# Artificial Intelligence Laboratory 4: Bayesian Network

*DT8012 (HT16) Halmstad University*
*5th of December 2016*

This lab is designed to introduce you the use of Bayesian Network (BN). By the end of this lab session you will:

- Learn how to estimate BN parameters from data
- Learn how to use BN for reasoning and decision making

For this lab, you will use BN in two different domains:

- **Smart Grids**: using failure history data of an electricity grid to estimate the probability of the duration of a blackout (electricity outage) to be short or long
- **Poker Player**: using historical data of player's betting actions to estimate hand strength and strategy of other agents

**Before you arrive at the lab:**

You should read through the entire document and look up anything you don't quite remember from the lectures.

**After the lab:**

Within two weeks after this lab session you must hand in your **report** and **source code**. Your report should explain your results, and **how** you achieved them. Make sure you include any relevant information to your explanation. Your code, developed in order to solve the exercises proposed here, should be well commented and self-explanatory.

The deadline for your report and code submission is **December 20**. Send your report as a pdf document and your code as one zip file to hassan.nemati@hh.se. The name of your files should include your first and last names, e.g. HassanNematiReport.pdf, HassanNematiCode.zip. Write the name of the lab in the title of your email e.g. Lab 4. Do not forget to write your name and your group mate full name.

# Introduction

**Environment setup**

The only required software tool is Python. You need to install 'libpgm' library for this lab. Libpgm is a tool that makes BNs easy to use. Download it from here and install it. If you are using IDE like PyCharm, it can be easily installed within 'Preference -> Project Interpreter' menu.

For more information about libpgm read: http://pythonhosted.org/libpgm/

- You need to learn how to create network skeleton by defining nodes (vertices), and connections (edges). See http://pythonhosted.org/libpgm/graphskeleton.html
- You need to learn how PGMLearner works. PGMLearner provides tools to generate BNs that are "learned" from a data set. See http://pythonhosted.org/libpgm/pgmlearner.html

# Task 1: Smart Grids

## Introduction

In this task, you will use failure history data of a smart grid as an input to compute probabilities for different outcomes (in this example, to estimate the probability of the duration of a blackout to be short or long).

$$\mathbf{P}(Outage\_duration = long \mid Information)$$

An efficient way to do this is by constructing a BN. A BN is a directed graph in which each node is annotated with quantitative probability information. The nodes are a set of random variables with a probability distribution which are connected to each other by a set of directed links (see Figure 1).

In the **Lab4.zip**, you can find the following resources for **task1**:

- **NetworkLearn.py**: an example code for learning the parameters of a BN from historical data.
- **Network_skeleton.txt**: an example graph skeleton.
- **Data_Sample.pkl**: the failure history data that keep records of specific conditions that a failure has happened. This pickle file stores 2000 observations. Each element is a dictionary contains values for 8 attributes (Figure 2 shows part of this data file). The attributes are: Number_of_Customers, Time, Day, Season, Weather, Demand_Factor, Overload, and Outage_Duration which have values listed by the following:

    - Number_of_Customers: Low, High
    - Time: Morning, Afternoon, Evening, Night
    - Day: Weekdays, Weekend
    - Season: Spring, Summer, Autumn, Winter
    - Weather: Cold, Warm
    - Demand_Factor: Low, Medium, High
    - Overload: Yes, No
    - Outage_Duration: Less_than_1H, More_than_1H

