1 Fitting a naive Bayes PGM to the data

delay	delay(0)	delay(1)	delay(≥ 2)	delay(NA)
age(20-23)	0.186	0.143	0.353	0.438
age(≤ 20)	0.769	0.714	0.647	0.5
age(>23)	0.0441	0.143	0.0	0.0625

Table 1: CPD for age and delay

Task 1.1

The group is the one of age(> 23). It is significant because this datum provides information about the total amount of people enrolled in the bachelor's whose age is greater than 23.

Task 1.2

Probabilities are equal when using the ratio and when using the fit. The results are the following:

- Probability for $delay \rightarrow 0 \rightarrow 0.769811320754717$
- Probability for $delay \rightarrow 1 \rightarrow 0.10566037735849057$
- Probability for $delay \rightarrow >=2 \rightarrow 0.06415094339622641$
- Probability for $delay \rightarrow$ NA $\rightarrow 0.06037735849056604$

The fit function computes the CPD over *delay*. The ratio function computes the relative probability. As *delay* does not depend on any variable, the relative probabilities in *delay* obtained from the ratio will be the same as those obtained from the CPD.

2 Probability queries (inference)

Task 2.1

According to the results obtained: $p(delay = 0|age <= 20) = 0.8010$. This is different from the previous question because now there is a prior acting, and the conditional probability given by this prior cannot be retrieved from the CPD since information flow goes in the opposite direction.

Task 2.2

Now, the condition provided is $delay = 0$. This can be directly obtained from the CPD, returning the following results:

- Age(20-23): 0.1863
- Age(<=20): 0.7696

- Age(>23): 0.0441

It can be seen that the most probable case is that if *delay* = 0, person's age will be lower or equal to 20 years old; and the least probable one is that the person belongs to the group of age over 23 years old.

Task 2.3

As it can be seen in table 1, probabilities match with those obtained at the beginning, when calculating the CPD. They must be the same since the results can be directly obtained from the CPD due to the information flow.

Task 2.4

The result obtained is the same: the function returns the maximum probability, which in this case corresponds to the age group <= 20.

3 Inverting all the edges

Task 3.1

Implemented in the provided code.

Task 3.2

This CPD has 4 different entries, which are the possible values *delay* can take: '0', '1', '>=2' and 'NA'.

Task 3.3

The number of entries, in this case, is given by the number of different values *delay* can take. Probabilities are calculated from the frequency of the values in the samples. Changing the samples will imply a modification in the values of the probabilities.

Task 3.4

When some case is missing, the probability is equal to zero. There are a total of 89 missing cases out of a total of 384 possible combinations (4 *delay* possible values times 96 different combinations of the values of the other variables).

Task 3.5

In this case, the CPD returns the following values:

- Probability for *delay* → 0 → 0.7611
- Probability for *delay* → 1 → 0.1116
- Probability for *delay* → >=2 → 0.0754
- Probability for *delay* → NA → 0.0520

The relative frequencies are the followings:

- Probability for *delay* → 0 → 0.769811320754717
- Probability for *delay* → 1 → 0.10566037735849057
- Probability for *delay* → >=2 → 0.06415094339622641
- Probability for *delay* → NA → 0.06037735849056604

The relative errors between these two sets of values are:

- Probability for *delay* → 0 → 0.01131617647
- Probability for *delay* → 1 → 0.05621428571
- Probability for *delay* → >=2 → 0.1753529412
- Probability for *delay* → NA → 0.13875

As it an be seen, the obtained values are slightly different. This is expected since we have inverted the information flow and now the probabilities of *delay* depend on several factors, as we have started with different conditional probability tables: *delay* now depends on its parent nodes.

4 Comparing accuracy

Task 4.1

Kullback-Leibler measures the difference between two probability distributions. Nevertheless, it is not considered a metric since generally the KL from p(x) to q(x) is not the same as KL from q(x) to p(x). Summing up, KL is not symmetric, so it cannot be a metric.

Task 4.2

Divergence is equal to infinity when some of the relative frequencies is equal to zero. KL is defined by:

$$D_{KL}(P||Q) = \sum_i \log\left(\frac{P(i)}{Q(i)}\right) P(i)$$

Therefore, for any of the probabilities equal to zero, the distance will be infinity.

Task 4.3

The first condition counts the amount of queries with number of evidence variables = n and with finite KL divergence for both models.

The second condition counts the number of queries with n evidence variables in which the KL measure is finite and smaller for the first model compared to the second one (being first model the one with *delay* as the parent node and the second one the one used in the previous section, which has *delay* as child node with all the others as parent nodes).

The third condition counts how many queries are there with n evidence variables in which the KL measure is finite and smaller for the second model compared to the first one.

The fourth condition is counting the number of queries in which the number of evidence variables is equal to 2 and either any of the models has an infinite KL.

The last condition is adding all the KL divergences for the first model.

