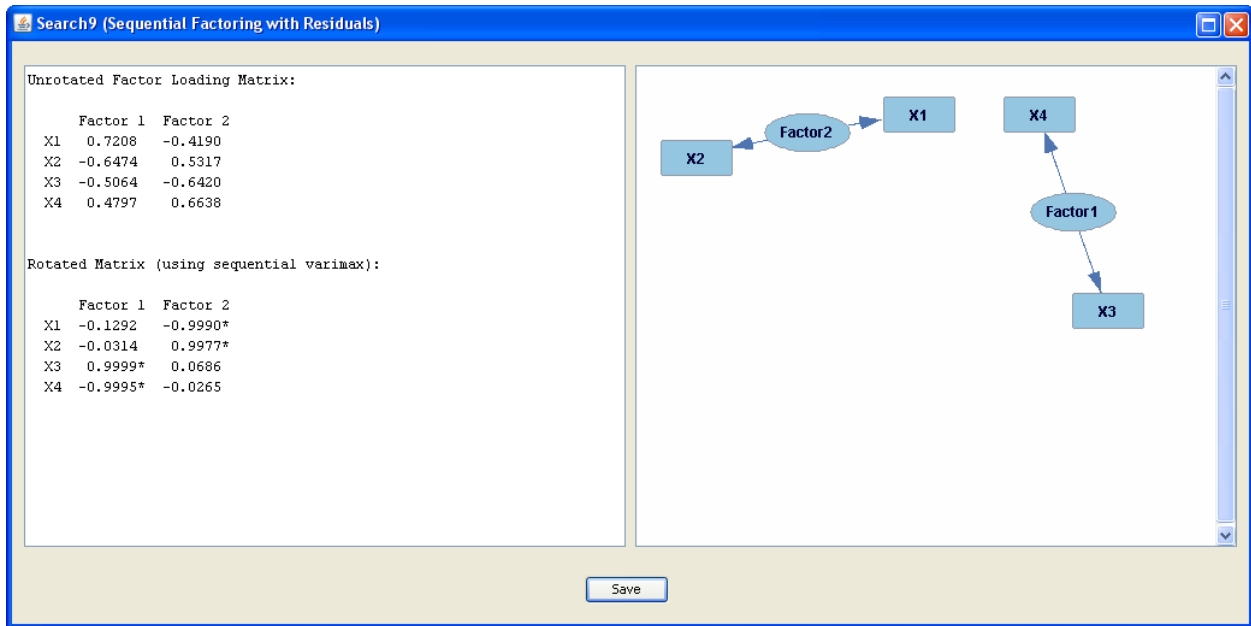
### Sequential Factoring with Residuals

Sequential factoring with residuals (SFR), like BPC, attempts to find clusters of variables in a data set with a latent common cause. It is a less accurate algorithm than BPC, and will occasionally find clusters where there are none. However, when clusters are present and clearly distinguished, it is reasonable accurate.

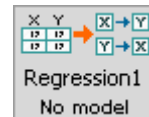
Take, for example, a data set with the following underlying causal structure:



When this data set is input into SFR, the following window results:



On the right is SFR's guess as to which variables are influenced by common factors. In this rather straightforward case, SFR is accurate. On the left are the factor loadings for the unrotated and rotated solutions.



The regression box in the main workspace looks like this:

### **Possible Parent Boxes of the Regression Box:**

- A data box
- A data manipulation box

### **Possible Child Boxes of the Regression Box:**

- A graph box
- A graph manipulation box
- A comparison box
- A parametric model box
- A search box

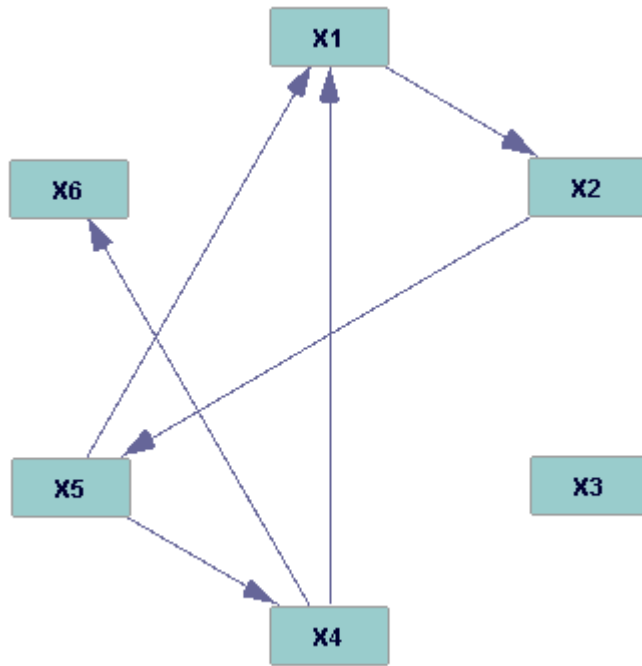
### **Using the Regression Box:**

The regression box performs regression on variables in a data set, in an attempt to discover causal correlations between them. Both linear and logistic regression are available in the regression box.

