university of groningen

faculty of science
and engineering

mathematics and applied
mathematics

Bayesian Networks and analysis with incomplete data

Bachelor's Project Mathematics

July 2020

Student: S.R. Ranft

First supervisor: Prof.dr. M.A. Grzegorczyk

Second assessor: Prof.dr. W.P. Krijnen

Abstract

A statistical pipeline for the analysis of incomplete data is theorised and applied, with missing data entries substituted for using multiple imputation by chained equations (MICE). Bayesian Networks are constructed for the purposes of multivariate analysis, and providing insight into conditional (in)dependencies between variables. Partial correlation is used predominantly, as both an investigative and diagnostic tool. The theories explore the merits of frequentists and Bayesian approaches, with the ensuing application conducted in a Bayesian framework.

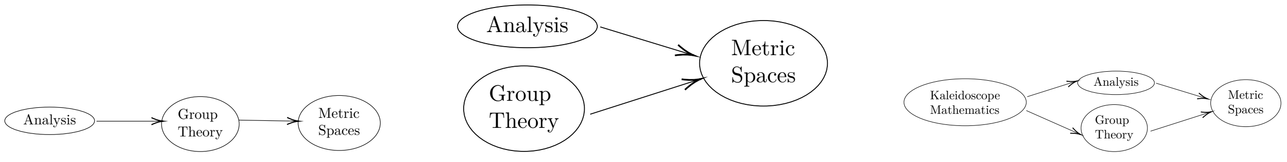
Acknowledgements

A special thank you to Prof.dr. Grzegorczyk for observing my interest in Bayesian statistics and suggesting Bayesian Networks as supplementary analysis tool to multivariate statistics. Additionally for providing me with tutelage and lecture notes on this topic, and aiding my understanding of MICE. Furthermore I extend gratitude towards Prof.df. Krijnen for evaluating me on this project, and for being a continuous source of statistical knowledge.

Contents

Introduction

It is interesting to investigate into the accruement of knowledge of Bachelor Mathematics students at the Rijksuniversiteit Groningen (RuG). By examining the correlation between grades and a Bayesian Network is proposed (a graph) to explain any causal relationships. It can be reasoned that grades are a reflection of knowledge a student retains from the course, and further that higher grades amounts to a higher retention of information. Most Bachelor Mathematics courses can be allocated by topic into subgroups of pure mathematics, statistics, computer science, and physics. For example, as the courses Analysis, Group Theory, and Metric Spaces are pure mathematics courses, it stands to reason that a student who achieves a high grade in one of these courses will do so in the other two courses. As the courses are undertaken in the order above, is there a causal relationship between the three courses, or is there an attributing variable. Figure 1a shows the dependency of these three courses according to their relation in time, and suggests that knowledge of Analysis is crucial in understanding the content of Group Theory, however once Group Theory has been passed the knowledge of Analysis is no longer directly utilised in Metric Spaces. In this case, the knowledge of Analysis is absorbed or incorporated into the knowledge of Group Theory, that a student who passes Group Theory will also pass Metric Spaces, despite having not passed Analysis. In terms of Bayes Rule,



(a) *(b)* *(c)*

Figure 1

$$\mathbb{P}(\text{A, GT, MS}) = \mathbb{P}(\text{A}) \cdot \mathbb{P}(\text{GT} \mid \text{A}) \cdot \mathbb{P}(\text{MS} \mid \text{GT}),$$

where 'A' is Analysis, 'GT' is Group Theory, and 'MS' is Metric Spaces. In Figure 1b, only if a student has passed both Analysis and Group Theory with high grades is it expected for them to pass Metric Spaces also with high grades, and therefore

$$\mathbb{P}(\text{A, GT, MS}) = \underbrace{\mathbb{P}(\text{A}) \cdot \mathbb{P}(\text{GT})}_{\mathbb{P}(\text{A, GT})} \cdot \mathbb{P}(\text{MS} \mid \text{A, GT}).$$

That is, the level of knowledge accrued in Analysis and Group Theory directly influences the grade of Metric Spaces, and the distribution of grades in Analysis is independent of the grades of Group Theory. This suggests that there is no overlap in knowledge between these two courses, however the unification of this knowledge leads to a greater understanding of Metric Spaces. The last example in Figure 1c states that a high degree of knowledge of the course Kaleidoscope Mathematics ('KM') is required to achieve high grades in both Analysis and Group Theory, and that the grade of Metric Spaces depends only on the unification of knowledge of these two courses.

