Sample Book

Author

April 26, 2022

Abstract

Here I write the abstract

Contents

Chapter 1

Introduction

Reasoning with evidence in law is cool!

The main research question for this Master's Thesis is: can we automatically create good Bayesian Networks that reflect the ground truth of simulations? If so, can we use this simulation + Bayesian Network setup to investigate BN idioms and methods for law more generally, to see how well the probabilistic approach holds up?

Chapter 2

State of the Art

2.1 The problem with evidence

When we find evidence for a hypothesis that we have held in the back of our minds, we then find the hypothesis more likely. This is the basic idea behind reasoning with evidence. In a constellation of hypotheses and pieces of evidence, we want to construct a network that will lead us to believe as many true hypotheses as possible, given the evidence that we have.

However, evidence itself is elusive, and it's connection to hypotheses is as well - how can we be sure that some evidence supports some hypothesis, and even if we know that it does, how can we express how strong the piece of evidence is? Some evidence is very weak, and only after a tedious process of ruling out other factors and careful investigation and collection of other pieces of evidence, we can come to a conclusion about a hypothesis. On the other hand, some evidence is so strong that it leaves no room for doubt.

We all have intuitions about evidence strength - but can we make these intuitions precise? Additionally, can we manage the complex realities of weak evidence for many different hypothesis? These are questions that have guided this state of the art section.

In this section I will briefly discuss methods for reasoning with evidence, then transition to probabilistic reasoning with evidence, Bayesian Networks. I will discuss some problems with Bayesian Networks.

2.2 Reasoning with evidence

Argumentation graphs and their semantics. Scenario theory. Bayes Law. Hybrid Approach.

2.3 Bayes!

Probabilistic reasoning (Dahlman, simple). Bayesian Networks Fenton and Vlek - combining scenarios, idioms in Bayesian Networks.

2.4 Problems with the Bayesian Approach

There are many problems with the Bayesian Approach.

The most obvious is the problem of the numbers: where do we get them? We can get some of them from statistics, but we need to many numbers that we have to make some numbers up. This is not necessarily a bad thing - we put our (betting) money where our mouth is and assign numerical precise probabilities to situations that we preciously only had vague intuitions for. However, this brings about the veil of objectivity. By giving a probability to your intuitions, you have made your intuitions more precise and you can now reason with them, update on evidence using Bayes Law, and everything's great. In some domains, this is obviously okay. If you want to bet cents on world events and walk that fine line between calibration and discrimination for fun and profit, that's no problem. After all, there are incentives to abstain from the unclear, the stuff that might not have a specified answer, the vague. But when we are talking about using evidence in law, we are talking about exactly that domain - we're not making predictions about the price of oil in 6 months, or the outcome of the French election, with clear outcomes, clear procedures for measurement. Instead, we're trying to make predictions about crimes and crime scenarios, which are a lot vaguer, and strangely unobservable at times - things like motives, or behaviours that happen under specific circumstances, interlocking stuff with complex dependencies. We all have intuitions about evidence strengths in vague situations, but they are more difficult to make precise than the traditional 'forecasting' events. So trying to assign probabilities without a clear method of calibration, makes that they will be imprecise.

Level of granularity.

Independence relations. Selecting events.

Technical problems with Bayesian Networks. Precision. Evaluation. Sensitivity Analysis.

1. The problem of the reference class, which is also a problem within a 'clear' domain - but this is a fundamental philosophical problem that I will not discuss further.
2. We do not know the probabilities (frequency) of our variables in the first place.
3. We do not know how robust the network is to imprecise or wrong frequencies.
4. We do not know if the assumption of independence between any two variables holds. The assumption of independence is necessary in Bayesian Networks, otherwise the

complexity of the network becomes unmanageable.

Chapter 3

Creating Simulations to Evaluate Bayesian Networks.

