# Fundamentals of Probabilistic Data Mining
# Graded work 1: probabilistic graphical models

Shubham Agarwal
MSIAM Master 2 Data Science (2016-17), University of Grenoble Alpes
Nitika Verma
MoSIG Master 2 Data Science (2016-17), Grenoble INP

## 1 Part 1: properties of ARGES algorithm (Research)

**Presented by: Nitika VERMA**

**1. p.2 Give a formal definition of high-dimensional consistency.**
High-Dimensional context is when the number of nodes p may be much larger than sample size n, i.e. $p \gg n$. An algorithm is defined to be high-dimensional consistent when it consistently estimates the equivalence class and the skeleton of an underlying sparse DAG, as sample size $n \rightarrow \infty$, even if $p = p_n = O(n^a)(0 \leq a < \infty)$ is allowed to grow very quickly as a function of n [5]. The vertex set and graph become $\mathbf{V} = \mathbf{V}_n = (X_{n,1}, ..., X_{n,p_n})$ and $G = G_n$ respectively. The CPDAGs which represents $G_n$ are denoted by $C_n$, and the estimated CPDAGs using tuning parameter $\alpha_n$ are denoted by $\hat{C}_n(\alpha_n)$. The high dimensional consistency states that under some conditions, there exists a sequence $\alpha_n$ such that $P(C_n(\alpha_n) = \hat{C}_n) \rightarrow 1$ as $n \rightarrow \infty$ [3].

**2. p.2 Since it is claimed that both GES and ARGES are high-dimensional consistent, why to bother to use ARGES rather than GES?**
The paper proves that consistency of GES and Adaptively Restricted GES (ARGES) in sparse high-dimensional settings with multivariate Gaussian or nonparanormal distributions. But, GES does not scale well to large graphs due to which arises the need for hybrid algorithms based on GES which are consistent in certain high-dimensional settings and scale well to large sparse graphs. Adaptively restricted GES (ARGES) scales well to sparse graphs with thousands of variables.

**3. p.7 Give the definition of BIC.**
Bayesian information criterion, was introduced by Schwarz (1978)[10] as an asymptotic approximation to a transformation of the Bayesian posterior probability of a candidate model. In large-sample settings, the fitted model favored by BIC ideally corresponds to the candidate model which is a posteriori most probable; i.e., the model which is rendered most plausible by the data at hand. By choosing the fitted candidate model corresponding to the minimum value of BIC, one is attempting to select the candidate model corresponding to the highest Bayesian posterior probability. The BIC-score is the sum of a likelihood term and a penalty term which penalizes complex networks and is decomposable and equivalent. The penalty term of BIC is more stringent than the penalty term of AIC. Consequently, BIC tends to favor smaller models than AIC. It has

the following formulation

$$BIC(G,D) = logP(D \mid G, \theta^{ML}) - \frac{1}{2}Dim(G)logN \tag{1}$$

where $D$ is the dataset, $\theta^{ML}$ are the parameter values obtained by likelihood maximisation, and where the network dimension Dim(G) is defined as follows: As we need $r_i - 1$ parameters to describe the conditional probability distribution $P(X_i \mid Pa(X_i) = pa_i)$ , where $r_i$ is the size of $X_i$ and $pa_i$ a specific value of $X_i$ parents, we need $Dim(X_i, G)$ parameters to describe $P(X_i \mid Pa(X_i))$ with $Dim(X_i, G) = (r_i - 1)q_i$ where $q_i = \Pi_{X_j \in Pa(X_i)} r_j$ [7]. Thus, the dimension of the Bayesian network is defined by

$$Dim(G) = \sum_{i=1}^{n} Dim(X_i, G) \tag{2}$$

The BIC criterion is a special case of the $l_0$-penalized log-likelihood score. In particular, the BIC score of a DAG $H_n$ equals $2nS_{\lambda_n}(Hn, Dn)$ with $\lambda_n = log(n)$

**4. p.7 Prove that BIC is decomposable (make the model parametrisation explicit).** For any scoring criterion $S(G,D)$, we say that $S$ is decomposable if it can be written as a sum of measures, each of which is a function only of one node and its parents. [2] In other words, a decomposable scoring criterion $S$ applied to a DAG $G$ can always be expressed as:

$$S(G,D) = \sum_{i=1}^{n} FamScore(X_i \mid \mathbf{Pa}_i^G, D) \tag{3}$$

Bayesian Information Score (BIC) can be decomposed as :

$$\begin{aligned} BIC(G,D) &= logP(D \mid G, \theta^{ML}) - \frac{1}{2}Dim(G)logN \\ &= \sum_i \sum_k \sum_j m_{ijk} log \frac{m_{ijk}}{\sum_i m_{ijk}} - \sum_i \frac{Dim(X_i, G)}{2} logN \\ &= \sum_i BIC(X_i \mid \mathbf{Pa}_i^G, D) \end{aligned} \tag{4}$$

where $Dim(X_i, G)$ is defined in the question 3 and the family score for BIC is :

$$BIC(X_i \mid \mathbf{Pa}_i^G, D) = \sum_k \sum_j m_{ijk} log \frac{m_{ijk}}{\sum_i m_{ijk}} - \frac{Dim(X_i, G)}{2} logN \tag{5}$$

This proves that BIC is decomposable.

**5. p.8 Define what structural equation models (SEMs) are. How are they related to probabilistic graphical models?**

Structural equation modeling (SEM) uses various types of models to depict relationships among observed variables, with the same basic goal of providing a quantitative test of a theoretical model hypothesized by the researcher. The purpose of SEM is to examine a set of relationships between one or more Independent Variables (IV) and one or more Dependent Variables (DV). Structural equation modeling is also known as 'causal modeling' or 'analysis of covariance structures' [11].

