Bachelor's Project

Exploring Prosperity through Bayesian Networks: A Data-Driven Analysis of the Legatum Prosperity Index

Author:
María Cuder Vera

Student Number:
S4730437

1st Supervisor:
Prof. M. A. Grzegorczyk

2nd Supervisor:
dr. W. P. Krijnen

Bachelor's Project
To fulfill the requirements for the degree
of the Bachelor of Science in Mathematics
at the University of Groningen

July 27, 2025

Abstract

This thesis examines how Gaussian Bayesian Networks (GBNs), a type of probabilistic graphical model that assumes continuous variables have a Gaussian distribution, can model the relationships among the 14 continuous pillars of the 2023 Legatum Prosperity Index. We use the `bnlearn` R package and apply score-based hill-climbing, a search algorithm for structure learning, using AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) scoring methods. For each, we test several restart settings (1, 10, 100, 1000) to assess how stable and sensitive the model structures are.

Building on this methodology, we evaluate the models in two ways: structurally, by looking at edge counts, v-structures, Markov blankets, Structural Hamming Distance (SHD), and the Jaccard index; and predictively, using log-likelihood and 5-fold cross-validation. The AIC-G models produce denser networks and slightly better predictions, while the BIC-G models are sparser and more stable. Results from cross-validation show only small differences in performance, suggesting that the main structural findings are reliable, even when model complexity changes.

Acknowledgements

I would like to begin by expressing my gratitude to my supervisor Prof. M. A. Grzegorczyk, for his steady support and encouragement throughout this journey. His expertise and patience were essential in bringing this thesis to life.

I would also like to thank my mother and my best friend Almu. They have supported me through all the doubts and challenges I have faced during this degree. I truly believe I could not have finished this degree without you by my side.

To Inés, María, Edu, and Ivet: as I have told you before, I am here entirely because of you. Thank you for your endless patience, kindness, and for always believing in me, even when I doubted myself. Your friendship and willingness to help me through so many courses mean the world to me. I am deeply grateful for your presence in my life.

And finally, to all the friends I have made during this degree, from Llopis and Lucía, to those who joined along the way like Paula, Nuria, Pepe, and Chey. I wish I could name every one of you. Each of you has made these years brighter and filled them with laughter, support, and unforgettable memories. I am truly grateful to everyone who has crossed my path over the last four years. Even though I cannot mention every name, please know that your friendship has meant so much to me.

Finishing this bachelor thesis means more to me than it may seem. So I want to end by thanking the slightly younger version of myself, the one who didn't give up. These four years have been intense, but in the end, it's all been worth it.

Contents

1 Introduction

The 2023 Legatum Prosperity Index reports that global prosperity has leveled off for a third consecutive year, largely due to institutional and democratic regression. Despite this stagnation, access to housing, education, and healthcare has improved in many parts of the world [1]. However, the poorest countries continue to fall further behind. This persistence of stagnation and the widening gap emphasize the need to examine the dimensions of prosperity as interconnected and mutually dependent.

To understand this period of stagnation, it is important to view prosperity as a complex and multifaceted phenomenon shaped by institutional, social, and economic factors. Prosperity extends beyond material wealth to include education, effective governance, individual freedoms, and other elements that support well-being. The Legatum Institute describes true prosperity as:

> *"True prosperity is when all people have the opportunity to thrive by fulfilling their unique potential and playing their part in strengthening their communities and nations."* (p. 9) [1]

To measure prosperity, the Legatum Institute publishes the Global Prosperity Index annually, compiling national data across 12 pillars. The 2023 edition covers 167 countries, providing a comprehensive global view. These pillars are: safety and security, personal freedom, living conditions, health, education, governance, social capital, investment environment, enterprise conditions, market access and infrastructure, economic quality, and natural environment.

Each pillar consists of several elements, and each element is measured using specific, carefully selected indicators. This system enables each country to receive a grade for every pillar. The Legatum Institute sources indicators from reputable international datasets and assigns weights based on each indicator's relative importance. For example, the Safety and Security pillar comprises five weighted elements: War and Civil Conflict (20%), Terrorism (15%), Politically Related Terror and Violence (30%), Violent Crime (25%), and Property Crime (10%). The following example from the War and Civil Conflict element illustrates the types of indicators used to evaluate the Safety and Security pillar:

- War and Civil Conflict includes indicators such as two-sided conflict deaths (UCDP), civil and ethnic war occurrences (CSP), conflict-driven internal displacement (IDMC), and refugee counts by origin (UNHCR).

After scoring each pillar, the Index averages all 12 to compute a country's total prosperity, ensuring that every facet counts equally; no economic, social, or institutional dimension dominates. The Index benchmarks national performance, enables analysis, tracks change, highlights barriers, and, most relevant for this thesis, helps set national development priorities.

Managing numerous variables requires mathematical modeling, but interdependencies among indicators, such as education, income, safety, and health, complicate this task, as these factors influence each other in both subtle and obvious ways. Capturing these relationships requires a method that can represent both direct and indirect influences. Bayesian Networks (BNs) provide such a framework by using directed acyclic graphs (DAGs) to encode conditional dependencies, illustrating how a change in one factor can affect others. With Legatum Index data, BNs can help policymakers identify where interventions yield the greatest benefits. For example, if funding is limited but needs to improve both safety and economic quality, a BN might reveal that investing in education has the largest combined effect.

Often, we do not know all the relationships between variables, and manually building the network can be subjective and impractical. The challenge intensifies as the number of variables increases, causing the number of possible network structures to rise superexponentially. Structure learning in Bayesian Networks addresses this by using data-driven algorithms that employ scoring functions to balance model fit and complexity, helping to avoid overfitting to noise. Despite this, finding the optimal network remains computationally challenging, so most algorithms use greedy strategies to build the network step by step.

In this study, I use hill-climbing structure learning with two complementary scoring functions: the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). These criteria help balance model fit with complexity. AIC applies a lighter penalty for additional parameters, often resulting in richer and denser networks, while BIC uses a stronger penalty, favoring simpler, sparser models. By comparing results from both criteria, I can identify which dependencies between variables are robust; persisting regardless of the penalty applied, and which may be sensitive to the choice of scoring function.

Since hill-climbing is a greedy algorithm and can get stuck in local optima, I perform multiple random restarts (1, 10, 100, and 1000) for each scoring function. Running the search from different starting points enables a broader exploration of possible network structures. If certain dependencies consistently appear across both scoring methods and many restarts, I interpret them as stable features of the underlying data, rather than artifacts of the algorithm or overfitting.

This thesis investigates structure learning for global prosperity data, aiming to determine whether interpretable network structures emerge from the Legatum Index. It also examines how modeling choices, such as scoring functions and restart counts, influence the results. Specifically, we ask:

- How do AIC and BIC scoring criteria affect network complexity (number of edges, v-structures, and average Markov blanket size)?
- How does the number of random restarts (1, 10, 100, 1000) impact the stability of the learned structures, as measured by Structural Hamming Distance and Jaccard similarity?
- Which pillar-to-pillar dependencies emerge consistently across all scoring and restart settings, and which connections vary, highlighting areas of uncertainty?
- How do in-sample penalized-likelihood scores compare to 5-fold cross-validated log-likelihood, and do these performance metrics align with structural stability?
- Which combination of scoring function and restart count best balances explanatory richness (complexity) and parsimony (interpretability) for the Prosperity Index data?

To explore these questions, I use score-based structure learning with the hill-climbing algorithm, implemented via the `bnlearn` package in R. This approach is described in *Learning Bayesian Networks with the bnlearn R Package* [2]. I compare networks learned with AIC and BIC, study the effect of multiple random restarts, and evaluate each network by examining its structure and predictive performance using k-fold cross-validation.

1.1 Literature Review

Bayesian Networks (BNs) have become increasingly relevant in social science because they can represent complex, multidirectional relationships between variables. Unlike traditional regression methods, which often assume a single direction of causality, BNs provide a more flexible approach that helps capture the interdependencies commonly present in social data.

Alvarez-Galvez [3] demonstrate this with a case study using additive Bayesian networks to examine the relationship between socioeconomic status (SES) and self-rated health (SRH) in various European welfare systems. Their results suggest that traditional one-way approaches may miss important effects that depend on context or are reciprocal. By contrast, the BN framework can reveal hidden patterns and allows for more flexible associations, leading to a deeper understanding of how variables depend on each other. This thesis builds on that work by applying Bayesian networks to global prosperity data, specifically the Legatum Index, and discussing key methodological issues in learning BN structures.

A key issue in this process is choosing the right model, and scoring functions help balance model fit with complexity. The two most commonly used criteria are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), both of which are widely used in BN research and have strong theoretical foundations. As explained by Kuha [4], AIC prefers models with good predictive ability and lightly penalizes complexity, while BIC imposes a stronger penalty and aims to find the most likely true model from a Bayesian perspective. Using both criteria provides a broader understanding: when they agree, confidence in the chosen model increases; when they differ, it highlights uncertainty or potential instability in the model.

In this thesis, both AIC and BIC are used with multiple random restarts to test the stability and reliability of the learned models. This approach puts the theory discussed by Kuha [4] into practice, offering real-world insight into how different scoring methods affect the modeling of complex social data.

2 Theory

This section introduces the theoretical foundation for modeling complex variable relationships using Bayesian networks, focusing on their structure, learning algorithms, and model evaluation techniques.

2.1 Bayesian Networks

The theory in this section draws from *Bayesian Networks: With Examples in R* by Scutari and Denis [2], which introduces Bayesian networks (BNs) and covers structure learning, parameter estimation, and implementation in R.

A Bayesian network provides a concise way to represent the joint probability distribution of several random variables. It consists of two main elements: a probabilistic model and a graphical model. The graph structure shows both dependencies and independencies among variables. Formally, a Bayesian network consists of:

- A set of random variables $X = \{X_1, X_2, \ldots, X_p\}$ representing the quantities of interest. The joint probability distribution over all variables is called the *global distribution*, while the probability distribution for each variable X_i is referred to as its *local distribution*.
- A directed acyclic graph (DAG), $G = (V, A)$, where each node $v \in V$ represents a variable $X_i \in X$. Directed edges $a \in A$ indicate probabilistic dependencies between variables. When two nodes are not connected by an edge, the corresponding variables are statistically or conditionally independent, given some other variables.

In this study, the 12 variables from the Legatum Index are modeled as nodes in a directed graph illustrating their dependencies. Some variables directly affect others, while some are connected only indirectly through intermediate nodes. A direct dependency is shown as $A \to B$, meaning B depends on A, with A as the parent and B as the child. Indirect dependencies are represented by paths, such as $A \to E \to R$, where R is indirectly influenced by A via E. For proper probabilistic interpretation and factorization, the graph must be acyclic, it cannot contain directed cycles. This requirement ensures the network forms a valid *Directed Acyclic Graph* (DAG). Note that arcs indicate statistical dependencies, but do not always imply causality unless further evidence is provided.

One main advantage of Bayesian networks is that they simplify the modeling of joint probability distributions. Without the network structure, estimating the joint distribution for p variables would require an exponential number of parameters. However, the DAG encodes conditional independencies, enabling the global distribution to be broken into simpler components. Each variable depends only on its parents in the graph, resulting in the following factorization:

$$\Pr(\mathbf{X}) = \prod_{i=1}^{p} \Pr\left(X_i \mid \Pi_{X_i}\right) \tag{1}$$

Here, Π_{X_i} denotes the set of parents of X_i in the DAG. The acyclic nature of the graph ensures this factorization is valid. Each local distribution is thus simpler and easier to estimate, making this approach more efficient than the general chain rule, which involves more complex conditioning. When a variable has multiple parents, its conditional distribution depends on all parents collectively. In convergent (collider) structures at X_i, we cannot assume the parents are independent and must account for their joint distribution.

