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

M. van der Heijden and J. Kwisthout

December 2014

1 Introduction to the assignments

Welcome to the second series of assignments on knowledge representation and reasoning. In this series you will model a probabilistic burglary prediction system, 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 small report, including formalizations, 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.

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?

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

3rd of February 2015

2 Introduction AILog

Rather than a programming language such as Prolog, AILog is a knowledge-representation and reasoning system built on top of Prolog. The language one can use in AILog for modelling, however, is very similar to that of Prolog. The main facilities offered by AILog are:

- a knowledge-representation and reasoning system that supports different forms of *logical and probabilistic* reasoning;
- an *ask-tell interface* that allow users to communicate with AILog;
- *explanation facilities* allowing users to explore why and how the system did, or did not, derive a particular fact.

3 AILog exercises

3.1 AILog representation

As Prolog, AILog supports Horn clauses. However, it is preferred to use in AILog a syntax that is slightly different from Prolog. A clause has to conform to the following syntax:

clause `::=` *head* `<-` *body*.

In a *body* one is advised to use the ampersand '&' for logical conjunction rather than the comma '`,`'. This allows making a syntactic distinction between Prolog programs and AILog representations, although AILog is also able to parse Prolog syntax.

3.2 Using AILog

First one needs to download the `ailog2.pl` Prolog program; program available on blackboard.

Next, AILog must be loaded into Prolog:

```
?- [ailog2].

AILog Version 2.3.3. Copyright 2009, David Poole.
AILog comes with absolutely no warranty.
All inputs end with a period. Type "help." for help.
ailog:
```

One can load knowledge bases into AILog using the `load` command:

```
ailog: load 'example'.
```

However, it is also possible to communicate clauses to AILog manually by means of the *ask-tell command interface*. By means of the `tell` command, clauses are added to the knowledge bases:

```
ailog: tell parent(X, Y) <- mother(X,Y).
ailog: tell mother(juliana,beatrix).
```

One can query the knowledge base by means of the `ask` command:

```
ailog: ask parent(X, Y).
Answer: parent(juliana, beatrix).
```

If one wishes to inspect the content of the knowledge base, the command `listing` can be used:

```
ailog: listing.
parent(A, B) <- mother(A, B).
mother(juliana, beatrix).
ailog:
```

Note that the variable names X and Y are renamed.

- *Compare the* PROLOG *program* `exercise0.pl` *to the* AILOG *knowledge base* `exercise10.ail`. *What are the differences?*

- *Load the knowledge base* `exercise10.ail` *into* AILOG *and inspect its content.*

- *Ask questions to the knowledge base, and compare the results to those obtained for* `exercise0.pl`. *Do you notice differences?*

You may have noticed that AILOG does not go into infinite recursion; rather it used a depth-bound to limited recursion. One can increase the depth-bound interactively when necessary.

3.3 Explanation facilities

The user interface of AILOG allows determining `how` a particular fact was derived, `why` a particular fact was `not` derived, and `why` a particular fact was derived. This is done by the commands: `how`, `whynot`, and `why`, respectively. Have a look at the following example:

```
ailog: ask predecessor(X,Y).
Answer: predecessor(emma, beatrix).
  [ok,more,how,help]: how.
   predecessor(emma, beatrix) <-
      1: predecessor(wilhelmina, beatrix)
      2: ruler(emma)
      3: parent(emma, wilhelmina)
   How? [Number,up,retry,ok,prompt,help]: 2.
   ruler(emma) <-
      1: queen(emma)
   How? [Number,up,retry,ok,prompt,help]:
```

- *Try to understand the way these commands work from the* AILOG *user manual.*

- *Load the knowledge base* `exercise10.ail` *into* AILOG *ask questions and experiment with the explanation facilities.*

3.4 User interaction

AILOG also allows prompting users by means of the `askable` build-in predicate. For example, by replacing

```
father(hendrik, juliana).
```

by

```
askable father(hendrik, juliana).
```

in the file `exercise10.ail`, one obtains the following behaviour:

```
ailog: ask predecessor(X,Y).
Is father(hendrik, juliana) true? [yes,no,unknown,why,help]: yes.
Answer: predecessor(emma, beatrix).
  [ok,more,how,help]:
```

- *Change one of the facts in the file* `exercise10.ail` *to* `askable`*, and see what happens when you query the knowledge base.*

3.5 Probabilities

Read `http://artint.info/code/ailog/ailog_man_16.html`. It gives an overview of how to specify probabilistic knowledge using AILog. You may also consult `http://artint.info/code/ailog/ailog_code/ch14/leaving.ail` for an example of the 'leaving' network.