The structural equation model is an algebraic object. As long as the causal graph remains acyclic, algebraic manipulations are interpreted as interventions on the causal system. The Bayesian network is a generative statistical model representing a class of joint probability distributions, and, as such, does not support algebraic manipulations. However, the symbolic representation of its Markov factorization is an algebraic object, essentially equivalent to the structural equation model. Both Bayesian networks (BN) and Structural equation model (SEM) are graphical models that are able to model causality [1].

For $\epsilon^1, ..., \epsilon^4$ independently distributed standard Gaussian random variables, we write the linear SEM in matrix notation as $X = BX + \epsilon$, where $B$ is a lower triangular matrix of coefficients and $\epsilon = (\epsilon^1, ..., \epsilon^4)^T$. Thus $X = (I - B)^{-1}\epsilon$ and $X$ has a zero-mean multivariate Gaussian distribution with covariance matrix $\sum_0 = (I - B)^{-1}(I - B)^{-T}$. This linear SEM can be represented by the DAG $G_0$, where an edge $Xi \rightarrow Xj$ is present if and only if $B_{ji} \neq 0$ and then the weight of the edge $Xi \rightarrow Xj$ is $B_{ji}$.

**6. p.8 Can any model of the form $X = (I - B)^{-1}\epsilon$, where $\epsilon$ is some i.i.d. Gaussian sample and $B$ a lower triangular matrix, be rewritten as linear SEM? Why / How?**

We can write the model of the form $X = (I - B)^{-1}\epsilon$ as a linear SEM when $B$ is a strictly lower triangular matrix.

$$\begin{aligned} X = (I - B)^{-1}\epsilon &\iff (I - B)X = \epsilon \\ &\iff X = BX + \epsilon \end{aligned} \tag{6}$$

which is the matrix notation of recursive linear SEM and $\epsilon$ is some i.i.d. Gaussian sample independent of all $X_i$'s. $(I - B)$ is always invertible because B is strictly lower triangular and the above equation is well-defined. Thus,

$$\begin{aligned} X_1 &= \epsilon_1 \\ X_2 &= B_{2,1}X_1 + \epsilon_2 \\ &\cdots \\ X_n &= \sum_{i=1}^{n-1} B_{n,i}X_i + \epsilon_n \end{aligned} \tag{7}$$

We are able to write each $X_k$ as a deterministic function of other $X_i$'s plus a noise that is independent of every other random variable. The graph $G_0$ induced by this linear SEM will have edge $X_i \rightarrow X_j$ is present if and only if $B_{ji} \neq 0$. As B is a strictly lower triangular matrix, $G_0$ is a DAG.

**7. p.9 Does Remark 3.1 contradicts consistency of GES? Why?**

No, Remark 3.1 does not contradict consistency of GES. In fact using Example 1, the authors have shown consistency of GES and shown inconsistency of naive hybrid versions of GES restricting the search space to an estimated CIG or CPDAG- skeleton. They have referred these hybrid methods as RGES-CIG or RGES-skeleton. More precisely, they restrict the search space of GES by allowing an edge $X_i \to X_j$ for addition only if $X_i$ and $X_j$ are adjacent in the CIG or in the CPDAG-skeleton.

Remark 3.1 does not say anything about the consistency of GES while discussing about the inconsistency of hill-climbing DAG search, hill-climbing DAG search restricted to the CIG and hill-climbing DAG search restricted to the CPDAG-skeleton.
The authors then propose ARGES as compared to RGES hybrid methods for supporting their cause.

**8. p.10 Theorem 4.1. Draw $H$ and $G_0$ such satisfying item 2. Due to which independence property is not $H$ an I-map? Same question with item 3.**

Theorem 4.1 says that: A DAG H is not an independence map of G0 if and only if:

- $skeleton(G_0) \nsubseteq skeleton(H)$, or
- there exists a triple of nodes $(X_i, X_j, X_k)$ such that $X_i$ and $X_k$ are non-adjacent in $H$, $\pi_H(X_i, X_j, X_k)$ is a non-collider path, and $\pi_{G_0}(X_i, X_j, X_k)$ is a v-structure, or
- there exists a triple of nodes $(X_i, X_j, X_k)$ such that $\pi_H(X_i, X_j, X_k)$ is a v-structure and $X_i \not\perp_{G0} X_k \mid Pa_H(X_k)$, where without loss of generality we assume $X_i \in Nd_H(X_k)$.

As written in the paper, Proposition 27 and Lemma 28 of Chickering [2002] imply that if one of the first two conditions of Theorem 4.1 hold, then H is not an independence map. If the third condition of Theorem 4.1 holds, then $X_i \not\perp_{G0} X_k \mid Pa_H(X_k)$ and $Xi \perp_H X_k \mid Pa_H(X_k)$ (since $X_i \in Nd_H(X_k) \setminus Pa_H(X_k)$), and hence $H$ is not an independence map of $G_0$.
Thus for item 2 we can propose the following DAGs as in Fig 1 and see that $H$ is not an i-map of $G_0$ since node $X_1 \perp X_2 \mid X_3$ in H while not in $G_0$.

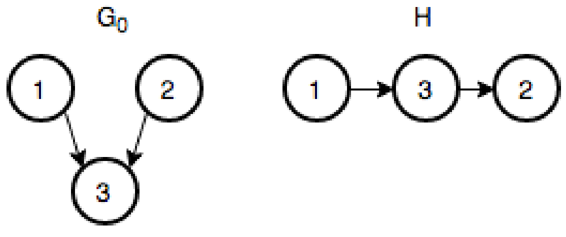


**Fig. 1.** $G_0$ and $H$ for item 2

Similarly for item 3, we propose simple DAGs as in Fig. 2

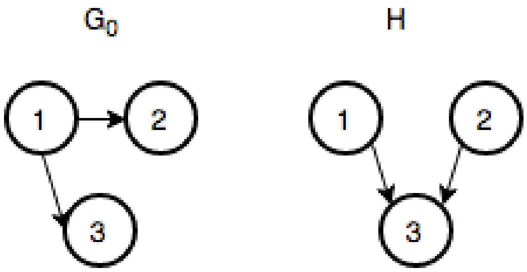


**Fig. 2.** $G_0$ and $H$ for item 3

Here $Pa(X_3) = \phi$ and we see that $H$ is not an i-map of $G_0$ since node $X_1 \perp X_2$ in H but not in $G_0$.

