# Project 2 KRR

Erman Acar, Andreas Sauter

November 2021

**Team-size:** For this project, you will have to make teams of size four. For canvas to know to know that you are working in groups, create a group (with title "Project 2 (PGM) groupnumber") in canvas and add the members.

## 1 Task 1: Developing a Bayesian Network Reasoner (40pts)

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` 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 an *BayesianNet* object. Here, we expect you to implement the following algorithms:

- ***d*-Separation:** Given three sets of variables $X$, $Y$, and $Z$, determine whether $X$ is independent of $Y$ given $Z$. (**5pts**)
- **Ordering:** Given a set of variables $X$ in the Bayesian network, compute a good ordering for elimination of $X$ based on the *min-degree* heuristics (**2pts**) and the *min-fill* heuristics (**2pts**). (*Hint: you get the interaction graph "for free" from the BayesNet class*)
- **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 (**5pts**).
- **Marginal Distributions:** Given query variables $Q$ and a possibly empty evidence $E$, compute the marginal distribution $P(Q|E)$ (**12pts**). *(Note that $Q$ is a subset of the variables in the Bayesian network $X$ with $Q \subset X$ but can also be $Q = X$.)*
- **MAP and MEP:** Given a possibly empty set of query variables $Q$ and an evidence $E$, compute the most likely instantiations of $Q$ (**12pts**).

Make sure to also provide some test cases in which you show that your methods work. 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 you code.

*Hint: Before implementing the marginal queries and MAP/MEP, it might be helpful to write methods for summing-out, multiplying factors and maxing-out.*

## 2 Task 2: Performance Evaluation (20pts)

Show the comparative average performance of your implementation on the aforementioned tasks (MAP, MPE) with different elimination order heuristics (min-order, min-fill vs. random order compared to one another) w.r.t. increasing size of variables (growing with 10 more variables or more each time).

by plots e.g., $x$-axis can time in seconds, while $y$-axis can be the number of variables.

*Hint: You can of course create such big BNs manually, but automatic generation would make your life much easier. This task will be graded according to the depth and elaboration of the analysis.*

# 3 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 to model a Bayesian network that represents the probabilities of the events in that example. We strongly advise you to only use variables which can either be true or false. 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:

- an a-priori marginal query.
- a posterior marginal query.
- one MAP and one MEP query.

We want you to report a diagramm + CPTs of the variables in the report (**6pts**). This should come along with a thorough explanation of the various variables and why you came up with the CPTs (**8pts**). You also have to report about 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 (**6pts**).

# 4 Report (20pts)

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 exposition of the report and **10pts** for the organization.

The final submission has to be uploaded to Easy-chair (the link will be provided soon). And each person has to review one report.

# 5 Summary on the Evaluation of the Project

Everybody in the group gets the same point. In summary, evaluation of the project is based on achieving the aforementioned tasks. Points weighting is as follows

- Implementing the Inference Engine (40 pts)
  - $d$-Separation (5 pts)

  - Ordering (4 pts) (a) min-degree (2 pts) (b) min-fill (2 pts )
  - Network Pruning (5 pts)
  - Computing a Marginal joint and posterior distribution (12 pts)
  - Computing MPE/MAP (12 pts)
- Performance Evaluation (20 pts)
  - Comparison of orderings (5 pts)
  - Comparison Network Sizes (15 pts)
- Modelling Use-Case (20 pts)
  - Story Explanation of the use-case (8 pts)
  - Quality of the use-case (6 pts)
  - Queries (e.g., Computing MPE/MAP, marginal (posterior), *d*-separation) (6pts)
- Quality of the scientific report (20 pts)
  - Exposition (10 pts)
  - Organisation (10 pts)

For questions, use discord, or canvas system or email to erman.acar@vu.nl and a.sauter@vu.nl

Best of Luck!
Erman & Andreas