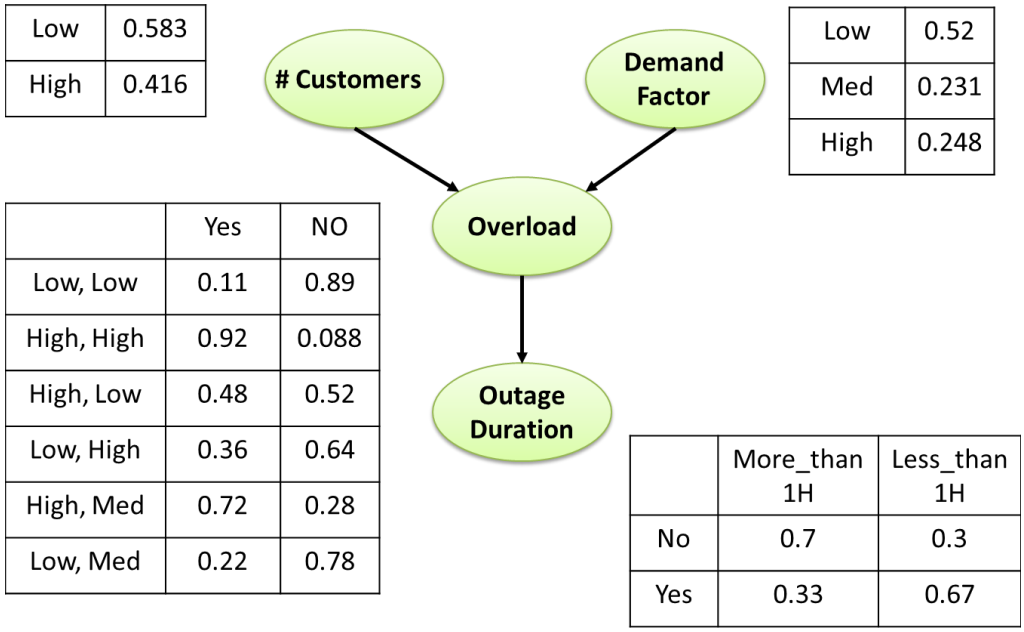


Figure 1- An example of smart grid Bayesian Network

| Season | Outage_Duration | Number_of_Customers | Overload | Weather | Time | Demand_Factor | Day |
|---|---|---|---|---|---|---|---|
| {u'Season': u'Autumn' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'Yes' | u'Weather': u'Cold' | u'Time': u'Morning' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Winter' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Evening' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Spring' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Evening' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Winter' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'No' | u'Weather': u'Warm' | u'Time': u'Morning' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Spring' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Morning' | u'Demand_Factor': u'Low' | u'Day': u'Weekend'} |
| {u'Season': u'Winter' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Morning' | u'Demand_Factor': u'Medium' | u'Day': u'Weekdays'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Warm' | u'Time': u'Evening' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Spring' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Evening' | u'Demand_Factor': u'High' | u'Day': u'Weekend'} |
| {u'Season': u'Summer' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'Yes' | u'Weather': u'Warm' | u'Time': u'Evening' | u'Demand_Factor': u'High' | u'Day': u'Weekend'} |
| {u'Season': u'Winter' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Night' | u'Demand_Factor': u'Low' | u'Day': u'Weekdays'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'Yes' | u'Weather': u'Cold' | u'Time': u'Night' | u'Demand_Factor': u'Low' | u'Day': u'Weekend'} |
| {u'Season': u'Spring' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'Yes' | u'Weather': u'Cold' | u'Time': u'Afternoon' | u'Demand_Factor': u'High' | u'Day': u'Weekend'} |
| {u'Season': u'Summer' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'Yes' | u'Weather': u'Warm' | u'Time': u'Afternoon' | u'Demand_Factor': u'Medium' | u'Day': u'Weekend'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'Yes' | u'Weather': u'Warm' | u'Time': u'Evening' | u'Demand_Factor': u'High' | u'Day': u'Weekend'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Afternoon' | u'Demand_Factor': u'Medium' | u'Day': u'Weekdays'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'More_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Morning' | u'Demand_Factor': u'Low' | u'Day': u'Weekend'} |
| {u'Season': u'Summer' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'Low' | u'Overload': u'No' | u'Weather': u'Cold' | u'Time': u'Morning' | u'Demand_Factor': u'Medium' | u'Day': u'Weekend'} |
| {u'Season': u'Autumn' | u'Outage_Duration': u'Less_than_1H' | u'Number_of_Customers': u'High' | u'Overload': u'Yes' | u'Weather': u'Cold' | u'Time': u'Evening' | u'Demand_Factor': u'High' | u'Day': u'Weekdays'} |

Figure 2- Part of the input data file (Data_Sample.pkl)

The overload failure is considered to have a direct impact on the duration of an electricity outage. Overload is a type of failure caused by an increase in electricity demand in which this increment is not tolerable by the components in the grid.

