RZN: Assignment 3, Learning with Bayesian Networks

Alperovich Galina

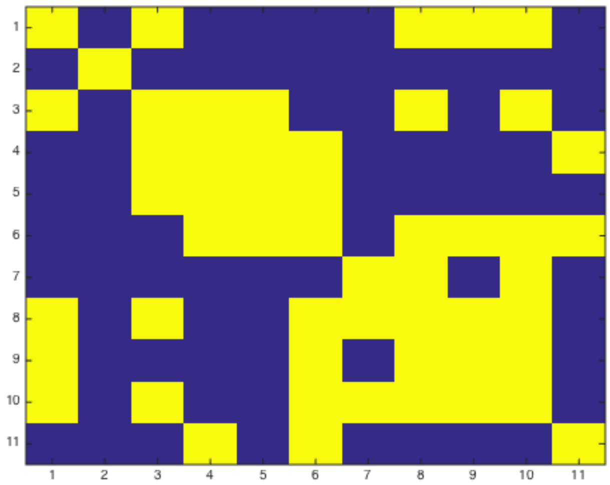
January, 2016

Contents

1 Understanding the data

To understand the data we need to find reliable method which will help us to explore connections between variables. Since all our variables are categorical ones, we can use chi-square test in order to test the hypothesis about independance. We have 11 variables, that means we need to run the test for each pair and see on p-value. If it is small then we can reject the null hypothesis that two variables are independent.

Let's divide our data into two data sets: that one which has no missing data and another one which has ones. Then we can run chi-square test for each pair of variables. Then we can distinguish that variables which are strongly related (not independent). On the picture below you can see matrix with p-values for each pairs (p-values ≤ 0.01 were included, that meant at the 1% significance level, we reject the null hypothesis that variables are independent).



There is a number notation for the columns/rows names:
1. AverageSpeed
2. Country
3. DangerLevel
4. NumberAccidents
5. NumberFatalities
6. NumberJourneys
7. PoliceActivity
8. RoadConditions
9. Season
10. Weather
11. Weekend

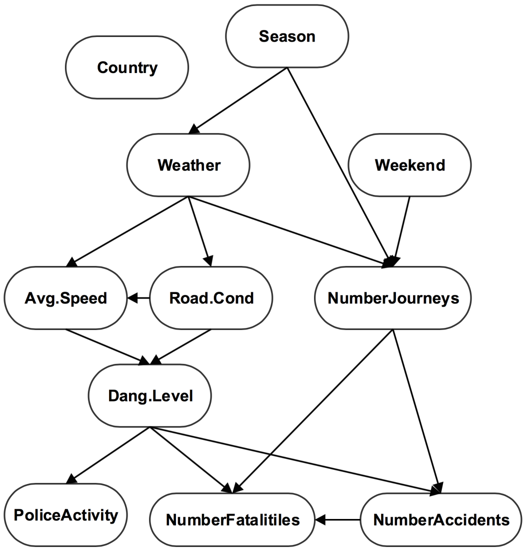
For each row (variable) let's consider all variables with p-value ≤ 0.01:

-AverageSpeed: DangerLevel, RoadConditions, Season, Weather
-DangerLevel: AverageSpeed, NumberAccidents, NumberFatalities, RoadConditions, Weather
-NumberAccidents: DangerLevel, NumberFatalities, NumberJourneys, Weekend
-NumberFatalities: DangerLevel, NumberAccidents, NumberJourneys
-NumberJourneys: NumberAccidents, NumberFatalities, RoadConditions, Season, Weather, Weekend
-PoliceActivity: RoadConditions and Weather.
-RoadConditions: AverageSpeed, DangerLevel, NumberJourneys, PoliceActivity, Season, Weather
-Season: AverageSpeed, NumberJourneys, RoadConditions, Weather
-Weather: AverageSpeed, DangerLevel, NumberJourneys, PoliceActivity, RoadConditions, Season
-Weekend: NumberAccidents, NumberJourneys

Based on chi-square test we can conclude that variable Country is independent with all others in our dataset.

