

# Demo Home Work

Peter Lang ([peterlang@mailbox.hu](mailto:peterlang@mailbox.hu))

MY\_NEPTUN

## Marks for the demo:

Comment about the purpose of this task.

Solution of the task.

Comment about how to implement it in the software.

# 1. task: Variable definitions

This part is where the expert-knowledge of the field is needed. It is the first step when we are implementing a model for a specific task, the experts tell us what the variables are and how those depend on each other.

Now, you will be the experts. You have to define the whole model, the dependency graph and all the probabilities. Try to model a real world case.

## Desirable model:

A desirable model consists of **5-8 nodes**, every node should have multiple parents and/or multiple children, so **no nodes with only one parent and one child**, unless it represents an a priori node with no parent. Also having some undirected cycles is highly recommended (note that the model must be a DAG). If possible **don't add more than 3 parents to a node**, try to create a model where the CPTs (Conditional Probability Table) won't contain too many rows (max 8-10) as it will only make your work take more time.

One or two sentences about the model's purpose is needed just to know what it is all about.

My model is about my mother's "apple pie baking" habit. The purpose of the model is to find out the probability that an apple pie is waiting for me at home in various cases.

List the variables in a short and expressive way like this. Keep it in mind that someone's going to read your homework, so try to keep it readable and tidy.

Also, **use the same names in the software** so we can check your answers.

## Variables:

- **Autumn:**
  - **Represents:** the current season is autumn.
  - **Value range:** binary (true, false)
  - **Depends on:** nothing
- **EconomicRecession:**
  - **Represents:** the current economic state is in recession.
  - **Value range:** binary (true, false)
  - **Depends on:** nothing
- **ApplePrice:**
  - **Represents:** the current price of apple.
  - **Value range:** low, normal, high
  - **Depends on:** Autumn, EconomicRecession
- **MomMood:**
  - **Represents:** my Mom's mood.
  - **Value range:** sad, happy
  - **Depends on:** EconomicRecession

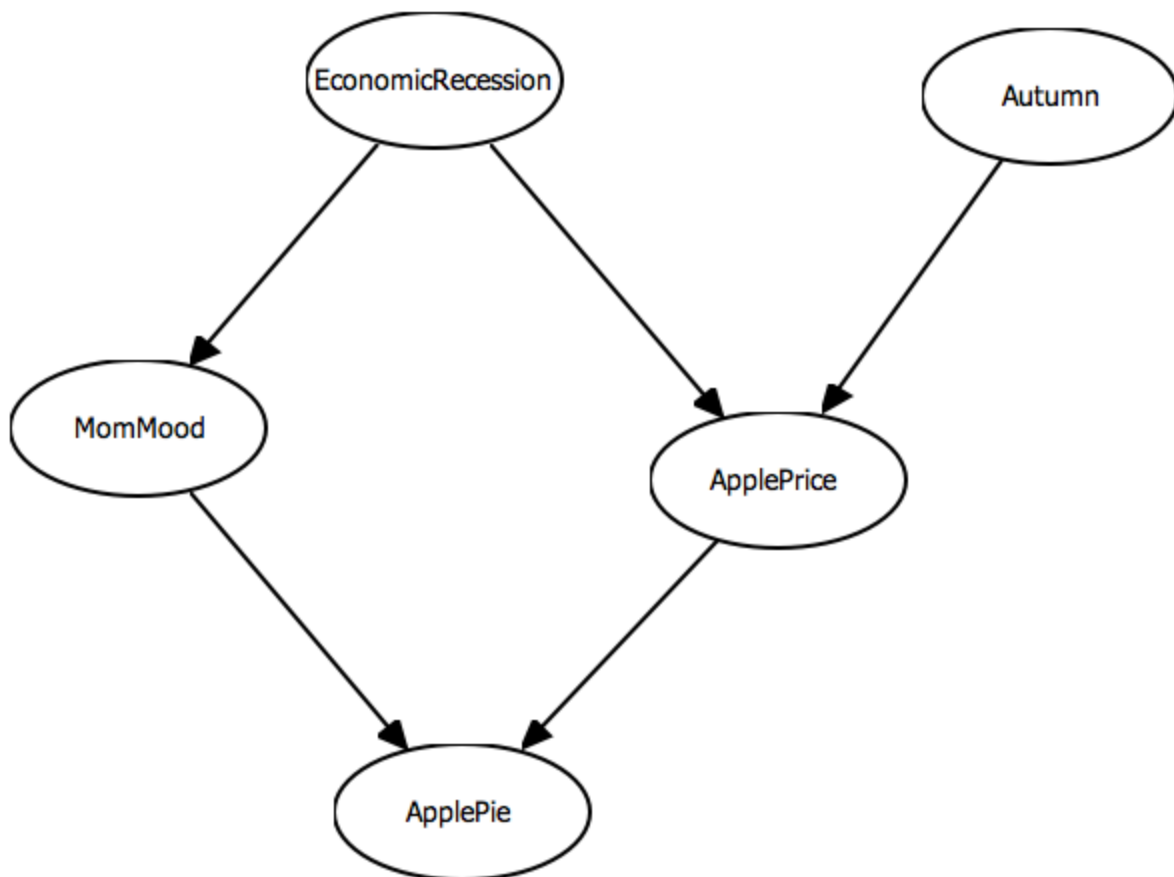
- **ApplePie:**
  - **Represents:** Mom baked apple pie for me.
  - **Value range:** binary (true, false)
  - **Depends on:** ApplePrice, MomMood