In Figure 1, we constructed an example of BN model based on some of these factors and calculated the parameters. In this task you need to **find a better BN** to represent the failure history and specify **which factors have influence on outage duration** (hint: the outage duration should be the last node in the BN).

**Task 1:**

a) Try to construct different networks (at least 2 more networks). To do this you need to create a txt file similar to network_skeleton.txt. You need to define all 8 attributes as vertices (V), and then decide the connectivity between vertices as the edges (E). You need to draw the networks with the corresponding parameters in your report.

b) By using the function 'discrete_constraint_estimatestruct' in PGMLearner you can find a model directly from the input data and without specifying the structure of the network. Use this function to find the connectivity between vertices. You need to draw the network with the corresponding parameters in your report. Try to explain the difference between your networks (task 1-a) and this network.

c) (extra credit) Find a way to measure the accuracy of BN and then compare BNs created in task 1-a and task 1-b. Explain the comparison method and show the results in your report.

d) (extra credit) create a network by connecting the nodes which you intuitively know there should not be any connection between them e.g. "the number of customers affects the weather". In your network you need to have at least 3 edges of such example. Then compare the accuracy of your network with the network in task 1-b. Draw the network and explain the results in your report.

# Task 2: Poker Project

## Introduction

You can also apply BN to poker agent project. **Lab4_poker_data.txt** (within **task2** folder) contains 200 poker game samples. Each sample is a part of betting between two agents. Data using following format:

```
#Sample, P1Hand P1HandRank P1Coin, P2Hand P2HandRank P2Coin, P1Action P1BettingAmount P2Action P2BettingAmount, P1Action P1BettingAmount P2Action P2BettingAmount,...
```

Each line starts with sample number. **P1Hand** represents player1's hand category, **P1HandRank** represents the rank of the hand category (e.g if the hand is 'highcard', **P1HandRank** is highest card within; if the hand is 'OnePair', P1HandRank is the rank of the pair), **P1Coin** represents the amount of money that player1 possess. **P2Hand**, **P2HandRank**, **P2Coin** are the same information for p2. This is followed by the action sequence of two agents.

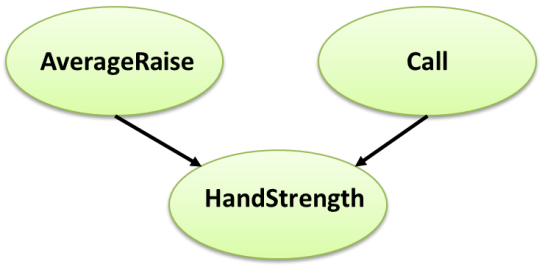


Figure 3- An example of poker game Bayesian Network

**Task 2:**

a) Take a look at the given network skeleton in **Poker_Network.txt** (Figure 3), calculate attributes (e.g. one of the attributes can be average amount of money raise by an agent, **AverageRaise** in given network, the value could be numerical or descriptive e.g. *'high*/*low'*) and estimate BN parameters. You need to write the the corresponding parameters of the network in your report.

b) Come up with at least two more attributes and construct different networks (at least 2 more networks). You need to draw the network with the corresponding parameters in your report. Try to explain the difference between your networks (task 2-a) and this network.

c) (extra credit) Compare the quality of the BN in task 2-a with networks in task 2-b. Explain the comparison method and show the results in your report.

## Grading criteria

- Pass: Complete all the tasks except: 1-c, 1-d, and 2-c.
- Extra credit: Complete the task 1-c, 1-d, and 2-c (within the 2 weeks deadline).
- Submission deadline December 20.