CMP9794M Assessment Item 1: FAQ (version 1.1)

This is a complementary document to the assessment brief of the Advanced AI assignment, which aims to provide further details on how to solve the proposed tasks.

How to solve Task 1?

Solving task 1 requires writing a configuration file for at least one of the datasets provided. A baseline structure that has been suggested is that of a Naïve Bayes classifier. Later on, you can use of a randomly/semi-randomly/manually generated and/or a learnt one.

Once you have defined the structure, you should estimate the prior and conditional probabilities using the program `CPT_Generator.py` provided since the workshop of week 2.

One you have your configuration file including structure and parameters, you can use it for probabilistic inference. For that, you can use the program `BayesNetExactInference.py` provided during the workshop of week 2 or the program `BayesNetInference.py` provided during the workshop of week 4. Whilst the former only supports Inference by Enumeration, the latter supports both Inference by Enumeration and Rejection Sampling. You must decide which version you want to use in order to provide results accordingly. Example commands can be found in the workshop tasks of those two weeks, which will allow you to answer the probabilistic queries listed in page 1 of the briefing document. Note that if you want to use discretised values in your probabilistic queries (due to using discrete Bayes nets), simply identify the original value (in `*_data-original-train.csv`) and then observe its discretised version in the corresponding file called `*_data-discretized-train.csv`.

The program `ModelEvaluator.py`, delivered during the workshop of week 3, was originally developed to show an implementation of performance metrics using Naïve Bayes classifiers. This program has been extended for your convenience to support discrete Bayes Nets, which can be found in the materials of week 5 see file `workshop-w5-updated-ModelEvaluator.zip`. But the metrics LL and BIC are currently only generated for Naïve Bayes classifiers (due to them being an example implementation), and you are expected to extend LL and BIC for Bayes nets if you want to make use of them for any Bayes net.

Note that the code has been released incrementally (not independently), which means that week 5 for example includes previous weeks.

How to solve Task 2?

An example implementation for training and testing Gaussian Processes has been provided in the workshop of week 5. You should be able to apply it to data with continuous variables (namely `*-original-train.csv` and `*-original-test.csv`). In addition to that, you could also apply it to discrete data if you wish to do so, but this is optional.

If your PC runs out of memory when running `GaussianProcess.py` (delivered in week 5), your choices are for example and among others:

1. Use a smaller training set. You can randomly select M training examples instead of N, where M < N. For example, M could be only a quarter or a third of the whole dataset. This simple option implies ignoring a substantial amount of training data.
2. A less simple option consists in splitting the training data into K splits for training K-1 models, one model per data split, then find the best model using a validation set corresponding to split K. For example, assuming a K=5 would let you train 4 different Gaussian Process classifiers (one per split) which have to be evaluated using the evaluation split (i.e., K=5). The model with the highest evaluation performance can be declared as the best model and subsequently be applied to the test set.
3. Use a library such as GPytorch or GPFlow to find whether they are able to cope with both datasets of this assignment – and with what performance results.

The lecture of week 5 provided some other ideas on what to do as part of this task.

How to evaluate the performance of my trained Bayesian networks?

An implementation of different metrics has been provided to you in the program `NB_ModelEvaluator.py`. Look for function `compute_performance()`, which implements the following metrics: Balanced Accuracy, Area Under Curve, Brier Score, KL Divergence, Training Time, and Inference Time. Note that you are expected to train your Bayesian networks using training data and to evaluate them using test data. The later should neither be used for training nor for hyperparameter finetuning. The test set should only be used for testing – as the name suggests.

The program `ModelEvaluator.py` part of `workshop-w5-updated-ModelEvaluator.zip` has been extended to support the evaluation of both Naïve Bayes and Bayes Nets classifiers. In the code, look for flag `useBayesNet` to set your choice accordingly.

How can I get high marks?

Coding-wise, you could implement yourself an algorithm for probabilistic inference and/or for structure learning. Any of the algorithms discussed during lectures can be targeted, but if you want to implement a different algorithm not seen during lectures feel free to do so. But please be realistic with your ambition, your abilities, your progress so far, and the deadline.

Report-wise, write a clear and crafted report aligned to the brief requirements. Whilst the CRG provides indicators of mark categories, the following is a rough indication of your potential mark: the more you show your understanding for the methods of the assignment, the more you adhere to the brief requirements, and the more comprehensive your experimental results and analyses are – the higher your mark.

What to include in the report?

You are free to layout the structure and content of your report. The following is a suggestion. Explain conceptually how you solved each of the proposed tasks, report the results obtained using the provided datasets/methods/metrics, and analyse the experimented methods accordingly in order to draw conclusions of the performance of your methods (i.e., Bayesian networks, Gaussian processes). You are expected to make use of references not only by listing them but also by citing them whenever appropriate. Consider the tables below as a suggestion to fill them out based on your experimental results. But use the template provided (IEEE format) and avoid using your own template.