You can create new variable groups but those are not necessary in such small models. For every type of **Value range** you have to define new variable types (Right Click -> Variable Types...). The binary (true, false) variable type is provided as default.

## 2. task: Structure of the Bayesian network

This is the part where you create the Dependency Graph for your model. Normally an expert is needed here or given enough samples a structure can be learned too. Independencies can be hard to find out but a causal model is a good estimation.

In the previous task I've already defined my dependencies so here I'm just inserting a picture of the graph.



If your causal model is straightforward then not a single word is needed here. If you are not sure whether your consultant will fully understand it you can write a few sentence about the problematic connections, for example:

My Mom's general mood is affected by the economy. If she's happy, she's more likely to bake apple pie.

During autumn apple is cheaper but it is also affected by the economy.

This is the part where you have to create your variables (Nodes) on the program's canvas and create the dependencies between them (Directional Edge).

The Mac edition can be buggy, I had to switch to Inference Mode and back to Editor Mode in the Operations menu item to make the Palette appear on the right side. You're going to use Chance node with CPT and Connection.

If you click on a node there's a Properties tab below where you can actually name the variable and select the proper variable type.

### 3. task: Quantify the Bayesian network

This is the last part of creating your model. Initial estimation's can be done by experts but usually the probabilities are calculated from samples by statistical methods. Your task here is to make up probabilities for your CPTs (Conditional Probability Table).

It is desired that the probabilities you find out somewhat correlates to a real world example. It is very important NOT TO USE probabilities like 100% or 0% as there won't be any "probability models" anymore. That will reduce it to simple logical model. If you want to make logical connections always smooth the model a bit (i.e. use high/low probabilities like 95% or 5%).

We are curious about the CPTs and their values so **add nicely formatted tables here**. To make yourself clear why you chose these probabilities please make some note of it.

Keep in mind that you have to create a CPT for every variable and for every combination of their parents values. Also, **if any of the rows won't sum up to 1 it's a major problem!** (See: Total Probability Theorem about this)

P(Autumn)	True	False
	0.25	0.75

In Hungary, it's autumn about 25% of the time.

P(EconomicRecession)	True	False
	0.1	0.9

Economy is in a very bad shape nowadays. :(

P(ApplePrice Autumn,EconomicRecession)	Low	Normal	High
Autumn= <b>false</b> , EconomicRecession= <b>false</b>	0.3	0.5	0.2
Autumn= <b>false</b> , EconomicRecession= <b>true</b>	0.15	0.35	0.5
Autumn= <b>true</b> , EconomicRecession= <b>false</b>	0.7	0.2	0.1
Autumn= <b>true</b> , EconomicRecession= <b>true</b>	0.5	0.3	0.2

Apple price is mainly affected by the season.

P(MomMood EconomicRecession)	Sad	Happy
EconomicRecession= <b>false</b>	0.2	0.8
EconomicRecession= <b>true</b>	0.4	0.6

My Mom is generally a happy person, even the economy can hardly make her sad.