The factorized form in (1) provides a structured model of the global distribution that requires fewer parameters. As a result, Bayesian networks are well suited for high-dimensional problems, such as modeling prosperity indicators. This structure also enhances interpretability and computational efficiency by making relationships between variables clear.

Having established the link between the probabilistic and graphical components, we can now provide a formal definition:

Definition 2.1 (Bayesian Network)**.** Given a joint probability distribution P over a set of variables $\mathbf{X}$, and a DAG $G = (\mathbf{X}, A)$, we say that $\mathcal{B} = (G, \mathbf{X})$ is a *Bayesian Network* if and only if G is a minimal I-map of P. That is, G encodes exactly the set of conditional independencies in P, and no arc in G can be removed without violating that property.

This definition highlights the key idea behind Bayesian networks: using a graph to encode conditional independencies so the full joint distribution can be constructed from local conditional distributions. The resulting model is both compact and expressive, making it particularly useful for studying complex systems such as global prosperity indicators.

Since the dataset in this study contains continuous variables, we use *Gaussian Bayesian Networks* (GBNs). In a GBN, each variable is assumed to be normally distributed, and conditional distributions are modeled using linear Gaussian regression. Given its parents Π_{X_i}, each variable X_i follows:

$$X_i = \beta_0 + \sum_{X_j \in \Pi_{X_i}} \beta_j X_j + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

This assumption allows for parameter estimation using standard linear regression. The resulting joint distribution defined by the DAG is multivariate normal, and the factorization property remains valid. GBNs are particularly useful for socioeconomic data, where variables are continuous and often exhibit linear relationships.

An important theoretical concept in Bayesian networks is the idea of equivalence classes. Often, multiple DAGs encode the same conditional independencies, these are called *Markov equivalent*. While their edge directions may differ, the available data alone cannot distinguish between them without additional information, such as time order or interventions. This raises the question: if many DAGs are equivalent, which one should we choose?

To address this issue, we use a *Completed Partially Directed Acyclic Graph* (CPDAG) to represent equivalence classes of DAGs. A CPDAG contains:

- Directed edges for relationships that are identifiably directional across all equivalent DAGs, and
- Undirected edges for relationships whose directionality cannot be determined from the data alone.

A CPDAG summarizes all DAGs within an equivalence class. For example, if A and B are connected by an undirected edge $A - B$ in a CPDAG, then both $A \to B$ and $B \to A$ are consistent with the data, with no evidence favoring one direction. If a directed edge $A \to B$ appears, every DAG in the class contains that arc.

CPDAGs are important in structure learning because most algorithms, especially score-based or constraint-based ones, can only find the network up to its equivalence class. Therefore, when evaluating or visualizing Bayesian networks, it helps to look at both the original DAG and its CPDAG to see which dependencies are clear and which are ambiguous.

2.2 Structure Learning

Many algorithms have been developed to learn the structure of Bayesian networks, drawing on concepts from probability theory, information theory, and optimization. Despite their varied theoretical foundations, these methods are typically classified into three categories: *constraint-based*, *score-based*, and *hybrid* approaches.

Constraint-based algorithms, such as the PC (Peter-Clark) algorithm, use statistical tests of conditional independence to determine the network structure. By identifying which variables are conditionally independent, these algorithms construct a graph that results in a partially directed acyclic graph (PDAG), representing the set of networks consistent with the observed data.

Score-based algorithms, in contrast, search through possible network structures using a scoring function that evaluates how well each structure fits the data. Methods like hill-climbing and tabu search are commonly used, often relying on criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC).

Hybrid approaches combine the strengths of both methods: they first use independence tests to narrow the search space and then apply a scoring function to select the optimal structure from the remaining possibilities.

In this thesis, we focus exclusively on *score-based structure learning*, a choice that offers several advantages. First, score-based methods provide a clear, quantitative means of balancing model complexity with data fit. Second, they enable direct comparison of models learned with different scoring rules, such as AIC and BIC. Third, they are well-suited for investigating how algorithm parameters, like the number of restarts, affect convergence and model stability, which are key interests in this study.

The decision to use score-based learning is also supported by Scutari et al. [5], who found that score-based algorithms perform well in terms of accuracy and robustness across numerous benchmark datasets. Specifically, greedy search methods using AIC or BIC demonstrated strong computational and statistical performance. These findings further justify the use of score-based learning for analyzing complex systems such as the Legatum Prosperity Index.

While effective, score-based structure learning depends on several assumptions to ensure that the resulting model is both statistically valid and interpretable:

- Each node in the network stands for a unique, observable random variable. Redundant, derived, or deterministic variables should not be included.
- The joint distribution of the variables is strictly positive, meaning all possible combinations of variable values have non-zero probability. This allows the model structure to be identified.
- Observations are assumed to be independent and identically distributed (i.i.d.). When dependencies exist over time or space, extensions such as dynamic Bayesian networks are more appropriate.

- Dependencies among variables are assumed to be described by conditional independence statements, which are shown in the graph structure.

2.2.1 Hill-Climbing and Model Selection

In this study, we use the *hill-climbing* algorithm for score-based structure learning. Hill-climbing is a procedure that incrementally modifies a Directed Acyclic Graph (DAG) to enhance the model's fit to the data. At each step, the algorithm evaluates possible additions, removals, or reversals of a single edge, while maintaining acyclicity at all times. The aim is to find the DAG G that maximizes a chosen scoring function $f(G : D)$ based on the observed data D:

$$G^* = \arg\max_{G \in \mathcal{G}_n} f(G : D), \tag{3}$$

where $\mathcal{G}_n$ denotes the space of DAGs over n variables.

Typically, the algorithm starts with an initial graph, often the empty DAG, and systematically examines all graphs obtainable by a single edge modification, selecting the one with the highest score. This process repeats until no further single change increases the score, indicating a local optimum has been reached. Because hill-climbing only considers immediate improvements, it may miss the global optimum. To address this, we use multiple *random restarts*, launching the algorithm several times from different random starting graphs. This strategy increases the chances of discovering better structures [6].

The main steps of the hill-climbing algorithm are outlined below:

Algorithm 1 Hill-Climbing Algorithm [2]

1. Choose an initial network structure G over $\mathbf{V}$, typically the empty DAG.
2. Compute its score, $Score_G = Score(G)$.
3. Set $maxscore = Score_G$.
4. Repeat while $maxscore$ improves:
   (a) For all possible arc additions, deletions, or reversals that preserve acyclicity:
      (i) Compute the score of the modified network G^*:
      $$Score_{G^*} = Score(G^*)$$
      (ii) If $Score_{G^*} > Score_G$, set $G = G^*$ and $Score_G = Score_{G^*}$.
   (b) Update $maxscore$ to the new value of $Score_G$.
5. Return the final DAG G.

One key advantage of hill-climbing is that it works well with *decomposable scoring functions*. These scoring functions let us calculate the total score by adding up the contributions from each node and the set of its parents:

$$f(G : D) = \sum_{i=1}^{p} f_D(X_i, \mathrm{Pa}_G(X_i)), \tag{4}$$

where $\mathrm{Pa}_G(X_i)$ denotes the set of parents of node X_i in graph G. Since each modification impacts only one node's parents, the score can be updated quickly and efficiently, making the algorithm

practical even for larger graphs.

In this thesis, we use two classic penalized-likelihood criteria: the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both are based on the model's log-likelihood, which quantifies how well the structure explains the observed data. The log-likelihood for a dataset $D = \{x_1, x_2, \ldots, x_n\}$ and a Bayesian network B with parameters $\hat{\theta}$ is defined as:

$$\log \mathcal{L}(\hat{\theta}) = \sum_{t=1}^{n} \log P_B(x_t \mid \hat{\theta}). \tag{5}$$

However, models with more parameters often achieve higher log-likelihoods, potentially leading to overfitting. To counteract this, AIC and BIC incorporate a penalty for model complexity. Their general form is:

$$\text{Score} = \log \mathcal{L}(\hat{\theta}) - \alpha \cdot \text{df}, \tag{6}$$

where df is the number of free parameters in the model, and α sets how strongly complexity is penalized.

The **Akaike Information Criterion (AIC)** adopts $\alpha = 2$, yielding:

$$\text{AIC} = -2 \log \hat{\mathcal{L}} + 2p, \tag{7}$$

where p is the number of parameters. AIC is based on ideas from information theory and tries to choose models that are close to the actual process that generated the data. It is asymptotically efficient, meaning that as we get more data, AIC tends to pick models that predict new data well, even when the true model is complicated or unknown [4].

The **Bayesian Information Criterion (BIC)** imposes a stronger penalty, with $\alpha = \log n$, where n is the sample size:

$$\text{BIC} = -2 \log \hat{\mathcal{L}} + p \log n. \tag{8}$$

BIC is based on Bayesian ideas and approximates a Bayes factor. It is consistent, meaning that as the amount of data increases, BIC will usually select the correct model, provided that the right model is among the candidates. BIC prefers simpler, easier-to-interpret models and is often used when we value simplicity and clear explanations [4].

Choosing between AIC and BIC changes how complex and robust the final network is. AIC tends to include more connections to better fit the data, which can capture subtle patterns but may make the model harder to interpret. BIC usually leads to simpler graphs that might generalize better and show only the most reliable relationships.

By applying both scoring rules (AIC and BIC) with hill-climbing and running the search multiple times from different starting points, we can observe how our assumptions influence the learned structures. Consistent results across settings suggest robust relationships between variables, while discrepancies highlight areas where the learning process is sensitive and may require more careful modeling or expert input.

2.3 Model Evaluation

To evaluate the quality and usefulness of the learned Bayesian networks, we consider both their structural properties and their predictive performance.

2.3.1 Structure Evaluation

Beyond predictive accuracy, this thesis examines whether the learned Bayesian networks are structurally meaningful, stable across modeling choices, and interpretable within a broader context. While log-likelihood metrics assess model fit, they do not reveal if the learned structure; the pattern of dependencies, is reliable. Because structure learning algorithms can return just one of many equivalent graphs, it is essential to evaluate the learned DAGs and their equivalence classes to identify which relationships are robust and which may be artifacts.

As discussed in Section 2.1, different DAGs can encode the same set of conditional independencies, forming an equivalence class represented by a Completed Partially Directed Acyclic Graph (CPDAG). In a CPDAG, edges are directed only if the data clearly support them, often due to v-structures; otherwise, the edges remain undirected. By focusing on CPDAGs, we examine the features that can be reliably identified regardless of the specific edge directions chosen.

To evaluate the structural properties of these networks, we use both global and local perspectives:

- **Global structure:** Skeletons and v-structures. The skeleton of a network, the set of edges without regard for direction, reveals which variables are directly linked. Comparing skeletons from models learned using AIC and BIC helps identify stable dependencies. We also examine v-structures (triplets such as $X_i \rightarrow X_k \leftarrow X_j$), since these are the only edge directions uniquely identifiable from observational data. Consistent v-structures across runs suggest robust conditional dependencies.
- **Local structure:** Markov blankets. For any given node, its Markov blanket consists of its parents, children, and the other parents of its children. This set best predicts the variable. Markov blankets are valuable because they highlight the closest influences and effects. Tracking changes in Markov blankets across different scoring functions and random starts allows us to assess local variation and model stability.

To measure the similarity between the learned networks, we use two widely accepted metrics:

- The Structural Hamming Distance (SHD) counts the number of edge operations; additions, deletions, or reversals, needed to transform one DAG into another. SHD is computed between CPDAGs. For two graphs, G_1 and G_2 defined on the same set of nodes, SHD is given by:

$$\text{SHD}(G_1, G_2) = \#(\text{additions}) + \#(\text{deletions}) + \#(\text{reversals}). \tag{9}$$

  A lower SHD indicates greater structural similarity between the networks.