$$\mathbb{P}(\text{KM, A, GT, MS}) = \mathbb{P}(\text{KM}) \cdot \mathbb{P}(\text{A} \mid \text{KM}) \cdot \mathbb{P}(\text{GT} \mid \text{KM}) \cdot \mathbb{P}(\text{MS} \mid \text{A, GT}).$$

In order to construct a causal or predictive Bayesian Network, the data must be complete in that there must be no missing values. These missing values must be treated with careful list-wise deletion or by imputing the missing values using multiple imputation chained equations (MICE) [14]. The classification of missingness is important in the assumptions of the MICE method; the common classifications are missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR). MAR or MCAR classified missing values can be readily imputed as the reason for missingness is independent of the data, e.g. it is not dependent on the variable. For instance, a student who misses an exam due to weather can more accurately impute their expected result from the data than a student who purposely missed the exam due their lack of preparation (individual effects bias). Additionally, there may be a systematic reason for the missingness of a particular variable, e.g. the course may be mandatory only for an unpopular minor and thus only a small subset of students from the sample would have valid data entries. Missingness due to individual or systemic bias falls under the MNAR classification and requires additional modelling assumptions.

Literature Review

Using Bayesian Networks to analyse students' academic performance for predictive purposes is not unique, and has previously been conducted by H. and K. Itoh, and K. Funahashi from the Nagoya Institute of Technology, Japan [6, 7]. The purpose of these studies was to provide bachelor students with academic advice at the start of their second year so that they might alter their learning techniques, improve their academic performance, and prevent students from abandoning their studies. The authors developed a Bayesian network model for forecasting course grades in the second year using the grades of first-year courses, compared the predictive accuracy with a linear predictive model, and deduced the probability of a student requiring academic advice in two different ways: deviation in grade-point-average (GPA) from the first to the second year, and deviation in GPA with regards to specialised second-year courses.

This paper is written with the intent to be understood by any educator (inline with the purpose of the journal), and hence utilises simple language (jargon-less). The report would benefit greatly from a brief introduction into the proposed benefits of a Bayesian framework and outlining the benefits of a posterior updating technique for non-linear and non-normal data. It is not clear how the authors' method could be implemented using other data, as the report only discusses the merits of the method for their data, and thus it can be evaluated as a limitation in itself, as this contradicts the purpose: real-life applicability. This project aims to remedy this by defining the statistical methods in depth with the intent that such analyses and predictions can be replicated on alternate data sets.

The theoretical framework for constructing a Bayesian Network using the GSA is outlined in Koski and Noble's book [9]. Chapters 1–5 of this book summarise (in part) the courses Statistical Reasoning, Stochastic Models, and Statistics, which is assumed knowledge at this point. Chapter 6 is aptly named *learning the graph structure* and provides a clear understanding of performing the GSA, which is referred to as the "K2 structural learning algorithm" in §6.3.3. Additionally, lecture notes and `R` codes from the course Statistical Genomics, which teaches the use of Bayesian Networks and the GSA to provide probabilistic information on gene expression, and will act as a supplementary learning tool to the book.

Numerous online resources exist for understanding how to properly implement MICE, and those which have proved most knowledgeable and immediately instructive are authored by Prof. Dr. Stef van Buuren from the Universiteit Utrecht [1, 14]. Moreover, the vignettes provide educational examples from which broadly applicable methods are eventually drawn [15]. Great emphasis is placed on the validity of these sources due to the authors' expertise on MICE, and in particular that Prof. Dr. van Buuren is the author and maintainer of the package `mice`.

Research question and problem statement

With regards to the previous research [6, 7], this thesis investigates the possibility of prescribing a global statistical pipeline which analyses multivariate data sets for the purposes of constructing causal or predictive Bayesian Networks. In particular, the attention of this problem is focused on the realistic setting of incomplete data sets and if imputing the missing data can decently maintain the underlying conditional (in)dependencies.

The first chapter focuses on the theoretical framework for designing such a pipeline, which can be partitioned into two phases. The first phase defines univariate and multivariate analysis techniques of the data frame, and appropriately utilising the resulting statistics in the construction of the imputation procedure. In the second phase, the mathematical reasoning for developing a Bayesian Network, and associated inferences, are outlined in the context of continuous Gaussian or discrete Multinomial data types.

