

Assignments of the subject "Probabilistic Methods"

Topic 3. Bayesian Network Construction

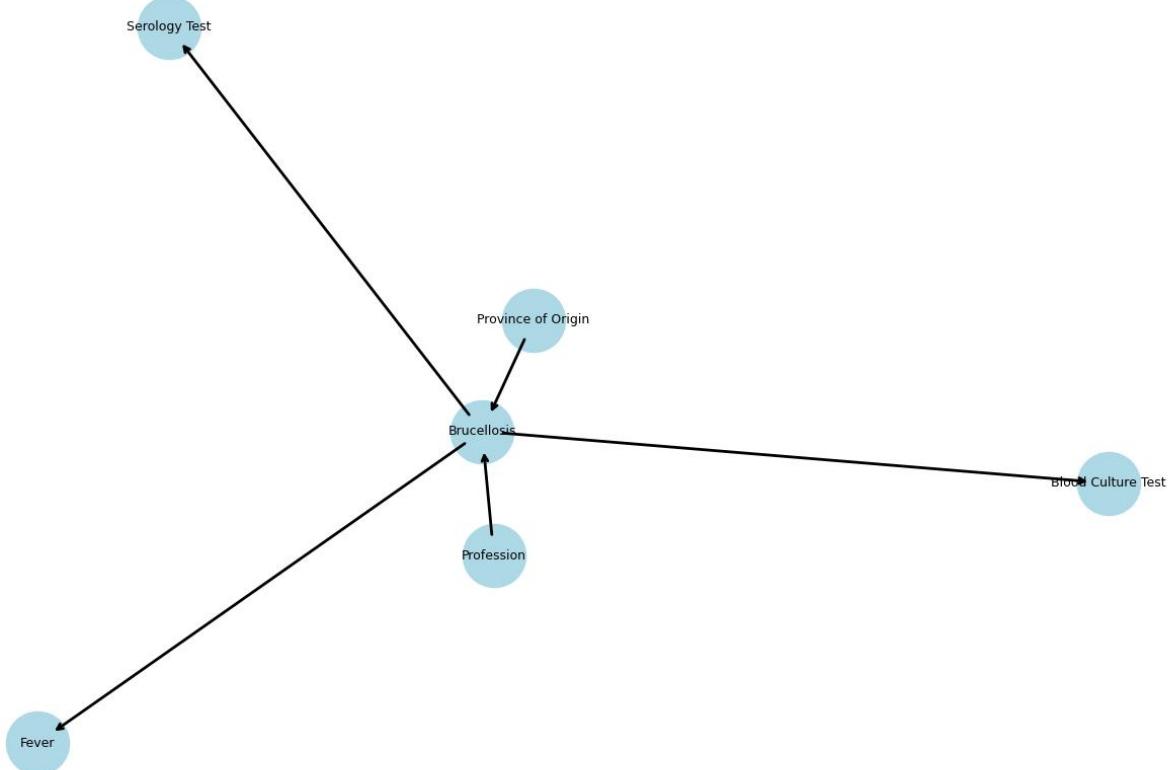
Exercise 3.1.

Draw a causal graph for each of the following problems:

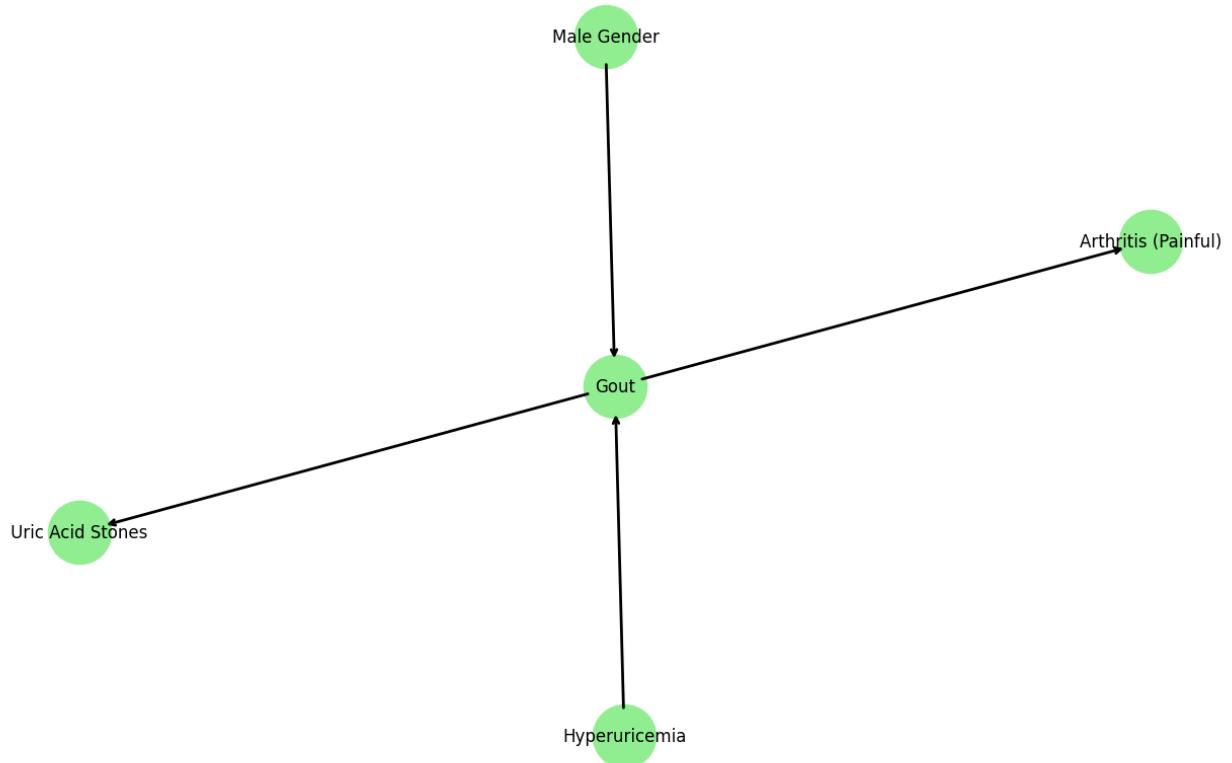
1. The province of origin and the profession are two of the risk factors for brucellosis. One of the symptoms of this disease is fever. The two most common tests to diagnose this disease are blood cultures and serology.
2. Gout is a disease that mainly affects adult males. Almost all people with it have a history of hyperuricemia, but only 5% of patients with hyperuricemia develop gout. The first manifestation of acute gout is extraordinarily painful arthritis. Half of the uric acid stones found are also due to this disease.
3. (Example proposed in [1]) Metastatic cancer can cause serum-calcium elevation and is, in turn, one of the possible causes of brain tumor. Both elevated serum calcium and brain tumor can produce coma. The brain tumor can also produce severe headaches.
4. (Adapted from [3]) A patient's dyspnea may be due to, among other causes, tuberculosis, lung cancer, or bronchitis. The fact that the patient has recently been to Asia increases the likelihood that he or she will suffer from tuberculosis, and the fact that he or she is a smoker is a risk factor for both lung cancer and bronchitis. A positive X-ray result indicates the presence of tuberculosis or lung cancer.

Solution 3.1.