P(ApplePie MomMood,ApplePrice)	True	False
MomMood= <b>Sad</b> ,ApplePrice= <b>Low</b>	0.4	0.6
MomMood= <b>Sad</b> ,ApplePrice= <b>Normal</b>	0.2	0.8
MomMood= <b>Sad</b> ,ApplePrice= <b>High</b>	0.05	0.95
MomMood= <b>Happy</b> ,ApplePrice= <b>Low</b>	0.5	0.5
MomMood= <b>Happy</b> ,ApplePrice= <b>Normal</b>	0.25	0.75
MomMood= <b>Happy</b> ,ApplePrice= <b>High</b>	0.1	0.9

Mom really likes to bake apple pie, she does it pretty often.

We are on a low budget so apple price has a greater impact if there's an apple pie home than my Mom's mood.

Generally it's better to do this part in the program first so you can't forget about any of the variable combinations and it also makes sure that the rows sum up to 1. You have to double click on the variables and the CPT's are all there filled with default values. Just adjust these values to reasonable ones.

## Quantify with decision trees (optional)

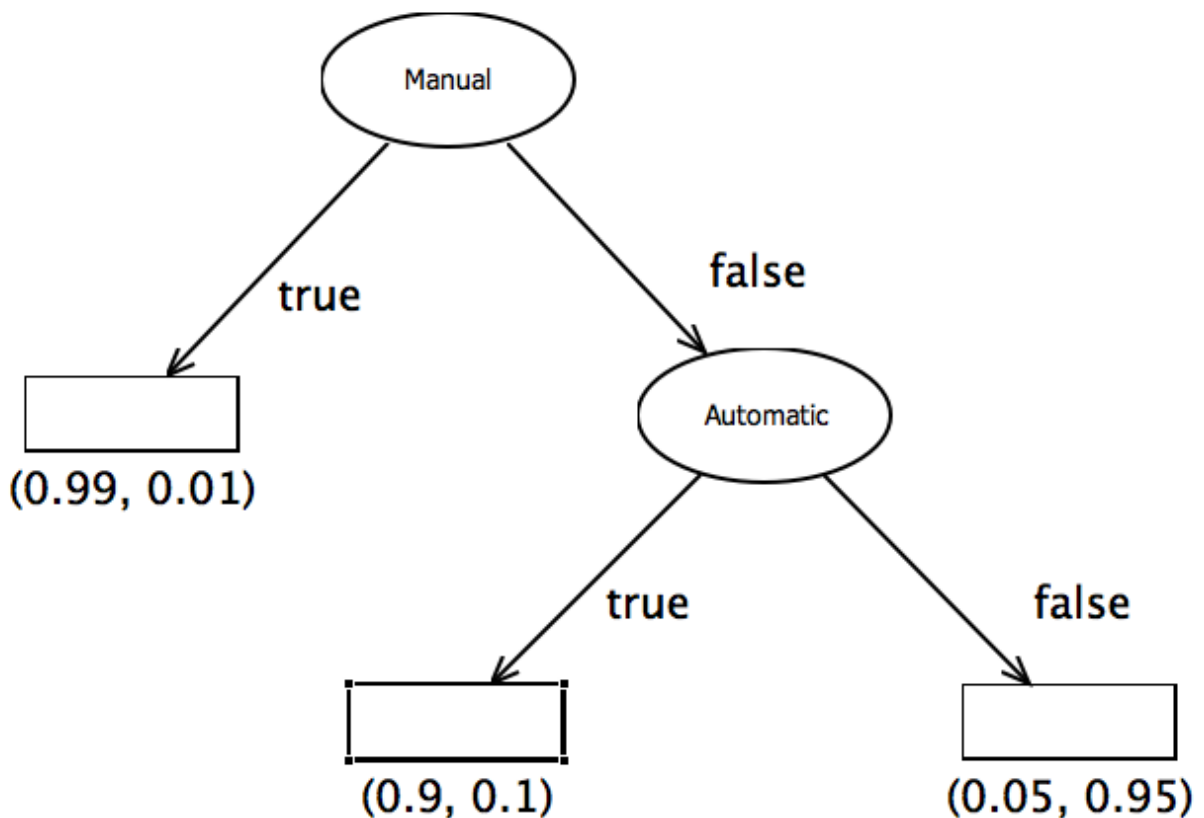
Note that using CPT is not the only option to quantify the network. Another approach is to use **decision trees** which can be used to express contextual (in)dependencies. It is available in the software so feel free to use it. It can be also mixed standard CPTs.

