

Introduction

PRONEL© is an European project ESPRIT performed by four societies: Siemens AG (Germany), Schlumberger (France), Hugin (Denmark), and Tiga Technologies (France). The aim of this project is to propose a software, which provided a Bayesian network construction by using database and expert knowledge in order to help the decision maker. The Pronel BN Learner is a tool for constructing Bayesian networks from data. With the Pronel BN Learner tool you can load a database table and select the columns you want to represent variables/nodes in the Bayesian network. Then, you can perform a structural learning task that will give you a Bayesian network structure which may contain some uncertainty. Through some simple interactions with the user, the final Bayesian network structure is determined after which the underlying probabilities can be learned.

Starting the Pronel BN Learner Tool

When you start the Pronel BN Learner tool, the main window opens without any data loaded and everything looks empty. In Figure 1 the main window is shown.

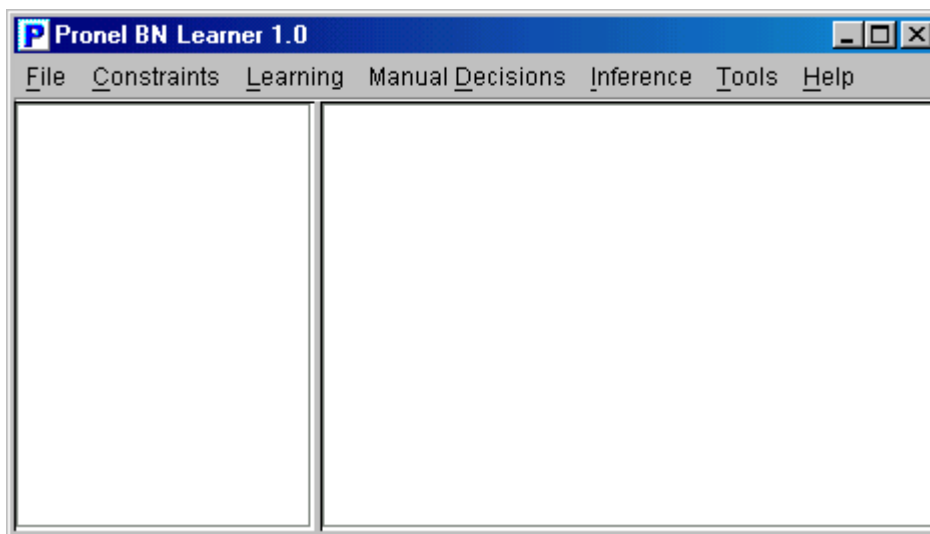


Figure 1: The main window before data is loaded.

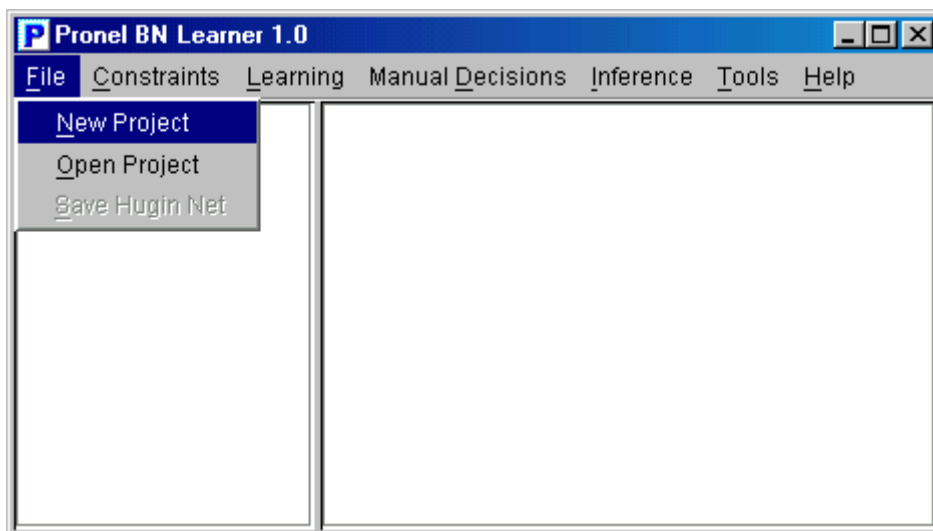


Figure 2: Create a project or load an existing one.

Figure 2 shows the 2 possibilities : to open an existing project by using the Open Project menu, or to create another one by selecting New Project menu.

Load Data

The first dialog of the database preprocessor is the "Load Data" dialog shown in Figure 3. Here you can open a text database by providing a name in the "Text file to open" field. You can also specify an installed ODBC database by clicking "Open DataBase". There are also a number of different descriptive fields you can fill in if you like.

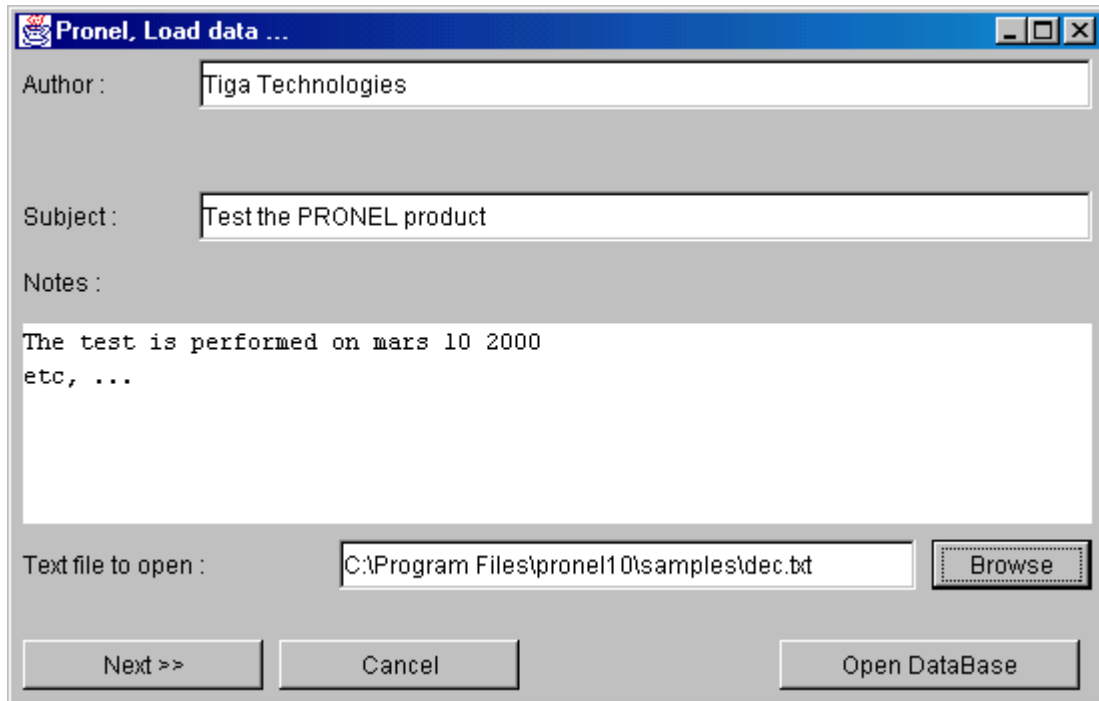


Figure 3: Specifying the database and project information.