The subsequent chapter applies the theoretical techniques to a data set of student grades from the Bachelor of Mathematics programme at the Rijksuniversiteit Groningen. The results of MICE and the Bayesian Networks are then discussed, and future applications are contemplated with reference to the effects of the current COVID-19 situation due to the online learning environment.

Theory

With the purpose to ensure the validity of inferences, a solid foundation in the statistical methods utilised in the `R` packages `mice` and `bnlearn` must be established. Additionally, the implementation in `R` of respective measures are described with reference to their suitability for different data types or scenarios.

Multiple Imputation by Chained Equations (MICE)

To rectify the issues of MNAR, `NA` values in a set must be replaced with values which preserve the properties of the (joint) distribution of the original set. Consider an $n \times p$ data frame, Y, then if only one variable of Y, Y_j for some $j \in \{1, \ldots, p\}$, contains `NA` values then univariate imputation methods are required. The complete variables $Y_{-j} = Y \setminus Y_j$ are utilised in the imputation of the incomplete variable Y_j, however if more than one variable is incomplete then multivariate imputations are required. To illustrate and identify the completeness of variables, the influx and outflux are calculated (c.f. §4.1.3 [1]) and displayed using `fluxplot()` from the `mice` package. Complete variables have influx equal to zero and outflux equal to one, and entirely incomplete variables have influx equal to one and outflux equal to zero. The ideal situation for imputation of missing data is for the outflux to be higher than the influx, and for the sum of influx and outflux to be equal to one, rather than less than.

Univariate analysis

The reliability of the imputation depends not only on the predictor matrix, but also on the imputation methods (c.f. Table 1, §3.1 [14]). Most imputation methods depend on the normal distribution, and therefore care must be taken in ensuring correct classification of variable distributions. The Shapiro-Wilk test of normality [10] can be implemented using the `stat.desc()` function from the `pastecs` package, where the assumption of normality is rejected when the p-value (`normtest.p`) is less than a prescribed value, e.g. 0.05. In addition, the function `ggqqplot()` from the package `ggpubr` displays the QQ-plots of each variable which illustrates any deviation from normality, for example bimodal and skewed distributions. If a variable's distribution is significantly different from normal, then it is more appropriate to use predictive mean matching (PMM; `pmm` or `midastouch`). Moreover, PMM methods are suitable for discrete or semi-continuous data as the imputations are drawn from a subset of observed values, and therefore adhere to any rounding or interval-censoring of the data (c.f. §3.7.3 [1]). The imputation method can be implemented globally (to all variables), e.g. `meth = "pmm"`, or individually selected for differing variable types, e.g. for two variables `meth = c("pmm", "norm")`. The pros and cons of different methods are tabulated below.

Method	Description	Type	Pros	Cons
`mean`	Mean imputation	Univariate	Simplicity Unbiased for the mean (MCAR)	Underestimates the variance Biases correlation to zero Biased for the mean (MAR) Alters the distribution
`pmm`	Predictive Mean Matching imputed value is taken from subset of observed values whose predicted value is close to the predicted missing value	Multivariate	Robust to data transformations Implicit model - robust to misspecification Not reliant on normality	
`midastouch`	Midas touch Improved `pmm`	Multivariate	Improvement on `pmm` for small samples	
`norm.predict`	Predictive imputation (Bayesian linear regression) imputation = prediction	Univariate	Unbiased regression estimates (MAR) Good approximation given by R^2	Dependent on Normal distribution Over-inflates correlations Underestimates the variance Harmful to statistical inference
`norm.nob`	Predictive imputation (Non-Bayesian linear regression) imputation = prediction + noise	Univariate	Preserves the original distribution and correlations	Dependent on Normal distribution Symmetric and constant error term restrictive
`norm`	Bayesian Normal linear regression imputation = prediction + noise + parameter uncertainty	Multivariate	Preserves the original distribution and correlations	Dependent on Normal distribution
`norm.boot`	Calculated from bootstrap sample imputation = prediction + noise + parameter uncertainty	Multivariate	Preserves the original distribution and correlations	Dependent on Normal distribution

Table 1

The table includes only continuous variable imputation methods, however there exists imputation methods suitable for discrete variable types, e.g. binary or multinomial (c.f. §3 [1]). A full list of imputation methods available in the `mice` package is given in Table 6.1 from §6.3.1 of [1].