A Bayesian network is designed to express conditional (in)dependencies that is given a set of variables two set can be independent which gives us a way to greatly reduce the size of a model but there are other ways we can further reduce the size.

Contextual independency means that two set of variables are independent given some values of other variables. For example: there is an alarm system where there are automatic sensors and a manual operator. If the manual operator sounds the alarm it overrides the automatic sensors. So let the CPT be:

P(Alarm Manual, Automatic)	True	False
Manual= <b>false</b> , Automatic= <b>false</b>	0.05	0.95
Manual= <b>false</b> , Automatic= <b>true</b>	0.9	0.1
Manual= <b>true</b> , Automatic= <b>false</b>	0.99	0.01
Manual= <b>true</b> , Automatic= <b>true</b>	0.99	0.01

You can see that when the operator sounds the alarm (Manual=true) the Alarm does not depend on the automatic sensors anymore which means the values are the same for every value of Automatic. The following decision tree shows us the reduced representation:



**This is a generic case that independencies only occur given some values.** Given more parental nodes the reduction can be measured as comparing the ordinary probability tables to the conditional Bayesian representation.

This example model can be seen in [decision\\_tree\\_example.xml](#)

To use the Decision Tree representation change the original CPT nodes to **Chance node with decision tree**, which is on the Editor mode's Palette.

Double click on the decision tree node and another tab will pop out with the inner representation of the node and a rectangle which represents a single row of a CPT that is now applied to any parental value combination.

On the right side you can see a Palette with the parent nodes. Drag and drop it on a rectangle to create a split in the decision tree. Click on a decision node and press Delete (or Fn+Backspace on Mac) to delete the node. Thus you can the decision tree.

The leaves represents rows of the original CPT for the given parental value combinations.

#### 4. task: Inference and sensitivity analysis

The previous tasks were about the steps to create a model. Now it's time to use it.

The whole purpose of making networks like this is to make inferences. For example: you work for an insurance company, and you want to know how much the customer should pay for his life insurance that it will be profitable. You've already paid lots of money for a Bayesian network to make such predictions, so you give in as many Evidences as you can. Data from other insurance companies, from doctors, etc. Other variables that you can't fill in will have **a priori** distributions (like the probability for Autumn is 25% if we don't know what day it is). That insurance model preferably will give you a distribution over time so you can make calculations about the customer's life expectancy so the bank can make more profit of him.

First click Operations menu item -> Inference mode. On the right side you can Watch variables, and set Evidences with right click. Below is the Inference Watch where can press the Sample button. Pressing the button you can see the estimation for the probabilities of the watched variables given the evidences.

I made the following inferences:

Note that we need the exact query (inference) you made, so stick to the "scientific" notation. Also **explain your queries** so we can check that you know what you were doing.

What is the probability that there will be an apple pie when nothing else is given?

$$P(\text{ApplePie}=\text{true}) = 0.3$$

In autumn what is the chance that I will have an apple pie?

$$P(\text{ApplePie}=\text{true} \mid \text{Autumn}=\text{true}) = 0.385$$

My mom is sad and there's no apple pie on the table. What is the chance that there's an economic recession?

$$P(\text{EconomicRecession}=\text{true} \mid \text{MomMood}=\text{Sad}, \text{ApplePie}=\text{false}) = 0.195$$

In autumn, my mom is waiting for me with a steaming apple pie. What's the price of apple?

$P(\text{ApplePrice} \mid \text{ApplePie}=\text{true}, \text{Autumn}=\text{true})$	Low	Normal	High
	0.845	0.13	0.025

And what if it's summer?

$P(\text{ApplePrice} \mid \text{ApplePie}=\text{true}, \text{Autumn}=\text{false})$	Low	Normal	High
	0.5	0.425	0.075

The Sensitivity Analysis of  $P(\text{ApplePie}=\text{true} \mid \text{EconomicRecession}, \text{Autumn})$

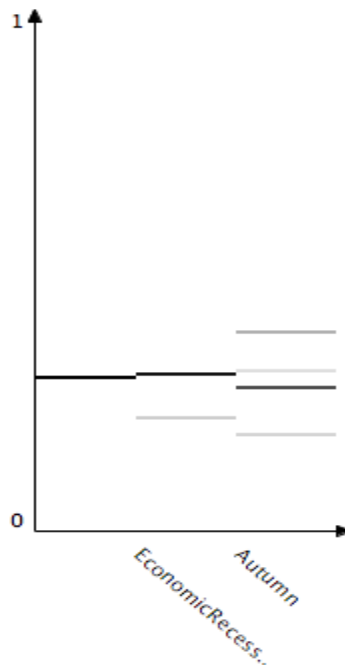


**Sensitivity analysis** is when you are interested in the following question: How much impact can an added Evidence have on the Target's distribution?