If you click the Open DataBase button (Figure 4), a dialogue box appears an ask you to enter the Database source name and the name of the table you want.

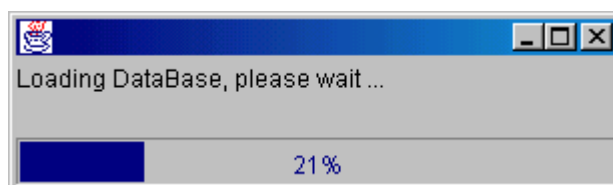
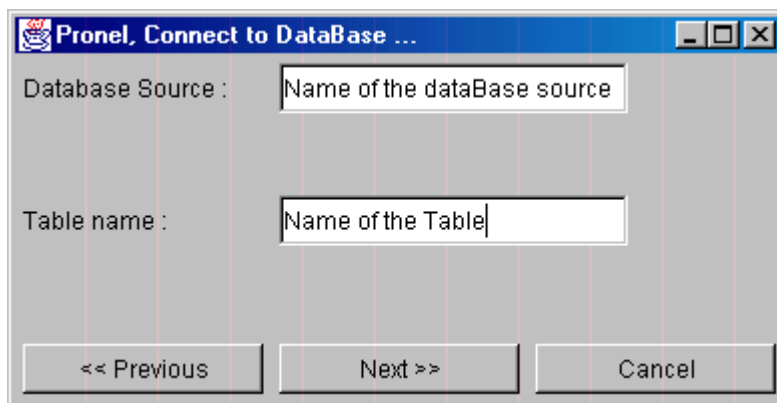


Figure 4: Loading database.

Specify Separators

After data has been loaded, the next step is to specify how the database columns are separated. This is shown in Figure 5. Use one of the standard separators or specify another one - here the space character.

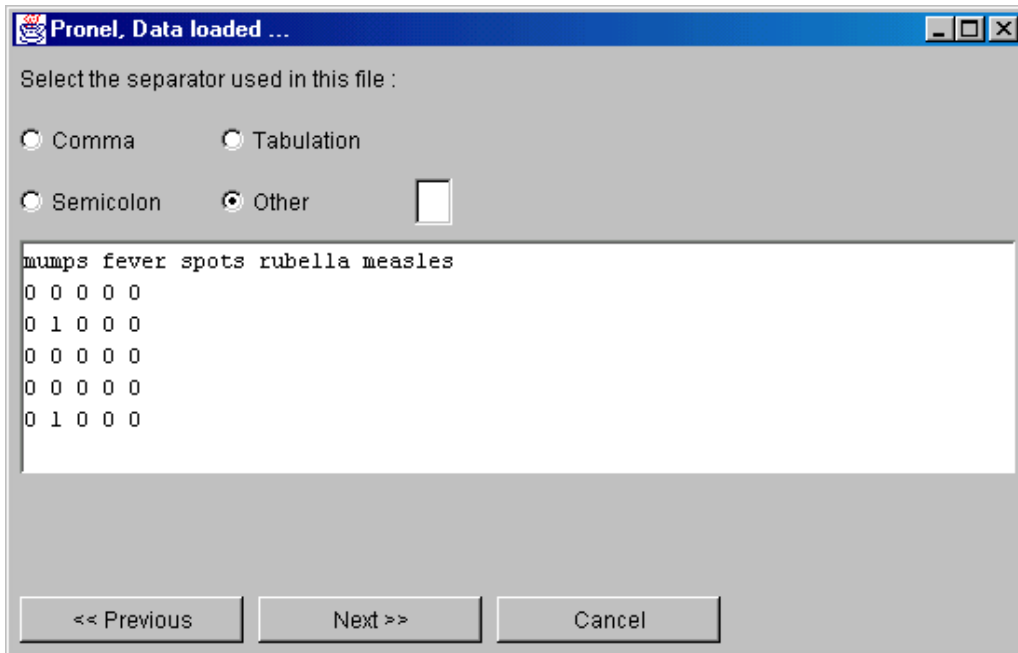


Figure 5: Specify column separators in the text database.

This will allow the preprocessor to identify the different columns each corresponding to a variable/node in the Bayesian network. The next dialog will allow you to specify only some of the columns.

