Project 2 (BNs)

The tasks will be done in groups of 3 students, again. However, this time you can pick your group-mates.

The deadline for submitting the final report (again via EasyChair) will be on Friday 16.12 at 18.00.

Task 1: Developing a Bayesian Network Reasoner (45pts)

With this task we want you to deepen your knowledge and intuition about some reasoning tasks in a Bayesian network. As we want you to focus on the actual reasoning methods in Bayesian networks, we already provided you with some code that facilitates basic housekeeping methods like loading a network from an .BIFXML file. To use it (which we strongly advise you to do), git fork **https://github.com/sa-and/KR21_project2. git (https://github.com/sa-and/KR21_project2.git)** and carefully read the README.md file.
We also provided an (almost) empty reasoner class BNReasoner in which you can implement all your methods using a BayesianNet object. Note that you are encouraged to use methods that you already implemented. E.g. you can use factor multiplication for computing marginals. We expect you to implement the following algorithms:

- **Network Pruning**: Given a set of query variables Q and evidence e, node- and edge-prune the Bayesian network s.t. queries of the form P(Q|E) can still be correctly calculated. (**3.5pts**)
- **d-Separation**: Given three sets of variables X, Y, and Z, determine whether X is d-separated of Y given Z. (**4pts**)
- **Independence**: Given three sets of variables X, Y, and Z, determine whether X is independent of Y given Z. *(Hint: Remember the connection between d-separation and independence)* (**1.5pt**) **EDIT:** For this part, please assume that the BN is faithful. A BN is faithful iff for any set of variables X, Y, Z: X independent of Y given Z => X d-separated Y given Z
- **Marginalization:** Given a factor and a variable X, compute the CPT in which X is summed-out. (**3pts**)
- **Maxing-out:** Given a factor and a variable X, compute the CPT in which X is maxed-out. Remember to also keep track of which instantiation of X led to the maximized value. (**5pts**)
- **Factor multiplication:** Given two factors f and g, compute the multiplied factor h=fg. (**5pts**)
- **Ordering**: Given a set of variables X in the Bayesian network, compute a good ordering for the elimination of X based on the min-degree heuristics (**2pts**) and the min-fill heuristics (**3.5pts**). *(Hint: you get the interaction graph "for free" from the BayesNet class.)*
- **Variable Elimination:** Sum out a set of variables by using variable elimination. (**5pts**)
- **Marginal Distributions**: Given query variables Q and possibly empty evidence e, compute the

marginal distribution P(Q|e). Note that Q is a subset of the variables in the Bayesian network X with $Q \subset X$ but can also be Q = X. (**2.5pts**)

- **MAP:** Compute the maximum a-posteriory instantiation + value of query variables Q, given a possibly empty evidence e. (**3pts**)
- **MEP**: Compute the most probable explanation given an evidence e. (**1.5pts**)

Make sure also to provide some test cases in which you show that your implementations work (**5.5 pts**). For that, you can use the example Bayesian networks which you can find in the ”testing” folder. This includes two examples from the lecture as well as an extra example. It is not allowed to use already existing packages for inference on Bayesian networks to accomplish the tasks. Along with the report, you have to submit your code.

Task 2: Performance Evaluation (25pts)

The goal of this task is to get a better insight into the complexities of inference on BNs. More specifically, we are interested in how different methods influence the runtime of the inference. Interesting comparisons could be min-fill ordering vs. min-degree ordering, variable elimination vs. naive summing-out, small BNs vs. big BNs, pruned networks vs. unpruned networks, etc. Your task is to pick *two* interesting comparisons and perform an extensive analysis w.r.t. the runtime. If you feel like you want to compare dimensions that are not in these suggestions, you are free to do so.

This task will be graded based on the quality of your experimental setup (**5pts**), the correctness of the results (**2.5pts**), and how extensive your comparison is (**5pts**) for each of the two comparisons.

Task 3: Modelling a Use-Case (20pts)

In this part, you will model a use-case with a Bayesian network and answer interesting queries with the reasoner you have built in the first task. We want you to think of a real-world example and model a Bayesian network that represents the probabilities of the events in that example. We strongly advise you to only use binary variables. You will have to adhere to the following constraints for the model:

- It has at least 10 nodes.
- It has at most 3 root nodes.
- Nodes, which are not root nodes, have between 2 and 4 incoming edges.

We suggest you first try to model it simply with pen and paper. Once you are happy with the model, you should write the model in the .BIFXML format. See the README.md file of the git repository for a more detailed description of the format. There you can also find some examples. Once you have written it in .BIFXML format, you should be able to use the implementation of your

reasoner to answer interesting queries. For example, if you modeled a ”traffic” model, you could query the probability of being late for work given it is rainy and a weekend. Make sure you include at least:

- a prior marginal query.
- a posterior marginal query.
- a MAP and a MEP query.

We want you to report a diagram + CPTs of the variables in the report **(7pts)**. This should come along with a thorough explanation of the various variables and why you came up with the CPTs **(9pts)**. You also have to report the queries you investigated. You should discuss whether the results correspond with your expectation and document interesting insights the queries gave into your modeled problem **(4pts)**.

Report (10pts)

We expect you to hand in a report about the tasks in PDF format. Whatever is not reported in the report, will be considered as not done (with the exception of the code in Task 1). The report must be at max. 10 pages (excluding the Bayesian network model) with a page margin of at least 1in and font size 12pt. (LNCS format: https://www.springer.com/gp/computer-science/lncs) You will get 10pts for the overall quality of the report. The final submission has to be uploaded to Easy-chair (the link will be provided soon). And each person has to review one report.