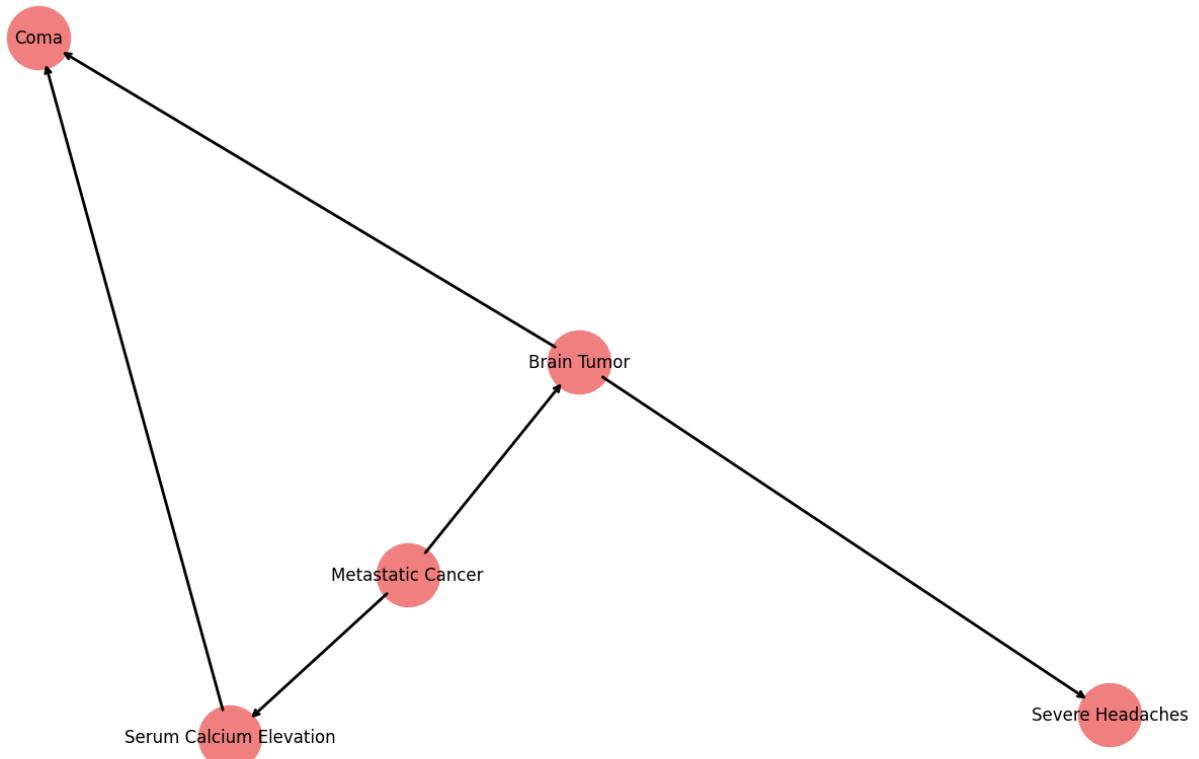
1) Causal Graph for Brucellosis



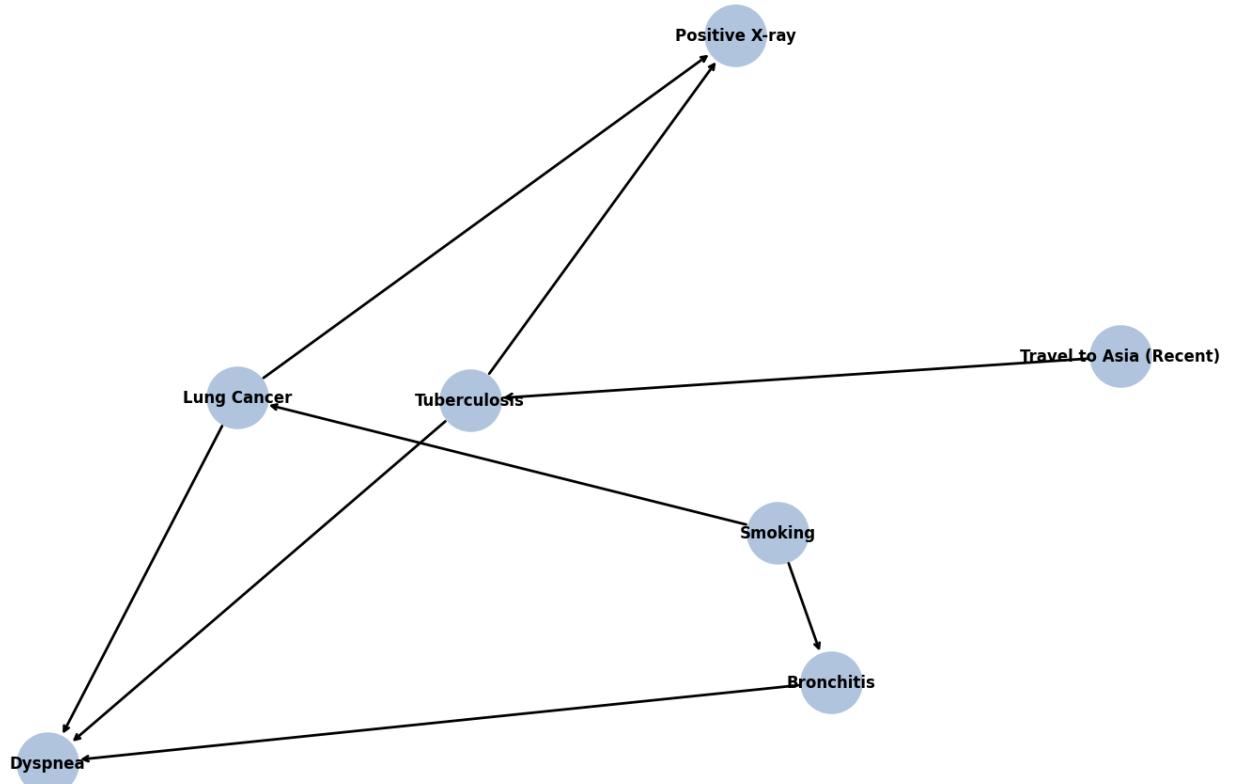
2) Causal Graph for Gout Risk Factors and Manifestations



3) Metastatic Cancer



4) Dyspnea



Exercise 3.2.

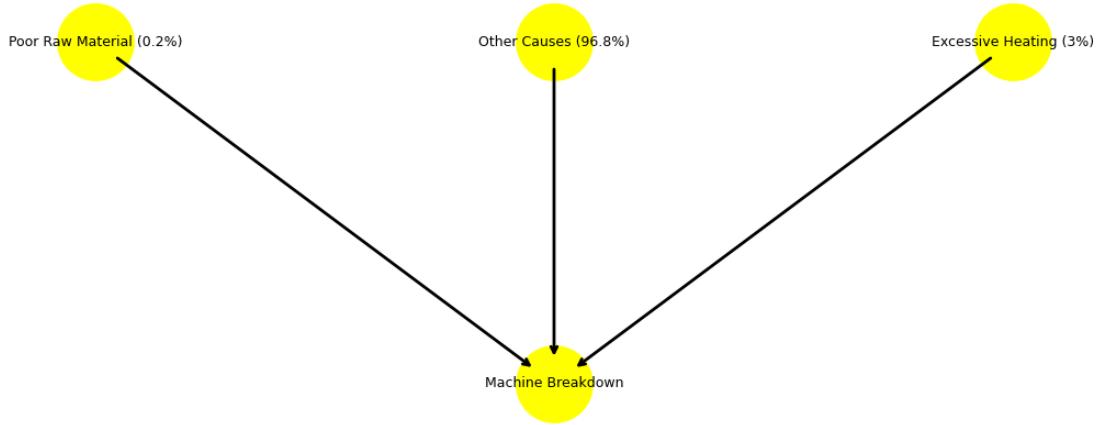
A machine for the production of plastic parts can break down due to two main causes: the poor quality of the raw material used, which occurs in 0.2% of cases, and excessive heating of the lathe, which occurs in 3%. Poor raw material quality causes a breakdown in 8% of cases, while heating causes it in 2%. There are other minor causes, which can produce a breakdown with a probability of 0.0001.