3.1 Introduction

In this part, I explain the general method for creating simulations with automatic Bayesian Networks, as well as how to evaluate these Bayesian Networks. The general process of creating simulations to evaluate Bayesian Networks is illustrated in Figure 3.1. We start by defining a simulation with agents, and then have reporters report on that simulation in the 'Experiment' stage. Relevant events in simulations are collected by the reporters in each run. The collection of runs is used by an automatic Bayesian Network constructor algorithm to construct a Bayesian Network automatically in the 'Building BN' stage. Finally, the constructed Bayesian Network is evaluated with respect to the criteria in the 'Evaluate BN' stage. These three stages are explained in this section. Specific instances of simulations and networks I created with this process are the subject of the next chapters.

3.2 Setting-up an Experiment

An experiment consists of a simulation and reporters. Reporters are defined separately from the simulation because they are not inherent to it - they are defined with respect to what we want to know about the simulation.

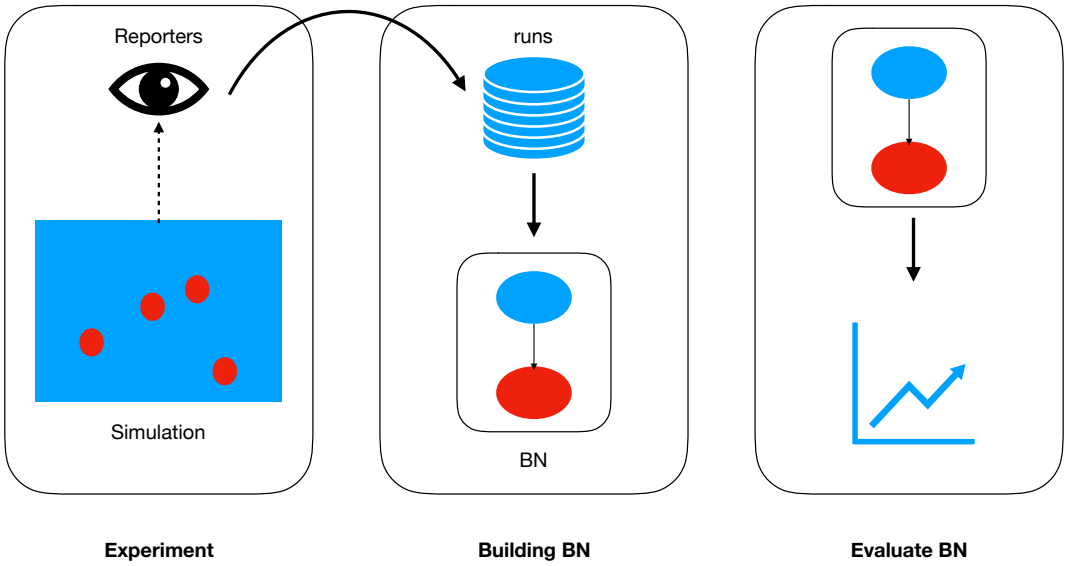

Figure 3.1: Method for evaluating automatically generated Bayesian Networks from simulations.

3.2.1 Simulation

Structure of a simulation

In the simulation, we are simulating some sort of criminal scenario - a theft, usually. Or we are simulating purely for the theoretical things. The simulation can be as precise as necessary, but there are certain things that need to be present: we need to have states that happen, there needs to be evidence for those states as well. The granularity of the simulation and its complexity depend on the modeller and her requirements.

Spatial and Non-spatial simulations.

I'm discussing two types of simulations: spatial simulations and non-spatial simulations. In a spatial simulation, an agent's behaviour is mediated with respect to their environment - eg, agents cannot pass through buildings, or cannot see agents that are far away, or can only steal from another agent when that agent is nearby. In a non-spatial simulations, agents can behave and interact with each other, but this is happening without any environment, hence we are simulating an abstraction. In concrete terms, you can think about a non-spatial

simulation as a communication game.

For this project, that means that spatial simulations are more complex and interesting than non-spatial simulations, as there are more possibilities for variety.

