Knowledge Representation and Reasoning
BKI312 (2014-2015)
Assignments (Series 2)

M. van der Heijden and J. Kwisthout

December 2015

1 Introduction to the assignments

Welcome to the second series of assignments on knowledge representation and reasoning. In this series you will model a mission to Mars, and you will use probabilistic explanations for visual information.

Marks and Time This assignment is marked out of a total of 100 percent, and it contributes a total of 20 percent towards your overall grade for the course knowledge representation and reasoning. The distribution over the sub-assignments is as follows:

Assignment 2-1: 50 percent
Assignment 2-2: 50 percent

The estimated amount of time for the average student can be up to roughly four full days of work per student. This varies since most students work in teams of two. The four days are effort asked for, but in the case much more is needed to complete only the required parts, contact the teachers.

Submission See the individual descriptions of the assignments for detailed descriptions of what to hand in. We expect all written answers in the form of a report, including formalizations, explanations, pictures, and possibly small code fragments to illustrate your answers. Each assignment needs to be covered in a separate section (or chapter) of your report. Your final score will depend on the quality of both the code and the report.

In addition: for each of the two assignments, answer the following questions: i) how much time did it take you to finish it? ii) if you would have to change aspects of the assignment: what would they be and why? iii) what do you think of ProbLog usability, are there any debug tools, e.g. more proof explanation tools, that you think would be useful? Summarised answers to (iii) may be shared (anonymously) with the ProbLog mailing-list, please disregard this question if you don't want this.

Code files are to be submitted in a zip-file, properly named. All submissions should be done through Blackboard.

Good Luck!

The deadline for submission is

2nd of February 2016

2 Introduction to ProbLog

ProbLog is similar to Prolog, but in recent implementations is no longer directly based on a Prolog interpreter. As you have seen in the lectures, ProbLog allows you to add probabilistic information to statements in logic. ProbLog supports a large subset of Prolog syntax, sufficient for most purposes, but not including the more obscure Prolog features. Keep this is mind when writing ProbLog programs. On the probabilistic reasoning front, ProbLog offers a wide range of possibilities, documented in a series of tutorials on the ProbLog website: `https://dtai.cs.kuleuven.be/problog/tutorial.html`

2.1 Using ProbLog

One of the options for using ProbLog is the web-based editor on the website of the University of Leuven: `https://dtai.cs.kuleuven.be/problog/editor.html` This will run your code on their servers (not ideal for larger projects).

Otherwise you can do a local install on your own machine, for which you need Python. See `https://dtai.cs.kuleuven.be/problog/` for installation instructions. Although it runs on Windows, Linux is easier and gives you the most optimised inference engine. ProbLog can be used by executing one of the following commands from the ProbLog folder:

- `problog-gui.cmd`
- `problog-cli.py [explain] <model>`
  (or `problog` when installed through `pip`)

The first command opens the ProbLog editor in a browser window, which allows interactive editing and evaluation of your model. The second takes a ProbLog model (composed in your favourite editor) as argument and returns the answers to all the queries in the model. Additionally, it can provide explanations of models without evidence. By adding the `explain` option, proofs of the queries with the appropriate probabilities will be given. Be aware that for complicated programs that use the advanced features of ProbLog these explanations may not be as helpful as they are for straightforward ProbLog.

3 Assignment 2-1: Probabilistic Mars missions

In this first assignment, you will do some probabilistic knowledge engineering in ProbLog and compare the representation with a Bayesian network. First read the following story:

> There have been several missions to Mars to establish, among others, whether it would be possible for microbial life to develop on Mars. Specifically, three major conditions need to be fulfilled in order for life to be possible: favourable atmospheric conditions, the presence of liquid water, and particular chemical nutrients required for microbial life. The probability of life developing if all conditions are met is estimated to be 0.01. A priori probabilities of these conditions being met are 0.1, 0.05 and 0.05 respectively, according to some expert.
>
> The presence of chemical nutrients can be tested by performing spectrography tests, for example on soil samples taken by the Mars-rover Curiosity. These measurements are performed with a spectrometer which has a sensitivity and specificity of 90%. If liquid water has existed on Mars this would lead to particular observable geologic features with 60% probability. However, these could also have been caused independently by some other process, known to occur on Mars. This process is estimated to cause the geological features with a probability of 20%. If microbial life were present there would be a 90% chance that this would produce elevated levels of methane, which would otherwise only have 3% chance to be observed. Methane can also be measured using a spectrometer, this one with a sensitivity of 97% and specificity of 92%.

The tasks below ask you to analyse and model the story, the results of which should be reflected in your report (coherently, not as an enumeration of answers).