Multivariate analysis

In addition to the choice of univariate imputation method, predictors of imputations are selected to increase accuracy. On the first run of `mice`, if the predictor matrix is not specified then each variable is used in the prediction of every other variable. The predictor matrix is a $p \times p$ binary data matrix where a "1" in cell (i, j) indicates that variable j is used in the prediction of variable i, and a "0" indicates it is not used, for $i, j = 1, \ldots, p$ variables. If variable i is complete, then `mice` silently defines all entries in row i to be zero. The accuracy of imputed values can be increased by carefully selecting predictor variables which are strongly associated by altering the predictor matrix.

Measures of association identify the monotonic relationship between two variables and is dependent on the variable type, the most relevant statistics of association are tabulated below. If two variables have a strong monotonic relationship then there is an equally strong predictive ability of missing values, and consequently should be selected as predictors for each other.

Statistic	Pearson's r	Spearman's ρ	Kendall's τ
Description	Ratio of covariance to product of standard deviations $r = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}}$ measures linear relationship	Pearson's r of ranked data non-parametric (non-linear monotonic relationship) greater suited for tied ranks than τ	non-parametric (ordinal relationship) $\tau = \frac{n_c - n_d}{n_0}$ n_c - number of concordant pairs n_d - number of discordant pairs $n_0 = n(n-1)/2$

Table 2: Correlation statistics computed in `R` using `correlation()` from the `correlation` package, implemented with `meth = "auto"`, `bayesian = TRUE`, `partial_bayesian = TRUE`, `partial = TRUE`, and `robust = TRUE` [12].

These statistics of association are influenced by the underlying multivariate distributions and do not precisely express the unique bivariate relationship. The partial correlation coefficient resolves the influence of other variables in the bivariate relationship by regressing out the effects. The `R` package `correlation` contains the `correlation()` function which gives the partial regression coefficients for all three methods listed in Table 2, although the function allows for seven other main methods [12]. Additionally, this function allows the user to compute zero-order and partial correlations under a Bayesian framework, which is beneficial for lower sample sizes [3].

Partial Correlation

Consider a set of variables X, Y and $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_k)$, then compute the residuals of the regression of X on $\mathbf{Z}$, and the residuals of the regression of Y on $\mathbf{Z}$. The Pearson's correlation coefficient of the pair of residuals $r_{\delta,\varepsilon}$ is the Pearson's partial correlation between X and Y. In the simplest example take $k = 1$, $\delta := X - (\alpha + \beta Z)$ to be the residuals from the regression of X on Z, and $\varepsilon := Y - (\alpha^* + \beta^* Z)$ the residuals from the regression of Y on Z. The slope and intercepts of the regression line are computed as follows:

$$\beta = \frac{\mathrm{Cov}(Z,X)}{\mathrm{Var}(Z)}, \quad \beta^* = \frac{\mathrm{Cov}(Z,Y)}{\mathrm{Var}(Z)}, \quad \alpha = \bar{X} - \beta \bar{Z}, \quad \text{and} \quad \alpha^* = \bar{Y} - \beta^* \bar{Z}.$$

The covariance for the residuals δ and Y is determined as

$$\begin{aligned}
\mathrm{Cov}(\delta, Y) &= \mathrm{Cov}(X - (\alpha + \beta Z), Y) = \mathrm{Cov}(X,Y) - \beta\,\mathrm{Cov}(Z,Y) \\
&= \mathrm{Cov}(X,Y) - \left(\frac{\mathrm{Cov}(Z,X)}{\mathrm{Var}(Z)}\right)\mathrm{Cov}(Z,Y) \\
&= \sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}\left(\frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}} - \frac{\mathrm{Cov}(Z,X)}{\sqrt{\mathrm{Var}(Z)\,\mathrm{Var}(X)}} \cdot \frac{\mathrm{Cov}(Z,Y)}{\sqrt{\mathrm{Var}(Z)\,\mathrm{Var}(Y)}}\right) \\
&= \sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}\,(r_{XY} - r_{XZ} r_{YZ}).
\end{aligned}$$