- The Jaccard Index provides a normalized similarity measure by comparing edge sets. It is defined as:

$$\text{Jaccard}(G_1, G_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|}, \tag{10}$$

where E_1 and E_2 are the edge sets in the two networks. We calculate this index for both directed edges and undirected skeletons to show the difference in stability between edge presence and edge direction. A higher Jaccard index indicates the structures are more consistent.

These structural comparisons address two main questions: How do scoring criteria (AIC versus BIC) and the number of restarts affect the networks, and do relationships among variables remain stable across modeling assumptions? Typically, AIC leads to denser graphs, while BIC produces sparser and more interpretable models. By comparing CPDAGs, Markov blankets, and similarity metrics, we assess whether dependencies are consistent or sensitive to algorithm settings.

Structural evaluation helps validate the networks and identify robust, interpretable features in the Prosperity Index. If edges, v-structures, or Markov blankets persist across models, they likely reflect genuine patterns in the data rather than artifacts of optimization.

2.3.2 Predictive Evaluation

The structure of a Bayesian network illustrates which variables depend on each other, but it is also relevant to check how well it predicts unseen data. In practical applications such as the Legatum Prosperity Index, where models inform policy or strategic choices, strong predictive performance is crucial. Accurate predictions demonstrate that the learned statistical relationships generalize to new observations, not just the training data.

To quantify predictive performance, we assess both in-sample (using data the model was trained on) and out-of-sample (using new, unseen data) results. The in-sample metric is the *log-likelihood*, which measures how likely the observed data are under the fitted model: higher values indicate that the model better explains the data. For a Bayesian network B and dataset $D = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, the log-likelihood is defined as:

$$LL(B \mid D) = \sum_{t=1}^{N} \log P_B(\mathbf{x}^{(t)}), \tag{11}$$

where $P_B(\mathbf{x}^{(t)})$ is the probability of $\mathbf{x}^{(t)}$ under network B. In Gaussian Bayesian networks, this is computed using conditional densities from linear regressions for each node and its parents. Altogether, the network forms a multivariate normal distribution over all variables.

Models with higher log-likelihood values often exhibit greater complexity. To account for this, penalized log-likelihood measures such as AIC and BIC (see Section 2.2) are used. These criteria impose a penalty based on the number of free parameters: AIC typically favors models that fit the data better, while BIC applies a stronger penalty for complexity and tends to select models that generalize more effectively, as previously discussed.

To test the model's ability to predict new data, we use 5-fold cross-validation. This method measures out-of-sample performance by splitting the dataset into five equal parts. Each part is used in turn as a test set while the remainder forms the training set:

1. Partition the dataset D into disjoint folds $D_1, D_2, \ldots, D_5$.
2. For each $i = 1, \ldots, 5$:
    - Train the Bayesian network on $D \setminus D_i$ using the hill-climbing algorithm and a chosen scoring function.

- Estimate the parameters with maximum likelihood for the learned structure.
- Compute the log-likelihood of the held-out data D_i under the fitted model:

$$LL_i = \sum_{\mathbf{x} \in D_i} \log P_B(\mathbf{x}).$$

3. Average the five scores to obtain the *cross-validated log-likelihood (CV-LL)*:

$$\text{CV-LL} = \frac{1}{5} \sum_{i=1}^{5} LL_i.$$

This process provides an unbiased estimate of predictive accuracy on new data. As noted by Marcot and Hanea [7], 5-fold cross-validation offers a good balance between variance and bias for Bayesian networks and is manageable for moderate-sized datasets. Increasing k reduces bias further but also increases computational demands and the variance of the test error. We use $k = 5$ as a reasonable compromise.

Predictive evaluation tests the model's fit and reliability. If a network exhibits strong cross-validated log-likelihood and favorable AIC or BIC values, this suggests that the dependencies it identifies are genuine. For prosperity modeling, this is critical: the model must make accurate predictions for new cases to support analysis or guide policy.

We use log-likelihood, cross-validated log-likelihood (CV-LL), AIC, and BIC to measure predictive performance. By comparing these across models with different scoring functions and random restarts, we see the trade-off between complexity and generalizability. This approach helps pick Bayesian networks that are clear and robust for global prosperity analysis.

3 Methodology

3.1 Data

The dataset for this study is the *2023 Global Prosperity Index*, published on Kaggle by Aaron Norman.[1] It contains data from more than 160 countries and covers different aspects of prosperity, such as governance, education, health, safety, and economic quality.

In this thesis, we used only the 12 main **continuous prosperity pillars**, excluding categorical or country-level details such as region, country name, or regime type. This made it possible to use Gaussian-based methods without complications. The remaining variables were all continuous scores from 0 to 100, each showing an important part of national well-being.

Before modeling, the data went through several preparation steps. First, we removed any country with missing data in one or more of the 12 pillars, leaving 167 complete cases. Then, we standardized all variables to have a mean of zero and a variance of one. This helps with numerical stability in estimating the model. Visual checks, such as histograms and scatterplot matrices, showed that although the variables were not perfectly normal, they were continuous and unimodal enough to use Gaussian methods.

3.2 Modeling Framework

The analysis uses **Gaussian Bayesian Networks** (GBNs), a type of Bayesian network where each variable is modeled by a linear Gaussian regression, given its parent nodes. This approach does not require turning variables into categories and lets us directly model continuous relationships between indicators.

Mathematically, each variable X_i is assumed to follow a normal distribution given its parents Π_{X_i}, meaning:

$$X_i \mid \Pi_{X_i} \sim \mathcal{N}\left(\beta_0 + \sum_j \beta_j X_j, \sigma^2\right).$$

This setup gives a joint multivariate normal distribution for all variables. The parameters; regression coefficients and variances, are estimated using maximum likelihood. The model assumes linear, additive effects, approximate normality, and that the data points are independent and identically distributed (i.i.d.).

3.3 Structure Learning with `bnlearn`

Structure learning was done using the `bnlearn` package in R [2], using its version of the **hill-climbing (HC)** algorithm. HC is a greedy search method that changes the DAG step by step to improve a scoring function.

We used two scoring criteria: the **Akaike Information Criterion (AIC)** and the **Bayesian Information Criterion (BIC)**. AIC rewards better fit but uses a smaller penalty (2 parameters per edge), while BIC uses a larger penalty ($\log(n)$ per edge), making it more conservative. The models learned with these scores are called AIC-G and BIC-G, respectively.

To reduce the risk of getting stuck in local optima during hill-climbing, we ran the algorithm with several **random restarts**. For each scoring method, the learning was done with 1, 10, 100,

[1] https://www.kaggle.com/datasets/aaronnorman/2023-global-prosperity-index-w-region-politics

and 1000 restarts, each using a different random seed. This gave us eight final networks (and their respective 8 CPDAGs), making it possible to see how the choice of score and the number of restarts affect the learned structure.

For each model, we estimated the parameters using maximum likelihood under the Gaussian assumption, resulting in networks that have both structure (DAG) and parameters fully specified.

3.4 Evaluation Strategy

We evaluated the learned networks from both structural and predictive perspectives. Structurally, we measured the number of edges, v-structures (colliders), and average Markov blanket sizes in the learned DAGs to assess model complexity. To quantify similarity across learned graphs, we compared models using the **Structural Hamming Distance (SHD)** and the **Jaccard index**. Since different DAGs can encode the same independencies, we based these similarity metrics on CPDAGs (Completed Partially Directed Acyclic Graphs), which represent equivalence classes.

For predictive evaluation, we computed both the **in-sample log-likelihood** and the **5-fold cross-validated log-likelihood** on the learned DAGs, as these models include the complete set of parameters needed for prediction. Cross-validation provides an estimate of generalization performance and helps determine whether denser structures truly improve predictions on unseen data.

3.5 Software

All analyses were performed in `R` version 4.5.1. The key packages used were:

- `bnlearn` [2]: for Bayesian network structure learning and parameter estimation.
- `caret`: for implementing cross-validation.
- `tidyverse`: for data wrangling and visualization.

4 Results

This section evaluates the quality of the learned models from both structural and predictive perspectives. It explores how different choices in scoring function and search strategy affect the networks' complexity, stability, and ability to generalize beyond the training data.

4.1 Structural Evaluation

We begin by summarizing the main structural features of the learned Bayesian networks using both scoring criteria; AIC-G and BIC-G, across all four levels of random restarts (1, 10, 100, 1000). Our comparison is based on the learned DAGs, which are the fully specified models that incorporate edge directionality. Table 1 presents these features, including:

- Edges: The number of directed arcs in the learned DAG, reflecting model complexity and indicating how many direct dependencies are present.
- V-Structures: $X \to Z \leftarrow Y$, where X and Y are not directly connected. These structures are important for causal inference and represent edge directions that cannot be changed within the equivalence class.
- Markov Blanket (MB): For each node, the Markov blanket includes its parents, its children, and any other parents of its children. It is the smallest set of nodes needed to render a node independent from the rest of the graph. We report the average, minimum, and maximum sizes of Markov blankets.

Table 1: Summary of Structural Features Across Models

Model	Restarts	Edges	V-structures	Avg MB	Min MB	Max MB
AIC-G	1	39	78	8.83	6	11
AIC-G	10	39	78	8.83	6	11
AIC-G	100	41	72	8.83	6	11
AIC-G	1000	38	81	8.67	5	11
BIC-G	1	29	30	6.00	3	10
BIC-G	10	29	30	6.00	3	10
BIC-G	100	34	39	6.17	2	9
BIC-G	1000	34	36	6.00	3	9

Several clear trends emerge from this structural comparison. First, networks learned using AIC-G are consistently denser than those learned with BIC-G, showing more edges, more v-structures, and larger Markov blankets. This reflects the mathematical logic underlying the AIC and BIC penalties: AIC applies a fixed penalty per edge (2 parameters per edge), while BIC's penalty increases with sample size (scaling with $\log n$). As a result, BIC serves as a stronger regularizer and produces simpler graphs.

The difference in the number of v-structures is particularly notable: AIC models identify between 72 and 81 v-structures, while BIC models identify between 30 and 39. This indicates that the two models capture different patterns of conditional dependencies in the data. For example, in `CPDAG_AIC-G_1000`, many v-structures appear at nodes such as *education* and *governance*, which act as points where edges converge. In contrast, `CPDAG_BIC-G_1000` features fewer of these patterns and more chain-like or star-like structures, signifying fewer conditional dependencies.

A concrete example of this difference is seen in the node *governance*. In AIC-G models, it frequently appears as a collider (e.g., *personal freedom* → *governance* ← *education*), suggesting a role as a mediator or integrating factor. This specific v-structure appears in `CPDAG_AIC-G_10` and `DAG_AIC-G_100`. In BIC-G models, however, this pattern is often absent or replaced with simpler chains (e.g., *personal freedom* → *governance* → *education*), possibly reflecting more conservative causal assumptions.

The difference in Markov blanket size further supports this pattern. AIC-G networks have average MB sizes of 8.67–8.83, meaning each node is conditionally dependent on almost all other nodes. In contrast, BIC-G models have an average MB size of about 6.0, indicating greater conditional independence among nodes. This lower connectivity is also evident visually: in `DAG_BIC-G_1`, for instance, nodes such as *natural environment* or *infrastructure* appear on the edge of the network with fewer connecting edges.