**9. p.11 Definition 4.1. Using the example in Fig. 1, show a move allowed by ARGES-CIG (respectively, -skeleton) not allowed by RGES-CIG (respectively, -skeleton).**

We can show the search paths of RGES-CIG and ARGES-CIG as in Fig.3. The red ar-

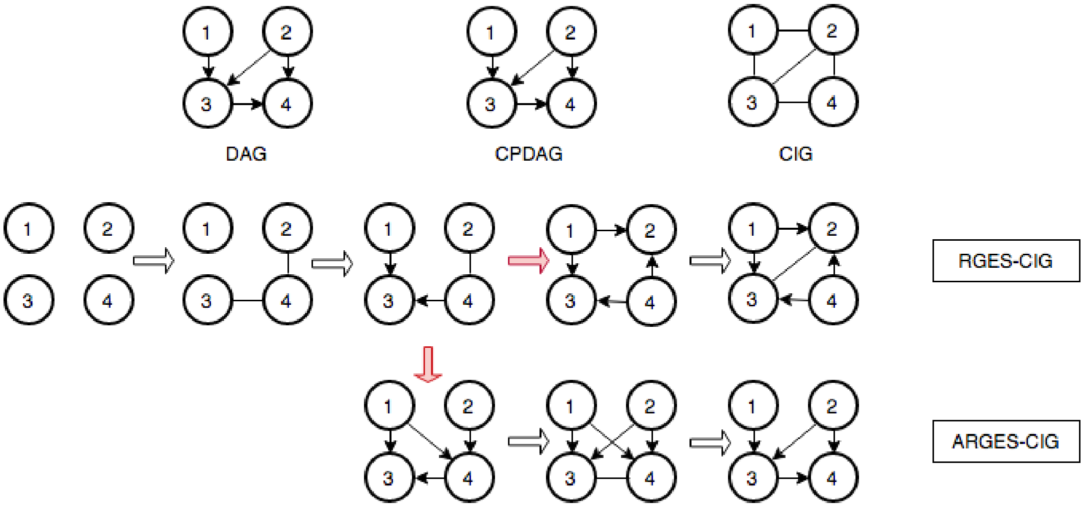


**Fig. 3.** $G_0$ and $H$ for item 3

rows mark the difference in both the algorithms showing permissible move for ARGES-CIG as compared to RGES-CIG.

Similarly, we can show a move as in Fig. 4 which is admissible for ARGES-Skeleton which is not permissible for RGES-Skeleton.

As given by Nandy et al., 2016 [**?**], the edge $X_i \rightarrow X_k$ (or $X_k \rightarrow X_i$) between a pair of non-adjacent nodes $(X_i, X_k)$ in a *CPDAG* C is admissible for C (ARGES-CIG) with respect to an undirected graph I if at least one of the following holds:

- $X_i$ and $X_k$ are adjacent in I; or
- There exists a node $X_j$ such that $(X_i, X_j, X_k)$ is a v-structure in C.

Also, the edge $X_i \rightarrow X_k$ (or $X_k \rightarrow X_i$) between a pair of non-adjacent nodes $(X_i, X_k)$ in a *CPDAG* C is admissible for C (ARGES-Skeleton) with respect to an undirected graph U if at least one of the following holds:

- $X_i$ and $X_k$ are adjacent in U; or
- There exists a node $X_j$ such that $(X_i, X_j, X_k)$ is an unshielded triple in C.

While for RGES-CIG or RGES-Skeleton, an edge $X_i \rightarrow X_j$ is admissible only if $X_i$ and $X_j$ are adjacent in the CIG or in the CPDAG-Skeleton. An unshielded triple $(X_i, X_j, X_k)$ is a v-structure if $X_i \rightarrow X_j \leftarrow X_k$. In Fig. 3, the edge $(1 \rightarrow 4)$ between the nodes 1 and 4 is admissible for ARGES-CIG because $(1 \rightarrow 3 \leftarrow 4)$ form a v-structure in the CPDAG. This move is not admissible in RGES-CIG as nodes 1 and 4 are not adjacent in the CIG. Similarly in Fig. 4, the edge $(1 \rightarrow 4)$ is admissible for ARGES-Skeleton as it forms an unshielded triple $(1 \rightarrow 3 \leftarrow 4)$ in the CPDAG, but it not admissible in RGES-Skeleton as nodes 1 and 4 are not adjacent in the CPDAG-Skeleton.

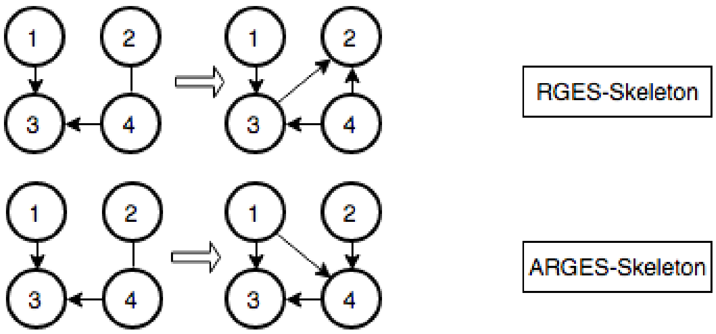


**Fig. 4.** $G_0$ and $H$ for item 3

**10. p.13 Notation 5.1. What does $p_n$ represent?**

We use $p_n$ to denote the number of covariates. The number of covariates $p_n = O(n^a)$ for some $0 \leq a < \infty$. This assumption allows the number of covariates to grow as any polynomial of the sample size, representing the high-dimensional setting.

