TDT4171 Assignment 3

Fredrik Veidahl Aagaard

March 2019

1 Introduction of the model

In this assignment I have chosen to model a decision support system for deciding which city to visit on an interrail trip. The scenario is that we are a group of friends on an interrail trip. We are currently in Prague, and we have booked a hotel in Zagreb in 6 days. We don't want to stay in Prague for 5 more days, and we don't want to go to Zagreb early, so we have decided to visit one of the three following cities: *Vienna, Bratislava, Budapest*.

We must then decide which city we want to visit, when to travel, and how to get there. We can choose from the cities above, and we can choose to travel on one of the next 5 days. We must also decide between our means of travel as we can take the standard train, which is stops more, but is more affordable, or we can take an express train going directly to the city of our choise.

To get a proper model, such that we get a good final decision, there are multiple factors that come into play. How many passengers are there on the train? How much does it cost? How comfortable is the train? In the following section we'll explore some of these questions further and make some assumptions about the scenario.

2 Assumptions

It is important to make some assumptions about the scenario such that it doesn't become overly complex, but still manages to function as a decision supprt system. We'll assume:

- The tickets for both trains are never sold out
- We must buy these tickets at full (or discounted price) as they don't come with the interrail pass, and that we will only buy 1 ticket each.
- We don't care about the cost from that city to our final destination *Zagreb*.
- To simplify we have made some rough increments on some of the variables. E.g. we only consider to travel during either the night or day.
- Discounts are binary, in that we either get a discount of 20%, or we don't.
- Delay and distance doesn't depend on technical errors, like the train not working, or a traincrash.
- The maximum delay is 60 minutes, and it is modeled in 20 minute increments. This is because delay is very hard to model.

- The two types of trains *standard* and *express* has the same speed, they only stop a different number of times.

- We pick the day we are travelling on Sunday (March 10th, 2019) before the first possible day to travel (Monday).

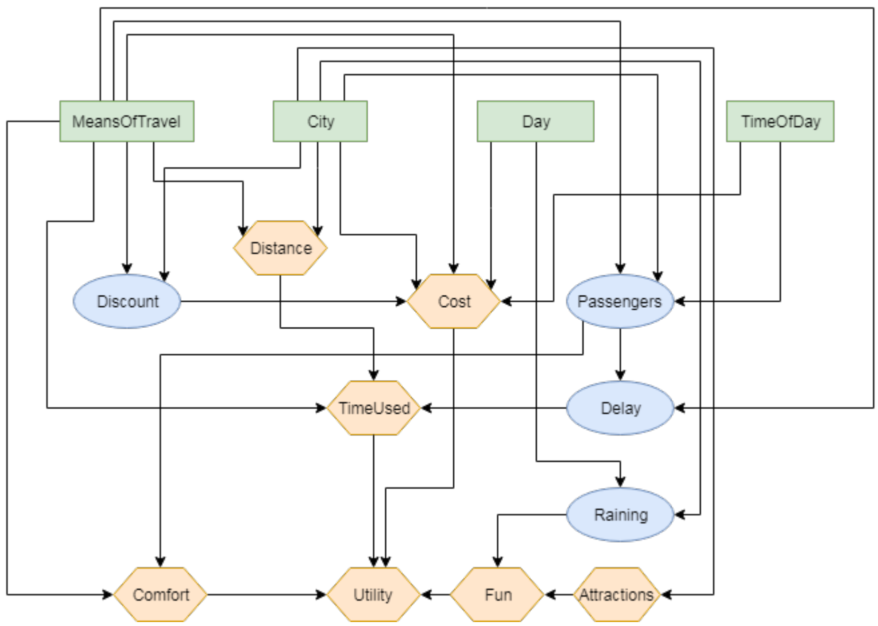

Figure 1: Graphical structure showing the interconnection between the decision nodes(green), chance nodes(blue), and utility nodes(yellow).