The simulations were programmed in MESA, a python package that is made for the creation of Multi-agent systems simulations. We define an environment that the agents can interact with, as well as agents that perform some behaviours. Specifics of simulations and agent behaviour are described in the following chapters.

Predictability and randomness

Where does the interest of the simulation come from? In one part, agents sometimes do things because they are commanded to do so by the computer [1]. This means that, at the start, a random number generator might 'decide' that some agent has a motive, since the random number generator generated a 1 instead of a 0. On the other hand, some randomness arises from interactions between agents, or between agents and their environments. This is where spatial simulations bring additional value compared to non-spatial simulations.

In non-spatial simulations, all agent states are essentially brought about by a combination of randomly generated numbers, and reasoning rules. For example, if an agent has a tendency to lie (randomly generated), and it has the opportunity for lying (brought about due to the current non-spatial simulation), then it will lie. Hence a combination of behavioural rules and randomly generated numbers results in a state of 'agent lies' of 1.

However, in spatial simulations, interactions and behavioural rules and randomly generated numbers are all brought together: if an agent is near another agent (chance interaction), and it has a tendency to steal (randomly generated), then it will attempt (but might not succeed) in stealing. Here the behavioural rule might lead to a more interesting/complex/complicated outcome than in the non-spatial simulation.

3.2.2 Reporters

In the simulation, certain states can be brought about. For example, an agent can succeed in stealing an object, or in lying, or in having a motivation (or in not having those things). We need a way to observe these states: this is where reporters come in. A reporter reports the outcome of a relevant event or state in the simulation, and is embedded in the code. If an event happens (or does not happen), the reporter reports that the event is true (or false). In essence, the reporter (R) is a random variable (RV) that maps an event (e) to a truth value:

[1] rephrase, this is always true lol.

$$R : e \rightarrow \{0, 1\}$$

Not all the states in the simulation have a reporter associated with it - otherwise I could build infinitely many reporters. I could have reporters for names, for $agent\_Q\_at\_x\_1\_y\_200$. Hence, I only created reporters for states I deemed relevant for the scenario that I am investigating. Here is a subjectivity gap. I can imagine that in my simulation of the Grote Markt (see later chapter), there is some part of the simulation that by chance geometry, lends itself to an easier job for the thief than another part of the simulation. If the thief and victim spawn near this point, then the probability of the thief succeeding will be higher. Increasing the granularity of the reporters might help us determine if there is a spot like this. However, I did not implement this level of granularity in the simulation (yet), because that is a local and specific part, and does not fit into the more global scenario description (the scenario of theft is spatially-free).

The global state of one run of the simulation, is the combination of all reporters.

$$G = (e_0 \rightarrow \{0, 1\} \times e_1 \rightarrow \{0, 1\} \times ... \times e_n \rightarrow \{0, 1\})$$

or,

$$G = R_1 \times R_2 \times ... \times R_n$$

for n reporters.

Then, we collect these global states over the number of runs that we do for each experiment, which results finally that the output O of this stage of the method, is a series of global states, one for each run:

$$O = (G_0, G_1, ...G_{runs})$$

3.3 Creating a Bayesian Network from a Simulation Automatically

The output of an experiment is the collection of runs O, where each run is the global state G of the simulation, as measured by the random variables R. Semantically, it fits that reporters are random variables, as the reporters become the nodes in the Bayesian Network.

Once we have a collection of states and runs, we can give this to an automated Bayesian Network learner, such as those implemented in pyAgrum. These learners can interpolate

a Bayesian Network using algorithms, such as Greedy Hillclimbing and K2. This results in automatic generation of the Bayesian Network which is solely based on the data that is collected in O.

For more detail on these algorithms watch this space.

3.4 Evaluating the Bayesian Network

We want to evaluate different aspects of the Bayesian Network: structural criteria, performance criteria and human criteria.