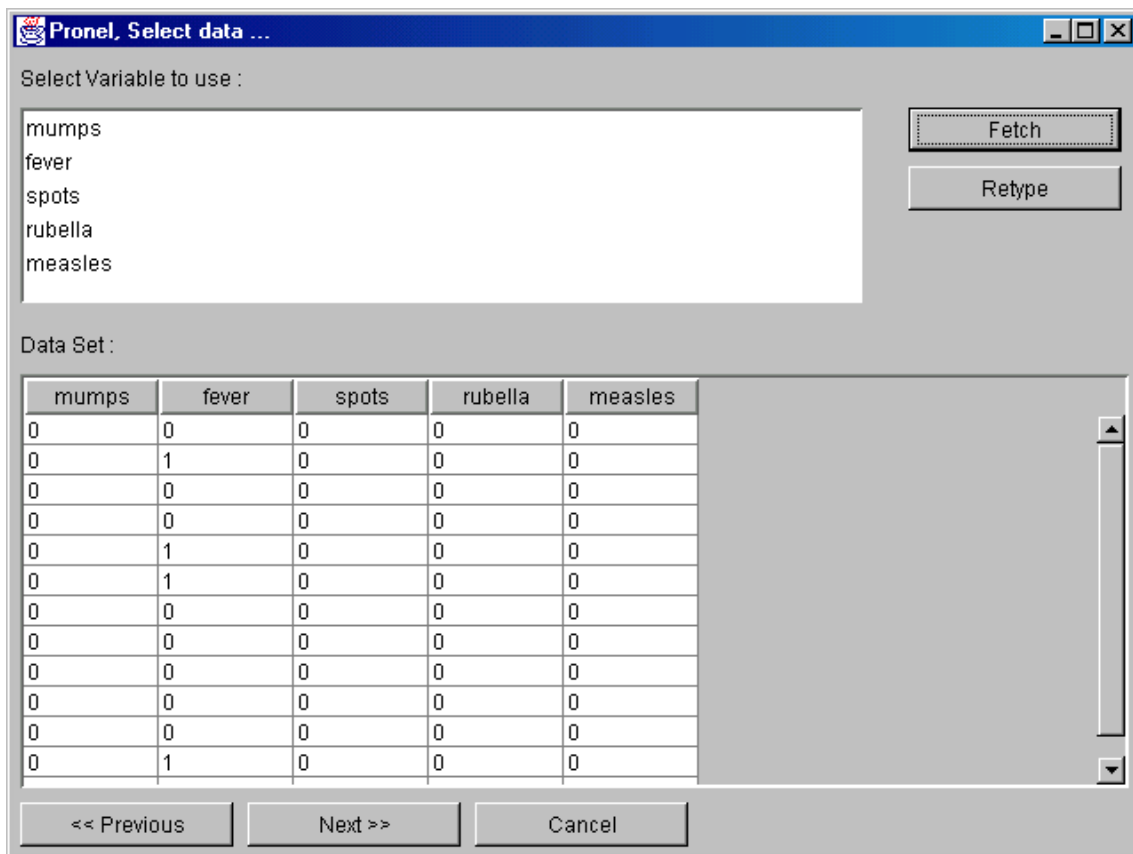


Figure 6: Fetching the nodes of interest.

All columns in the database initially correspond to variables/nodes in the Bayesian network. In the Select Data dialog (Figure 6), you can fetch exactly the nodes that you want.

When pressing the "Next" button, the data will be processed to determine the possible states of the variables/nodes of the Bayesian network. The result is a set of nodes without causal links on which learning can take place.

Learning the Structure

There are three steps in the structural learning process:

1. Specify constraints
2. *Learn* the structure
3. Resolve uncertain regions

Actually, only step 2 is the real learning step. The other steps are user interaction steps to assist the learning algorithm to get as good results as possible.

Step 1: Specify Constraints

Figure 1 shows the Pronel BN Learner tool after data has been loaded and before any learning has taken place. Now, we are able to tell the tool what we know about the domain we currently work with. This will help the learning algorithm in determining the right structure because it would decrease the number of possible edge combinations that it would otherwise have to check to determine the right structure.

E.g. imagine in the domain of a small diagnostic system for child deceases we want to specify that rubella is known to cause fever. Then, we should add a constraint link from “rubella” to “fever”.

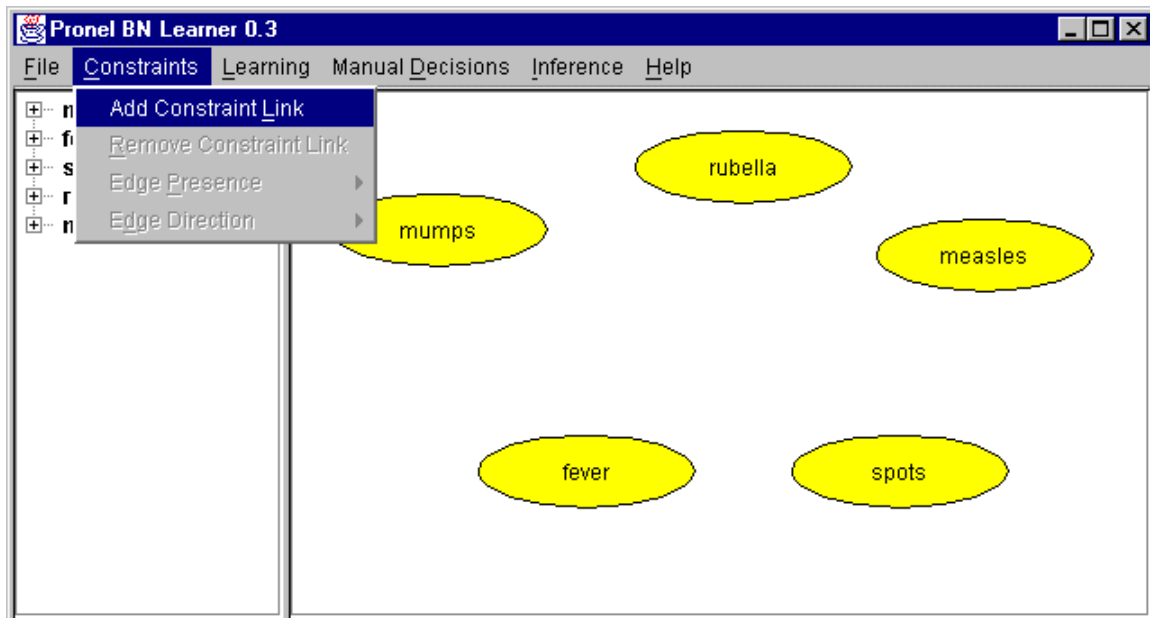


Figure 1: Adding a constraint link.

We do this by selecting “Add Constraint Link” from the “Constraints” menu and dragging the link with the mouse pointer.

Another constraint could be that we assume that the two decease nodes “rubella” and “measles” are not linked because they are assumed not to be able to cause each other.