**11 p.17 Could we estimate $c_n$ (hence $\lambda_n$) from real data? Why / How?**

$c_n$ is a lower bound on partial correlations with $c_n^{-1} = O(n^{d_2})$ for some $0 \leq d_2 < \frac{b_2}{2}$. This has been described as Assumption (A6) in the paper for proving high dimensional consistency of ARGES. The lower bound of partial correlations is unknown. The authors have defined $\lambda_n = \frac{1}{9} log(1 - c_n^2)$ Tuning the penalty parameter $\lambda$ of a scoring criterion of ARGES is a well-known practical problem and the authors recommend to apply the stability selection approach of Meinshausen and Bühlmann [2010] [9]. To get a feeling for good values in the domain of realistic parameter settings, we should fit a wide range of parameter settings and compared the quality of fit for different values.

## 2 Part 2: comparison of two algorithms for graph estimation (Industrial)

**Presented by: Shubham AGARWAL**

### 2.1 Simulated Data

For this question, we have to simulate a multi variate Gaussian model with perfect map as in Fig. 5 using bnlearn package in R [12].

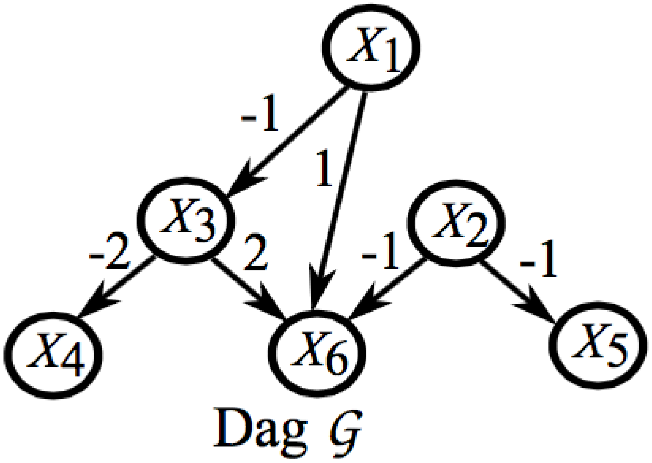


**Fig. 5.** Simulated Model

**Create some boolean function to decide whether the graph provided by some procedure such as gs or hc is equal to the true graph. Include this function in your report**

We can create the true graph by using the function model2network as

```
trueGraph = model2network(”[X1][X2][X3|X1][X4|X3][X5|X2][X6|X1:X2:X3”)
```

To check whether the graph produced by gs or hc, we have created a boolean function “compareWithTrueGraph” which uses *compare*() function of the bnlearn package. “isTrueGraphHC” is an example of calling this method, where we provide the true graph and the graph produced by the algorithm.

```
## Boolean function to decide whether graph
## is equal to true graph
compareWithTrueGraph = function(trueGraph,derivedGraph){
  stats = compare(trueGraph, derivedGraph)
  if(stats$fp !=0 | stats$fn !=0 |stats$tp ==0){
    return(FALSE)
  }
  else {return(TRUE)}
}
isTrueGraphHC = compareWithTrueGraph(trueGraph,bn.hc)
```

**Compare both estimated graphs with the true graph**

For taking a sample size of 40 to simulate the model as in Fig. 5 while using gs and hc algorithms from the bnlearn package we could not estimate the true graph as depicted in Fig. 6

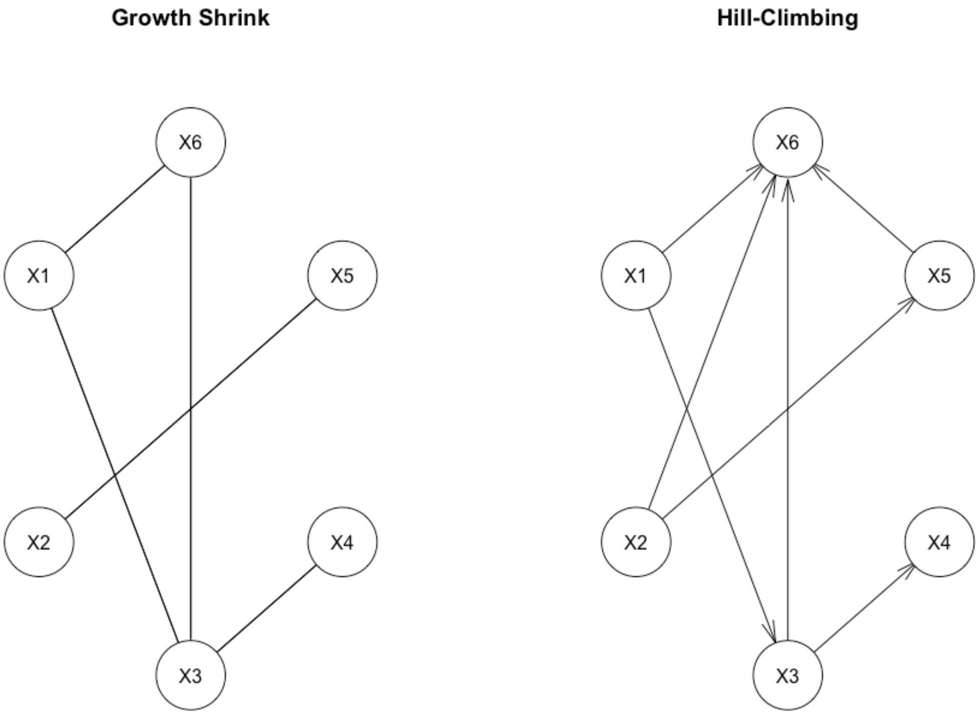


**Fig. 6.** Graph from gs and hc algorithm when sample size of 40

We can use the compare(target,current) function for the task where current is the graph that we want to compare with target graph. The function compare returns a list containing the number of true positives (tp, the number of arcs in current also present in target), false positives (fp, the number of arcs in current not present in target) and false negatives (fn, the number of arcs not in current but present in target).

| Algorithm | tp | fp | fn |
|---|---|---|---|
| GS | 0 | 5 | 6 |
| HC | 6 | 1 | 0 |

**Table 1.** Compare GS and HC when sample size 40

```
#Converting to dataframe results of compare
gsCompare = as.data.frame(compare(trueGraph, bn.gs))
hcCompare = as.data.frame(compare(trueGraph, bn.hc))
```

As we can see from Table. 1 and Fig. 6 HC algorithm is able to detect most of the edges while also false predicting edge $X_5 \rightarrow X_6$ when sample size =40. On the other hand GS algorithm performs poorly because it is generating an undirected graph (skeleton) and not providing direction sense on the edges. Thus, we get all the edges predicted as fp (=5) and fn as the number of all the edges (directed) in the original graph (=6).