1. Draw a Bayesian network diagram showing the three variables of the problem.
2. Calculate "by hand" (i.e., by applying the equations shown in [2]) the values in the conditional probability of failure table. If you use a canonical model, indicate which one you have chosen and what hypothesis you have used to be able to apply it.
3. Build a Bayesian network for this problem with OpenMarkov and capture the screen where this table is displayed, to check that the values match those calculated in the previous section.
4. When a breakdown occurs, what is the most likely cause? To answer, calculate "by hand" the probability a posteriori of each of the causes and then check that the values match those shown by OpenMarkov.
5. There has been a breakdown, but it has been verified that the raw material was of good quality. What is the probability that there has been warming? As in the previous case, calculate the probability "by hand" and compare it with OpenMarkov's.

Solution 3.2.

1)

1) Machine Breakdown Causes

**2, 3 & 4)**

We model the three "cause" events as a **mutually-exclusive**, exhaustive partition of the sample space:

- $C_1 = \{\text{poor raw material}\}$, with $P(C_1) = 0.002$
- $C_2 = \{\text{excessive heating}\}$, with $P(C_2) = 0.03$
- $C_3 = \{\text{other minor causes}\}$, with $P(C_3) = 1 - 0.002 - 0.03 = 0.968$

and the "failure" event F . We are given

$$P(F | C_1) = 0.08, \quad P(F | C_2) = 0.02, \quad P(F | C_3) = 0.0001.$$

1. Total probability of failure

By the law of total probability,

$$\begin{aligned} P(F) &= \sum_{i=1}^3 P(C_i) P(F | C_i) = 0.002 \cdot 0.08 + 0.03 \cdot 0.02 + 0.968 \cdot 0.0001 \\ &= 0.00016 + 0.0006 + 0.0000968 = 0.0008568. \end{aligned}$$

2. Joint probabilities $P(C_i \wedge F)$

$$P(C_i \wedge F) = P(C_i) P(F | C_i),$$

so

$$\begin{aligned} P(C_1 \wedge F) &= 0.002 \cdot 0.08 = 0.00016, \\ P(C_2 \wedge F) &= 0.03 \cdot 0.02 = 0.00060, \\ P(C_3 \wedge F) &= 0.968 \cdot 0.0001 = 0.0000968. \end{aligned}$$

3. Posterior probabilities $P(C_i | F)$

By Bayes' theorem,

$$P(C_i | F) = \frac{P(C_i \wedge F)}{P(F)} = \frac{P(C_i) P(F | C_i)}{P(F)}.$$

Numerically,

$$\begin{aligned} P(C_1 | F) &= \frac{0.00016}{0.0008568} \approx 0.1867, \\ P(C_2 | F) &= \frac{0.00060}{0.0008568} \approx 0.7003, \\ P(C_3 | F) &= \frac{0.0000968}{0.0008568} \approx 0.1130. \end{aligned}$$

4. Conditional-probability table

Cause	$P(C_i)$	$P(F C_i)$	$P(C_i \wedge F)$	$P(C_i F)$
Poor raw material (C_1)	0.002	0.08	0.000160	0.1867
Excessive heating (C_2)	0.030	0.02	0.000600	0.7003
Other minor causes (C_3)	0.968	0.0001	0.0000968	0.1130
Total	1.000	—	0.0008568	1.000

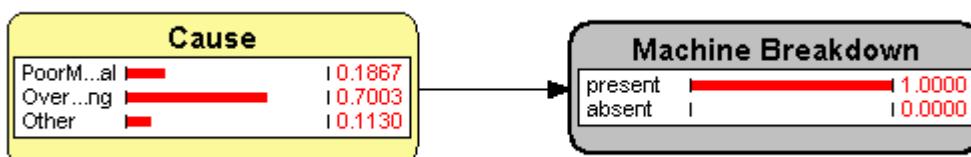
Model and hypothesis

We have used a **simple Bayesian (mixture) model** in which the causes C_i form a mutually exclusive and exhaustive partition of all events, and each one has its own conditional failure probability. Under that hypothesis we can apply:

1. **Law of total probability:** $P(F) = \sum_i P(C_i)P(F | C_i)$

2. **Bayes' theorem:** $P(C_i | F) = \frac{P(C_i) P(F | C_i)}{P(F)}$

No other dependencies or interactions between causes are assumed.



5)

We are asked to compute:

$$P(C_2 \mid F, \neg C_1)$$

where:

- F : breakdown occurs,
- C_1 : poor raw material,
- C_2 : excessive heating,
- C_3 : other minor causes.