These values can be used as a way to see how close each model is to the frequency when comparing with the other.

Task 4.4

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of 'inf'
1	100%	0%	2.2003202153404e-15	0.7698931216463456	12
2	81.25%	18.75%	0.1964272593863048	1.0490738726932298	38
3	33%	67%	0.20580125153705406	0.03627073187537378	39
4	0	100%	0.1922615863198352	0.0	48

Table 2: Calculations - target: *delay*

Analysing the table it is possible to observe that as N increases, performances of models 1 and 2 change. $N = 1$ shows that model 1 is always better than 2. Nevertheless, this value eventually decreases until model 2 becomes better in all cases, for $N = 4$.

It is necessary to remark that when using $N = 4$, the second model is being implemented. Therefore, information can be directly obtained from the CPD and this is why the second always outperforms the first one.

Task 4.5

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of 'inf'
1	86.11%	13.89%	0.722907844945035	1.5433633715860877	10
2	76.92%	23.08%	0.8136314814273684	1.4568907266781541	28
3	88.89%	11.11%	0.4019743347122403	0.5860525082782467	34
4	50.00%	50.00%	0.1331222385455232	0.12432247471690223	44

Table 3: Calculations - target: *age*

As it can be seen, a smoother transition between percentages according to the value of N can be observed now. In the previous case, probabilities could be directly obtained from the model, but in this case, this is not possible. This case is more complex and requires applying inference; this is why results are not presenting a pattern as defined as in the previous case and the first model performs considerably better when $N = 1$.

Task 4.6

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of 'inf'
1	98.63%	1.37%	0.9403945107152406	2.747775108744819	22
2	66.67%	33.33%	1.0936006350390703	2.301054067085836	65
3	66.67%	33.33%	0.5283421447710206	0.5791461319218795	83
4	60.00%	40.00%	0.19467646376113662	0.21432713508171417	95

Table 4: Calculations - target: *delay*, *age*

In this case, both variables *delay* and *age* as considered as targets. It can be seen how model 1 outperforms 2 in all cases. Model 1 was better in most of the cases before, so it makes sense that when joining then, model 1 keeps being the best.

6 Finding a better structure

Task 6.1

K2 score prefers simpler models, understanding by it those models which have a lower number of parent nodes. From here, it can be deduced that model 1 will be preferred, since it only has one parent node, while model 2 has four [1].

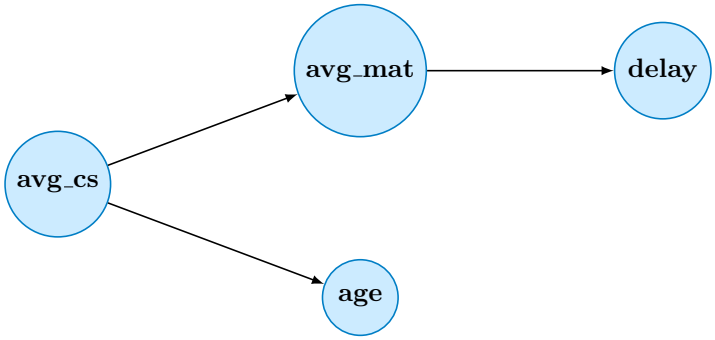
Task 6.2

Scoring methods have as goal measuring the performance of a network: i.e. - how well a network fits the data. They search over the whole space of possible configurations for the optimal one [2].

It is always necessary to have, besides training, a validation set. This allows to check how well the trained model fits data and also to avoid overfitting if the trained model does not properly generalize. Even if you selected the best possible model, it is necessary to confirm this fact.

Task 6.3

Selecting those with the highest score (less negative), the best configuration is the following:



For which the correspondent score is: -925.8214756938006.

It can be seen that $avg\_mat \perp\!\!\!\perp age$ given avg_cs, as well as $delay \perp\!\!\!\perp age$ given avg_cs.

Task 6.5

The result obtained by 'hill climb search' is the same that the one obtained by 'exhaustive search', which is the following:

$$[('avg\_cs','avg\_mat'),('avg\_cs','age'),('avg\_mat','delay')]$$

Nevertheless, when using BIC score, the result obtained is:

$$[('avg\_cs','avg\_mat'),('avg\_mat','delay')]$$

BIC score may underfit the model since it penalizes variables, even reaching to remove parameters. This is why the model obtained from the BIC score is simpler than the other one, and different from the one obtained in the previous task.

Task 6.6

Not in this case. As the dataset is not extremely big, is more convenient to directly compute the relative frequencies of data. When the model is complex, depends on a lot of variables, has huge amounts of samples..., in this cases a PGM is worth it; otherwise, it is more convenient to perform different methods and calculations.

References

[1] C. Borgelt and R. Kruse. An empirical investigation of the k2 metric. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 240–251. Springer, 2001.

[2] Y. Zhou. Structure learning of probabilistic graphical models: A comprehensive survey, 2011.