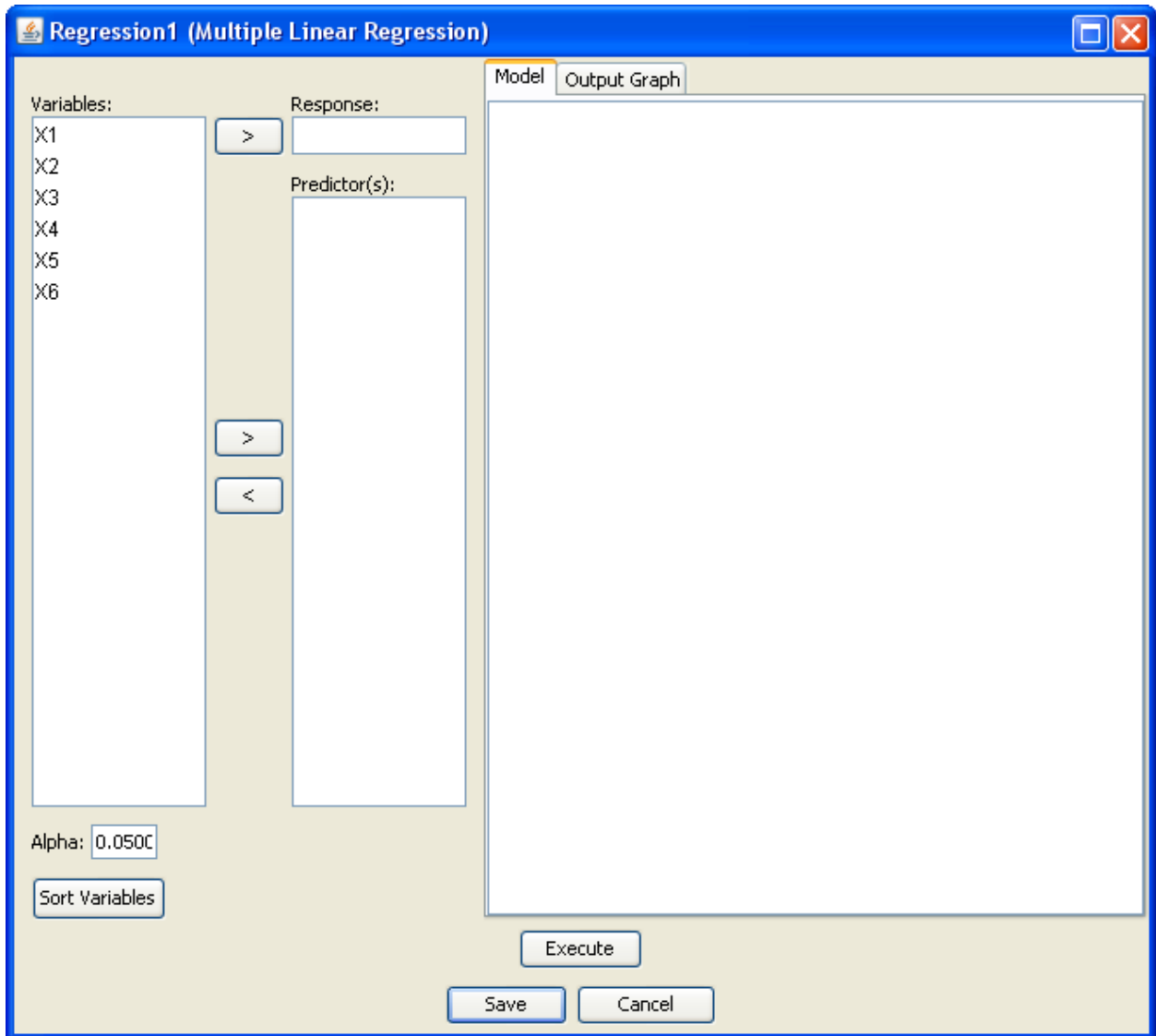
### Multiple Linear Regression

Linear regression is performed upon continuous data sets. If you have a categorical data set upon which you would like to perform linear regression, you can make it continuous using the data manipulation box.

Take, for example, a data set with the following underlying causal structure:



When used as input to the linear regression box, the following window results:



To select a variable as the response variable, click on it in the leftmost box, and then click on the top right-pointing arrow. If you change your mind about which variable should be the response variable, simply click on another variable and click on the arrow again. To select a variable as a predictor variable, click on it in the leftmost box, and then click on the second right-pointing arrow. To remove a predictor variable, click on it in the predictor box and then click on the left-pointing arrow.

Clicking “Sort Variables” rearranges the variables in the predictor box so that they follow the same order they did in the leftmost box. The alpha value in the lower left corner is a threshold for independence; the higher it is set, the less discerning Tetrad is when determining the independence of two variables.

When we click “Execute,” the results of the regression appear in the box to the right. For each predictor variable, Tetrad lists the standard error, t value, and p value, and whether its correlation with the response variable is significant.

The Output Graph tab contains a graphical model of the information contained in the Model tab. For the case in which X4 is the response variable and X1, X2, and X3 are the predictors, Tetrad finds that only X1 is significant, and the output graph looks like this:



Comparison to the true causal model shows that this correlation *does* exist, but that it runs in the opposite direction.

### Logistic Regression

Logistic regression may be run on discrete, continuous, or mixed data sets; however, the response variable must be binary (and therefore discrete). In all other ways, however, the logistic regression box functions like the linear regression box.