Click Operations -> Show Sensitivity Of Inference

Now use the "Inference Specification" panel on the right side! Set the SOI target and conditions. In my example, the SOI target is ApplePie=true and the SOI conditions are EconomicRecession and Autumn.

Click show and a strange diagram appears. The first line represent the target probability when no evidence is provided. The next few lines represents the different probabilities for different values of the first evidence, and so on.



You should also compute the values by normal inference.

P(ApplePrice   EconomicRecession)	True
EconomicRecession= <b>true</b>	0.315
EconomicRecession= <b>false</b>	0.221

Difference is: ~0.094

P(ApplePrice  Autumn)	True
Autumn= <b>true</b>	0.385

Autumn= <b>false</b>	0.273
----------------------	-------

Difference is: ~0.112

As we can see, the value of Autumn has a greater impact on the random variable ApplePie than EconomicRecession.

## 5. task: Compare differences of constructed and learnt models

The purpose of this task is to see how structural and parametric learning works. Our constructed models are usually casual, because that's how human mind works. Learnt models only use the probabilistic independencies that can be found in the sample set so the models we get can be very hard to understand for humans. Anyways, the learnt model is still understandable for humans and can be changed by experts or can be the basis of many considerations of the initial model.

Get back to Editor mode.

Sample generation: Operations -> Sample generation...

You have to give the Sample Size. Trying different sample sizes is the 7. task. The higher the number, the more it will correlate with your initial model. Don't make sample size more than 10000

**Important note: For sample generation's output, use Indices!**

**To map index - value, search for the Variable Types... Here you can see lists of variable types and their lists of values. The index used is the index of the value's row, starting from 0. (e.g.: default type: true - 0, false - 1)**

**Please rename the variables accordingly!**

Learning structure: Operations -> Structure learning...

Prior: CH/BDeu, whichever you want

Max. parent count: if you had N random variables, use N-1 or less

Max. permutation: needed for the optimization algorithm. 50-100 should be enough for small models.

Press start. If it asks whether it is the same model: press cancel, otherwise it will override your current model.

**Save the model and reopen it.** You will see default CPTs in the model.

Learning parameters: Operations -> Learning parameters...

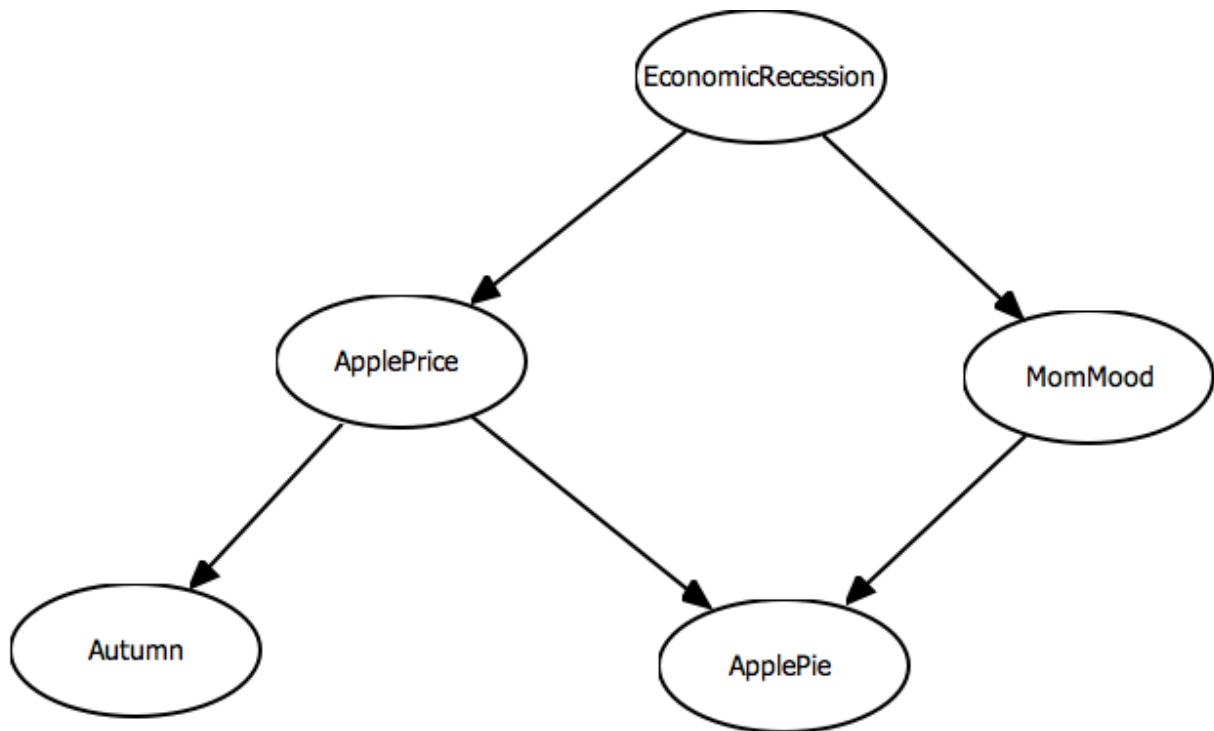
Choose the same Sample Set you used for structure learning and you are done!