The covariance for the pair of residuals δ and ε, $\sigma_{\delta\varepsilon}$, is given by the formula

$$\begin{aligned}
\mathrm{Cov}(\delta, \varepsilon) &= \mathrm{Cov}(X - (\alpha + \beta Z), Y - (\alpha^* + \beta^* Z)) \\
&= \mathrm{Cov}(X,Y) - \beta\,\mathrm{Cov}(Z,Y) - \beta^*\,\mathrm{Cov}(Z,X) + \beta\beta^*\,\mathrm{Var}(Z)
\end{aligned}$$

$$
\begin{aligned}
&= \operatorname{Cov}(X,Y) - \left(\frac{\operatorname{Cov}(Z,X)}{\operatorname{Var}(Z)}\right)\operatorname{Cov}(Z,Y) - \left(\frac{\operatorname{Cov}(Z,Y)}{\operatorname{Var}(Z)}\right)\operatorname{Cov}(Z,X) \\
&\qquad + \left(\frac{\operatorname{Cov}(Z,X)\operatorname{Cov}(Z,Y)}{\operatorname{Var}(Z)^2}\right)\operatorname{Var}(Z) \\
&= \operatorname{Cov}(X,Y) - \frac{\operatorname{Cov}(Z,X)\operatorname{Cov}(Z,Y)}{\operatorname{Var}(Z)} \\
&= \sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}\left(r_{XY} - r_{XZ}r_{YZ}\right).
\end{aligned}
$$

The variances of the pair of residuals, σ_δ^2 and σ_ε^2, are calculated using

$$
\begin{aligned}
\operatorname{Var}(\delta) &= \operatorname{Cov}(\delta,\delta) = \operatorname{Cov}(X - (\alpha + \beta Z), X - (\alpha + \beta Z)) \\
&= \operatorname{Var}(X) - 2\beta\operatorname{Cov}(Z,X) + \beta^2\operatorname{Var}(Z) \\
&= \operatorname{Var}(X) - 2\left(\frac{\operatorname{Cov}(Z,X)}{\operatorname{Var}(Z)}\right)\operatorname{Cov}(Z,X) + \left(\frac{\operatorname{Cov}(Z,X)}{\operatorname{Var}(Z)}\right)^2\operatorname{Var}(Z) \\
&= \operatorname{Var}(X) - \frac{\operatorname{Cov}(Z,X)^2}{\operatorname{Var}(Z)} = \operatorname{Var}(X)\left(1 - \frac{\operatorname{Cov}(Z,X)^2}{\operatorname{Var}(Z)\operatorname{Var}(X)}\right) \\
&= \operatorname{Var}(X)\left(1 - r_{XZ}^2\right);
\end{aligned}
$$

$$
\begin{aligned}
\operatorname{Var}(\varepsilon) &= \operatorname{Cov}(\varepsilon,\varepsilon) = \operatorname{Cov}(Y - (\alpha^* + \beta^* Z), Y - (\alpha^* + \beta^* Z)) \\
&= \operatorname{Var}(Y) - 2\beta^*\operatorname{Cov}(Z,Y) - {\beta^*}^2\operatorname{Var}(Z) \\
&= \operatorname{Var}(Y) - 2\left(\frac{\operatorname{Cov}(Z,Y)}{\operatorname{Var}(Z)}\right)\operatorname{Cov}(Z,Y) - \left(\frac{\operatorname{Cov}(Z,Y)}{\operatorname{Var}(Z)}\right)^2\operatorname{Var}(Z) \\
&= \operatorname{Var}(Y) - \frac{\operatorname{Cov}(Z,Y)^2}{\operatorname{Var}(Z)} = \operatorname{Var}(Y)\left(1 - \frac{\operatorname{Cov}(Z,Y)^2}{\operatorname{Var}(Z)\operatorname{Var}(Y)}\right) = \operatorname{Var}(Y)\left(1 - r_{YZ}^2\right).
\end{aligned}
$$

The correlation coefficient is the ratio of the covariance and standard deviations, and thus measures the degree of association between two variables.

$$
\implies r_{\delta\varepsilon} = \frac{\operatorname{Cov}(\delta,\varepsilon)}{\sqrt{\operatorname{Var}(\delta)\operatorname{Var}(\varepsilon)}} = \frac{r_{XY} - r_{XZ}r_{YZ}}{\sqrt{\left(1 - r_{XZ}^2\right)\left(1 - r_{YZ}^2\right)}}.
$$

The formula above is that of the partial correlation between X and Y in the case that $k = 1$, however if $k > 1$ then the formula can be determined.