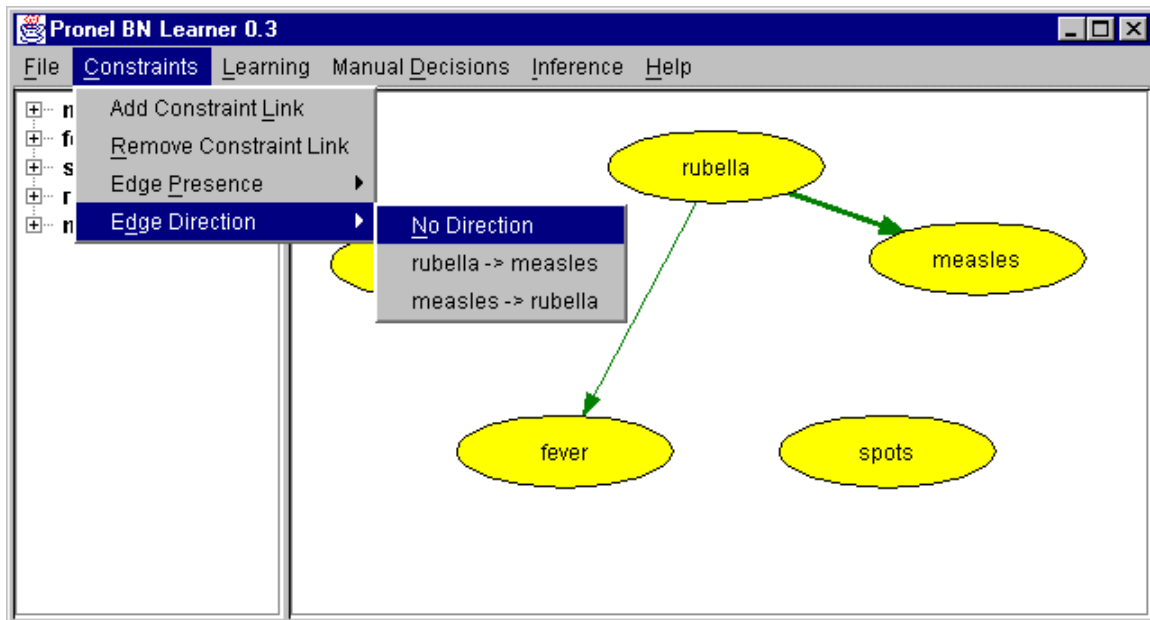


Figure 2: The constraint between “rubella” and “measles” should not have a direction.

To specify this, first add the constraint link as before. Then (as shown in Figure 2), set the direction of the link to “No Direction” by using the “Edge Direction” submenu of the “Constraints” menu. After this (as shown in Figure 3), select “Forbidden” from the “Edge Presence” submenu of the “Constraints” menu.

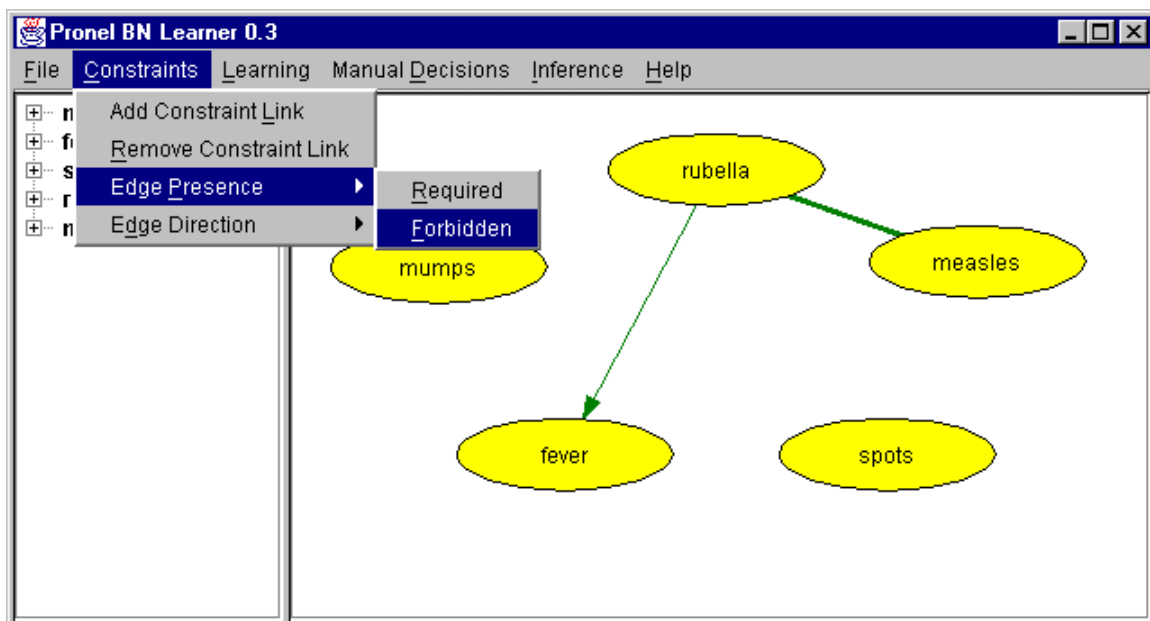


Figure 3: Set the presence of the constraint link to “Forbidden”.

The constraints representing “required” edges are color-coded green while “forbidden” edges are red.

Step 2: Learn the Structure

When no more constraints could be specified it is time to let the tool do its job and *learn the structure* from the data. This is where the most advanced programming has been done, but in the user interface of the Pronel BN Learner tool, it only takes a single menu selection to go through it (that’s how it often is in software development).

Figure 4 shows the Pronel BN Learner tool after a few constraints have been added. Now, the user wants to activate the structural learning algorithms. This is done by selecting “Learn Structure” from the “Learning” menu.

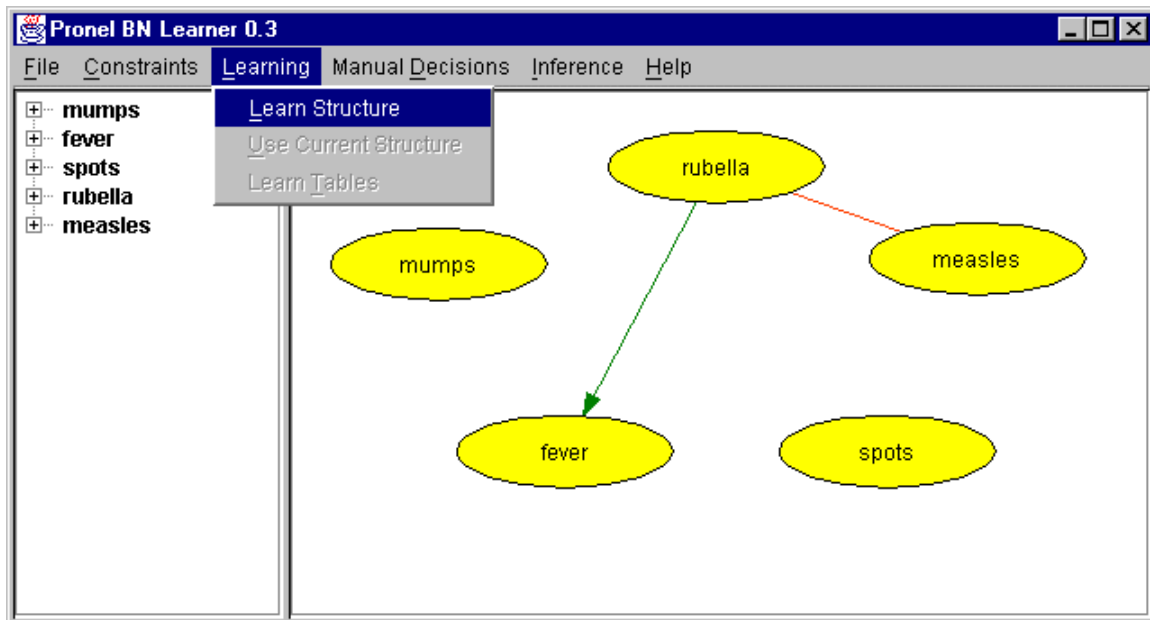


Figure 4: Structural learning is activated from the “Learning” menu.