Save the model again! Don't forget to attach the learnt models too!

The parameters I used for sample generation and learning:

Sample Size: 1000  
Prior: CH  
Max. parent count: 4  
Max. permutation: 50

The learnt structure:



As we can see, the structure resembles the original except for the Autumn and MomMood random variables. The CPTs are changed accordingly the new model which can be seen in the attached `applepie_learnt.xml`

Proper examination whether the two models are the same is out of this works reach but let us retry some queries:

$P(\text{ApplePie}=\text{true}) = 0.298$  (prev.: 0.3)

$P(\text{ApplePie}=\text{true} \mid \text{Autumn}=\text{true}) = 0.3825$  (prev.: 0.385)

$P(\text{EconomicRecession}=\text{true} \mid \text{MomMood}=\text{Sad}, \text{ApplePie}=\text{false}) = 0.229$  (prev.: 0.195)

You can see that except the third query no real difference can be observed to the original results which indicates the resemblance.

## 6. task: Analyse estimation biases

The following 2 tasks are optional but highly recommended, as the 1-5. tasks means the minimal

requirements.

When an expert is estimating the probability values of a CPT it is observed that the expert is usually **overconfident**. Overconfidence means that the values are “crispy” they tend to be 1.0 and 0.0. **Underconfidence** is the opposite when the values tend to be smooth that is closer to 0.5.

The task here is to analyse the effects of an overconfident and an underconfident model.

First, duplicate your initial model's xml file twice and rename them, so one will be your overconfident and the other will be your underconfident model, use these models from now on. As described you have to choose one or more variables and make some of the values smoother or crispier accordingly.

There are two ways of doing this. You can do it manually, just changing values or there is a dedicated menu item in Operation -> Estimation bias...

If you are using the tool, there are a few options there:

Bias: the measure of the bias (between 0 and 1)

Method: central - underconfidence, extreme - overconfidence

And finally, choose the variables you want to change.

Underconfidence:

I changed the ApplePrice and EconomicRecession variables to make my model underconfident.

P(EconomicRecession)	True	False
	0.3	0.7

P(ApplePrice Autumn,EconomicRecession)	Low	Normal	High
Autumn= <b>false</b> , EconomicRecession= <b>false</b>	0.3	0.45	0.25
Autumn= <b>false</b> , EconomicRecession= <b>true</b>	0.25	0.3	0.45
Autumn= <b>true</b> , EconomicRecession= <b>false</b>	0.6	0.25	0.15
Autumn= <b>true</b> , EconomicRecession= <b>true</b>	0.4	0.35	0.15

Reexecuting my queries, I got the following result:

P(ApplePie=true) = 0.278 (prev.: 0.3)

$P(\text{ApplePie}=\text{true} \mid \text{Autumn}=\text{true}) = 0.338$  (prev.: 0.385)