3.4.1 Structural Criteria

1. (temp) Events are ordered temporally - scenario-like.
2. (con) Evidence connects to hypotheses.
3. (exc) All events that are irrelevant are not included in the scenario BN.
4. (exh) All events that are relevant are included in the scenario BN.
5. (ind) Independent events are not connected to each other.

3.4.2 Performance Criteria

1. The accuracy of the network is not lower than the inherent randomness of the simulation.
2. The strong view: probabilities in the network correspond exactly to probabilities in the simulation.
3. The weak view: Updates on evidence in the network go in the same direction as updates on evidence in the simulation.
4. Sensitivity analysis shows that no simulation-irrelevant event influences some output [2]

3.4.3 Human Criteria

1. Do we think that a human can find these numbers?
   (a) How robust is the network against disturbances around the mean (problem with precision)?
   (b) How robust is the network around rounding to arbitrary intervals?

[2] rephrase

Now that we've established how the automatically generated Bayesian Networks are going to be judged, we can show cautiously in the next chapter how well they are holding up!

Chapter 4

A Simple Stabbing - Non-Spatial Simulations and their Bayesian Networks

4.1 Introduction

Here I talk about the credibility game and Charlotte Vlek's Stabbing Bayesian Network example.

In these non-spatial simulations, there are agents, and they interact with each other, but there is no environment for them to interact in. So the simulation is a pure combination of the probabilities assigned to each state transition (or something along those lines). In a sense, the probability for an agent to 'stab with knife' purely depends on the probability of 'motive' and 'opportunity'. Compared to a spatial simulation, where 'opportunity' is more complex, as it involves proximity to victim which arises from the simulation and not from some random number generator.

These non-spatial simulations are boring, but they are necessary first steps: after all, if we cannot make BNs out of these predetermined (the probabilities in each run are not predetermined, but the distributions that they are drawn from, and their thresholds, are), then the rest of our endeavours will be fruitless. On the other hand, if the process for creating and evaluating these simple simulations work well, then we can proceed to modelling more complex, spatial simulations.

In this part, I have two experiments. One simple experiment meant to be a replication of Charlotte Vlek's Bayesian Network in xxx [1], and another simple experiment mainly

[1] rephrase

meant to test the evaluation criteria for the Bayesian Networks, as outlined in the previous chapter.

4.2 Method

4.2.1 Vlek Networks

Take the Vlek networks from Vlek Jurix 2015.

The general story is: Jane and Mark had a fight, but Jane had a knife. Mark died.

Then, there are two specific scenarios that can explain why Mark died. In scenario one, Jane stabbed Mark, and then he died. In scenario 2, Jane threatened Mark with the knife, Mark hit Jane, Jane dropped the knife, Mark fell on the knife, and Mark died by accident. There are two separate networks for these scenarios.

I'm going to create two separate networks, and then also see if I can merge them, by creating a Jane-and-Mark-knife simulation, assigning some random probabilities that correspond to the story, and see what the K2 algorithm makes of it. Then I will also merge the two networks to see if the K2 can deal with mutually exclusive nodes (eg: 'Mark died by accident' should rule out 'Mark died by stabbing').

So, I created some logical rules, either atoms or rules, and we can process in the simulation these using standard forward chaining inference. Every atom has a prior probability, and every conclusion of a rule has a probability given the F/T state of the premises. At every step, the simulation checks which new sentences are true, applies a rule with a given probability, and counts the outcome.

This does mean that rules at the end are not triggered as often as rules in the beginning, even if they have the same trigger probability (they are on other sides of the chain, more needs to be have happened to conclude that Mark died).