Notably, the number of restarts does not significantly affect the total number of edges or the average MB size. However, it does influence the number of v-structures found: more restarts allow the hill-climbing algorithm to explore a greater variety of possible graphs, sometimes revealing better structures with subtle but important differences. For instance, the number of v-structures increases from 72 to 81 between `AIC-G_100` and `AIC-G_1000`, and from 30 to 39 between `BIC-G_10` and `BIC-G_100`. This demonstrates that multiple restarts help avoid getting stuck in local optima.

In summary, this structural overview highlights how the choice of scoring function and restart strategy together shape the final graph. AIC-G favors more expressive and densely connected networks, which may capture more details from the data. In contrast, BIC-G leads to simpler, potentially more robust structures. To further assess the consistency and differences between these structures, we next compare pairs of models using standard similarity measures.

4.1.1 Structural Hamming Distance and Jaccard Index

While the table above summarizes each model, a closer structural comparison is needed to assess stability and convergence. To do this, we examine how much the learned networks change as we vary the number of restarts and scoring functions, using two measures: Structural Hamming Distance (SHD) and the Jaccard index. SHD counts the edge changes (additions, deletions, reversals) required to turn one CPDAG into another, while the Jaccard index measures the proportion of shared to unique edges between two networks, considering both arcs and skeletons.

Table 2 presents the SHD values between pairs of CPDAGs. To understand model stability, we compare networks learned with the same scoring method but different restarts, as well as those learned with different scoring methods. For instance, when using the same scoring method, networks learned with 1 and 10 restarts (e.g., `aic-g_1` vs. `aic-g_10`) usually have very low SHD values, sometimes even 0. Adding more restarts can lead to modest improvements, for example, `aic-g_1` and `aic-g_100` differ by an SHD of 14. However, SHD increases much more when comparing networks with different scoring methods or very different numbers of restarts. For example, `aic-g_1` and `bic-g_1000` have an SHD of 28, indicating that their underlying model assumptions are quite different.

Table 2: Selected Structural Hamming Distances Between Learned Networks

Comparison	SHD
`aic-g_1` vs `aic-g_10`	0
`aic-g_1` vs `aic-g_100`	14
`aic-g_1` vs `bic-g_1`	11
`aic-g_100` vs `bic-g_100`	29
`bic-g_100` vs `bic-g_1000`	0

This pattern demonstrates that increasing the number of restarts can significantly affect the learned structure when the search space is large and the score function is complex, as with AIC. In contrast, BIC's stronger regularization often yields stable results with fewer restarts. The impact of these differences is further illustrated by examining specific structural changes: for example, the DAGs for `AIC-G_100` and `AIC-G_1000` (see Appendix) differ in important edges involving *social capital* and *economic quality*. New v-structures appear only when more restarts are used, indicating that the search explores the score space more deeply.

To complement the SHD analysis, we also examine the **Jaccard index** for both directed arcs and undirected skeletons. Before discussing the results, let us clarify their meaning: DAG skeletons indicate which dependencies exist, without considering direction, while directed arcs represent assumptions about how information or causality flows through the system.

The *directed Jaccard index* between two graphs, G_1 and G_2, is defined as:

$$\text{Jaccard}_{\text{dir}}(G_1, G_2) = \frac{|\text{Arcs}_{G_1} \cap \text{Arcs}_{G_2}|}{|\text{Arcs}_{G_1} \cup \text{Arcs}_{G_2}|},$$

while the *skeleton Jaccard index* uses the undirected versions of the same graphs:

$$\text{Jaccard}_{\text{skel}}(G_1, G_2) = \frac{|\text{Edges}^{\text{und}}_{G_1} \cap \text{Edges}^{\text{und}}_{G_2}|}{|\text{Edges}^{\text{und}}_{G_1} \cup \text{Edges}^{\text{und}}_{G_2}|}.$$

Here, $\text{Edges}^{\text{und}}$ means the edges are treated as *undirected*, even if the original graphs are directed.

Tables 3 and 4 display these similarity scores for all pairs of learned models, from which several important insights emerge.

Table 3: Jaccard Similarity of Directed Arcs Between Networks

	aic-g_1	aic-g_10	aic-g_100	aic-g_1000	bic-g_1	bic-g_10	bic-g_100	bic-g_1000
aic-g_1	1.00	1.00	0.51	0.38	0.68	0.68	0.37	0.37
aic-g_10	1.00	1.00	0.51	0.38	0.68	0.68	0.37	0.37
aic-g_100	0.51	0.51	1.00	0.28	0.28	0.28	0.24	0.24
aic-g_1000	0.38	0.38	0.28	1.00	0.37	0.37	0.51	0.48
bic-g_1	0.68	0.68	0.28	0.37	1.00	1.00	0.43	0.39
bic-g_10	0.68	0.68	0.28	0.37	1.00	1.00	0.43	0.39
bic-g_100	0.37	0.37	0.24	0.51	0.43	0.43	1.00	0.86
bic-g_1000	0.37	0.37	0.24	0.48	0.39	0.39	0.86	1.00

Table 4: Jaccard Similarity of Skeletons Between Networks

	aic-g_1	aic-g_10	aic-g_100	aic-g_1000	bic-g_1	bic-g_10	bic-g_100	bic-g_1000
aic-g_1	1.00	1.00	0.90	0.78	0.73	0.73	0.62	0.62
aic-g_10	1.00	1.00	0.90	0.78	0.73	0.73	0.62	0.62
aic-g_100	0.90	0.90	1.00	0.74	0.64	0.64	0.66	0.66
aic-g_1000	0.78	0.78	0.74	1.00	0.66	0.66	0.72	0.72
bic-g_1	0.73	0.73	0.64	0.66	1.00	1.00	0.77	0.77
bic-g_10	0.73	0.73	0.64	0.66	1.00	1.00	0.77	0.77
bic-g_100	0.62	0.62	0.66	0.72	0.77	0.77	1.00	0.93
bic-g_1000	0.62	0.62	0.66	0.72	0.77	0.77	0.93	1.00

These results reveal several important insights:

- **Within-scoring consistency:** AIC-G models are very stable across a small number of restarts (Jaccard = 1.00 between `aic-g_1` and `aic-g_10`), but this similarity drops with many restarts (e.g., 0.38 between `aic-g_1000` and `aic-g_1`), indicating that deeper searches uncover more network structures. In contrast, BIC-G consistently shows high similarity across restarts (such as 0.86 between `bic-g_100` and `bic-g_1000`), suggesting that BIC's stronger regularization makes further searching less necessary.
- **Between-scoring divergence:** When comparing models with different scoring functions, the overlap in structure decreases substantially. For instance, the directed-arc Jaccard index between `aic-g_100` and `bic-g_100` is only 0.24. This demonstrates that AIC and BIC can lead to very different network structures, not just minor differences.
- **Skeleton robustness:** The undirected skeletons are much more similar than the directed graphs. For example, the skeleton Jaccard index between `aic-g_1000` and `bic-g_1000` is 0.72, even though their directed overlap is only 0.48. This suggests that the fundamental conditional independence structure remains robust across scoring choices and restart counts, even if edge directions vary. For tasks focused on capturing core dependencies, such as variable selection or exploratory analysis, relying on the skeleton can provide stable insights despite variability in scoring or search strategy. This supports the idea that the CPDAG, representing equivalence classes, provides a more reliable summary than any single directed network.

These patterns are also evident in the appendix figures. For example, `CPDAG_AIC-G_1000` and `CPDAG_BIC-G_1000` share many skeleton components, such as close groupings around *health*, *education*, and *governance*, but differ in edge directionality. This highlights how model choice affects the interpretability of causal or directional hypotheses. For instance, AIC-G frequently directs edges into connected hubs like *governance*, while BIC-G produces clearer, simpler patterns with fewer v-structures. Thus, researchers analyzing edge directions should consider the influence of the score function on inferred causality.

The Jaccard similarities indicate that model complexity and the choice of scoring function are key factors influencing how much learned Bayesian network structures vary. These results also underscore the importance of caution when interpreting edge directions, especially when data is insufficient to clearly distinguish between different network structures.

Overall, these findings show that differences in structure, such as edge density, number of v-structures, and Markov blanket size, reflect the balance between model complexity and regularization imposed by the scoring functions. AIC-G tends to produce denser graphs with more conditional dependencies, while BIC-G yields sparser and more conservative models. These structural choices affect how information moves through the network, impacting both interpretability and the risk of

overfitting. However, denser graphs are not always better, their added complexity must be justified by improved predictive accuracy. In the next section, we examine whether these structural patterns lead to real improvements in generalization by comparing out-of-sample log-likelihood for all models.

4.2 Predictive Evaluation

While the structure shows which variables are related, it is also important to measure how well each model predicts numerical values. To do this, we look at both **in-sample fit** and **out-of-sample predictive performance**.

We measure in-sample fit using the **log-likelihood** of the data for each learned Bayesian network:

$$LL(B \mid D) = \sum_{t=1}^{N} \log P_B(\mathbf{x}^{(t)}),$$

where B is the learned network, and $D = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ is the dataset. Since our variables are continuous and our networks are Gaussian Bayesian Networks (GBNs), each part of the network is a linear Gaussian regression. The total likelihood is derived from the joint multivariate Gaussian distribution defined by the network.

Since log-likelihood can get higher as models become more complex, we also use **5-fold cross-validation**. This is a common method to test prediction on new data. For each fold, we train the model on 80% of the data and test it on the other 20%. We repeat this for all parts. This helps us see how well the model predicts and reduces the risk of overfitting. We use the average log-likelihood from all folds as our main score.

Table 5 summarizes these evaluation results by showing the cross-validated log-likelihood (CV-LL) and the score for each model, connecting the methodology discussed above to the specific outcomes.

Table 5: Predictive Performance Across Models

Model	Score Type	Restarts	Score Value	CV Log-Likelihood
AIC-G	1	1	-8724.0	-1746.3
AIC-G	10	10	-8724.0	-1746.8
AIC-G	100	100	-8723.0	-1746.2
AIC-G	1000	1000	-8721.0	-1748.3
BIC-G	1	1	-8809.0	-1747.2
BIC-G	10	10	-8809.0	-1747.5
BIC-G	100	100	-8803.0	-1747.0
BIC-G	1000	1000	-8803.0	-1747.9

AIC-G models consistently achieve better (less negative) log-likelihoods than BIC-G models, in both in-sample and cross-validation. This is expected, as AIC penalizes complexity less, allowing more arcs and higher likelihood scores. For example, `aic-g_100` yields the best cross-validated log-likelihood (-1746.2), closely followed by `aic-g_1` and `aic-g_1000`. However, performance differences across restarts are minor compared to structural differences, especially for AIC models.

These results show a common trade-off in statistical modeling: **predictive accuracy versus interpretability**. AIC-G models give higher predictive scores, but their structures are denser and

change more. BIC-G models are simpler, more stable, and probably easier to interpret. However, they lose a little in predictive performance.

One surprising result is that the range of cross-validated (CV) log-likelihood values is quite small across all models. Although the model structures, particularly AIC-G and BIC-G differ noticeably, their cross-validated performance is very similar. This may be explained by the relatively small dataset size ($n = 167$); each fold leaves about 132 training examples and only 33 test examples. With so few test cases in each fold, it is challenging to detect real differences in model performance.While using $k = 5$ folds is widely accepted for balancing bias and variance, the small sample size means this choice may not be ideal here. Fewer folds (e.g., $k = 3$) would produce larger test sets but more bias, whereas more folds (e.g., $k = 10$) would create very small test sets and higher variance. We selected $k = 5$ as a practical compromise, but it is possible that the limited dataset contributed to the similarity in cross-validation results across models.