The prior probabilities are:

$$P(C_1) = 0.002, \quad P(C_2) = 0.03, \quad P(C_3) = 0.968$$

and the conditional probabilities of failure given each cause are:

$$P(F \mid C_1) = 0.08, \quad P(F \mid C_2) = 0.02, \quad P(F \mid C_3) = 0.0001$$

Step 1: Recompute conditional priors under $\neg C_1$

Since C_1 did not occur, we renormalize the probabilities over the remaining possibilities:

$$P(\neg C_1) = 1 - P(C_1) = 0.998$$

Then:

$$P(C_2 \mid \neg C_1) = \frac{P(C_2)}{P(\neg C_1)} = \frac{0.03}{0.998} \approx 0.03006$$

$$P(C_3 \mid \neg C_1) = \frac{P(C_3)}{P(\neg C_1)} = \frac{0.968}{0.998} \approx 0.96994$$

Step 2: Compute joint probabilities with failure

We compute the joint probabilities of failure under the assumption that C_1 did not occur:

$$P(C_2 \wedge F \mid \neg C_1) = P(C_2 \mid \neg C_1) \cdot P(F \mid C_2) = 0.03006 \cdot 0.02 = 0.0006012$$

$$P(C_3 \wedge F \mid \neg C_1) = P(C_3 \mid \neg C_1) \cdot P(F \mid C_3) = 0.96994 \cdot 0.0001 = 0.000096994$$

Step 3: Total probability of failure given $\neg C_1$

$$P(F \mid \neg C_1) = P(C_2 \wedge F \mid \neg C_1) + P(C_3 \wedge F \mid \neg C_1)$$

$$P(F \mid \neg C_1) = 0.0006012 + 0.000096994 = 0.000698194$$

Step 4: Compute posterior probability

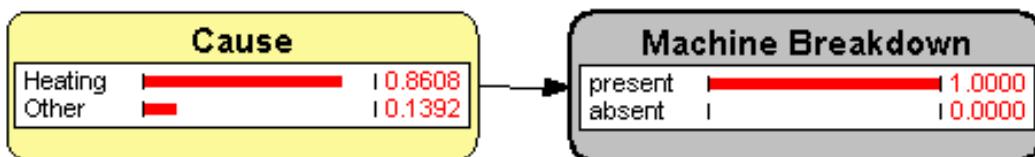
We apply Bayes' theorem to compute:

$$P(C_2 \mid F, \neg C_1) = \frac{P(C_2 \wedge F \mid \neg C_1)}{P(F \mid \neg C_1)} = \frac{0.0006012}{0.000698194} \approx 0.861$$

Final Answer

$P(C_2 \mid F, \neg C_1) \approx 0.861$

Thus, if a breakdown has occurred and it is known that the raw material was of good quality, the probability that excessive heating was the cause is approximately **86.1%**.

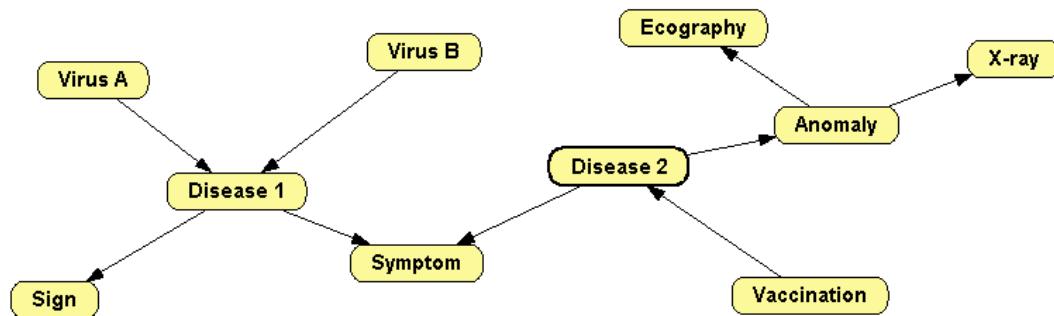
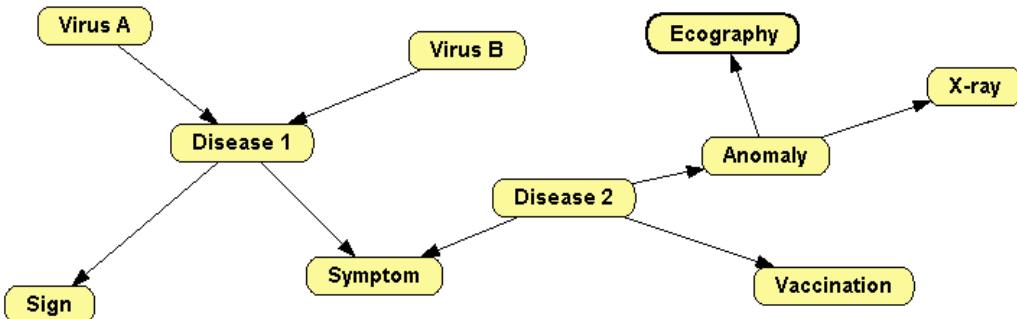


Exercise 3.3.

Using the BN-two-diseases.pgm network, generate a database of 10,000 virtual patients, as indicated in the OpenMarkov tutorial. Save it in xls format.