Probabilistic Query	Algorithm			
	Inf. By Enumeration	Rejection Sampling	Other algorithm	Another algorithm
P(target=0|write the evidence used here)	0.????	0.????	0.????	0.????
P(target=1|write the evidence used here)	0.????	0.????	0.????	0.????
P(outcome=0|write evidence used here)	0.????	0.????	0.????	0.????
P(outcome=1|write evidence used here)	0.????	0.????	0.????	0.????

Figure 1 Results of discrete Bayesian networks using baseline (Naïve Bayes) structures.

	Structure applied to Dataset X				
Metric	Structure1	Structure2	Structure3	Structure4	Structure5
Balanced Accuracy					
F1 Score					
Area Under Curve					
Brier Score					
KL Divergence					
Training Time					
Inference Time					

Figure 2 Results of Bayesian networks comparing different manual/random/learnt structures on dataset X.

	Gaussian Processes applied to Dataset Y				
Metric	Model1	Model2	Model3	Model4	Model5
Balanced Accuracy					
F1 Score					
Area Under Curve					
Brier Score					
KL Divergence					
Running Time					

Figure 3 Results of Gaussian processes comparing different models/implementations/kernels on dataset Y.

Is it okay to use any library?

Yes, no problem. Whilst using libraries is fine to confirm/compare results, you should try to implement something by yourself instead of only relying on libraries—for improved understanding of the concepts and algorithms covered in the module. Having said that, it is not mandatory to use the code provided during the workshops. But note that the materials of workshops have been provided with the purposes of exemplifying implementations of algorithms discussed and simplifying your solution to the proposed tasks in the assessment. In this way, you do not have to start your implementations of solutions from scratch. In addition to that, you are encouraged to make clear which are your contributions to solving the assessment tasks and to acknowledge the resources not implemented by yourself.

What to do when the code doesn't seem to work?

The code provided has been tested with multiple datasets including those of this assessment. There is one error that has been reported as part of the cardiovascular data, which is thrown due to an unexpected encoding in the feature `age`. If you encounter that, the following method can be adapted accordingly: `NB_Classifier.read_data()`

By adding the following line right after the for loop can solve the encoding problem:

```
line = line.replace('ï»¿', '')
```

Apart from the above, using the right names of random variables in configuration files and the right files can help to avoid errors.

What to do regarding Gaussian Bayes Nets?

The main materials for the assignment have been delivered in weeks 1 to 5. You are not expected to do much more beyond that point. Nonetheless, the code of the workshop of week 7 will provide support to train Bayesian networks using continuous data, which you will be able to use to generate additional results for your report if you wish to do so.

What referencing style to use in the report?

The template provided is IEEE-based, not Harvard-based. It is important to include and cite references using the recommended format.

When is this assessment due?

Please consult the Hand-in spreadsheet available via Blackboard.

Can I get the deadline of the assignment be extended?

Assessment deadlines are typically not changed. If you have a health problem or something serious that prevented you from working on your assignment, you can apply for an extension; but you will have to have evidence for your request. For that, go to **Blackboard > School of Computer Science module site > Useful Documents > Extension Documentation**. The delivery team cannot grant extensions because they are granted at school level.

I get an error when running BayesNetInference.py – what should I do?

You should run this program without any domain value for the target variable. Instead of trying to get a single probability such as P(Outcome=0|evidence) or P(target=1|evidence), run your program as in the command below. It should generate a probability distribution. From that distribution, you can take the probability you are interested in.

```
python BayesNetInference.py InferenceByEnumeration ..\config\config-
diabetes.txt "P(Outcome|Glucose=4,BMI=1,Age=5)"
```

In config files, which side of the | character when defining your structure represents the parents of a node?

The right hand-side are the parents of the variable on the left side. In other words, conditional probabilities can be read as P(QueryVariable|ParentsOfQueryVariable).

When implementing Hill Climbing, do we have to recalculate the CPTs (Conditional Probability Tables) for every "addition, deletion, or reversal" for evaluating effectiveness?

Yes. You need to recalculate the CPTs whenever the structure changes.

Why task 2 seems easier than task 1 in this assignment?

At first, it looks like task 2 is easier than task 1. But the application of Gaussian Processes to both datasets of this assignment can be challenging, especially for the larger dataset. So, don't underestimate task 2 and devote some attention to it instead of putting most of your attention to task 1. You have materials and recommendations on how to address both tasks.

I feel frustrated with the assignment, what can I do?

You should do the easy things first. Those activities do not require programming and are mostly experimental using the resources provided. It is advised that you analyse the first two questions above to identify what is easy vs. what is not. Again, the difficult things will require programming effort and will be more time consuming than the others. The latter are mostly experimental requiring running commands and understanding inputs and outputs.

Get in touch if you have any questions (hcuayahuitl@lincoln.ac.uk, rpolvara@lincoln.ac.uk).