2 Manually constructed BN

Information from previous section look reasonable and corresponds to our own experience. Let's draw the first picture with BN structure.



It is clear that Season affects on Weather, Weather affetcs on Avd.Speed (slower when it is bad weather and vice versa), Road Conditions (wet and dangerous when bad weather) and Number of Journeys (a lot of journeys when good weather and vice versa). Avg.Speed and Road Conditions affect of Dangerous Level (high speed and bad conditions are dangerous), Dangerous Level affects on Police Activity, Number of Fatalities and Number of Accidents. Also there is another variable Weekend which affects on Number of Journeys which affects of course on Number of Fatalities and Accidents (more journeys more accidents).

We will not include Country field to our made BN because it seems like it is independent with other variables.

3 Dealing with the input data

3.1 Train and test sets

We will split our data into two disjoint sets: training set and test set. This splitting procedure will help us to avoid overfitting because the model will be learned with train data and tested using fresh test data which was not used during learning stage.

Training set will be available both without missing values (complete) and original training with missing values. We will use these to training possibilities for building two different BN in order to compare the impact of missing values during the learning stage. Test set will be without missing values to check how the model fits the new data.

3.2 Missing data

There are many observations in our data which have one or several missing values in the fields. The first option to deal with it it is just omit these observations. As we will see below this way is not very good because we lose a lot of information between variables which are in that observations. Also we can insert the mode (most frequent value) instead of missing points. This way is better than previous one but this way does not take into consideration connections between different variables and can bring noise into data. Also we can fill missing point using kNN algorithm, it will help us to approximate these values better because we will consider all variables together connected. Also we can use not complete observations together with complete ones during learning stage of BN, which gives much better results.

4 Learn quantitative parameters

4.1 Complete and incomplete observations

First of all we will learn parameters of the network with our hand made structure (graph above). There are two ways how to do this: using only observations with complete data or using both complete and incomplete observations (we have prepared train sets for both cases).

As we can see on the table below second option gives us better performance (higher Log likelihood), because there is a lot of information in that incomplete observations despite the fact some fields are missing.

	Log likelihood
BN learned with data without missing observations	-9.8073e+03
BN learned with data with missing observations	-9.1100e+03

4.2 Estimated parameters

Let's consider the case when BN is learned from complete data (without missing values).

As an example we will take the node DangerousLevel which takes 2 possible values: low and high. On our graph this node has two parents: AverageSpeed and RoadConditions which also can take two values each. From our learned BN we can get conditional probabilities (our parameters) for all possible combinations of these variables.

RC	AS	P(DL=low)	P(DL=high)
bad	low	0.9753	0.0247
good	low	**0.9901**	**0.0099**
bad	high	**0.7253**	**0.2747**
good	high	0.9439	0.0561

As we can see, this table looks pretty reasonable: least dangerous situation is when the Road Conditions are good and Average Speed is low. This situation correspond to only 0.0099 probability of high Dangerous Level. The most dangerous situation is when Road Conditions are bad and Average Speed is high. When it happens probability of high Dangerous Level is very big - 0.2747. Other two cases show approximately the same result, only high Average speed increases probability of high Dangerous Level more than bad Road Conditions (0.0561 versus 0.0247).

5 Improvement of the structure

While experimenting we can improve our model from different sides. Firstly we can improve graph structure, secondary we can improve data we give as an input.

Our first experiment was the following: we build some structure by luck (actually based on personal experience and statistical test) and give only complete data, without missing values. Second experiment was a little bit improvement of the first: we add also values with missing fields and use EM algorithm to estimate parameters of the network. Third experiment was the following: we improved our graph structure with MCMC algorithms and give the same data, with missing values also. Next we have tried to use K2 algorithm with given topological order.

Each of these improvements gives its result, log likelihood on test data is increasing.

After MCMC new edges were added which were not considered in a previous structure: some of them extend understanding about variable dependence (for example Number of Accidents affects Number of Fatalities), some of them do not correspond with reality (for example Number of Jurneys affects Season). We will fix this contradictions and build new BN later.