Another possible reason for the similar CV results is that AIC-G models, which are often denser, may be overfitting the training data. While these models can capture more complexity and fit the training set better, this does not always lead to stronger performance on new data. With a limited sample size, overfitting becomes an even bigger concern because the evaluation process is less likely to penalize patterns that only exist in the training set. Although using $k = 5$ folds aims to balance training and test data, the small differences in CV log-likelihood show that neither overfitting nor how the data is split make a big difference in performance.

Finally, it is important to note that increasing the number of restarts leads to very little improvement in predictive performance. For instance, AIC-G models do not show any clear gains beyond 100 restarts. This means that while more restarts help explore different structures, the predictive benefit may stop increasing early, especially when the dataset size is fixed. In situations with small sample sizes like this one, prioritizing stability and interpretability can be more beneficial than striving for minor gains in predictive accuracy.

5 Discussion

This thesis examined whether clear, understandable Bayesian networks can be learned from global prosperity data, how much learned structures depend on the choice of scoring function and search method, and whether differences in structure affect predictive performance.

Building on the initial research questions, the results show that Bayesian networks can find useful links between prosperity indicators. Many important relationships, such as those among *education*, *health*, and *governance*, are present in most models, no matter which score function or restart number is used. This pattern suggests that the data contain strong statistical signals that structure learning can detect. For example, certain v-structures (like *governance* acting as a connection point between *education* and *personal freedom*) support the idea that these methods can uncover meaningful conditional dependencies that matter for policy and causal reasoning.

To better understand these findings, it is helpful to compare the AIC-G and BIC-G models. AIC-G graphs are denser, with more edges, more v-structures, and larger Markov blankets. This shows that AIC uses weaker penalties and is more likely to add edges that slightly improve fit. BIC, on the other hand, produces simpler and more stable structures across restarts. These structural differences are significant because they influence our understanding of the network. For example, in AIC-G, many nodes are in the Markov blankets of most others, showing broad conditional dependence. In BIC-G, the greater sparsity points to a smaller set of strong dependencies.

Furthermore, these structural patterns also appear in the Structural Hamming Distance (SHD) and Jaccard similarity scores. When comparing results within the same score type, BIC-G models are very consistent across different restarts, while AIC-G models change more, especially in edge directions. However, if we ignore edge direction and only look at the overall connections (skeletons), both scoring methods agree more. This means that most models find similar groups of dependent variables, even if they disagree on direction. This is especially important for those interested in associations rather than causality.

Shifting focus to predictive evaluation, the results give a more nuanced view. Even though the models have different structures, they all achieve similar cross-validated log-likelihoods. AIC-G models are slightly better overall, but only by a small amount. This probably happens because (1) the small sample size limits our ability to see real improvements; (2) denser models overfit a bit but do not predict much better; and (3) the Gaussian assumptions smooth out structural differences in prediction. Notably, increasing the number of restarts has little effect on predictive performance, even though it alters the structure. This means that unstable structures are not always bad for prediction, at least in this context.

Considering these predictive results along with the structural findings, the evidence shows that learning Bayesian network structures can produce useful and understandable models for global prosperity. However, the final results depend a lot on the choice of scoring function and search method. When we focus on the best-performing DAGs, the resulting structures can be quite different, suggesting they may not be fully reliable for precise interpretation and are best used for exploratory purposes. In contrast, when the analysis is based on CPDAGs and the overall network skeletons, the models are more consistent, supporting their use in finding general patterns of conditional independence.

In summary, AIC-G models tended to produce denser networks that captured more dependencies, but they were also more complex and less stable. In comparison, BIC-G models resulted in sparser, more consistent structures that were easier to interpret, yet still represented the key relationships

in the data. Although their structures differed, both approaches showed similar cross-validation results, likely because the small dataset made it hard to detect strong differences in how well they generalized.

Increasing the number of random restarts past 100 did not noticeably improve predictive accuracy, indicating that further restarts provided little added value with the dataset size available. Taken together, these findings answer our research questions and suggest that, when working with limited data, it is better to prioritize models that are interpretable and stable. If the focus is on uncovering as many relationships as possible, AIC-G is a good option; if clear interpretation and reproducibility are most important, BIC-G is the better choice.

Ultimately, these findings illustrate the importance of looking at both a model's structure and its predictive performance to truly understand its value. Structural comparisons reveal what the model assumes about the data, while predictive tests demonstrate how well it performs on new data. In this thesis, both views were necessary to determine whether learning structures from scores is a useful approach to studying prosperity.

6 Conclusion

This thesis explored whether score-based Bayesian network learning reveals meaningful relationships among prosperity indicators. Using Legatum Prosperity Index data, we trained Gaussian Bayesian Networks with AIC and BIC and examined how random restarts affect structure and prediction.

Our findings show that Bayesian networks can recover useful patterns of dependence across the pillars of prosperity. Building on the results, many relationship, such as those involving *education*, *health*, and *governance*, appear regularly across models, suggesting they reflect real statistical signals rather than random results from the algorithm. AIC-G models were typically denser, revealing more connections and v-structures, whereas BIC-G yielded simpler and more stable structures. Notably, predictive performance stayed similar across models despite these differences. This highlights a classic modeling trade-off: AIC-G provides richer but more complex models, while BIC-G prefers simpler and more interpretable networks with little loss in generalization.

These results highlight the value of Bayesian networks in social science research, particularly for complex datasets like the Prosperity Index, where indicators are interlinked. Their ability to find both direct and indirect dependencies makes them a useful tool for exploratory analysis, setting policy priorities, and understanding the network of factors that influence well-being. Moving forward, it is essential to recognize the key challenges that may arise in such analyses.

Limitations. However, important limitations remain. For instance, Gaussian Bayesian Networks assume the data follow a normal distribution. In our case, normality tests showed strong deviations from this assumption across all pillars (kurtosis about 4.8; Shapiro-Wilk $p < 0.0001$). While some patterns were consistent across models, this non-normality may have biased parameter estimates and affected scoring functions like AIC and BIC, which depend on normal log-likelihoods. In future work, one way to handle this is to use data transformations (such as Box-Cox or log) to reduce skewness and heavy tails. Another option is discretization: turning continuous scores into ordinal or quantile-based bins. This allows the use of discrete Bayesian networks, which can be more robust when normality does not hold. These approaches should be tried to check if the results here remain stable with different modeling choices. Addressing these limitations will ensure greater confidence in the findings.

Future Research. Future research could build on this study in several ways. First, analyzing multiple years of the Prosperity Index would help model structural changes over time. This dynamic approach would highlight which dependencies stay stable and which change, revealing long-term development trends. Second, combining Bayesian network results with qualitative research, like expert interviews, policy case studies, or regional reports, would add context. For instance, if *governance* is a key link, qualitative evidence could shed light on why. Using both methods connects data-driven findings to actual policy questions. Finally, applying structure learning to specific regions, such as Sub-Saharan Africa or OECD countries, could show if dependency patterns are global or context-specific. These extensions would make Bayesian networks more adaptable and useful for social science and development research.

In conclusion, Bayesian networks; despite their assumptions, are a promising method for analyzing complex data like the Legatum Prosperity Index. By integrating these methods thoughtfully and considering their limitations, we can look beyond single indicators and see how variables interact as a system. For social scientists, policymakers, and development analysts, such insights can lead

to more effective actions by showing which areas, such as *governance*, *education*, or *investment conditions*, are central in the network of prosperity.

References

[1] Legatum Institute and Our World in Data. *2023 Global Prosperity Index w/ Region Politics*. 2024. DOI: 10.34740/KAGGLE/DS/5375158. URL: https://www.kaggle.com/ds/5375158.

[2] Marco Scutari. "Learning Bayesian Networks with the bnlearn R Package". In: *Journal of Statistical Software* 35.3 (2010), pp. 1–22. URL: https://www.jstatsoft.org/article/view/v035i03.

[3] Javier Alvarez-Galvez. "Discovering Complex Interrelationships Between Socioeconomic Status and Health in Europe: A Case Study Applying Bayesian Networks". In: *Social Science Research* 56 (2016), pp. 133–143. ISSN: 0049-089X. DOI: 10.1016/j.ssresearch.2015.12.011.

[4] Jouni Kuha. "AIC and BIC: Comparisons of Assumptions and Performance". In: *Sociological Methods & Research* 33.2 (2004), pp. 188–229. DOI: 10.1177/0049124103262065. URL: https://doi.org/10.1177/0049124103262065.

[5] Marco Scutari, Cornelis E Graafland, and Marcel A J van Gerven. "Who Learns Better Bayesian Network Structures: Accuracy and Speed of Structure Learning Algorithms". In: *International Journal of Approximate Reasoning* 95 (2018), pp. 141–170.

[6] José A. Gámez, Juan L. Mateo, and José M. Puerta. "Learning Bayesian Networks by Hill Climbing: Efficient Methods Based on Progressive Restriction of the Neighborhood". In: *Data Mining and Knowledge Discovery* 22.1 (2011), pp. 106–148. ISSN: 1573-756X. DOI: 10.1007/s10618-010-0178-6. URL: https://doi.org/10.1007/s10618-010-0178-6.

[7] Bruce G. Marcot and Anca M. Hanea. "What Is an Optimal Value of k in k-Fold Cross-Validation in Discrete Bayesian Network Analysis?" In: *Computational Statistics* 36.3 (2021), pp. 2009–2031. ISSN: 1613-9658. DOI: 10.1007/s00180-020-00999-9. URL: https://doi.org/10.1007/s00180-020-00999-9.

A Appendix

A.1 DAGs and CPDAGs for All Learned Models

This appendix presents all Directed Acyclic Graphs (DAGs) alongside their corresponding Completed Partially Directed Acyclic Graphs (CPDAGs), which were learned using score-based structure learning with the hill-climbing algorithm. Each model is shown for both the AIC-G and BIC-G scoring criteria. Visualizations are provided for various numbers of random restarts (1, 10, 100, and 1000). These images help demonstrate the structural patterns identified by each configuration. To make the details easier to see, the images are rendered at a larger size.