Significance and effect size of partial correlation

The partial correlations provide information on the strength and direction of the unique relationship between two variables. In the context of multiple regression, the significance of the strength of the unique relationship can be factored into three levels: small, medium and large effect size (c.f. Case 1 §9 [2]). Partial correlations with medium or large effect size are highlighted as potential predictors in imputation.

small	medium	large
$pr^2 = \frac{0.02}{1 + 0.02}$	$pr^2 = \frac{0.15}{1 + 0.15}$	$pr^2 = \frac{0.35}{1 + 0.35}$

Table 3: Effect size interval partitions for determining the significance of partial correlations at three levels: small, medium and large [2].

In order to determine if there is a significant difference in partial correlation coefficients across imputation methods, the statistics are transformed using the Fisher-Z transformation,

$$
\tilde{z}\left[r_{XY.\mathbf{Z}}\right] = \frac{1}{2}\ln\left(\frac{1 - r_{XY.\mathbf{Z}}}{1 + r_{XY.m}}\right),
$$

where $r_{XY.\mathbf{Z}}$ is the partial correlation between variables X and Y (partialling out the effects of variable set $\mathbf{Z}$). The underlying hypothesis that there is a significant difference in the population between two partial regression coefficients is tested using

$$z = \frac{\tilde{z}(1)\left[r_{XY.\mathbf{Z}}\right] - \tilde{z}(2)\left[r_{XY.\mathbf{Z}}\right]}{\sqrt{\frac{1}{n_1-4} + \frac{1}{n_2-4}}} \sim \mathcal{N}(0,1),$$

where $\tilde{z}(1)\left[r_{XY.\mathbf{Z}}\right]$ is the transformed statistic for population one with sample size n_1, and $\tilde{z}(2)\left[r_{XY.\mathbf{Z}}\right]$ is the transformed statistic for population two with sample size n_2 [11]. The hypothesis $\rho(1)_{XY.\mathbf{Z}} = \rho(2)_{XY.\mathbf{Z}}$ is rejected in favour of $\rho(1)_{XY.\mathbf{Z}} \neq \rho(2)_{XY.\mathbf{Z}}$ if the sample statistic computed above exceeds $z^*_{\alpha/2}$, where $\mathbb{P}\left(|Z| > z^*_{\alpha/2}\,|Z \sim \mathcal{N}(0,1)\right) = \alpha$ is the accepted probability of a Type I error.

In order to test which imputation methods from a set of k are most suitable for a particular variable, pairwise comparisons are required and therefore comparisons are made on the association confidence intervals. The following represents the $(1-\alpha)\%$ confidence interval for the difference in the Fisher Z-transformed population correlations denoted $\tilde{z}(j)\left[\rho_{XY.\mathbf{Z}}\right] - \tilde{z}(j')\left[\rho_{XY.\mathbf{Z}}\right]$, for some $j \neq j' \in \{1,\ldots,k\}$:

$$\tilde{z}(j)\left[r_{XY.\mathbf{Z}}\right] - \tilde{z}(j')\left[r_{XY.\mathbf{Z}}\right] \pm \sqrt{\chi^2_{k-1}\left(1-\alpha\right)}\sqrt{\frac{1}{n_j-4} + \frac{1}{n_k-4}},$$

where $\chi^2_{k-1}\left(1-\alpha\right)$ is the $100\left(1-\alpha\right)$ percentage point of a central chi-square distribution with $(k-1)$ degrees of freedom. The upper bound UB and lower bound LB of the confidence interval for the transformed statistic are computed using the equation above. The inverse transformation yields the $(1-\alpha)\%$ confidence interval for the difference in the population correlations denoted $\rho(j)_{XY.\mathbf{Z}} - \rho(j)_{XY.\mathbf{Z}}$:

$$\left(\frac{1-e^{2\cdot LB}}{1+e^{2\cdot LB}}, \frac{1-e^{2\cdot UB}}{1+e^{2\cdot UB}}\right).$$

Bayes Factor

The aforementioned methods for determining significance and effect size are not infallible as they assume that the sample size of the data is "large enough" and that the Central Limit Theorem ensures the validity. Furthermore, many assumptions of frequentist methods assert an underlying normal distribution of the data, which is generally not true. For non-normal sample data with a small sample size, Bayesian methods for analysing (partial) correlation are more appropriate.