Load the file `exprob.ail` which contains the medical example you have seen during the lecture.

- ▶ *Investigate the effect of flu on myalgia, i.e. compute $P(myalgia)$ and $P(myalgia \mid flu)$. Make use of* `predict` *and* `observe`*.*

- ▶ *Now investigate the effect of sport on flu. Is this the correct behaviour? If not, how would you adapt the model?*

- ▶ *Flu, or influenza, is a virus which mutates constantly, so to represent this by* `flu` *is fairly imprecise. Use first-order logic in AILog so that the model can reason about different variants of the influenza virus, which all cause fever and myalgia. Include some additional rules to characterize subtypes of influenza and investigate the probability distribution by querying it. Note that you might need the prob_threshold command if probabilities are small (see the AILog manual).*

4 Assignment 2-1: Probabilistic representation and reasoning (and burglars)

In this first assignment, you will investigate how to define a Bayesian network using AILog and use the network to answer some queries.

1. Read the following story:

   > Mr. Holmes receives a telephone call from his neighbor Dr. Watson stating that he hears a burglar alarm sound from the direction of Mr. Holmes' house. If there is a burglar present (which could happen once every ten years), the alarm is known to go off 95% of the time. Preparing to rush home, Mr. Holmes recalls that Dr. Watson is known to be a tasteless practical joker. There's a 40% chance that Watson is joking and the alarm is in fact off. However, if the alarm is on, Holmes expects Watson to call 80 percent of the time. He decides to first call his other neighbor, Mrs. Gibbons, who, despite occasional drinking problems, is far more reliable. She may not have heard the alarm in 1% of the cases and is thought to erroneously report an alarm when it is in fact off in 4% of the cases.
   >
   > Mr. Holmes remembers having read in the instruction manual of his alarm system that the device is sensitive to earthquakes and can be triggered by one accidentally in 1 every five cases. A burglary and an earthquake can be seen as independent causes. Other causes which will trigger the alarm do not exist. The incidence rate for earthquakes is about once every 10 years. He realizes that if an earthquake had occurred, it would definitely be on the news. So, he turns on his radio and waits around for a newscast. Of course, sometimes the newscast can be mistaken. This will happen only once per 5000 broadcasts.

2. Draw a Bayesian network (BN) that captures the (in)dependencies in the story.

3. Write down the corresponding conditional probability tables (CPTs) and fill them out using the information in the story (discuss your design decisions).

4. Write an AILog program which defines the probabilistic knowledge associated with the network

5. Use AILog to answer the following queries:

   (a) the prior probability of a burglary

   (b) the probability of a burglary given that Watson called

   (c) the probability of a burglary given that Gibbons also reports it

   (d) the probability of a burglary given that the newscast reported an earthquake; explain why the probabilities change the way they do.

   (e) write down the most probable explanation for the observed evidence

6. Explain how AILog computes the probability of the alarm going off. Also compute this probability using variable elimination and compare to the computations of AILog. What are the differences and what are the similarities? Can you think of situations where you would prefer one method over the other?

7. Suppose now getting more information about burglaries. Mr. Holmes house is in a residential area consisting of 10,000 houses, in which (only) the burglars Joe, William, Jack, and Averall are active. Each day a burglar decides whether he wants to work or not, and on average this happens only 5 days a week. However, some of them will only burgle if some specific colleagues will join them, given by the following knowledge:

```
needs(joe, []).
needs(william, []).
needs(jack, [joe]).
needs(averall, [jack, william]).
```

i.e., Joe and William do not need anyone in particular, Jack only burgles if Joe is around, and Averall only burgles if both Jack and William are there. To avoid loneliness in the cold nights, the burglars will only work if they form a group of at least two. In addition, all active burglars on one night stay together (so they will never split-up in two groups). Finally, if they decide to burgle, then they will burgle 3 houses a night. Model this situation in AILog and derive the (new) probability that there is a burglary in Holmes' house.

8. Could you also represent this knowledge in a Bayesian network? If not, explain why not. If you can, show what this looks like and explain the advantages or disadvantages of the AILog representation compared to a Bayesian network representation.

5 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 reasoning 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 year a KeR 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

```
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 Marguerita, 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.

Modeling the domain in AILog Your domain (as well as the pictures-represented-as-sets-observations) needs to be modeled in AILog. Look up the manual of AILog to see which other features you can use. 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 `observe` in AILog.

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 AILog. 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 AILog 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 AILog. Finally, you need to (ask AILog 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 AILog
- 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 would need to proceed if one would like to extend it to large datasets or realistic, real-time operation.