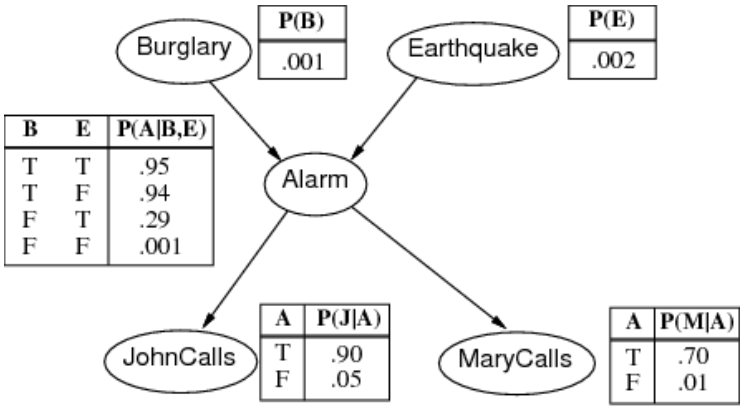
CMP9794M Advanced Artificial Intelligence – Workshop Week 4

UNIVERSITY OF LINCOLN

Summary: In this workshop you will carry out calculations related to Bayesian networks using approximate inference. In particular, you will use an implementation of Rejection Sampling algorithm discussed during the previous lecture. The class diagram from previous workshops has been updated, see file `workshop-w4/doc/CMP9794_BayesNets_ClassDiagram.pdf`. The main update is in class BayesNetExactInference, which is now called `BayesNetInference` to support both exact and approximate inference. The workshop materials from this week (Zip file in Blackboard under week 4) do not require any software dependencies to install this time.

Task 1: **Approximate inference using the Alarm network**

a. Using the Alarm Bayes net discussed in the previous lecture, also shown here,



answer the probabilistic queries $P(B|J = true, M = true)$ and $P(B|J = true, M = true)$ using 1000 sampled events. Example commands:

- `python BayesNetInference.py RejectionSampling ..\config\config-alarm.txt "P(B|J=true,M=true)" 1000`
- `python BayesNetInference.py RejectionSampling ..\config\config-alarm.txt "P(E|J=true,M=true)" 1000`

Run the commands above and observe the differences across runs.

b. Since your results will strongly depend on the number of sampled events, you need to be aware of an appropriate number of samples to get reliable estimates. To do that, try out different numbers of samples and compare the results via 5 different runs for the query $P(B|J = true, M = true)$. Use a spreadsheet such as the one below to record your results. Your workshop materials have the following template for your convenience: doc/ SamplingExperiment.xlsb

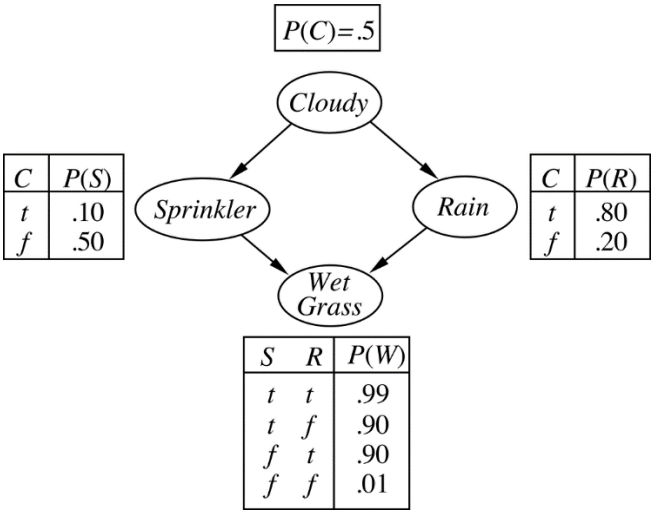
	A	B	C	D	E	F	G	H	I
1	**ALARM network**								
2		N=100		N=1000		N=10000		N=100000	
3	Run	true	false	true	false	true	false	true	false
4	1								
5	2								
6	3								
7	4								
8	5								
9	AVG	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
10	STD	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!

c. Using your knowledge acquired above in the task above, run the same program but using the number of samples N that you consider the most appropriate. Did you get similar results as compared to those with exact inference?

- `python BayesNetInference.py InferenceByEnumeration ..\config\config-alarm.txt "P(B|J=true,M=true)"`
- `python BayesNetInference.py InferenceByEnumeration ..\config\config-alarm.txt "P(E|J=true,M=true)"`
- `python BayesNetInference.py RejectionSampling ..\config\config-alarm.txt "P(B|J=true,M=true)" $YourNumberOfSamples`
- `python BayesNetInference.py RejectionSampling ..\config\config-alarm.txt "P(E|J=true,M=true)" $YourNumberOfSamples`

Task 2: **Approximate inference using the Sprinkler network**

a. Using the Sprinkler Bayes net discussed in the previous lecture, also shown below, answer the following probabilistic queries using Rejection Sampling with 1000 samples. This requires you to specify a config file for such a network – as done in week 2. Look at example config files in your workshop materials if needed, under the config folder, and save it in that folder.
P(S|C=true)=
P(R|C=true)=
P(W|S=false,R=true)=



b. Repeat the experiment of Task 1.b but for the query $P(W|S = false, R = true)$. Fill the spreadsheet mentioned in task 1 but this time using the table for the Sprinkler network. What observations can you notice from these two experiments, e.g., is the number of samples that you thought appropriate the same for both networks and queries?

c. Using your knowledge acquired above in the task above, run the same program but using the number of samples N that you consider the most appropriate. Did you get similar results as compared to those with exact inference?
   - `python BayesNetInference.py InferenceByEnumeration ..\config\config-sprinkler.txt "P(B|J=true,M=true)"`
   - `python BayesNetInference.py InferenceByEnumeration ..\config\config- sprinkler.txt "P(E|J=true,M=true)"`
   - `python BayesNetInference.py RejectionSampling ..\config\config-sprinkler.txt "P(B|J=true,M=true)" $YourNumberOfSamples`
   - `python BayesNetInference.py RejectionSampling ..\config\config-sprinkler.txt "P(E|J=true,M=true)" $YourNumberOfSamples`

After running the code above and verifying your calculations, you may want to inspect the class `BayesNetInference.py` to see how the algorithm `Rejection Sampling' was implemented.

Task 3: **Homework**

Implement the algorithm `Likelihood Weighting' and/or `Gibbs Sampling' discussed in the previous lecture and compare the inference results against those of Tasks 1 and 2. To facilitate this task, you can reuse and extend the class `BayesNetInference.py` and related files. You can also make your own separate implementation if that is your preference. Is your newly implemented algorithm faster to execute than Rejection Sampling and with reasonable approximate inferences?