	Log likelihood
Original BN (learning without missing data)	-9.8073e+03
Original BN (learning with missing data)	-9.1100e+03
Improved with MCMC (learning with missing)	**-9.0769e+03**
Improved with K2 (learning with missing)	-9.1057e+03

6 Generate a network structure from scratch

Let's build the network from scratch and learn the parameters. We can do it with the following ways: use MCMC algorithm for learning structure and then estimate parameters or we can use EM structural algorithm which will do everything for us (for this algorithm we have used another version of toolbox with this function). We can not use K2 algorithm because it needs topological sorting which we don't have in general case.

Here is results for both approaches. As we can see approach with MCMC is better.

	Log likelihood
EM structure learning (learning with missing data)	-9.3213e+03
MCMC structure learning (learning with missing data)	-9.0802e+03

7 Select optimal network

We have considered several different ways to build our Bayesian network. Let's mention them one more time:

- **BN1**: Structure made by hand, parameter learning from only complete data, without missing data (ML)
- **BN2**: Structure made by hand, parameter learning from both complete and missing data (EM)
- **BN3**: Structure made by hand, improvement with MCMC, parameter learning from both complete and missing data (MCMC + EM)
- **BN4**: Structure made by K2 with topological order from structure by hand, parameter learning from both complete and missing data (K2 + EM)
- **BN5**: Structure made by EM, parameter learning from both complete and missing data (EM)
- **BN6**: Structure made by MCMC, parameter learning from both complete and missing data (MCMC + EM)

On the table below you can see results for Log Likelihood, BIC and Bayesian score.

	Log likelihood	BIC	Bayesian
BN1	-9.8073e+03	-9.3213e+03	-9.4501e+03
BN2	-9.1100e+03	-9.3213e+03	-9.3457e+03
BN3	-9.0769e+03	-9.2158e+03	-9.2362e+03
BN4	-9.1057e+03	-9.2249e+03	-9.2394e+03
BN5	-9.3213e+03	-9.3821e+03	-9.3904e+03
BN6	-9.0802e+03	-9.2054e+03	-9.2204e+03
BN3*	**-9.0773e+03**	**-9.2512e+03**	**-9.2818e+03**

As we can see the best result gives us network BN3 where we build network structure by hand and then improved it with MCMC. This network has some additional edges compared to BN2. Some of them are really improvement and show additional relations between variables, some of them contradicts with reality (Weather affects Season and Number of Journeys affects Season). Since other edges looks very good and reasonable, we will try to change direction exactly for these two edges because there is a dependence between them, only direction is different. This modified BN3 we will call **BN3***. Results for this model you also can see on the table above.

So summarizing all results we can conclude that network **BN3*** shows us the best result. **Let's recall the algorithm of building it:**

1. Chi-squared test (pairwise) helped us to understand strong relations between variables and build first structure.
2. Then we have improved our BN structure with MCMC algorithm.
3. Drawing by hand this new structure we have found two badly directed edges and changed its directions
4. Then we build BN based oh new corrected structure and learned parameters with EM algorithm using both complete and incomplete data (missing).

8 Illustrate the inference

Let's show an example of inference. For example we want to know probability of NumberFatalities in the worst ever situation. Let's construct the evidence in terms of our variables.

1'Season' (**winter** spring summer fall)
2'Weather' (**bad** good)
3'RoadConditions'(**bad** good)
4'AverageSpeed' (low **high**)
5'DangerLevel' (low **high**)
6'PoliceActivity' (regular **increased**)
7'Weekend' (working **weekend** holiday)
8'NumberJourneys' (low **high**)
9'NumberAccidents' (low medium **high**)
10'NumberFatalities' (?)
11'Country' (**US** UK Europe)

If we calculate probability of NumberFatalities with this evidence we will get as follows:

P(NumberFatalities = "low" |evidence) = 0.1523
P(NumberFatalities = "medium" |evidence) = 0.3192
P(NumberFatalities = "high" |evidence) = 0.5285

This probabilities looks nice. Probability of NumberFatalities being high is very high because all fields increase it.

Let's consider another example. What if we want to know probability of DangerLevel in the best ever situation?