- Draw a Bayesian network (BN) that captures the (in)dependencies in the story.
- Write down the corresponding conditional probability tables (CPTs) and fill them out using the information in the story (discuss your design decisions).
- Write a ProbLog program which defines the probabilistic knowledge associated with the network.
- Use ProbLog queries to investigate the behaviour of the model in various circumstances (i.e. observed evidence). How do observations influence the probability of microbial life on Mars?
- Query ProbLog for the probability of the geological features (assuming no evidence). Now compute the value by hand and explain how ProbLog arrives at the answer. Besides the solution given by the *explain* capability of ProbLog, do you see a different way to compute it? (this may depend on how you implemented it, if this question makes no sense for your implementation, describe in detail how the correct answer can be computed in your implementation).
- Compare the BN and ProbLog representations, consider advantages and disadvantages of using either.
- Can you extend your representation to use the first-order capabilities of ProbLog to model multiple spectrography measurements? And to allow multiple experts, that each have different opinions on what the prior probabilities for the major conditions should be? How do these changes affect the probability of finding microbial life on Mars?
- How do these first-order extensions relate to your representation as a BN?

Modelling dynamics The model we have now is rather static, as it does not take time into account. In the following tasks you will make a dynamic model of whether life will survive on Mars. Since this is a rather complicated matter, we will look at a simplified model taking only into account how many of the required conditions are satisfied at any given time point. Assume that life is present in the initial situation and that all three conditions mentioned earlier are met. The probability of survival at time t given the previous situation with a number n of satisfied conditions, is given by the following formula:

$$P(\text{life}(t) \mid \text{life}(t-1), \text{conditions}(t-1) = n) = \begin{cases} 1 - \frac{1}{n+1} & n > 0 \\ 0 & \text{otherwise} \end{cases}$$

Since we are looking at processes on planetary time scales, each time point can be considered to be a million years or so, but this is inconsequential for our model. The final piece of information you need is the dynamics of the conditions. The probability that the number of satisfied conditions remains the same from t to $t+1$ is 0.8. Furthermore, the probability of an increase or decrease by one in the number of satisfied conditions is 0.1 for both.

Now consider these tasks:

- Construct a dynamic model in ProbLog from the information in the paragraph above.
- Given that $t = 0$ is the initial situation, query the probability of life at $t = \{1, 2, 5\}$.
- What do these probabilities imply for the survival of life on Mars in the long run?

As before, the results should be included (and discussed) in your report.

Optional Clearly both the static part and the dynamic part can be made a lot more realistic by including additional information and modelling interactions in more detail. Other options for extensions would be relating multiple measurements to movement of the Curiosity rover, or any other cool thing that you can come up with that makes sense in the context. A bonus of at most 10% on your final grade can be obtained (grades higher than 100% are not possible).

4 Assignment 2-2: Visual representations and reasoning

In this final task, you are going to experiment with logical knowledge representation of visual information and the accompanying reasoning styles to find out what is actually depicted. You have lots of freedom to choose your own domain, i.e. you own images.

In the lecture we have seen a simple example of aerial sketch map recognition. The aim of this assignment is to model (probabilistic) visual features, logical axioms about the visual domain, and to reason about specific pictures. Especially the rules defining how image objects come about from scene objects is important.

Picking a domain There are lots of different domains to choose from. In the lecture we have seen examples of houses (based on windows and doors), kitchen design, general architectural design, aerial sketch maps, shapes (e.g. a puppet) made out of basic shapes (such as squares and triangles) and so on. But, one can also think of cartoon-like pictures, maps, line drawings and so on. Note that – because you are using the power of logic – you can make models a bit more interesting with more complex definitions. For example, a stick figure consists of a `torso`, two `leg` objects on each side, and two `arm` objects on each side. Defining *spatial* predicates such as `leftof(X,Y)` and `above(X,Y)` may be useful for some domains.

You can choose any domain (but choose a fun or nice one, of course). The main **requirement** is that it should not be much simpler than the example used in the slides (the aerial sketch maps by Poole at the end of the vision-lecture), in terms of number of rules and components. You should be able to argue that your application is roughly (at least) as complex as that one. Another **requirement** is that there is considerable uncertainty in the rules.