$P(\text{EconomicRecession}=\text{true} \mid \text{MomMood}=\text{Sad}, \text{ApplePie}=\text{false}) = 0.4755$  (prev.: 0.195)

$P(\text{ApplePrice} \mid \text{ApplePie}=\text{true}, \text{Autumn}=\text{true})$	Low	Normal	High
	0.759 (prev.: 0.845)	0.195 (prev.: 0.13)	0.046 (prev.: 0.025)

$P(\text{ApplePrice} \mid \text{ApplePie}=\text{true}, \text{Autumn}=\text{false})$	Low	Normal	High
	0.524 (prev.: 0.5)	0.373 (prev.: 0.425)	0.103 (prev.: 0.075)

Adjusting the values and rerunning the queries is not enough. We have to draw conclusions from the data that corresponds to our beliefs.

When we had ApplePie as the target for our query the chance is lowered. That is because previously we had a bias on the ApplePrice to be Low and on the EconomicRecession to be false. Both of them used to effect positively the chance of an ApplePie.

However, as we softened the variables ApplePrice and EconomicRecession directly, we can see that queries about them are smoother too.

Repeat the same for Overconfidence, preferable use other variables for it. My demo homework is missing the overconfidence part, that is because the process should be the same.

## 7. task: Effects of model uncertainty and sample size on learning

Here, you should generate multiple Sample Sets from the initial model with different Sample Sizes and different probabilities. Learn this network for a given learning configuration and document the differences.

Two things can be shown here:

- Lowering the sample size means that original model is harder to learn. For this, it is best to repeat the 5. task with a sample size that is 1-2 magnitude less but all the other parameters should be the same. You should see, that there's more difference in this learnt graph and the original.
- A model with smoother probability values is harder to learn. If you have made an underconfident enough model in the 6. task you can use it here. Repeat the 5. task with this model and leave all the parameters as you had in the 5. task. You should see a more different graph than one learnt in the 5. task. If it is the same, try to make the model more underconfident.

### Undersampled Model:

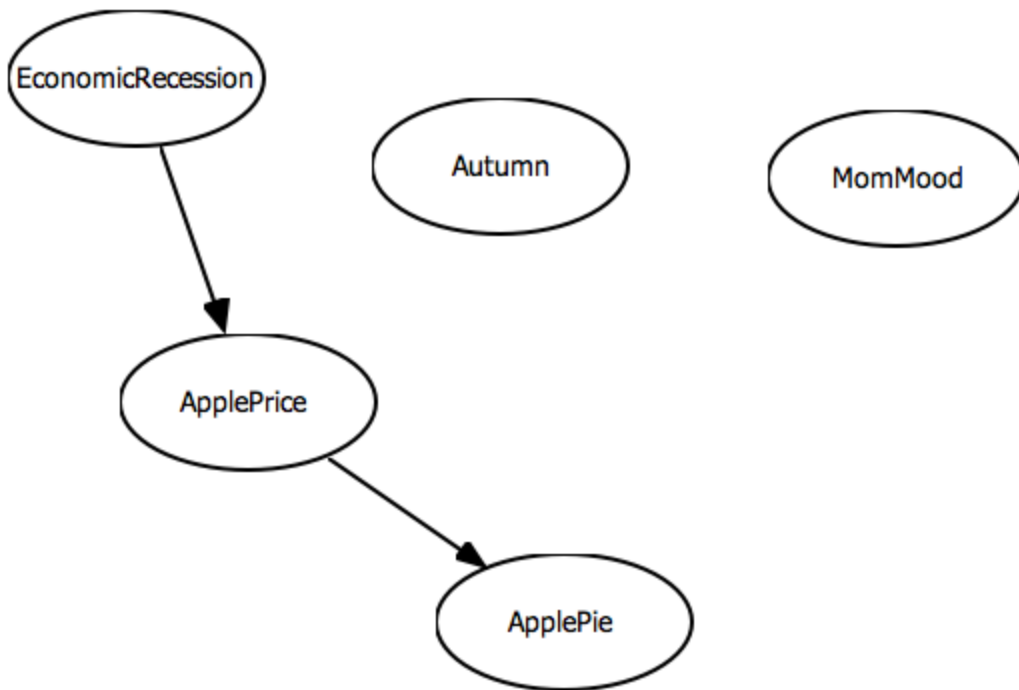
Parameters:

Sample Size: 100

Prior: CH

Max. parent count: 4

Max. permutation: 50



With this many differences, remarks beyond the normal comparison are highly appreciated. Like Autumn and MomMood are totally independent of all others. Note that you should also compare the result of some queries

You can see that now the model is completely different. Some nodes became isolated, like Autumn and MomMood. This means that now there is no dependency between them and any other variables.