1'Season' (winter **spring** summer fall)
2'Weather' (bad **good**)
3'RoadConditions'(bad **good**)
4'AverageSpeed' (**low** high)
5'DangerLevel' (?)
6'PoliceActivity' (**regular** increased)
7'Weekend' (**working** weekend holiday)
8'NumberJourneys' (**low** high)
9'NumberAccidents' (**low** medium high)
10'NumberFatalities' (**low** medium high)
11'Country' (US UK **Europe**)

If we calculate probability of DangerLevel with this evidence we will get as follows:

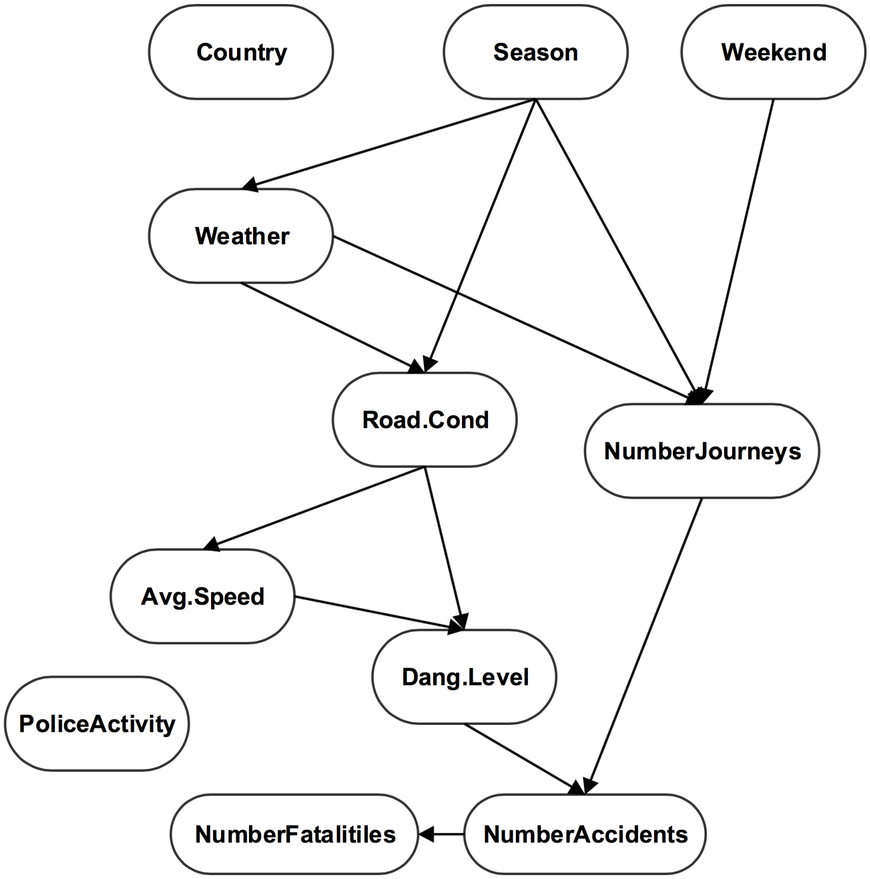
P(DangerLevel = "low" |evidence) = 0.9946
P(DangerLevel = "high" |evidence) = 0.0054

This looks also reasonable, probability of DangerLevel being low is extremely high 0.9946, because the situation is good by all fields.

With this built Bayesian Network we can do very cool things. We can use it for prediction of unknown fields knowing other fields, also we can fill missing values of original data set and learn BN again and again. Unfortunately it is easier said than done because all computations are very heavy and takes a lot of computer time.

9 Summary

In this report we have built and learned different Bayesian networks for car crash dataset. In general we have constructed 7 different networks with different modifications and improvements. Model with the best result are constructed using MCMC optimization of hand made network structure (also after MCMC we have edited graph a little bit), exact algorithm of building is explained in section 7.
Final network structure can be seen on the picture below.


Country
Season
Weekend
Weather
Road.Cond
NumberJourneys
Avg.Speed
Dang.Level
PoliceActivity
NumberFatalitiles
NumberAccidents