When we doubled the sample size, i.e. to 80, HC algorithm could predict the true structure of the target graph while GS still could not predict the graph and performed badly even compared to HC with the sample size of 40.

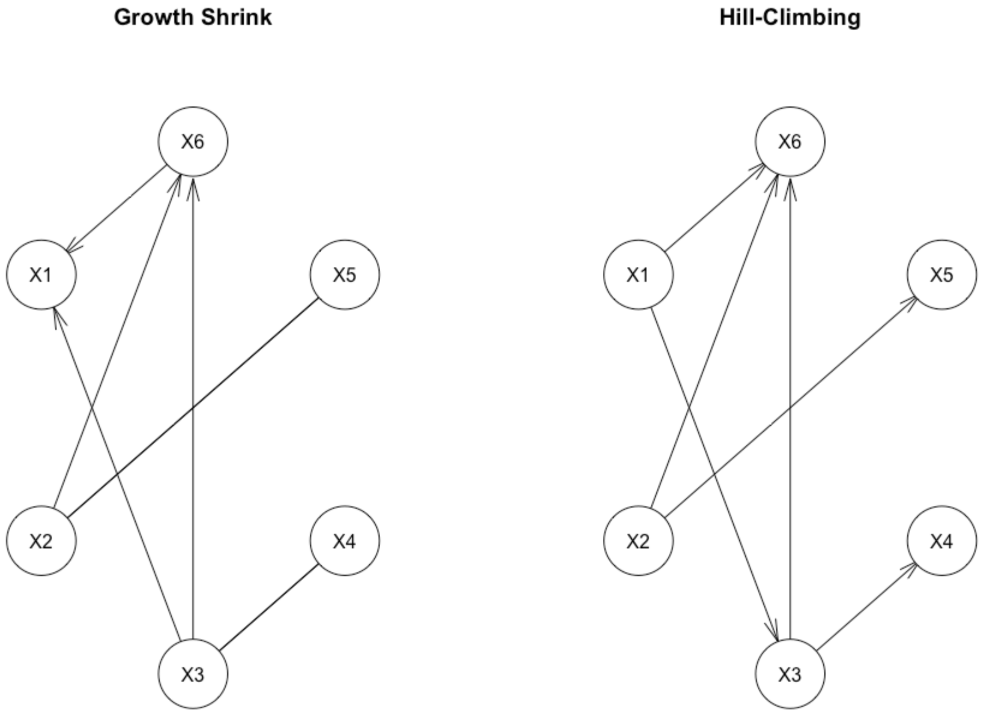


**Fig. 7.** Graph from gs and hc algorithm when sample size of 80

| Algorithm | tp | fp | fn |
|---|---|---|---|
| GS | 2 | 4 | 4 |
| HC | 6 | 0 | 0 |

**Table 2.** Compare GS and HC when sample size 80

Even after increasing the sample size to c(100, 1000, 10000) the GS could not predict the true graph while correctly predicting the skeleton. Basically, it was not able to assign the direction of the edge for $X_1 \rightarrow X_3$, $X_2 \rightarrow X_5$ and $X_3 \rightarrow X_4$ while correctly predicting for the other edges. Thus having tp=3,fp=3 and fn =3. On the other hand, HC algorithm converged for all the sample sizes and predicting true graph.

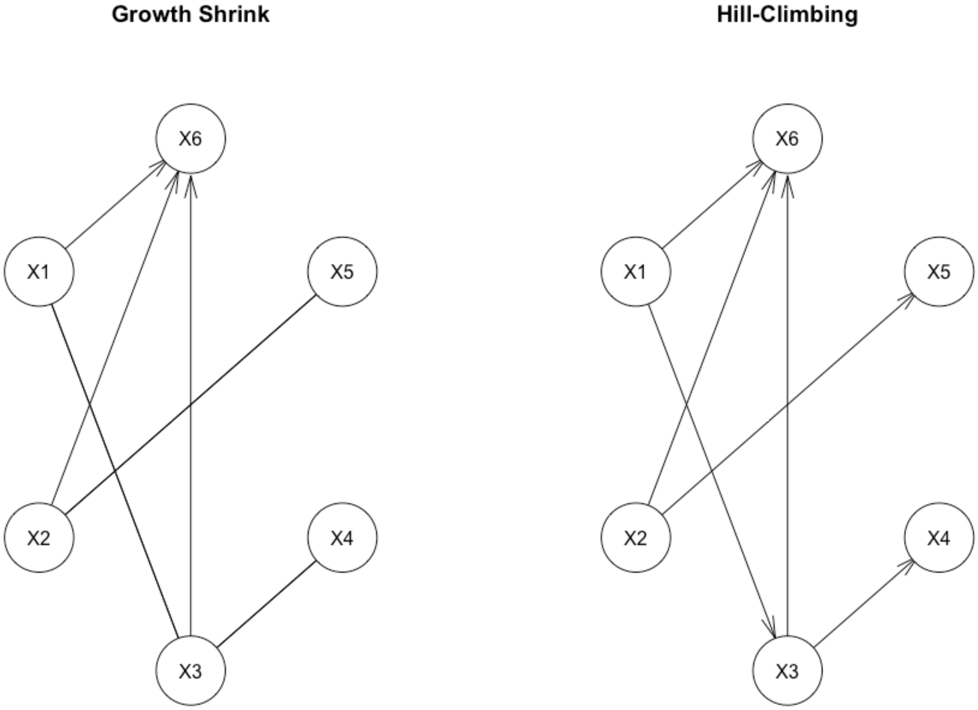


**Fig. 8.** Graph from gs and hc algorithm when sample size of 100/1000/10000

Thus, we can say that HC algorithm is able to predict the simulated graph of the model even with not much amount of simulated data while GS could not predict the target graph even with data amount in terms of $10^5$ while correctly predicting the skeleton though. This can be due to the fact that GS is constraint based algorithm while HC is score based algorithm. Constraint based algorithm works by first finding the skeleton of the graphical model and thus GS could thus correctly produce the skeleton of the graphical model. At the second step, GS algorithm could identify the v-structures and thus could provide direction to those edges $X_1 \to X_6 \leftarrow X_2, X_2 \to X_6 \leftarrow X_3$

.