Consider a set $\{X, Y, \mathbf{Z}\}$ of (non-normal) variables, then partial correlation can be treated as a parameter of the joint distribution of all variables. A posterior probability model for the partial correlation, now treated as a random variable dependent on some prior distribution with certain hyperparameters and also dependent on the likelihood of the data. The null hypothesis H_0 assumes that the partial correlation between X and Y (with the effects of variable set $\mathbf{Z}$ removed) in the population is zero; the alternative H_1 is that the variables have a population partial correlation significantly different from zero. The null hypothesis is tested using Bayes' Factor, which is defined as

$$BF_{10} := \frac{\mathbb{P}\left(\text{data}\,|H_1\right)}{\mathbb{P}\left(\text{data}\,|H_0\right)} = \frac{\mathbb{P}\left(H_1\,|\text{data}\right)\mathbb{P}\left(H_0\right)}{\mathbb{P}\left(H_0\,|\text{data}\right)\mathbb{P}\left(H_1\right)}.$$

A Bayes' Factor greater than one indicates a greater likelihood of the data given the alternative hypothesis and implies that the null hypothesis should be rejected in favour of the alternative (c.f. §1.7 [9]). This test is implemented in `R` using the `correlation` function from the package `correlation`, with options `method = "auto"`, `partial = TRUE`, `bayesian = TRUE`, `partial_bayesian = TRUE`, and `robust = TRUE` [12]. The `correlation` function supports the use of Pearson's, Spearman's and Kendall's methods, as well as additional types which are listed on `CRAN`. The option `robust = TRUE` rank-transforms the data prior to any computation and when used in conjunction with `method = "auto"` ensures the correct method selection for any data type. The `correlation` function allows two other methods of testing: probability of direction (p-direction) and region of practical equivalence (ROPE). The p-direction test is most similar to the frequentists approach of hypothesis testing except that 89% is a more stable acceptance region than 95%.

Bayesian Network

Bayesian Networks illustrate joint (conditional) probability models among variables in a given data sample. The variables represent nodes in the network, with directed edges between the nodes displaying conditional dependencies between the variables. The potential number of directed edges in a graph increases exponentially as the number of nodes increases, and therefore it is useful to determine the likelihood of a particular graph given the sample data input. The likelihood of the graph, $\mathbb{P}(\text{graph} \mid \text{data})$, is computed using Bayes' Rule as the posterior probability as is given below. The marginal likelihood, $\mathbb{P}(\text{data} \mid \text{graph})$, updates the prior belief, $\mathbb{P}(\text{graph})$, to reflect the true probability.

$$\overbrace{\mathbb{P}(\text{graph} \mid \text{data})}^{\text{posterior}} = \frac{\mathbb{P}(\text{data, graph})}{\mathbb{P}(\text{data})} = \frac{\overbrace{\mathbb{P}(\text{data} \mid \text{graph})}^{\text{marginal likelihood}} \cdot \overbrace{\mathbb{P}(\text{graph})}^{\text{prior}}}{\mathbb{P}(\text{data})}$$

A graph of nodes connected by directed edges is a directed acyclic graph (DAG), and if some edges are undirected then it is called a completed partially directed acyclic graph (CPDAG). Denote graph G in the set of all possible graphs $\mathcal{G}$, and data set D where $D_{j,i}$ is the jth observation of node i (c.f. §2 [9]). If we consider each graph to be equally likely then the prior belief is $\mathbb{P}(\text{graph}) = 1/|\mathcal{G}|$ where $|\mathcal{G}|$ is the total number of graphs (c.f. §6 [9]). Noting that

$$\mathbb{P}(\text{data}) := \sum_{G \in \mathcal{G}} \mathbb{P}(D, G) = \sum_{G \in \mathcal{G}} \mathbb{P}(D \mid G) \cdot \mathbb{P}(G),$$

i.e. the distribution of the data is independent of the graphs such that $\sum_{G \in \mathcal{G}} \mathbb{P}(D \mid G) = 1$, then it holds that

$$\mathbb{P}(\text{graph} \mid \text{data}) \propto \mathbb{P}(\text{data} \mid \text{graph}) = \int \mathbb{P}(D \mid \mathbf{q}, G) \cdot \mathbb{P}(\mathbf{q} \mid G) \, \mathrm{d}\mathbf{q},$$

where $\mathbf{q}$ is a vector of unknown parameters. It is assumed that the parameters q_i in $\mathbf{q}$ are independent such that $\mathbb{P}(\mathbf{q} \mid G) = \prod_{i=1}^{n} \mathbb{P}(q_i \mid G)$, then parameters depend only on the current node X_i and it's parent nodes $\text{pa}(X_i)$.