4.2.2 Behavioural rules for the simulation

Global

Jane and Mark had a fight (80, 20)

Jane had a knife (30, 70)

Scenario 1

Jane had a knife and **Jane and Mark had a fight** → **Jane stabbed Mark with a knife** (99, 1) (1, 0)

Jane stabbed Mark with a knife → **Mark died** (20, 80)

Scenario 2

Jane had a knife and **Jane and Mark had a fight** → **Jane threatened Mark with a knife** (97, 3) (1, 0)

Jane threatened Mark with a knife → **Mark hit Jane** (5, 95)

Jane dropped the knife → **Mark fell on the knife** (99, 1)

Mark fell on the knife → **Mark died by accident** (40, 60)

Mark died by accident → **Mark died** (0, 100)

This was done using forward chaining, with a time index, which means that a rule could only be triggered at one time (otherwise the probabilities get messed up). So we have a forward chaining rule, which means that if the premises of some rule are true, there are no excluding facts true, the timestep is correct, and the random number generator generated a number that is lower than the probability threshold, we find that the conclusion is true, and add it to our found facts. We keep doing this.

Since Vlek does not give probabilities in her paper, I'm just making some up. The logical sentences are reporters in the simulation, and then I let the simulation run for many times (10,000 times). Then there came out three different Bayesian Networks, one for scenario 1, one for scenario 2, and one for the combined network (Figure 4.1, Figure 4.2, Figure 4.3).

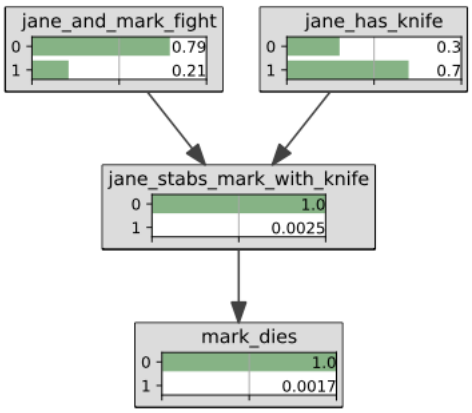

Figure 4.1: Automatically generated BN with K2 and the above forward chaining rules, scenario 1.

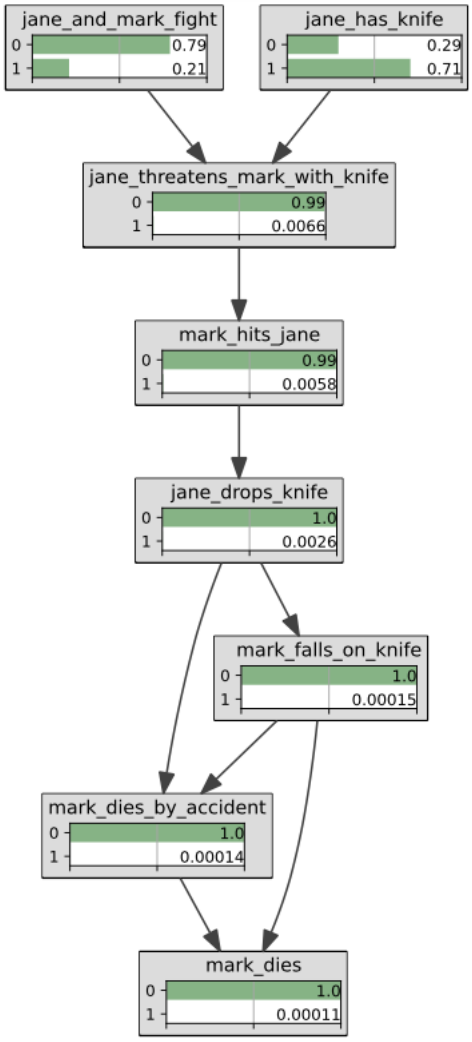

Figure 4.2: Automatic generated BN with K2 and above forward chaining rules, scenario 2

