1.1

P(age | delay>=2) = 0

It is significant because it is 0, meaning there are no samples in the data which can lead to problems estimating a probability. But “excluding” something with probability 0 instead of having a small probability close to 0 can be a big negative impact for the resulting model accuracy.

1.2

Yes, the marginal distribution of the delay corresponds to the relative frequency in the data.

Probabilities are relative frequencies over infinite samples which states why they are related and here given the (limited) data.

Additionally, MLE in Bayes Nets with Table CPDs can be treated as separate MLE problems given each possible assignment. And for multinomial cases we can just count the cases and state the empirical average.

2.1

P(delay=0 | age<=20) = 80,10%

In 1.1 it was P(age | delay).
The variable and evidence are exchanged.

2.2

Lowest: P(age>23 | delay=0) = 4,41%

Highest: P(age<=20 | delay=0) = 76,96%

2.3

Yes, the probabilities and relative frequencies are the same.

They should be the same. See 1.2.
Because the same sufficient statistics result in the same likelihoods, here counts per category for multinomial distribution.

2.4

MAP result is “age <= 20” is the same as in 2.2

MAP includes the prior. MLE is MAP with a uniform prior.
If the priors is sufficiently different to change the respective likelihoods, the results of MLE and MAP can be different.

3.2

3*2*4*4*4 = 384 entries, i.e. combinations for CPD of delay

3.3

The data set has 265 samples.

Since 384 combinations > 265 samples, not every combination can have a sample.

If a combination is not present for every value of delay it results in a uniform distribution, having the least assumptions about the data.

If one value is present this has 100% while the rest has 0%.

See image. (Cursor placement is not important.)

age	age(20-23)	age(20-23)
avg_cs	avg_cs(2<3)	avg_cs(2<3)
avg_mat	avg_mat(2<3)	avg_mat(2<3)
gender	gender(0)	gender(1)
delay(0)	0.25	1.0
delay(1)	0.25	0.0
delay(>=2)	0.25	0.0
delay(NA)	0.25	0.0

3.4

See 3.3

58 / 96 combinations have a uniform distribution.

3.5

The worst relative error is observed for both delay { “1”, “>=2“ } error = 1.38778e-17.

Should an exact inference always return the exact relative frequencies in data for this PGM structure? Why or why not?

In theory for a perfect specified PGM exact inference should lead to the exact answer. But here we give the model a structure and give it a (too small) amount of data. Hence assumptions about the

independencies and the structure of the PGM care made, but can be wrong, which can lead to errors as briefly detailed in 1.1.

4.1

The KL divergence is a measure of difference between two distributions.

It is not a metric because

1. it is not symmetric KL(X || Y) != KL(Y || X)
2. it does not satisfy the triangle inequality

4.2

The error comes from division by 0 since no samples are present for some constellations.

4.3

- n=2
- [len([r for r in divs2 if len(r[0][1])==n]),
    - condition on the number of prior conditioned variables
    - also present in the other queries
- len([r for r in divs2 if len(r[0][1])==n and r[3] < r[5]]),
    - condition on which model had the smallest divergence => 1 better
- len([r for r in divs2 if len(r[0][1])==n and r[3] > r[5]]),
    - condition on which model had the smallest divergence => 2 better
- len([r for r in divs if len(r[0][1])==n and not(math.isfinite(r[3]) and math.isfinite(r[5]))]),
    - condition to check for infinite divergences
- sum(r[3] for r in divs2 if len(r[0][1])==n)])
    - sum of divergence for a model over filtered queries

4.4

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of inf
1	55/55	0/55	-4.0051e-16	3.529	0
2	22/34	12/34	4.0212	5.6310	7
3	16/38	22/28	8.5358	6.449	19
4	0/26	25/26	7.9163	0	21

M1 starts clearly better and wins very often early, but then worsens with increasing evidence components. While M2 shows the inverted pattern and improves with increasing components.

The number of cases with infinite divergence increases with increasing components.

4.5

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of inf
1	49/57	8/57	6.2146	7.3018	0
2	17/33	16/33	6.7553	5.9541	13
3	5/23	18/23	13.1173	4.5774	22
4	3/15	12/15	12.1909	2.3514	37

Same as 4.4

4.6

N	M1 wins %	M2 wins %	Sum div M1	Sum div M2	Number of inf
1	94/98	4/98	5.9847	14.1627	0
2	53/88	35/88	13.6832	21.6293	8
3	26/70	44/70	26.5884	17.0036	29
4	2/36	34/36	12.5675	2.4925	71

Same as 4.4

5.1

K2 score higher is better i.e. less negative.

[-1037.7184908064387, -1096.3658651519336]

Model 1 is better.

5.2

What is the main idea with structure scoring methods and how are using them better than training on 100% of the data and doing no validation?

“Assign a score to each Bayesian network structure according to a scoring function and find the structure that optimizes the score. […] It is computationally more efficient to use less data since both the running time and memory usage are exponential in the number of variables in the worst case.”

Liu, Z., Malone, B. & Yuan, C. Empirical evaluation of scoring functions for Bayesian network model selection. *BMC Bioinformatics* **13** (Suppl 15), S14 (2012). https://doi.org/10.1186/1471-2105-13-S15-S14

Additionally, training on 100% of the data and doing no validation does not guarantee that the model will generalize well to new, unseen data, therefore using a common train-test split to simulate “new data” is helpful.

5.3

-925.8214756938007 [('avg_cs', 'avg_mat'), ('avg_cs', 'age'), ('avg_mat', 'delay')]

It suggests a conditional independence of (avg_mat, delay) and (age) given avg_cs.

And avg_cs and delay, given avg_mat.

In these cases the nodes are d-separated:

- children are independent given their parents
- child given parent is independent of previous history of parent

See drawn graph at the end.

5.6

Reflect over the use PGMs on the given dataset. Are using PGMs worth it in this case?

PGMs are useful for complex systems with many relationships, in cases with missing or unseen data and high dimensional data. But for that they also need large datasets. In this case the structure is not complicated, and our dataset is not large with <300 samples. Additionally, see the number of missing samples for combinations, which makes the lack of data again visible.

Therefore I would say no, they are not worth it in this case.