A.1.1 AIC-G Models

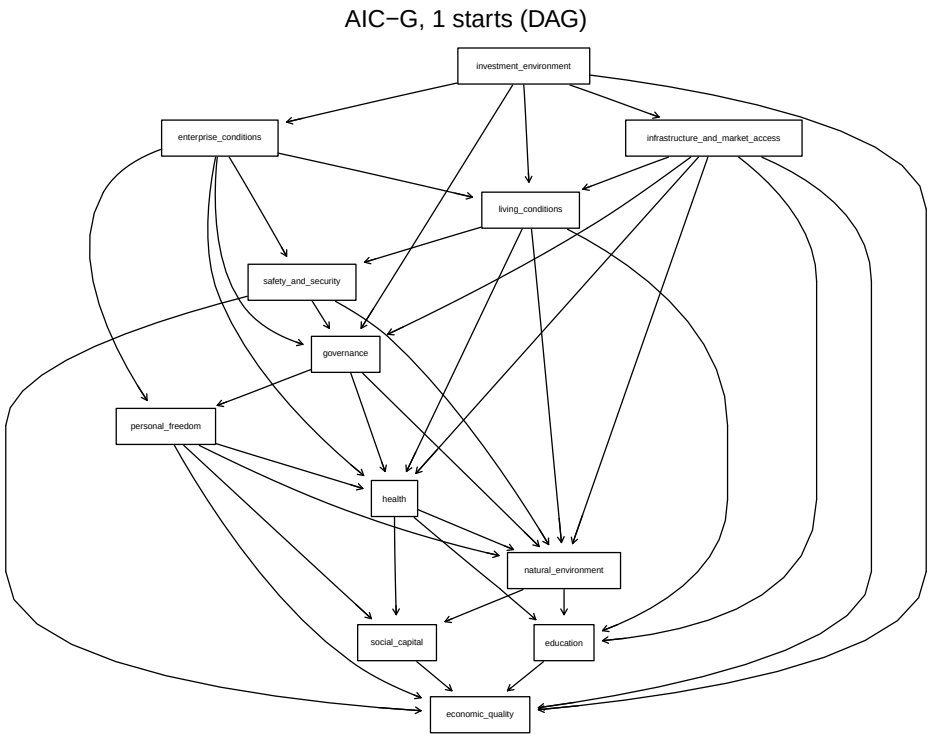

Figure 1: DAG - AIC-G with 1 restart

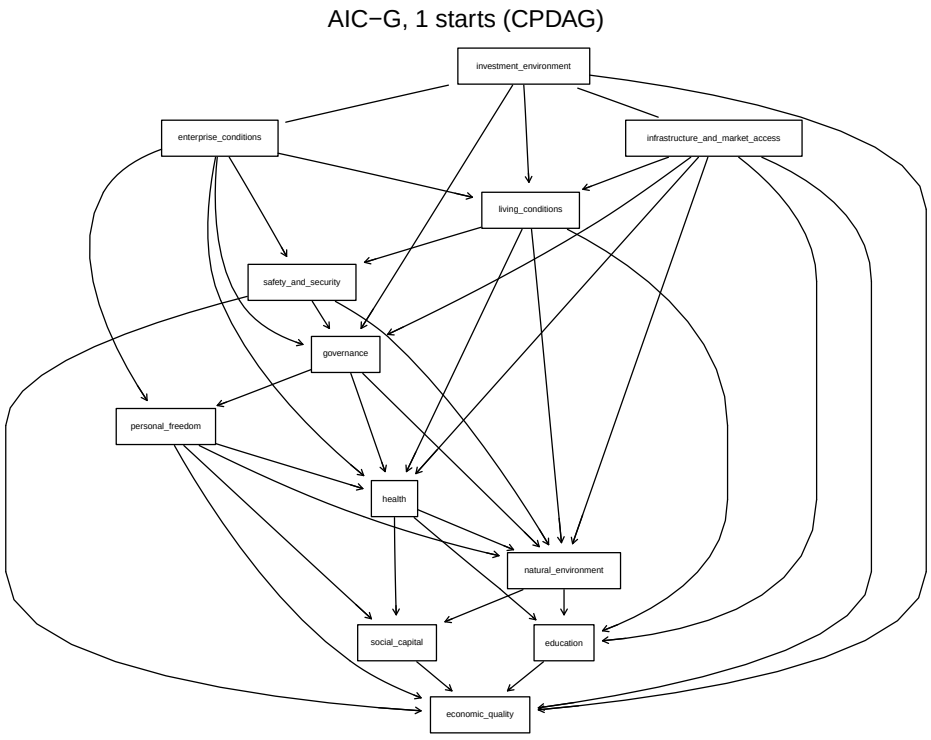

Figure 2: CPDAG - AIC-G with 1 restart

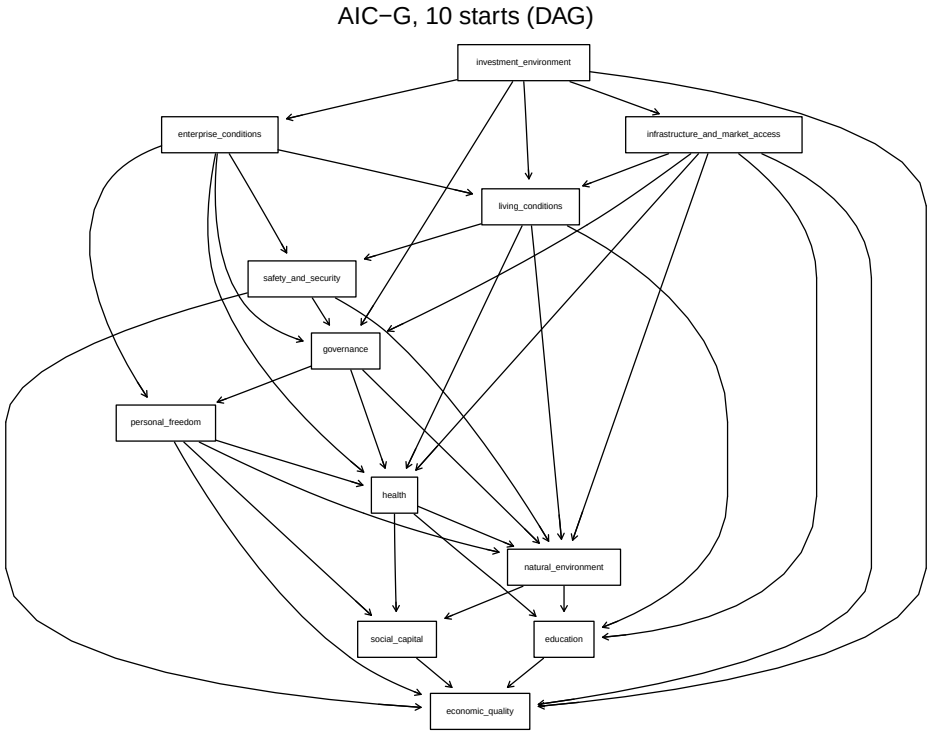

Figure 3: DAG - AIC-G with 10 restarts

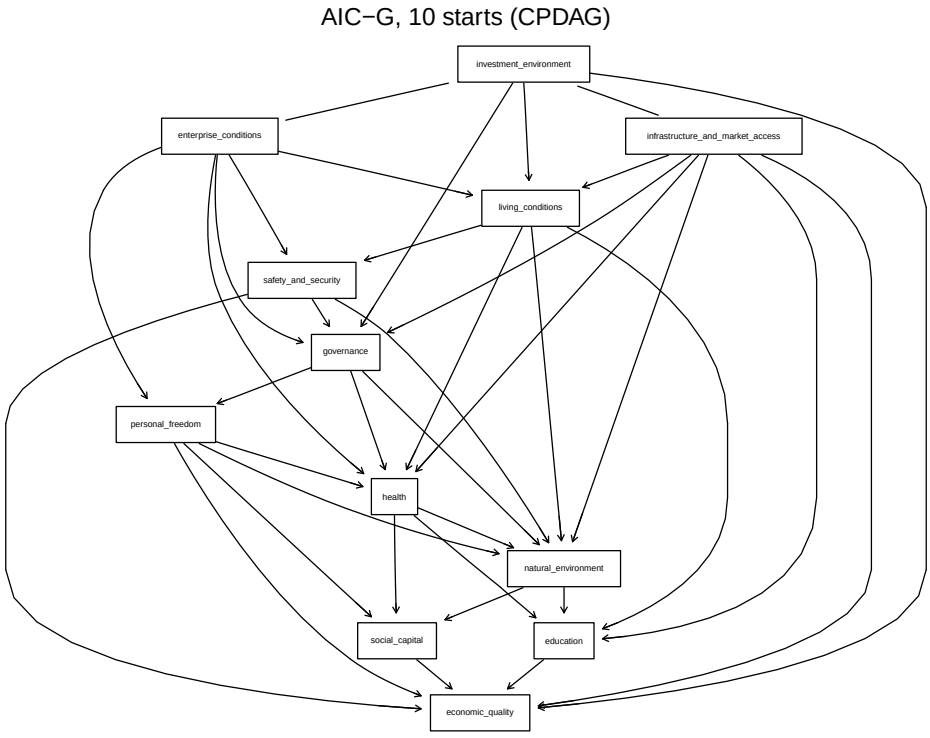

Figure 4: CPDAG - AIC-G with 10 restarts

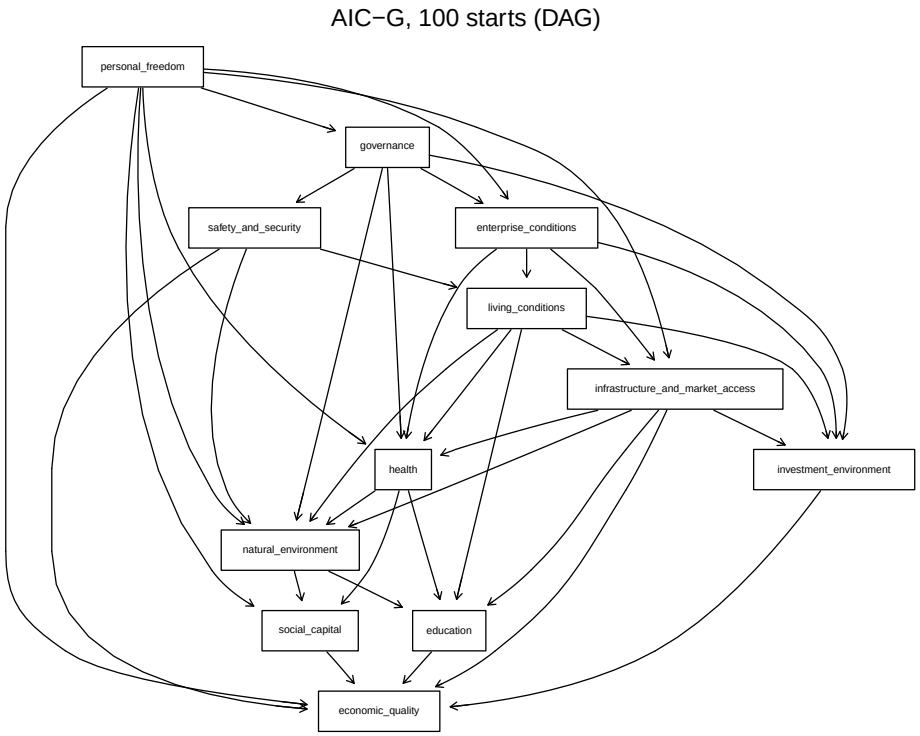

Figure 5: DAG - AIC-G with 100 restarts

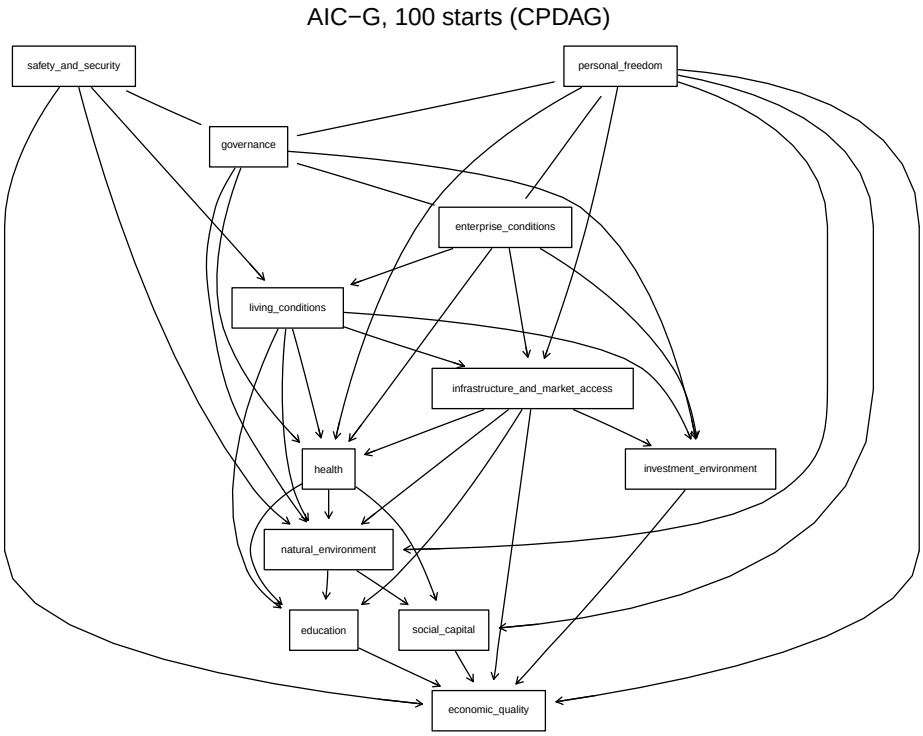