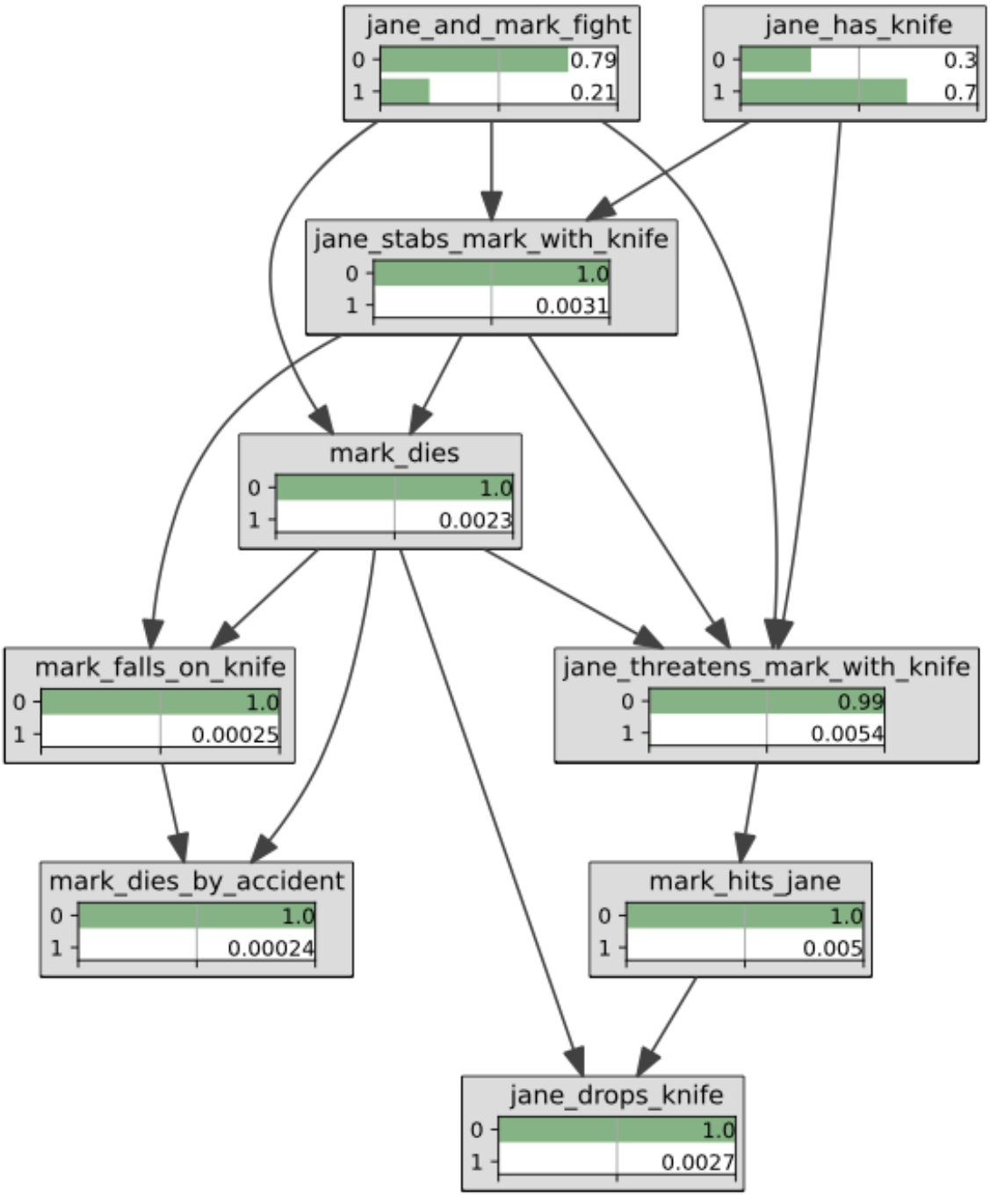

Figure 4.3: Automatically generated BN with rules from both scenarios included.

4.3 Results - Figure comparison to Vlek 2015

If we only look at the ordering of the Bayesian Network, we can see that for Figure 4.1, the sub-scenario structure is the same as in Vlek 2015 Figure 3. There's no scenario node constraining the network, all the information is contained in the network, no scenario node needed (todo: make the nodes the exact same names).

