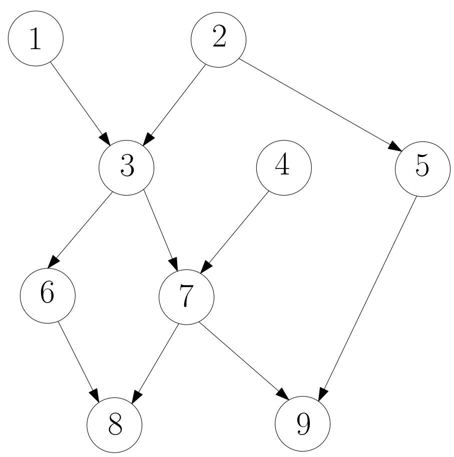
Data Mining 2020
Exercises Bayesian Networks

Exercise 1: Independence Properties of Bayesian Networks

Consider the following directed independence graph.



(a) Give the factorization of $P(X_1, X_2, \ldots, X_9)$ corresponding to this independence graph.

Construct the appropriate moral graphs to check whether the following conditional independencies hold:

(b) $6 \perp\!\!\!\perp 7$

(c) $6 \perp\!\!\!\perp 7 \mid 3$

(d) $6 \perp\!\!\!\perp 7 \mid 8$

(e) $2 \perp\!\!\!\perp 9 \mid \{5, 7\}$

(f) $2 \perp\!\!\!\perp 9 \mid \{3, 5\}$

(g) $5 \perp\!\!\!\perp 8$

(h) $5 \perp\!\!\!\perp 8 \mid 3$

Exercise 2: Learning Bayesian Networks

In structure learning of Bayesian networks one often uses a score function to determine the quality of a network structure, in combination with a hill-climbing local search strategy. One popular score function is BIC (Bayesian Information Criterion):

$$\mathrm{BIC}(M) = \mathcal{L}(M) - \frac{\ln n}{2} \dim(M),$$

where $\mathcal{L}(M)$ denotes the value of the loglikelihood function of model M evaluated at the maximum (also called the loglikelihood score), $\dim(M)$ denotes the number of parameters of model M, and n denotes the number of observations in the data set.
We want to construct a model on the following data set on 3 binary variables:

	X_1	X_2	X_3
1	1	1	0
2	1	0	0
3	1	0	0
4	1	0	0
5	0	0	0
6	0	1	1
7	1	1	1
8	0	1	1
9	0	0	1
10	0	0	1

The initial model in the search is the mutual independence model (corresponding to the empty graph).

(a) Give the maximum likelihood estimates of the parameters of the mutual independence model.

(b) Compute the loglikelihood score of the mutual independence model. The loglikelihood score is the value of the loglikehood function evaluated in the maximum. Use the *natural* logarithm in your computations.

(c) Give all neighbours of the current model, assuming a neighbour can be obtained by either: adding an edge, removing an edge, or reversing an edge. Which of these neighbour models are equivalent? **Note:** Define the skeleton of a directed graph as the undirected graph obtained by dropping the directions of the edges. Two models are equivalent if and only if they have the same skeleton and the same v-structures.

(d) Would adding an edge from X_1 to X_2 (or vice versa) improve the BIC score? Explain.

(e) Consider the neighbour model obtained by adding an edge from X_1 to X_3. Is this model preferred to the initial model on the basis of the BIC-score? Explain.

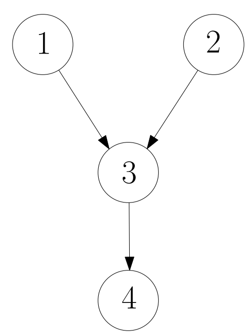
Exercise 3: Learning Bayesian Networks

This exercise is similar to exercise 2; it just gives you more practice.

We are constructing a model on the following data set on 4 binary variables:

	X_1	X_2	X_3	X_4
1	1	1	0	0
2	1	0	0	1
3	1	0	0	0
4	1	0	0	1
5	0	1	0	1
6	1	1	1	1
7	1	1	1	0
8	0	1	1	0
9	0	0	1	0
10	0	0	1	0

Suppose the current model in the search has the following structure:



(a) Give the maximum likelihood estimates of the model parameters.

(b) Compute the loglikelihood score for the given model and data set. Use the *natural* logarithm in your computations.

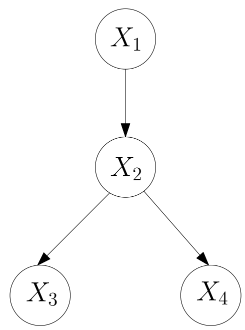
(c) Compute the BIC score of this model on the given data set.

(d) Give all neighbours of the current model, assuming a neighbour can be obtained by either: adding an edge, removing an edge, or reversing an edge. Which of these neighbour models are equivalent?

(e) Consider the neighbour model obtained by adding an edge from X_1 to X_4. Is this model preferred to the current model? Explain.

Exercise 4: Essential Graph

Construct a graph from the DAG below as follows: orient all edges whose direction is fixed in the equivalence class that the DAG belongs to, and make edges bi-directional if there are two members in the equivalence class which have edges in opposite directions. The resulting graph is called the *essential* graph. Recall that two DAGs belong to the same equivalence class iff they have the same skeleton and the same immoralities (v-structures). Hint: it doesn't suffice to check if you remain in the same equivalence class if you turn a single edge around!



Exercise 5: Structure Learning

We perform a greedy hill-climbing search to find a good Bayesian network structure on 5 variables denoted A, B, C, D, and E. Neighbour models are obtained by adding, deleting, or reversing an edge. We start our search from the empty graph. In step 1 of the search we find that adding the edge $A \to D$ gives the biggest improvement in the BIC score. Which Δ scores do we need to compute in step 2?