1. Open the database to view its contents and capture a snippet of the screen.
2. Apply the *hill climbing* algorithm using OpenMarkov interactively to learn a Bayesian network from this database. Capture an image of the learned network. Explain, as far as possible, why this algorithm is not able to recover the original network (the rather complicated question; it will not be possible for you to find a completely satisfactory answer, but it will be useful to reflect on it).
3. Repeat the previous section for the PC algorithm. Apply the PC algorithm to different values of the significance level, α , for example, 0, 0.001, 0.01, 0.1, 0.5, 0.7, 0.9, 0.99, 1.¹ Plot the number of links in the network as a function of α and explain the results obtained, especially for the extreme values, 0 and 1.

¹Be careful not to confuse the *level of significance* with the parameter of *generalized Laplace correction*. In the literature on Bayesian network learning, both are usually represented by the letter α , but their meaning and use are completely different: the significance level is used in the PC algorithm to learn the structure of the network, while the parameter of the generalized Laplace correction is used in all learning algorithms to estimate the probabilities when we already have the structure of the network.

Solution 3.3.**1)****2)**

Why the arc between Disease2 and Vaccination flips

1. **Score-equivalent networks.**

Most score-based learners (BIC, BDeu) cannot distinguish between two DAGs in the same *Markov equivalence class*. If flipping an arc does not change the set of conditional independencies, the score is identical, so hill-climbing may settle on either orientation.

2. **Lack of v-structure.**

In the original model, Disease2 is the only parent of Vaccination (no collider or moral-graph change if you reverse the link). As long as there's no "shielded collider" ($A \rightarrow C \leftarrow B$), the two orientations are indistinguishable from pure observational data.

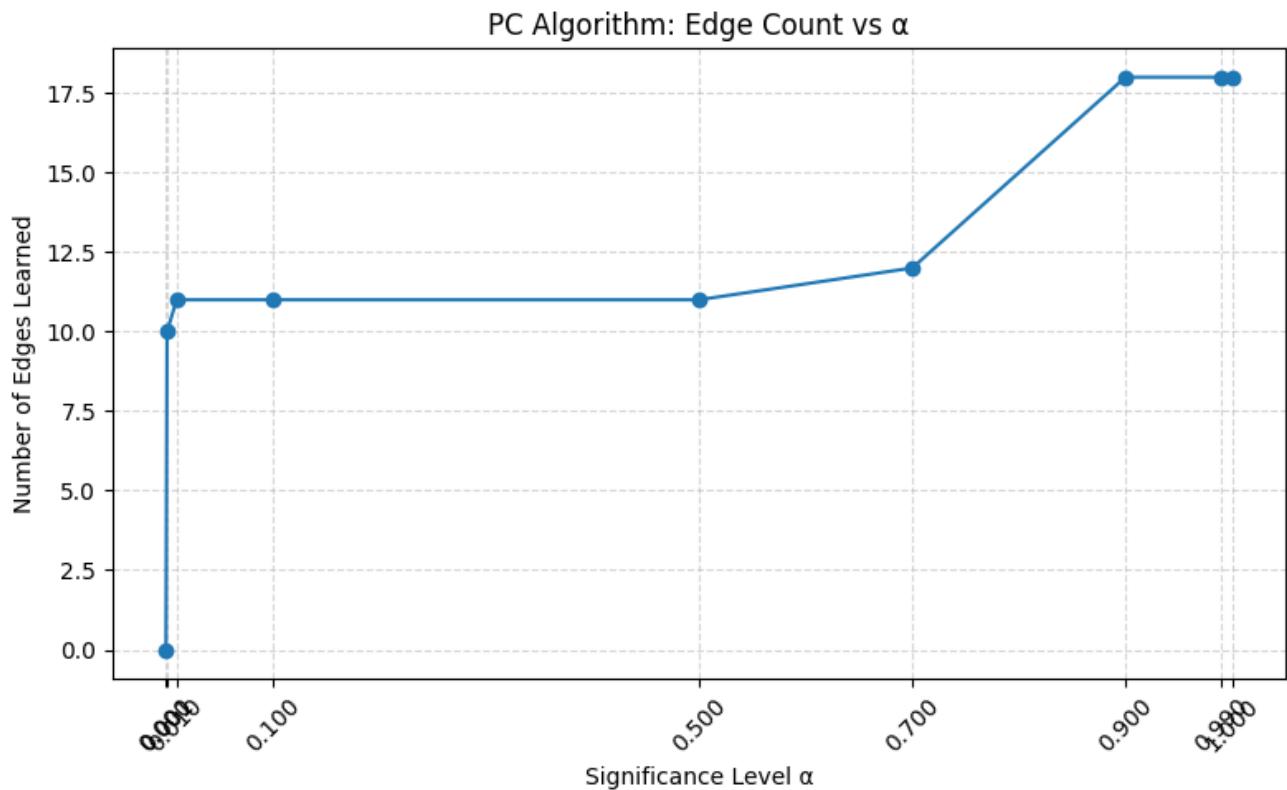
3. **No background constraints.**

Without expert-driven arc-direction constraints (e.g. "Vaccination cannot be caused by Disease2"), the learner is free to reverse that link.