Previous years we had a similar assignment. Some examples are:

- **Nijntje**, in which various items of the famous Nijntje-drawings were used to derive whether Nijntje is actually on the picture. For example (in AIlog)

```
ailog: observe shirt.
Answer: P(shirt|Obs)=0.8.
  [ok,more,explanations,worlds,help]: ok.
ailog: observe ~juwel.
Answer: P(~juwel|Obs)=0.34.
  [ok,more,explanations,worlds,help]: ok.
ailog: observe ~ears.
Answer: P(~ears|Obs)=0.6.
  [ok,more,explanations,worlds,help]: ok.
```

- **Flags**: if you see some white and some red, which country could it be?
- **Pizzas**: if you observe some onion, and it seems likely that there's some cheese there too, would it be possible that this is just a Pizza Margherita, or could it be a Hawaian as well? And which is more likely?
- **Vehicles**: if you observe one wheel, and another one maybe, and you seem to observe a frame, could it be a bicycle, or is it a motorcycle? Or did you miss other features and is it still likely that it could be a car?
- **Electronic devices**: if you see some buttons, a screen and a particular configuration (relational, spatial) of them, can you find out what you are looking at?

Other examples include the game *League of Legends*, sandwiches based on observed ingredients, building floor plans, flying objects which could be either plane or bird, and many more.

An additional interesting example is a bachelor thesis which was written by Sil van de Leemput in 2013 on the use of a probabilistic logical system to interpret Duplo buildings from 3D camera input. The BNAIC-paper can be found at `http://bnaic2013.tudelft.nl/proceedings/papers/paper_61.pdf`. The system uses a *deductive* approach to assemble pieces of coherent material into blocks, and blocks into buildings. The uncertainty comes from the camera input.

Modelling the domain in ProbLog Your domain (as well as the pictures-represented-as-sets-observations) needs to be modelled in ProbLog. An easy way to have multiple pictures in the database is to annotate picture observations with the picture number; for example `line(l1,picture1)`. That way, one can query about specific pictures by querying all observations for a specific picture. Another way could be to use `evidence` in ProbLog.

It is advised to invest a lot of thinking time in how to create the model. In the general abductive case, you should create a causal theory that couples *observations* and *causes*. It is very tempting to write something like this:

```
duck :- quacks, flies, beak, wings, ...
```

But now let us imagine one sets *prior probabilities* on the items like *flies* and *wings*. That entails that the more evidence we have for a situation, the less likely it becomes to be a duck! It also creates problems with the firing of the rule. In an abductive setting, the proper way to represent such things (deterministically and without variables here only) is something like this:

```
quacks :- duck.
flies :- duck.
beak :- duck.
wings :- duck.
...
```

In this way, things become more likely when there is more evidence. So... be careful to think about which questions you are going to ask to the system, and how they should be answered. If you choose for a more *deductive* solution, then you might need to use the first style of rules. The constraint-based approach (filling in the values of a fixed set of variables) discussed in the lecture may require yet another type of representation when implemented in ProbLog. Furthermore, not all combinations with negative conditions will be allowed; many probabilistic logical systems have difficulties with negation, so sometimes it helps to model `not(A==B)` as `notEqual(A,B)`, for example. For specific domains you have much freedom on how to choose to define your predicates.

In general, think of *how* the system will reason: you will give it a couple of *observations* and then it needs to employ the causal theory (i.e. your ProbLog rule base) to come up with possible explanations in terms of *assumables*. If I *observe* two lines crossing I might infer a possible *explanation* that it is a cross. The lines are observed, but the crossing is inferred and the cross itself only assumed...

Inference on specific, hand-chosen instances Inference on the pictures in your domain consists of supplying the observations belonging to a specific picture, and computing explanations for them. A **requirement** is that your input pictures contain enough ambiguities to ensure that *multiple* explanations exist for some instances (or parts of instances). Thus, in the report we want to see examples where we see the observations accompanied by what the algorithm computes as possible explanations (with probability) of what we actually *see* on the picture.

What to do So, summarizing, you need to choose a domain and create or gather some images. Then, you need to model your domain of pictures and implement it in ProbLog. Finally, you need to (ask ProbLog to) compute explanations for the pictures, i.e. interpretations of what is on the pictures. Show for some well-chosen pictures which explanations are possible, and what the most likely explanations are for the picture (just like the examples on the slides). Additionally, you can show some interesting queries, or funny interpretations, or unexpected results as well. Of course, you need to describe and specify your domain completely in your report. Hint: start small (just some small aspects of your domain, and test implementation and inference on these, and then make things larger and more complex).

In a more structured way, you are required to describe clearly and fully:

- The domain and the individual images.
- The representation of observations, assumables and the causal theory between them in ProbLog.
- The style of reasoning employed (abductive, deductive) and what considerations have played a role.
- The queries used to infer information about individual images.
- The results of the queries and reflections on their effectiveness and intuitions behind them.
- General reflections on your model, how appropriate it is for this domain, how effective it is and how you would need to proceed if one would like to extend it to large datasets or realistic, real-time operation.