3 Variables

In this section we'll try to discuss and quantify the variables shown in Figure 1. Table 1 shows an overview of all the variables, their names, types, values, and wether the variable is observed/known *right before* our decisions are made.

We will in the following subsections take a closer look, and make further assumptions, of the internal variables, that is all nodes in Figure 1 except for the decision nodes. Allthough all variables will be covered, some will be covered in more or less detail as I see fit. This is mainly so that this paper doesn't become too long.

Name	Type	Values	Observed
City	Decision	{Vienna, Bratislava, Budapest}	No
MeansOfTravel	Decision	{Standard train, Express train}	No
Day	Decision	{Monday,Tuesday,Wednesday,Thursday,Friday}	No
TimeOfDay	Decision	{Day, Night}	No
Discount	Chance	{Yes, No}	No
Passengers	Chance	{Few, Medium, Many}	No
Delay	Chance	{0, 20, 40, 60} (in minutes)	No
Raining	Chance	{Yes, No}	No
Attractions	Utility	{Few, Some, Many}	Yes
Comfort	Utility	{Little, Medium, High}	No
Distance	Utility	Distance (in kilometers)	Yes
TimeUsed	Utility	Time (in minutes)	No
Cost	Utility	Money (in NOK)	No
Fun	Utility	{Little, Some, Much}	No

Table 1: Variables and decisions used in the DSS.

3.1 Discount

We assume that the probability of getting a discounted ticket doesn't change day to day, but rather on a weekly basis. And that we can also quantify this probability in the given season we are in. In a more realistic scenario this probability does indeed change with time. Table 2 shows some probabilities that I see fitting. These are very low as train tickets tend to be of a set price, but there may be discounts due to overbooking, underbooking etc.

	Vienna	Bratislava	Budapest
Standard train	0.08	0.03	0.03
Express train	0.01	0.008	0.008

Table 2: Probability of getting a discounted ticket.
P(Discount | MeansOfTravel, City)

3.2 Passengers

This is a variable governing how many passengers there are on the train. This ranges from an empty train, *Few*, to a tightly packed train where every seat is taken and the corridors are filled with standing people, *many*. I have here chosen to not to take the travel day into account, as this doesn't seem to make that much of a difference in the summer. To find the probabilites here I have only anecdotal evidence from my own interrail trip. Table 4 reflects my experience with the differen type of trains and at what time they go.

		Vienna	Bratislava	Budapest
Standard train	Day	(0.1, 0.4, 0.5)	(0.2, 0.4, 0.4)	(0.1, 0.2, 0.7)
	Night	(0.8, 0.15, 0.05)	(0.85, 0.13, 0.02)	(0.7, 0.15, 0.15)
Express train	Day	(0.4, 0.4, 0.2)	(0.4, 0.4, 0.2)	(0.3, 0.4, 0.3)
	Night	(0.9, 0.09, 0.01)	(0.85, 0.14, 0.01)	(0.78, 0.12, 0.02)

Table 3: Probabilities for how many passengers are on the train.
P(Passengers | MeansOfTravel, City, TimeOfDay)

3.3 Delay

The delay, as stated in Section 2, is hard to model, and I have chosen here to model it with increments of 20 minutes, up to an hour. Table 4 tries to show a somewhat realistic scenario based on how crowded it is (people getting on and off the train at different stops) and which type of train is taken.

	Passengers		
	Few	Medium	Many
Standard train	(0.4, 0.3, 0.2, 0.1)	(0.2, 0.6, 0.1, 0.1)	(0.2, 0.2, 0.3, 0.3)
Express train	(0.7, 0.1, 0.1, 0.1)	(0.6, 0.1, 0.2, 0.1)	(0.2, 0.2, 0.4, 0.2)

Table 4: Probabilities for delay on the trip.
P(Delay | MeansOfTravel, Passengers)

3.4 Raining

This variable is governing wether it is raining or not given a city and day. This is a binary variable and only depends on the two conditionally independent input variables `Day` and `City`. I have chosen to use data from *The Weather Channel* (1) for the days March 11 to March 15. This data is shown in Table 5. This weather forecast is of course modelled using a complex model, which in turn has its own Bayesian Network. Although it is a complex model that is used I assume that it is highly simplified when displaying the probabilities to the users (e.g. rounding off to nearest 10%).