Take-away

Hill-climbing recovers the **skeleton** and the **unshielded colliders** (v-structures) correctly, but **cannot uniquely orient edges** that are neither compelled by data nor by prior constraints. In the two-diseases example, the Vaccination–Disease2 link lies in this ambiguous region, so the learned DAG flips it.

3)



The PC algorithm starts with a fully connected graph and removes edges based on conditional independence tests. The significance level α controls how easily the algorithm removes edges. When α is very low (like 0 or 0.001), it's very strict—almost all edges are removed because it only keeps connections with very strong evidence. That's why you saw nearly zero edges in those cases.

As α increases to moderate levels (around 0.01 to 0.7), the algorithm becomes more permissive, retaining more meaningful edges. This is the range where you typically see the most reliable structure—around 10 to 12 edges in your case.

At high α values (0.9 and above), the algorithm hardly removes any edges, so the graph becomes almost fully connected. At $\alpha = 1$, all tests pass, so no edges are deleted at all, resulting in a complete graph with 90 directed edges.

This pattern shows that small α values can lead to underfitting (too few edges), while large α values can cause overfitting (too many edges). The stable middle range is where the algorithm likely captures the true structure best.

Exercise 3.4.

The following pages collect the answers of two versions of ChatGPT when asked what the main differences are between the K2 and PC algorithms. Indicate the main successes and the main errors of the answers.

ChatGPT 3.5 (29-3-2024)

You

What are the main differences between the K2 algorithm and the PC?

ChatGPT

The K2 algorithm and the PC algorithm are two widely used methods in the context of machine learning and inference of Bayesian network structures. Here are the main differences between them:

1. Methodology :

- *The K2 algorithm is a comprehensive search algorithm that explores all possible Bayesian network structures in hierarchical order.*
- *The PC (Constraint-based) algorithm is based on the search for conditional independences in the data to eliminate connections that are not possible given certain constraints.*

2. Computational Complexity :

- *K2 is usually more computationally expensive, as it explores all possible structures in a systematic way.*
- *PC tends to be more efficient in terms of computation time as it relies on detecting conditional independences in the data to constrain possible structures.*

3. Use of Restrictions :

- *K2 does not rely on explicit constraints, but constructs the Bayesian network according to a specific order of variables.*
- *PC uses constraints based on conditional independences found in the data to eliminate arcs that are not possible given these constraints.*

4. Results :

- *K2 can find an optimal Bayesian network within the defined search space, but this depends on the order of the variables and may not be optimal in all cases.*
- *PC does not guarantee an optimal Bayesian network but can provide a more accurate structure by relying on conditional independence relationships in the data.*

5. Flexibility and Fit :

- *K2 is less flexible in terms of adapting to different data types and structures.*
- *PC is more flexible as it is based on the independence relationships observed in the data.*

In summary, while the K2 algorithm is more comprehensive but less efficient, the PC algorithm is faster and more flexible, although it does not guarantee the overall optimization of the Bayesian network. The choice between these algorithms depends on the specific characteristics of the data and the goals of the modeling.



You

What other differences are there?

ChatGPT

In addition to the differences mentioned above, there are other aspects that distinguish the K2 algorithm from the PC algorithm in the context of Bayesian network learning:

6. Data Domain Dependency :

- K2 generally does not need prior information about the dependencies between variables in the data domain. It simply follows a systematic process to find the best Bayesian network structure.
- PC requires conditional independence analysis to be performed on the data to identify potential relationships between variables. This implies that the user needs to have some knowledge of the domain and the expected relationships between the variables.

7. Ability to Handle Incomplete or Noisy Data :

- K2 can struggle to handle incomplete or noisy data, as it comprehensively explores all possible structures and can be sensitive to data quality.
- PC can be more robust against incomplete or noisy data because it relies on detecting conditional independences in the available data.

8. Scalability :

- K2 can become computationally expensive as the number of variables in the dataset increases, as you must explore all possible network structures.
- PC can be more scalable, especially on large datasets, since its execution time is not so much dependent on the number of variables, but on the complexity of conditional independence relationships in the data.

9. Sensitivity to Parameter Choice :

- K2 generally does not require parameter tuning and follows a deterministic process to find the optimal network structure.
- PC can be sensitive to the choice of parameters, such as the level of statistical significance used to determine conditional independences in the data.

ChatGPT Plus

(3-1-2025)

Question:

What are the main differences between the K2 algorithm and the PC?

ChatGPT Plus

K2 and PC algorithms are methods used in Bayesian network learning, but they have fundamentally different approaches and characteristics. The main differences are:

1. General Approach:

- **K2** is based on direct search and scoring. It builds the Bayesian network structure in a greedy way, iteratively selecting the best parents for each node based on a scoring function. It assumes that nodes are preordained, meaning the user must provide a rough causal order of the variables.
- **PC (Peter-Clark)** is a constraint-based algorithm. It builds the network structure using conditional independence tests to infer relationships between nodes. It does not

require a prior causal order, discovering the structure directly from data.