**2.2 Real Data: asset returns**

**1. Read Sections 1,2 and 4 in Drton and Maathuis (2016). Ideally if you had an implementation of every algorithm in Section 4, which one would you use to estimate the DAG in the asset return data set? Provide a detailed justification of your choice.**

The paper by Drton and Maathuis (2016) provided five different methods to estimate DAG:

- Exact Score-based search
- Greedy Score-based search
- Constraint based methods
- Hybrid Algorithms
- Structural Equation Models With Additional Restrictions

As described each algorithm makes some assumptions to identify DAG. In general, G is not identifiable from the distribution of X, but we can identify its Markov equivalence class, or equivalently, its CPDAG. Thus, many structure learning methods aim to learn the CPDAG instead. We treat exact score-based search in problems of moderate dimensionality and more broadly applicable methods are based on greedy search or conditional independence tests, as well as hybrids of these two approaches. By imposing additional assumptions, we can use SEM models that allow identification of the DAG. We will consider each of these methods and would provide a justification for its feasibility on the asset return dataset.

(Exact) Score-based approaches learn a DAG by determining the graph G that optimizes a specified score Q(G,x) such as BIC. Finding an optimal DAG, or possibly CPDAG, is hard due to the large search space and the acyclicity constraint but we can use different methods such as Dynamic Programming. While the computational and memory requirements are exponential in $\mid V \mid$, this approach is feasible for problems with up to roughly 30 nodes. As for our data, we have only 8 nodes, these methods can be used.

For large graphs, exact search is infeasible, and one can turn to greedy search such as Greedy Equivalence Search (GES). Due to the greedy search, GES will typically not find the global optimum of the score given data x with sample of size n but we can find the global optimum with probability converging to 1 as $n \to \infty$, if the score is decomposable, score-equivalent and consistent. As we have limited amount of data, GES would not be an optimum choice for finding DAG.

Constraint-based methods seek to find a DAG that is compatible with the conditional independencies seen in the given data set. They learn by first finding the skeleton of the graph and then find directions by first finding the v-structures. As we found in previous question, these algorithms can correctly learn the CDPAG but fail to learn the DAG. Thus, using constraint based methods would not be a suggested option for asset return data set.

Hybrid algorithms combine ideas from constraint-based and score- based methods, by employing a greedy search over a restricted space, often determined using conditional independence tests. Hybrid algorithms scale well with respect to the number of variables and exhibit good estimation performance. Nandy et al introduced a new hybrid algorithm, called Adaptively Restricted GES which was proved consistent in classical and high-dimensional settings. Hybrid algorithms would be an ideal choice for finding DAG in the asset return data set.

SEMs can be used to identify the unique DAG by imposing certain assumptions as discussed more in detail in Part 1 Question 5 and 6. In the case of non-Gaussian errors, independent component analysis can identify B up to scaling and permutation of the columns.

Thus, hybrid and exact score based algorithms would be ideal choice for asset return data to learn the DAG.

**2. Use file "Returns250d.txt" to create a data frame with only the 8 assets listed above. Remove the lines with missing values.**

The dataset provided consisted of 5039 observations for 252 variables. We use 'dplyr' package in R to subset particular columns using the 'select' function and then use 'na.omit' to remove missing values. Thus, we get a dataframe of 5037 observations of 8 variables.

```
assetFrame = assetFrame %>% select(AIR.FRANCE.KLM,ALCATEL.LUCENT,
AXA,FAURECIA,GAUMONT, GEODIS, PPR,UNION.FINC.FRANC.)
assetFrame = na.omit(assetFrame)
```

**3. Estimate directed graphs using the gs and hc procedures (Scutari, 2010) and plot their graphs**

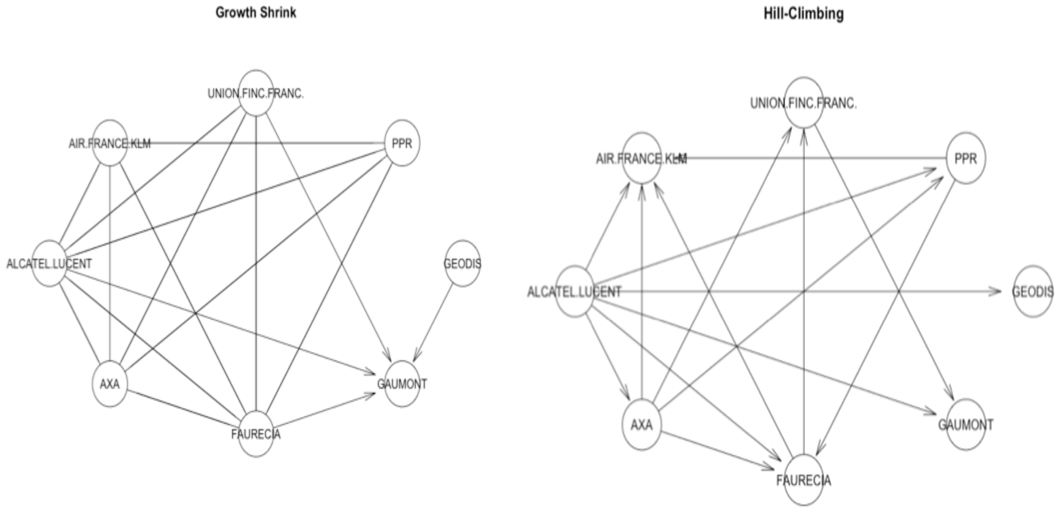


**Fig. 9.** Directed graphs using the gs and hc procedures for asset return dataset

**4. Find a marginal and conditional independence relationships between two variables or set of variables found by gs but not by hc, and conversely. Use another method to determine if these relationships hold or not.**