After the structural learning algorithms have done their work, a structure has been generated. This is shown in Figure 5. Because of the constraints this structure is actually ready to use. There are no uncertainties, and we would be able to go directly to the inference part. However, we shall jump back and see what happened if no constraints were specified.

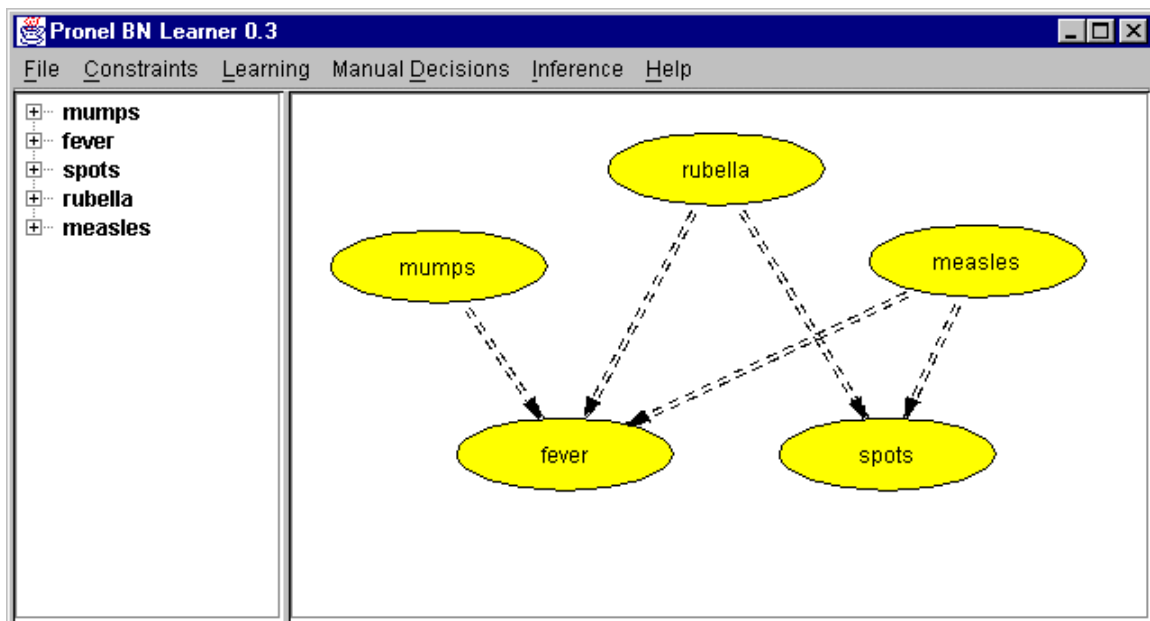


Figure 5: The learning process have build a structure based on the data and constraints.

Let’s assume we didn’t specify any constraints at all. Then, the structural learning algorithms would have less knowledge about the domain, and more uncertainty in the resulting structure would be expected.

Figure 6 shows the learned structure of the domain when no constraints had been specified. (Probably) because of too little data, the learning algorithms have not been able to determine all edges and their directions. In the display the double line edges represent certain edges where the direction may or may not be determined. An example of a certain edge where the direction is not determined is the “rubella”-“spots” edge. Uncertain edges are represented by a single line edge. These are also color-coded to tell which are related. We call this *uncertain regions*. The learned structure can have several uncertain regions but in the case shown here, only one is present: The blue region. The links of an uncertain region are related in the sence that if one is known to be certain, the others would probably not be present any more (all relationships in the data between the nodes can be explained by the certain link).

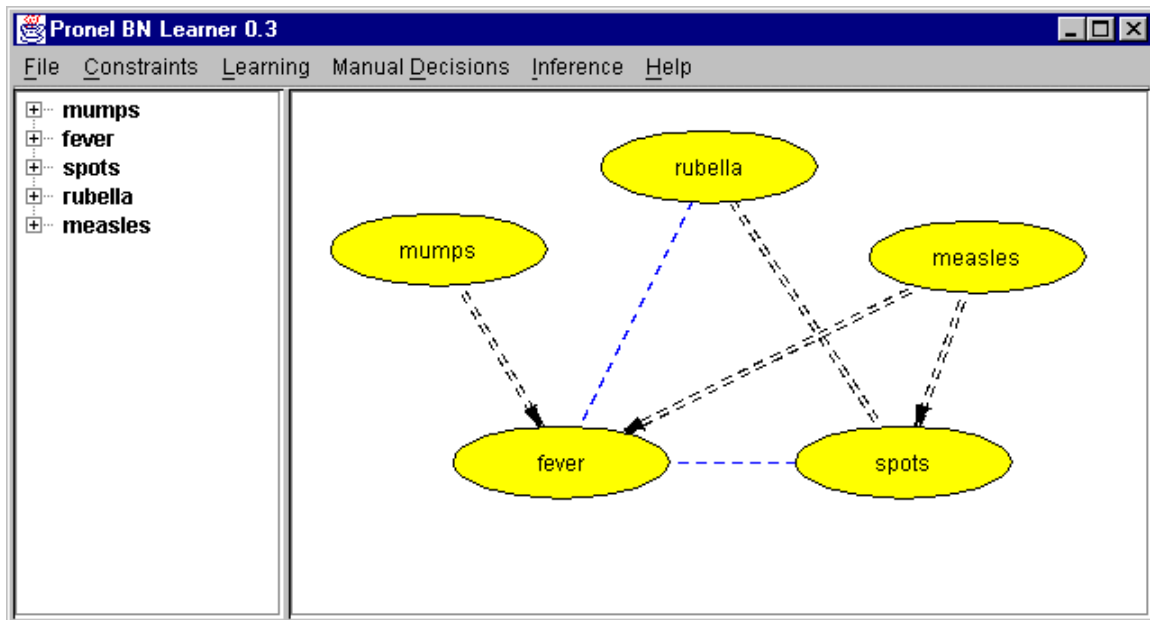


Figure 6: The result of learning the structure without constraints. Double line edges represent certain links (directed or undirected), and single line edges represent uncertain links. Uncertain links always belong to an *uncertain region*. These are color-coded each in their own color. In this case, only a blue uncertain region is present.

Step 3: Resolve Uncertain Regions

To get a proper Bayesian network, you must resolve the uncertain regions. You do this through the actions found in the “Manual Decisions” menu.