Figure 6: CPDAG - AIC-G with 100 restarts

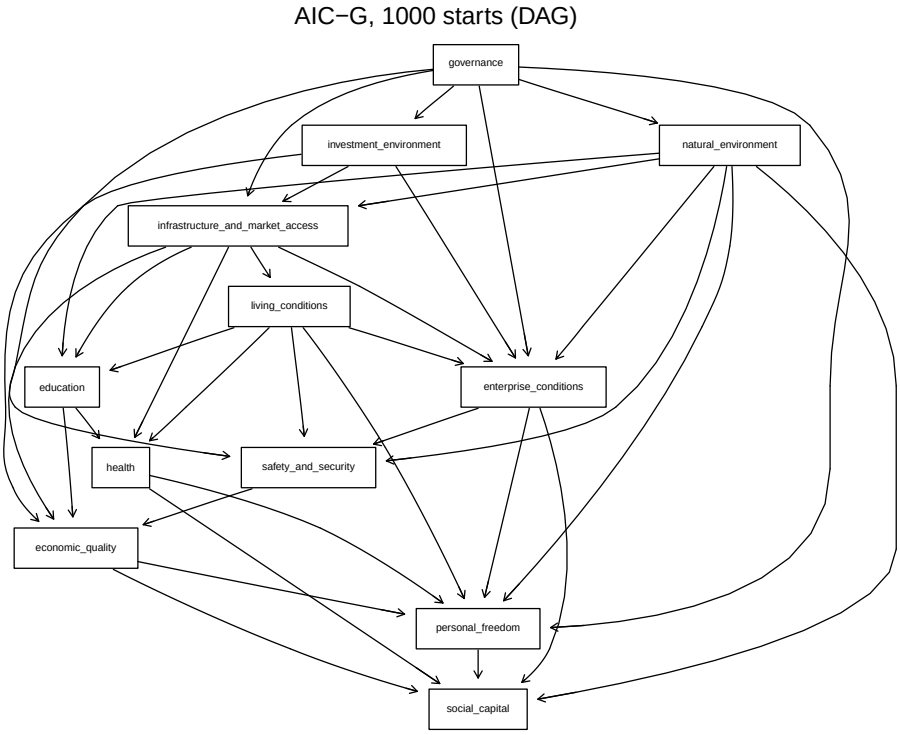

Figure 7: DAG - AIC-G with 1000 restarts

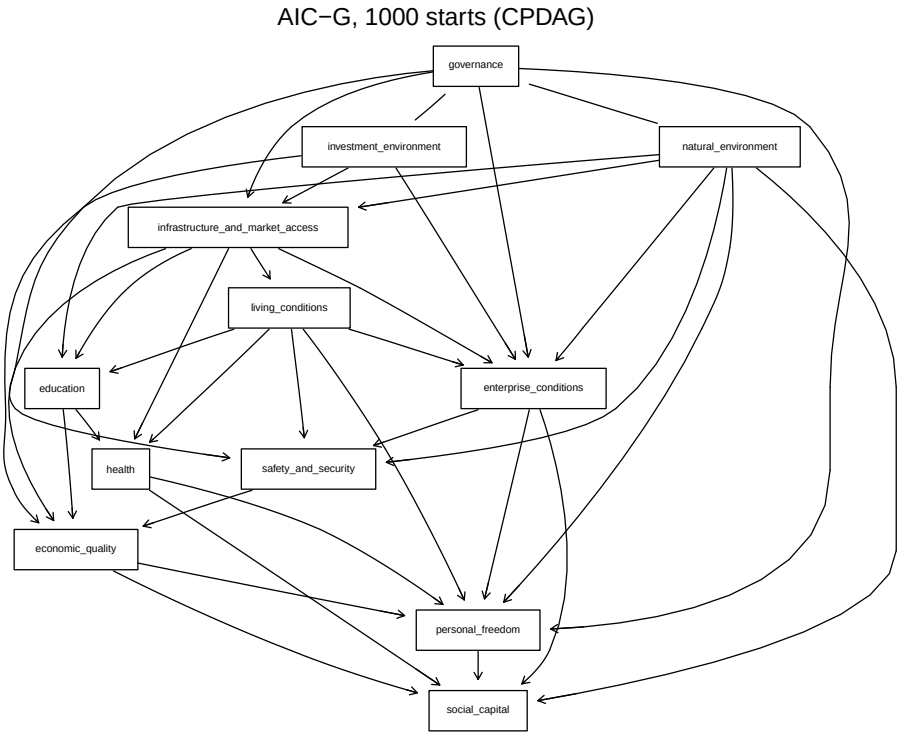

Figure 8: CPDAG - AIC-G with 1000 restarts

A.1.2 BIC-G Models

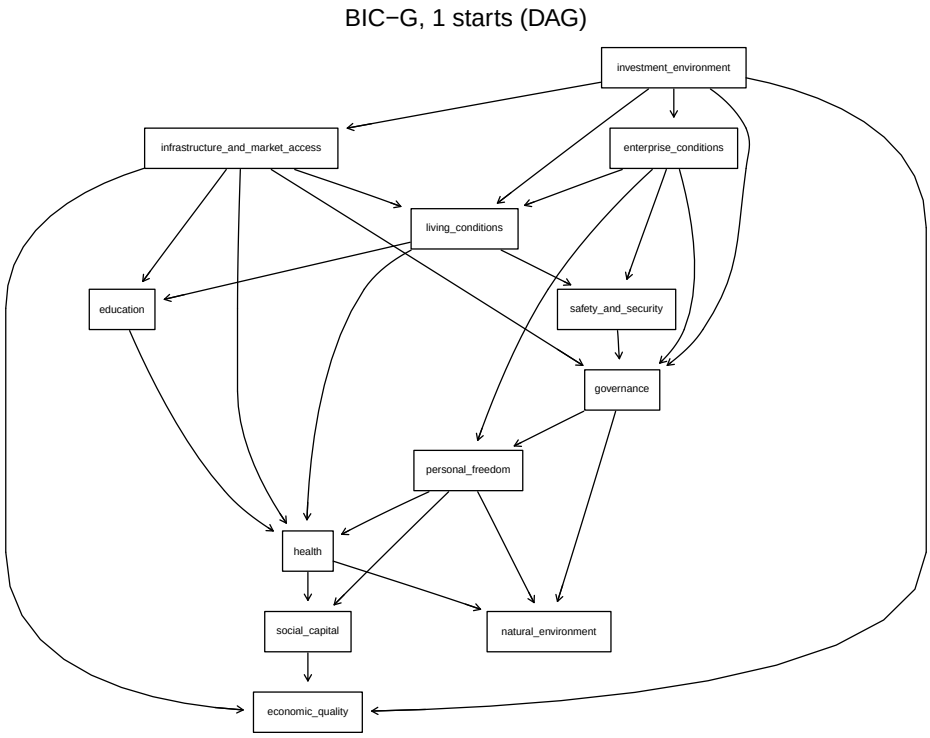

Figure 9: DAG - BIC-G with 1 restart

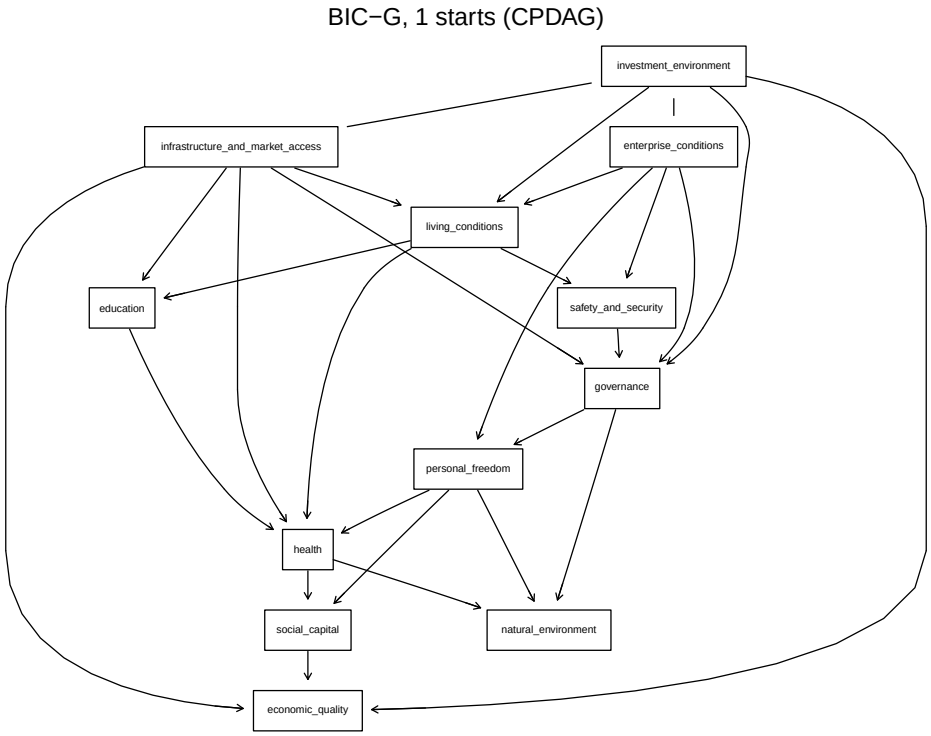

Figure 10: CPDAG - BIC-G with 1 restart

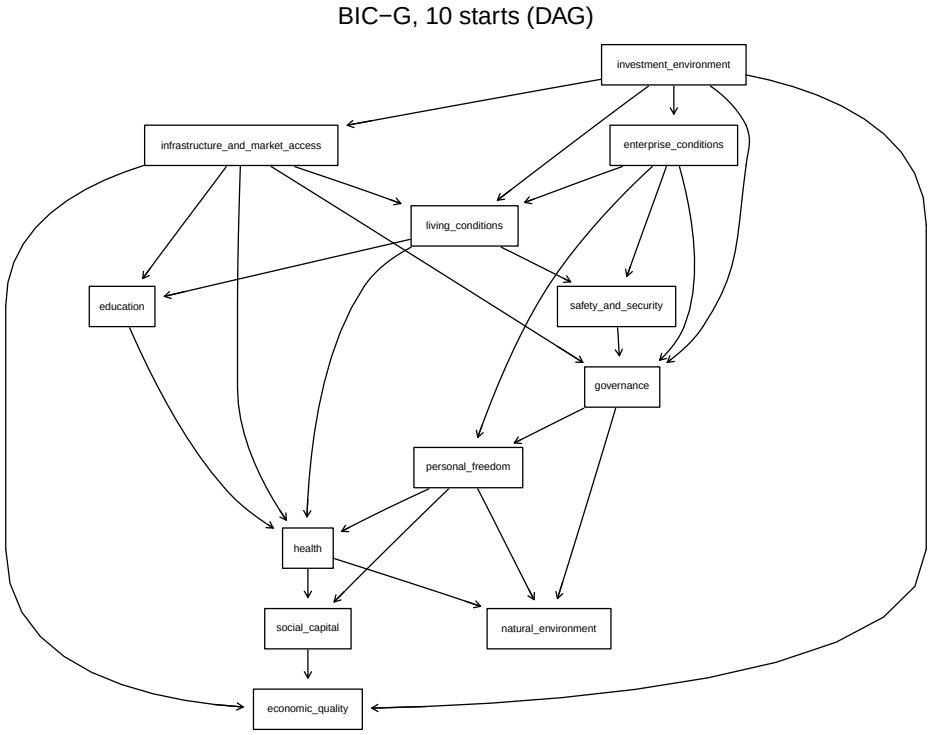