For questions 4,5 and 6 we can use ci.test method to determine marginal and conditional independence which gives us a p-value to reject the null hypothesis. We will take a threshold of 5 % to make a decision on independence

```
ci.test(x, y, z, data)
```

**For marginal independence:**

GS states that GEODIS and UNION.FINC.FRANC should be independent but HC does not. By the method of ci.test we find that GEODIS and UNION.FINC.FRANC are dependent.

```
        Pearson's Correlation

data:  GEODIS ~ UNION.FINC.FRANC.
cor = 0.030559, df = 5035, p-value = 0.0301
alternative hypothesis: true value is not equal to 0
```

GS states that GEODIS and AIR.FRANCE.KLM should be independent but HC does not. Conversely, we find that GEODIS and AIR.FRANCE.KLM are independent.

HC does not find any marginal independence between two variables.

**For conditional independence:**

HC says that GEODIS and GAUMONT should be independent given ALCATEL.LUCENT while GS doesn't. By ci.test we find GS to be true.

GS says that GEODIS and AIR.FRANCE.KLM should be independent given PPR while HC does not. By ci.test we find GS to be true.

**5. Find a marginal and conditional independence relationships between two variables or set of variables given another set of variables found by both gs and hc Use the method in question 4 to determine if these relationships hold or not.**

**For marginal independence:**

HC does not find any marginal independence between two variables.

**For conditional independence:**

Both say that GEODIS and PPR should be independent given ALCATEL.LUCENT which was found true by ci.test.

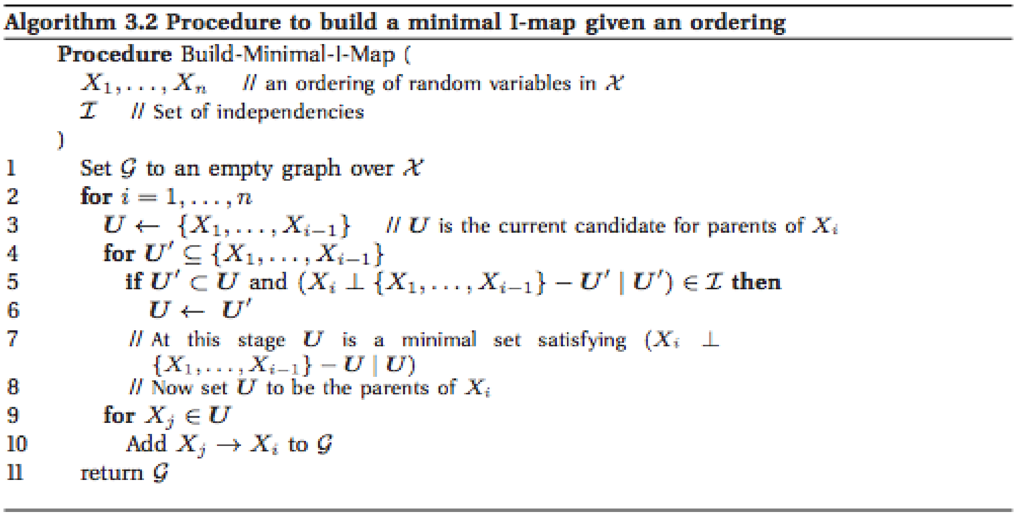
**6. Same question as 5 with marginal/conditional dependence instead of independence.**

Both say ALCATEL.LUCENT and GAUMONT dependent which was found to be true by ci.test.

Both say GEODIS and UNION.FINC.FRANC conditionally dependent given GAUMONT. It can be easily shown that they are conditional dependent by ci.test.

**7. What minimal I-map would you propose?**

A graph K is a minimal I-map for a set of independencies I if it is an I-map for I, and if the removal of even a single edge from K renders it not an I-map. [6] provides an algorithm for finding minimal I-map. As described minimal I-map would not be unique and it would depend on the initial choice of the ordering. We assume we are given a predetermined variable ordering, say, $\{X_1, ..., X_n\}$. We now examine each variable $X_i, i = 1, ..., n$ in turn. For each $X_i$, we pick some minimal subset U of $\{X_1, ..., X_{i-1}\}$ to be $X_i$'s parents in G. More precisely, we require that U satisfy $(X_i \perp \{X_1, ..., X_{i-1}\} - U \mid U)$, and that no node can be removed from U without violating this property. We then set U to be the parents of $X_i$. Fig. 10 is a snippet of the algorithm proposed by [6].



**Algorithm 3.2 Procedure to build a minimal I-map given an ordering**

```
Procedure Build-Minimal-I-Map (
  $X_1, \ldots, X_n$    // an ordering of random variables in $\mathcal{X}$
  $\mathcal{I}$   // Set of independencies
)
  Set $\mathcal{G}$ to an empty graph over $\mathcal{X}$
  for $i = 1, \ldots, n$
    $U \leftarrow \{X_1, \ldots, X_{i-1}\}$   // $U$ is the current candidate for parents of $X_i$
    for $U' \subseteq \{X_1, \ldots, X_{i-1}\}$
      if $U' \subset U$ and $(X_i \perp \{X_1, \ldots, X_{i-1}\} - U' \mid U') \in \mathcal{I}$ then
        $U \leftarrow U'$
      // At this stage $U$ is a minimal set satisfying $(X_i \perp$
        $\{X_1, \ldots, X_{i-1}\} - U \mid U)$
      // Now set $U$ to be the parents of $X_i$
    for $X_j \in U$
      Add $X_j \rightarrow X_i$ to $\mathcal{G}$
  return $\mathcal{G}$
```