This variable can be observed, but we have assumed that we make a decision *before* any of the coming days. Thus the probabilities changing with each passing day will not matter in this case.

	Vienna	Bratislava	Budapest
Monday	0.2	0.2	0.4
Tuesday	0.0	0.0	0.0
Wednesday	0.2	0.3	0.2
Thursday	0.1	0.1	0.1
Friday	0.1	0.5	0.5

Table 5: Probabilities for it raining. P(Raining | City, Day)

3.5 Attractions

This variable is governing the number of attractions in a given city. This can be hard to model as we must first define what an attraction is. In this model I define an attraction as a place to visit according to Ranker's top 100 places to visit in Europe (2). Table 6 summarizes how many of these attractions are in each city.

	Vienna	Bratislava	Budapest
Attractions	5	0	2

Table 6: Number of attractions in each city

3.6 Comfort

This variable is governing the level of comfort on the train given the type of train and how many passengers are on the train.

3.7 Distance

This variable is governing the distance we have to travel by train given a city and type of train. The distances are modelled differently depending on type of train based on anecdotal evidence from my own interrail trip where a train to Budapest went through Vienna, where a more direct route existed.

3.8 TimeUsed

This variable is governing the time spent travelling by train between Prague and a given city. This is a function of `MeansOfTravel`, `Distance`, and `Delay`. To get an estimate of `TimeUsed` we would add the expected time from `MeansOfTravel`, which is only concerned about how often the train is stopping, `Distance` which is only concerned about the distances (note we assume the speed of the train is the same for both trains), and `Delay`, which is only concerned about how long the train is delayed.

3.9 Cost

This variable is governing the cost of just the train trip. This is the most connected node in our network besides `City`, which makes it hard to model, and we would get a huge table since this is a function of `MeansOfTravel`, `City`, `Day`, `TimeOfDay`, and `Discount`.

3.10 Fun

This variable is governing the amount of fun we will have given the number of attractions and wether it rains or not.

3.11 Utility

When we want to get the final utility of the network we have to consider the other utility nodes that are connected to it, that being `Cost`, `Comfort`, `TimeUsed`, and `Fun`, and weigh these against each other. To do this we would have to scale the previous variables against each other to normalize them and get them on the same scale. We can then define our utility function as some combination of these and compare them to a *Golden Standard*.

4 Evaluating the Model

After we have defined our utility function as said above we can set each decision node such that it behaves as a chance node which have been observed, and calculate the resulting utility of the network. We can then compare that to a *Golden Standard*. This standard is chosen by me, and is given by Table 7, and shows what action is expected that the utility function returns in some given cases.

MeansOfTravel	Day	TimeOfDay	Expected Action
Standard Train	Wednesday	Day	Vienna
Standard Train	Friday	Day	Bratislava
Express Train	Monday	Night	Vienna
Express Train	Tuesday	Night	Vienna
Standard Train	Thursday	Day	Budapest

Table 7: Golden Standard when choosing city to travel to

5 Further Work

This model is far from perfect and complete. To get a better performing model, there would need to be more empirical evidence, as opposed to my anecdotal evidence. A good utiliy function would be requiered, and a well defined standardization of `Cost`, `Comfort`, `TimeUsed`, and `Fun` used to compute the utility would also have to be defined.

References

[1] The Weather Channel, "Weather forecast," 2019. [Online]. Available: https://weather.com/

[2] Island Hopper Dan, "The top must-see attractions in europe," 2019. [Online]. Available: https://www.ranker.com/list/best-places-to-visit-in-europe/island-hopper-dan