Figure 11: DAG - BIC-G with 10 restarts

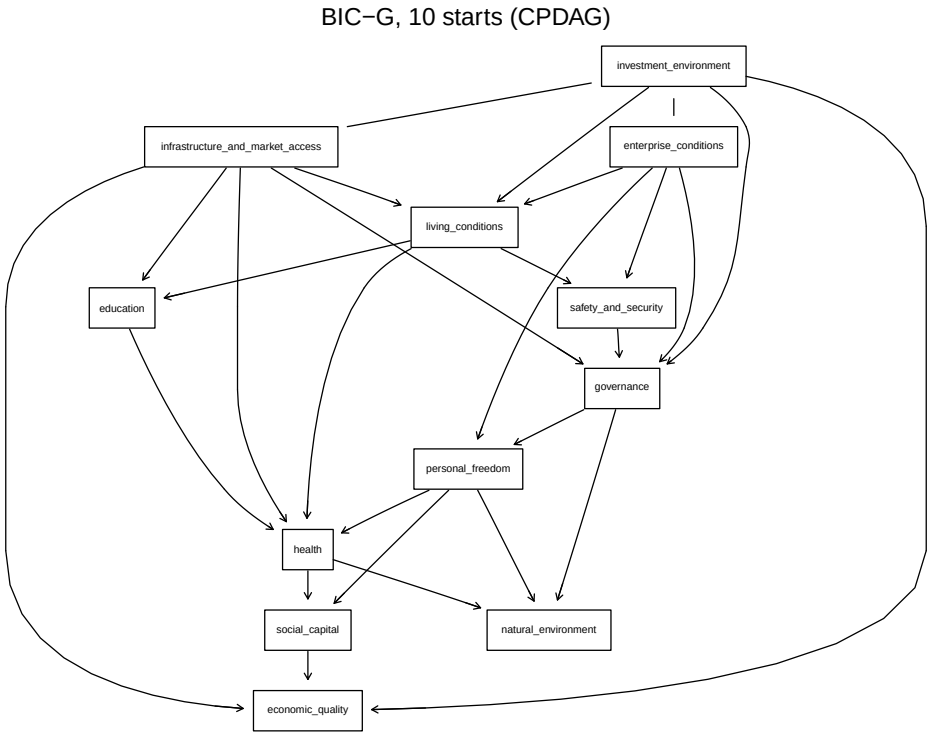

Figure 12: CPDAG - BIC-G with 10 restarts

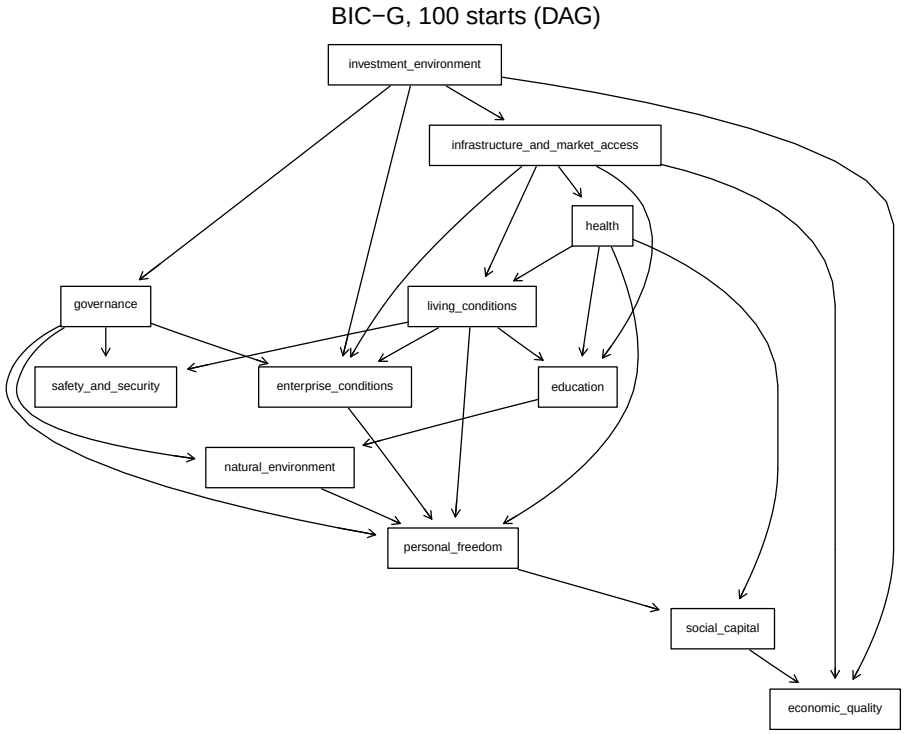

Figure 13: DAG - BIC-G with 100 restarts

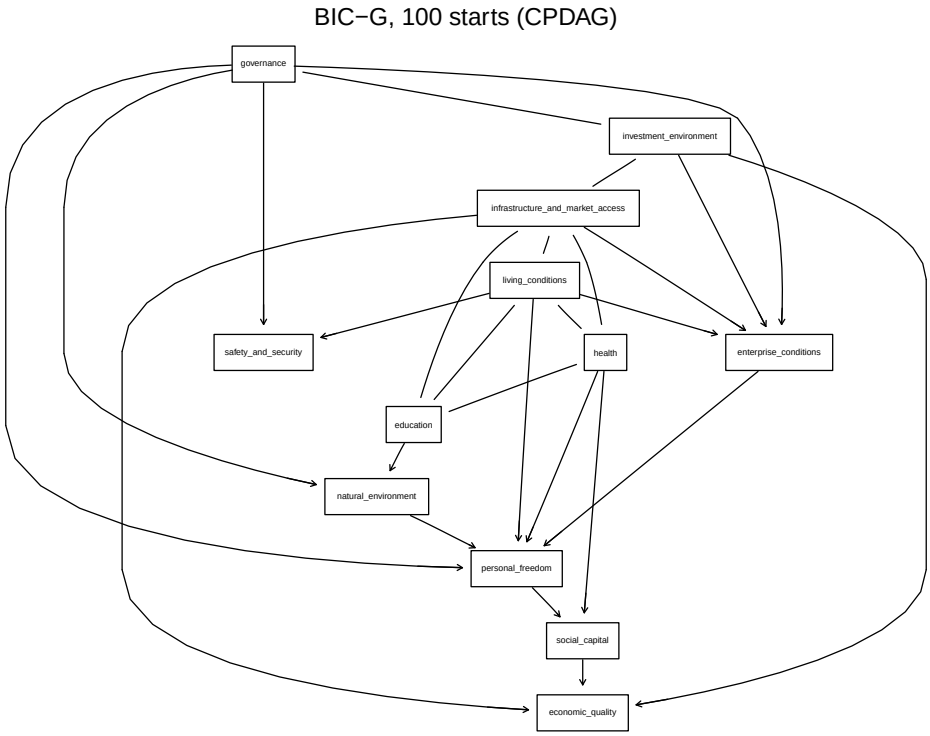

Figure 14: CPDAG - BIC-G with 100 restarts

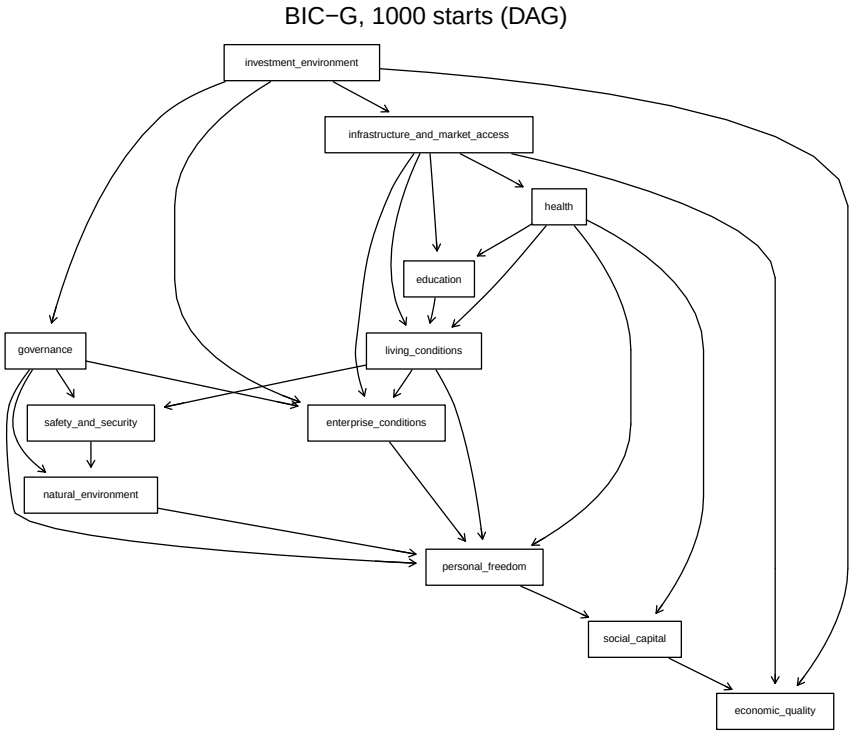

Figure 15: DAG - BIC-G with 1000 restarts

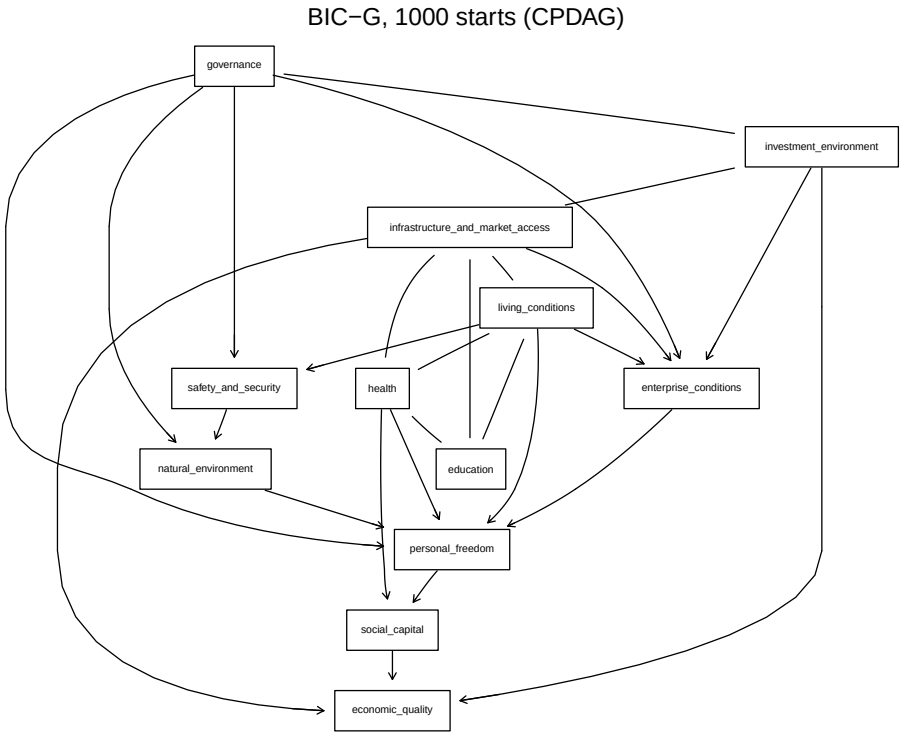

Figure 16: CPDAG - BIC-G with 1000 restarts