**Fig. 10.** Algorithm to find minimal I-map. Taken from [6]

Also, by the Theorem 4.1 from the part above, A DAG H is not an independence map of G0 if and only if:

- $skeleton(G_0) \nsubseteq skeleton(H)$, or
- there exists a triple of nodes $(X_i, X_j, X_k)$ such that $X_i$ and $X_k$ are non-adjacent in $H$, $\pi_H(X_i, X_j, X_k)$ is a non-collider path, and $\pi_{G_0}(X_i, X_j, X_k)$ is a v-structure, or
- there exists a triple of nodes $(X_i, X_j, X_k)$ such that $\pi_H(X_i, X_j, X_k)$ is a v-structure and $X_i \not\perp_{G0} X_k \mid Pa_H(X_k)$, where without loss of generality we assume $X_i \in Nd_H(X_k)$.

Working on the same grounds, we fixed the initial ordering to be (ALCATEL.LUCENT, GEODIS, AIR.FRANCE.KLM, AXA, FAURECIA, UNION.FINC.FRANC, PPR, GAUMONT). Thus, initially we set G to be empty graph. At first step, we add ALCATEL.LUCENT. At the second step, we add GEODIS. From the conditional independence test, we find that GEODIS $\not\perp$ ALCATEL.LUCENT. Thus, we cant remove an edge between GEODIS and ALCATEL.LUCENT and ALCATEL.LUCENT would be the parent of GEODIS. At the next step, we add AIR.FRANCE.KLM. With ALCATEL.LUCENT as choice of parent (U in algorithm) and the conditional independence test, we find that $AIR.FRANCE.KLM \perp GEODIS \mid ALCATEL.LUCENT$ with a p-value of $6\%$ (considering threshold of $5\%$ for independence test). Thus, we can remove the edge between GEODIS and AIR.FRANCE.KLM. Similarly, we follow the whole algorithm to arrive at the I-map as in Fig. 11

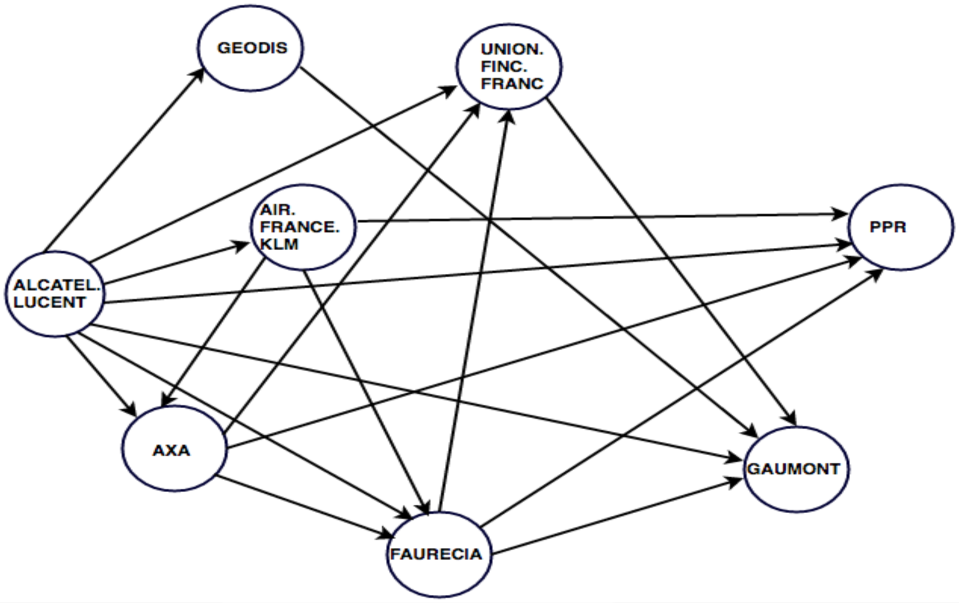


**Fig. 11.** Proposed I-map for an initial choice of ordering.

# References

1. Bottou, Lon, et al. “Counterfactual reasoning and learning systems: the example of computational advertising.” Journal of Machine Learning Research 14.1 (2013): 3207-3260.
2. Chickering, David Maxwell. “Optimal structure identification with greedy search.” Journal of machine learning research 3.Nov (2002): 507-554.
3. Colombo, Diego, and Marloes H. Maathuis. “Order-independent constraint-based causal structure learning.” Journal of Machine Learning Research 15.1 (2014): 3741-3782.
4. Drton, Mathias, and Marloes H. Maathuis.“Structure Learning in Graphical Modeling.” arXiv preprint arXiv:1606.02359 (2016).
5. Kalisch, Markus, and Peter Bühlmann. “Estimating high-dimensional directed acyclic graphs with the PC-algorithm.” Journal of Machine Learning Research 8.Mar (2007): 613-636.
6. Koller, Daphne, and Nir Friedman. “Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series).” (2009).
7. Leray, Philippe, and Olivier Francois. “BNT structure learning package: Documentation and experiments.” Laboratoire PSI, Universit et INSA de Rouen, Tech. Rep (2004).
8. Maathuis, Marloes H., Markus Kalisch, and Peter Bühlmann. “Estimating high-dimensional intervention effects from observational data.” The Annals of Statistics 37.6A (2009): 3133-3164.
9. Meinshausen, Nicolai, and Peter Bühlmann. “Stability selection.” Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72.4 (2010): 417-473.
10. Schwarz, Gideon. “Estimating the dimension of a model.” The annals of statistics 6.2 (1978): 461-464.
11. Schumacker, Randall E., and Richard G. Lomax. “A beginner’s guide to structural equation modeling.” Psychology Press, 2004.
12. Scutari, Marco. “Learning Bayesian networks with the bnlearn R package.” arXiv preprint arXiv:0908.3817 (2009).