However, there are differences between Vlek 2015 Figure 4, and the automatically generated BN here. Figure 4 in Vlek 2015 is replicated below in Figure 4.4 (todo: ask permission?? or just remove). The automatically generated BN is very linear and can be interpreted in a purely temporal way: mark hits jane, jane drops the knife, and due to jane dropping the knife, and mark falling on it, mark dies by accident - and if mark dies by accident, mark dies. The probabilities for all these events (from jane dropping the knife on), are ridiculously low, and don't really make sense (should interpret the small probabilities as e, and not as actual numbers I guess, due to underflow?).

In Vlek's paper, we see a subscenario: we have the subscenario "Mark died by accident", which contains events such like mark hit jane, which leads to jane dropping the knife, and mark falling on the knife, and then dying. The coparents of Mark died are the same in this network as in the automatically generated BN, however, we once-again miss the scenario-like construction where mark hitting jane, jane dropping the knife, and mark falling on the knife are connected as part of a subscenario, rather than their "own" nodes.

Why to choose for subscenarios when it is not required? Tomorrow I will look up why the scenario construction was used and I will see if it does something that I miss? Coherence? But that also travels up the chain? But something with d-separation probably. I'll read the Vlek 2015 again i guess.

And then I will also reflect on the evaluation section, and also show a plot of the changing prior, and maybe do the accuracy etc.

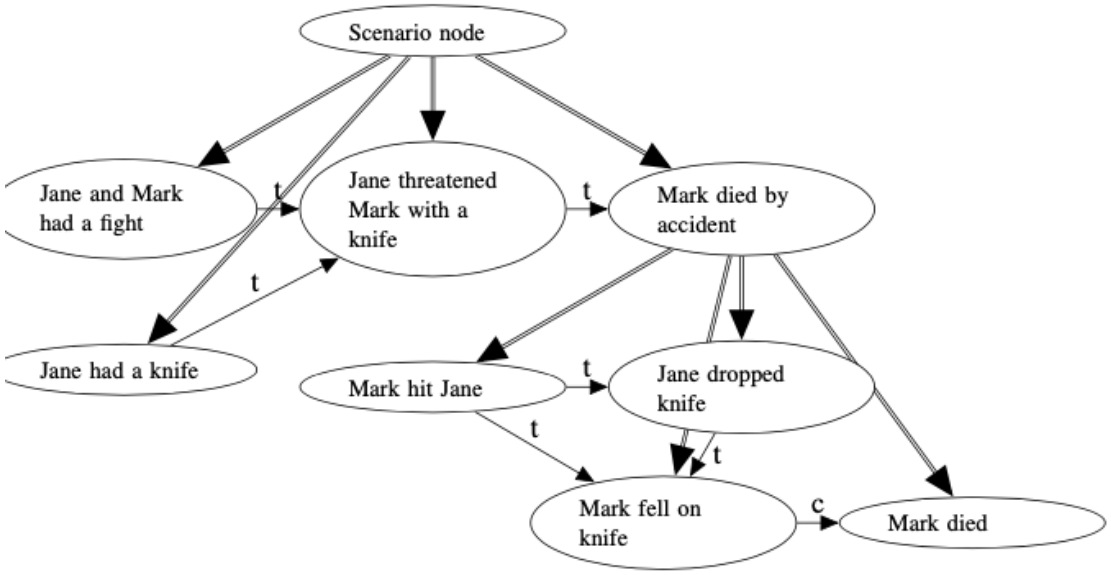

Figure 4. A Bayesian network structure for the second scenario

Figure 4.4: Vlek BN.

Chapter 5

A Simple Robbery - Spatial Simulations and their Bayesian Networks

5.1 Introduction

Here I talk about the robbery example with the evidence. Also model reference class explicitly just for fun here.

Chapter 6

The Grote Markt - investigating idioms with the Island Prior

6.1 Introduction

Here I talk about the Grote Markt and the more complex simulations that I created for it, including the island prior.

Chapter 7

Conclusion

7.1 Conclusion

Here I draw some conclusions.