In Figure 7, the uncertain link between “rubella” and “fever” has been selected and now the direction is set through the “Set Direction” submenu of the “Manual Decisions” menu. By setting the direction, you also indirectly specify that the link is certain. Doing this will result in a *propagation* which will run through the graph to see if this solves any other uncertainties in the network (in this case, we would at least expect that the blue region would be resolved).

Figure 8 shows the network after setting the direction of the link from “rubelle” to “fever”. The result is similar to the network in Figure 5. This shows that the same can be achieved by using constraints before the learning step and by using manual decisions after the learning step. To get the best results – especially in cases where data is sparse – it is recommended that the user sets as many constraints as possible before learning.

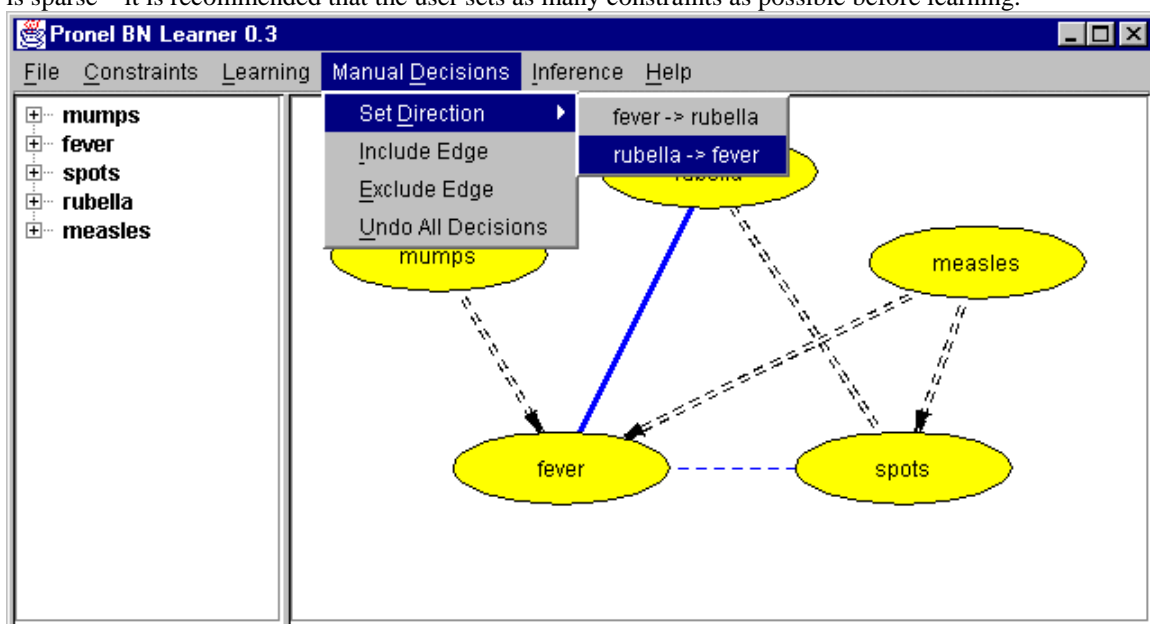


Figure 7: The direction of the uncertain link from “rubella” to “fever” is set.

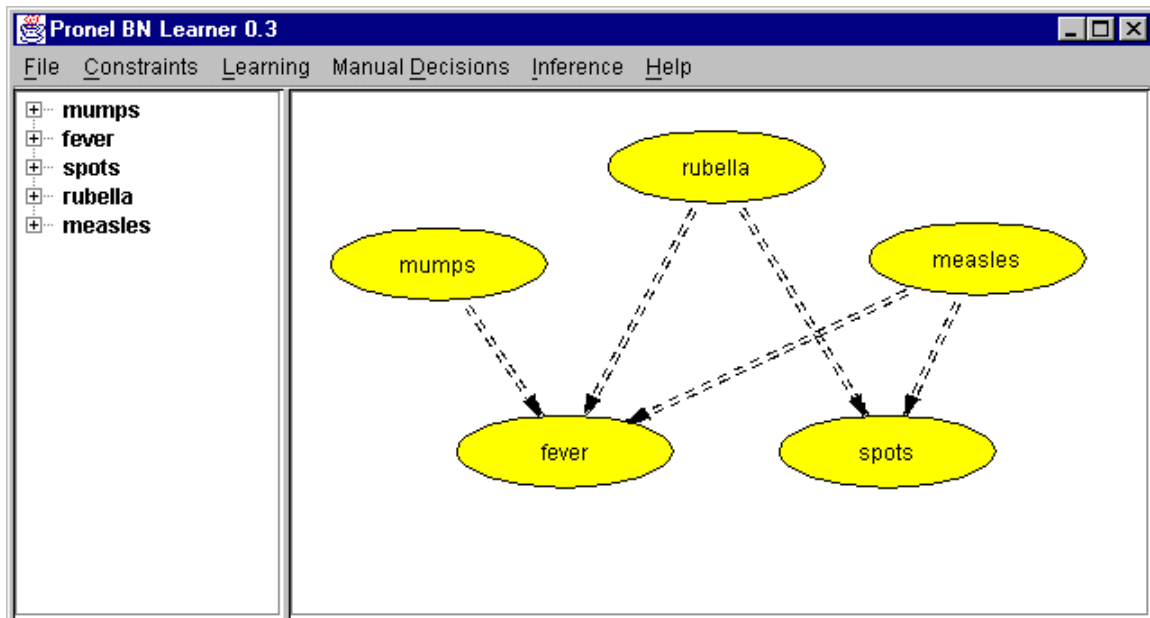


Figure 8: The result of setting the direction of the link from “rubella” to “fever”. This resolves everything and a network similar to the one in Figure 5 has been generated.

Learning Probability Tables and Using the Network

When the structure of the Bayesian network has been determined it is time to learn the initial probabilities. The first step is to tell the system that we want to use the current structure. This is done by selecting “Use Current Structure” from the “Learning” menu (see Figure 9).