2. **Independence Testing:**

- **K2** does not use conditional independence tests. It relies solely on scoring metrics (such as BIC or AIC) to determine the structure.
- **PC** uses conditional independence tests to identify relationships between nodes and determine connections. It requires a sufficiently large dataset for these tests to be reliable.

3. **Computational Complexity:**

- **K2** is generally more computationally efficient because it limits the parent search to a predefined set and uses a local search. However, its efficiency depends on the maximum number of parents allowed for each node.
- **PC** is more computationally expensive due to the need to perform multiple conditional independence tests across all combinations of conditional variables. Its complexity increases exponentially with the number of variables and the depth of conditional tests.

4. **Data Robustness:**

- **K2** works well with complete data and allows incorporation of prior knowledge via scoring metrics. However, it is sensitive to the predefined order of variables—performance may degrade if the order doesn't reflect true causal relationships.
- **PC** is more robust in detecting true independences and dependencies, but its performance can degrade with noisy data or small sample sizes where independence tests become unreliable.

5. **Resulting Structure:**

- **K2** generates a fully oriented Bayesian network (DAG - Directed Acyclic Graph).
- **PC** initially produces a partially oriented structure (CPDAG - Completed Partially Directed Acyclic Graph), which can be further refined into a DAG.

6. **Applications:**

- **K2** is useful when there is prior knowledge about the causal order of variables and the goal is efficient model scoring.
- **PC** is preferable when there is no prior knowledge and the goal is to discover causal structure directly from data via statistical dependencies.

Solution 3.4.

1. ChatGPT (standard) answer

Successes

- Correctly identifies the high-level distinction
 - K2 as a *score-based*, greedy search that requires a user-supplied variable order.
 - PC as a *constraint-based* algorithm that uses conditional-independence tests to prune edges.
- Notes dependency on variable ordering
 - K2's output quality depends critically on the initial node ordering.
- Mentions optimality vs. guarantees
 - K2 can find a (local) optimum under its scoring metric and order restrictions.
 - PC does *not* guarantee finding a globally optimal structure under a score.
- Touches on complexity trade-offs
 - K2 can be expensive if the allowed parent set is large.
 - PC's runtime is broadly governed by the number and size of conditional-independence tests.

Errors

1. Overstates K2's search scope

"explores *all possible* Bayesian network structures in hierarchical order."

- In reality, K2 performs a *greedy* (not exhaustive) search: it never revisits earlier parent-sets once it moves on.

2. Mischaracterizes PC's efficiency

"PC tends to be more efficient... its execution time is not so much dependent on the number of variables."

- PC's worst-case complexity is *exponential* in the number of variables and in the size of conditioning sets.

3. Ambiguous claims about flexibility

"PC is more flexible as it is based on the independence relationships observed in the data."

- Both approaches require quantitative data; "flexibility" isn't a standard comparative metric here.

4. Robustness to noise/incompleteness

"PC can be more robust... because it relies on detecting conditional independences."

- Conditional-independence tests are *highly* sensitive to sample size and noise; PC can suffer badly with limited data.

2. ChatGPT Plus answer

Successes

- Clearly states the core methodological divide
 - K2: greedy, score-based search with a preordained order.
 - PC: constraint-based via CI tests, no order required.
- Mentions CPDAG vs. DAG output
 - PC initially outputs a partially oriented graph (CPDAG), which must be completed or refined.
- Acknowledges sample-size dependency
 - PC needs "large enough dataset for independence tests to be reliable."

Errors

1. Self-contradictory on independence tests

- First: "PC ... does *not* use conditional independence tests."
- Immediately after: "It uses conditional independence tests..."
- (Clearly a copy-paste or editing glitch.)

2. Incorrect claim about K2's parameter-tuning

"K2 generally does not require parameter tuning"

- In practice you must choose a scoring metric (BIC/AIC) and a max-parents bound, which are hyperparameters.

3. Complexity descriptions muddled

- Under "PC": "depends on the maximum number of parents allowed for each node" — that bound is usually a *K2* parameter, not PC's.
- PC's complexity comes from the size and number of conditioning-sets, not a predefined parent limit.

4. Formatting/content omissions

- Several bullets under K2 and "Data Robustness" are blank or incomplete, suggesting missing content.

Bottom-Line Takeaways

- What both got *right*
 - K2 ≈ greedy, score-based with required node order; PC ≈ constraint-based via CI tests without order.
- Common misconceptions to avoid
 - K2 is *not* exhaustive—its greedy nature is exactly what limits its search.
 - PC's runtime can explode with many variables and large conditioning-sets.
 - Neither method is intrinsically "robust" to noise; both suffer under small or noisy samples, albeit in different ways.

References

- [1] G. F. Cooper. *NESTOR: A Computer-Based Medical Diagnostic Aid that Integrates Causal and Probabilistic Knowledge*. Ph.D. Thesis, Dept. Computer Science, Stanford University, STAN-CS-84-1031, 1984.
- [2] F. J. Díez. *Introduction to probabilistic graphical models*. UNED, Madrid, 2007.
- [3] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. **50**:157–224, 1988.