As we compare some queries made against this network and the one learnt in the 5. task, we can see how important the role of sample size is.

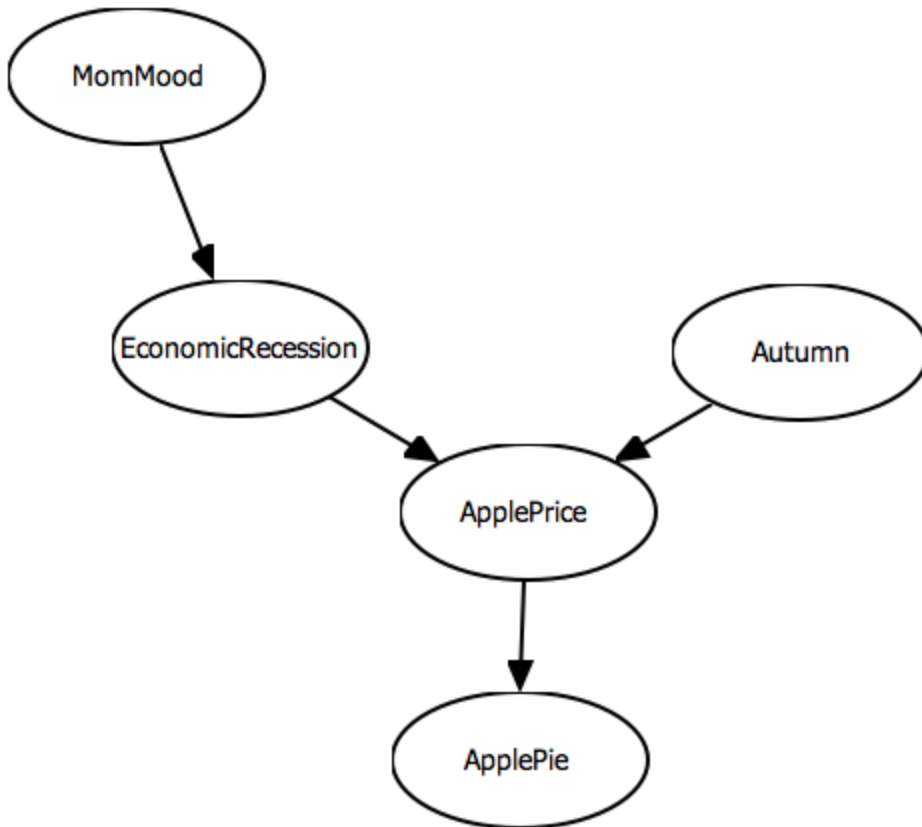
	Original	Learnt (1000 samples)	Learnt (100 samples)
$P(\text{ApplePie}=\text{true})$	0.3	0.298	0.31
$P(\text{ApplePie}=\text{true} \mid \text{Autumn}=\text{true})$	0.385	0.3825	0.31

P(EconomicRecession=true   MomMood=Sad, ApplePie=false)	0.195	0.229	0.106
---	-------	-------	-------

With the data from the 5. task, we can see a tendency, that simple queries without evidences tend to be more punctual than more complex queries with more evidences as the learnt model is becoming less and less accurate.

**Underconfident Model:**

Parameters:  
 Sample Size: 1000  
 Prior: CH  
 Max. parent count: 4  
 Max. permutation: 50



At least there is no isolated nodes, but the structure hardly resembles the original one. However, comparing our queries one can notice that the values are similar:

	Original underconfident	Learnt underconfident
--	-------------------------	-----------------------

$P(\text{ApplePie}=\text{true})$	0.278	0.277
$P(\text{ApplePie}=\text{true} \mid \text{Autumn}=\text{true})$	0.338	0.336
$P(\text{EconomicRecession}=\text{true} \mid \text{MomMood}=\text{Sad}, \text{ApplePie}=\text{false})$	0.475	0.494

Finally, we can conclude that both the model's probability distributions and sample size plays a big role in learning the model properly.