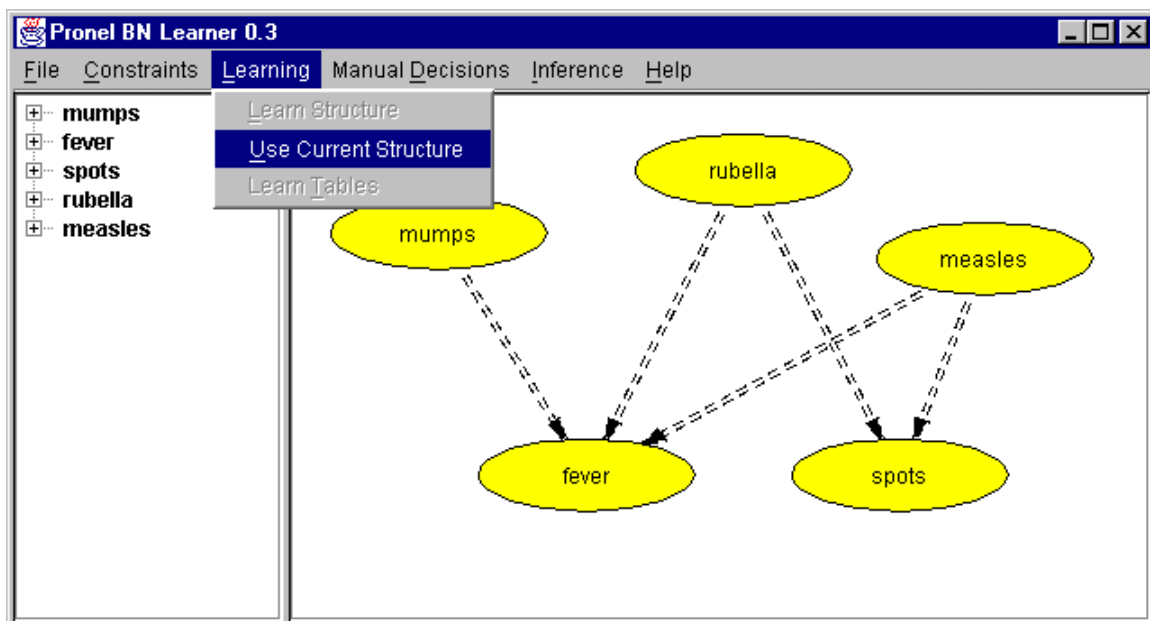


Figure 9: The current structure is selected as the one to work with.

After this, the display of edges changes. Now, all edges are directed and they are single line and solid. To get a usefull Bayesian network, the only thing that remains is learning the probability tables. This is done by selecting “Learn Tables” from the “Learning” menu (see Figure 10).

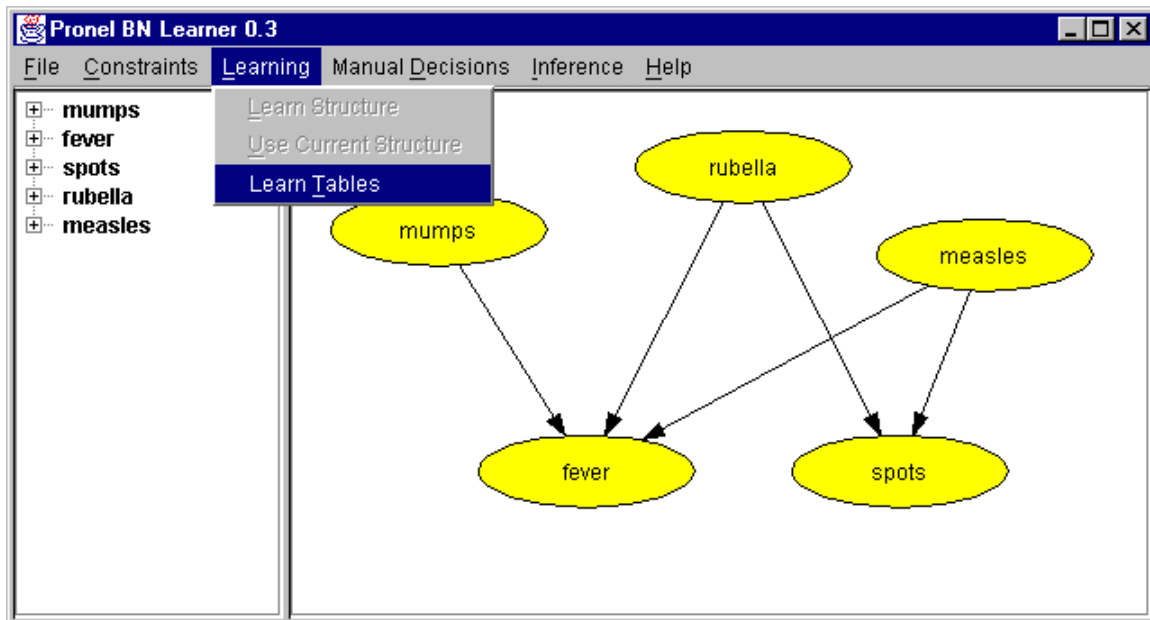


Figure 10: The probability tables are learned from data by selecting “Learn Tables” from the “Learning” menu.

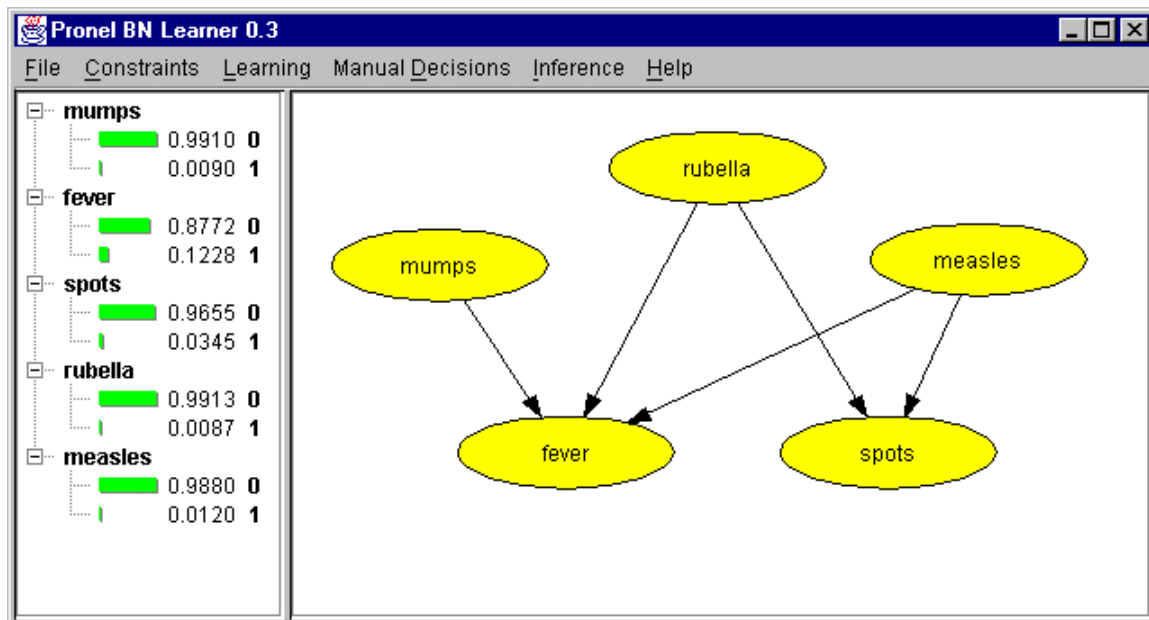


Figure 11: When probability tables have been learned, the initial probabilities of all nodes can be read in the node list on the left.

The Bayesian network shown in Figure 11 is ready to use. Now, you can choose either to use it within the Pronel tool as it is, or to export it in Hugin NET format which can be loaded from the Hugin tool. If you only want to use it as it is, the Pronel BN Learner tool has the basic features to enter evidence and propagate it to update probabilities. If you want to edit it further or investigate the generated tables, you can do this by loading the network with the Hugin tool. *Exporting to Hugin NET format* is done by choosing “Save Hugin Net” from the “File” menu.

Figure 12 shows how evidence is entered. Observed states can be selected either by double-clicking them in the leftmost *node list* pane or by selecting them (with a single click) and then choosing “Select State” from the “Inference” menu.

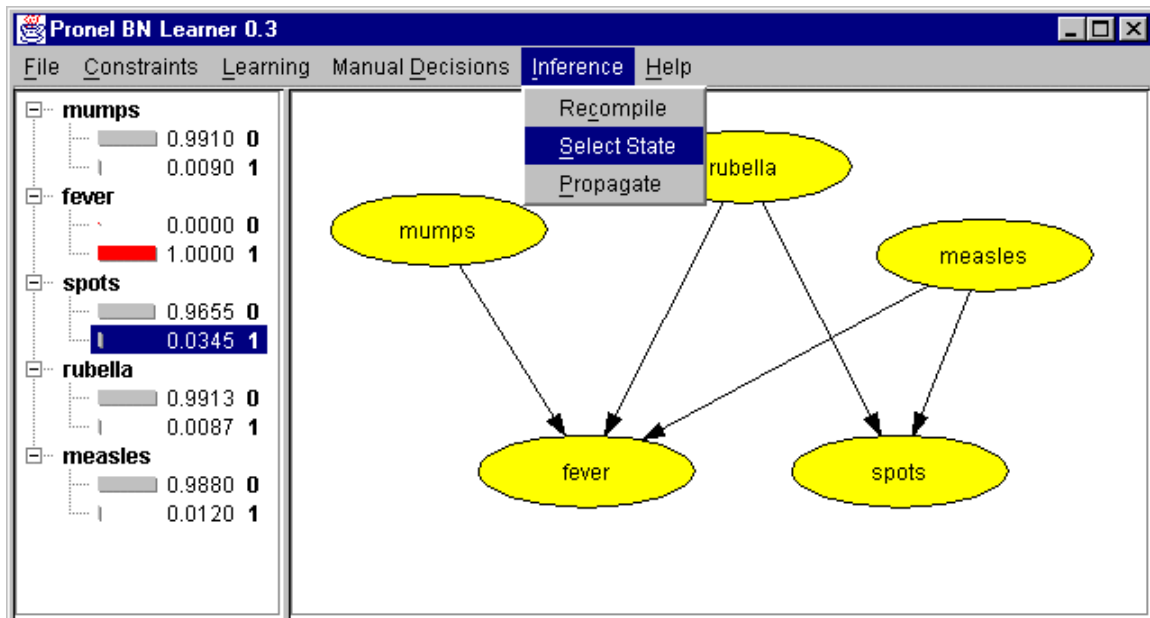


Figure 12: You can select observed states of the nodes by double-clicking them in the node list on the left or by selecting them and then selecting “Select State” from the “Inference” menu.

After evidence has been entered, this should be propagated to update beliefs in all nodes. This is done by selecting “Propagate” from the “Inference” menu. The result of this propagation can be seen in .

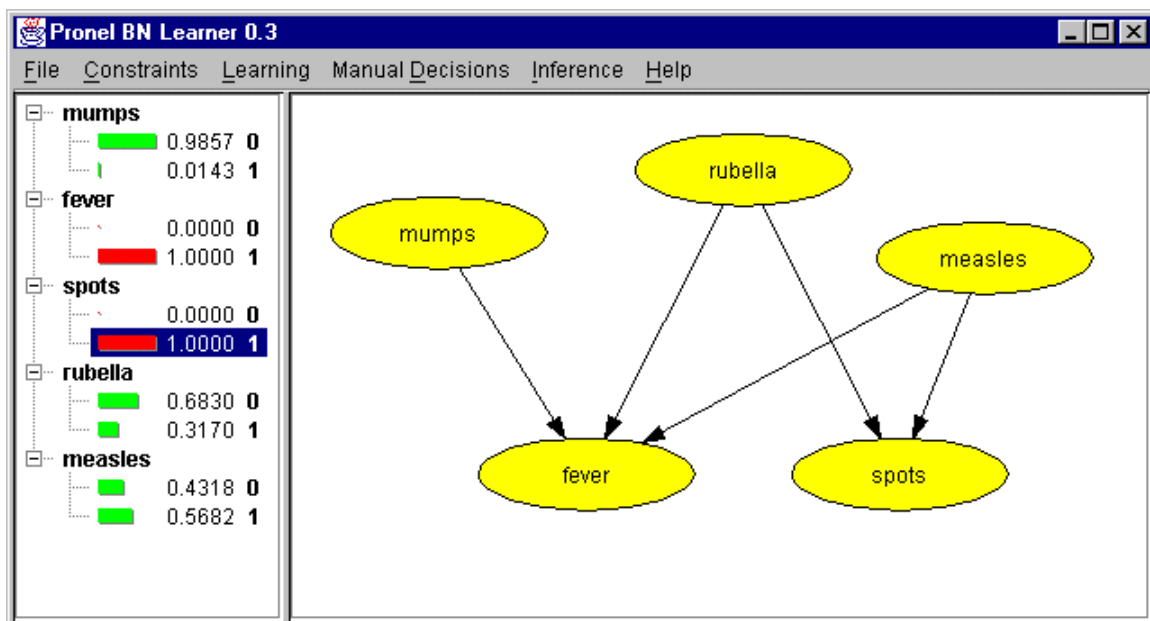


Figure 13: The result of “Propagate” is updated probabilities for all nodes in the node list. When probabilities are up-to-date their belief bars are green (or red if evidence is entered). Otherwise they are gray.