$$\begin{aligned}
\implies \mathbb{P}(\mathbf{q} \mid G) &= \prod_{i=1}^{n} \mathbb{P}(q_i \mid X_i, \text{pa}(X_i)) \\
\implies \mathbb{P}(D \mid G) &= \int \cdots \int \prod_{i=1}^{n} \left[\prod_{j=1}^{m} \mathbb{P}\left(D_{j,i} = X_i \mid \text{pa}(X_i) = D_{j,\text{pa}(X_i)}, q_i\right) \cdot \mathbb{P}(q_i \mid X_i, \text{pa}(X_i)) \right] \mathrm{d}q_1 \ldots \mathrm{d}q_n \\
&= \prod_{i=1}^{n} \underbrace{\int \prod_{j=1}^{m} \mathbb{P}\left(D_{j,i} = X_i \mid \text{pa}(X_i) = D_{j,\text{pa}(X_i)}, q_i\right) \cdot \mathbb{P}(q_i \mid X_i, \text{pa}(X_i)) \, \mathrm{d}q_i}_{\Psi_i\left(\text{pa}(X_i), D^{\{X_i\}}, D^{\{\text{pa}(X_i)\}}\right)}.
\end{aligned}$$

This leads to the assertion that the posterior distribution is also dependent only on the current node X_i and it's parent nodes $\text{pa}(X_i)$.

$$\implies \mathbb{P}(\text{graph} \mid \text{data}) \propto \prod_{i=1}^{n} \Psi_i\left(\text{pa}(X_i), D^{\{X_i\}}, D^{\{\text{pa}(X_i)\}}\right).$$

The following sections outline the approach to Bayesian Network building algorithms under two different model assumptions: Bayesian Gaussian and Bayesian Dirichlet models.

Bayesian Gaussian (Normal-Wishart) model

For the Bayesian Gaussian model, two model assumptions are imposed on the domain variables X_i, such that $X_1, \ldots X_n \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ are normally distributed with a vector of means $\boldsymbol{\mu}$ and covariance matrix $\Sigma = W^{-1}$, where W is the precision matrix. Moreover the unknown parameters, $\boldsymbol{\mu}$ and W, are distributed Normal-Wishart, such that $\boldsymbol{\mu} \sim \mathcal{N}\left(\boldsymbol{\mu}_0, (\boldsymbol{\nu} W)^{-1}\right)$ and $W \sim \mathcal{W}(\alpha, T_0)$, where $\alpha > n + 1$ is the degrees of freedom. The respective distributions are given below, dependent on the hyperparameters $\boldsymbol{\mu}_0$, $\boldsymbol{\nu}$, α, and T_0.

$$f(\boldsymbol{\mu}, W \mid \boldsymbol{\mu}_0, \nu, \alpha, T_0) = \mathcal{N}\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}_0, (\boldsymbol{\nu} W)^{-1}\right) \cdot \mathcal{W}(W \mid \alpha, T_0);$$

$$f\left(\boldsymbol{\mu} \mid \boldsymbol{\mu}_0, (\nu W)^{-1}\right) = (2\pi)^{-n/2} (\nu W)^{1/2} \exp\left\{ \frac{-(\boldsymbol{\mu} - \boldsymbol{\mu}_0)^T \nu W (\boldsymbol{\mu} - \boldsymbol{\mu}_0)}{2} \right\};$$

$$f\left(W \mid \alpha, T_0\right) = c(n,\alpha)\left|T_0\right|^{-\alpha/2}\left|W\right|^{(\alpha-n-1)/2}\exp\left\{\frac{-\operatorname{tr} T_0 \cdot W}{2}\right\},$$
$$\text{where}\quad c(n,a) := \left[2^{\alpha n/2}\pi^{n(n-1)/4}\prod_{i=1}^{n}\Gamma\left(\frac{\alpha+1-i}{2}\right)\right]^{-1}.$$

As T_0 and W are matrices, the operator $|\cdot|$ on these matrices is the determinant operator. Let G_c denote the complete DAG, where each pair of nodes are connected such that all pairs of variables are stochastically independent, there are no unconditional dependency relations, and there is maximal number of edges $n(n-1)/2$. Then the marginal likelihood of the data set D given the complete DAG G_c is

$$\mathbb{P}\left(D \mid G_c\right) = (2\pi)^{-nm/2}\left(\frac{\nu}{\nu+m}\right)^{n/2}\frac{c(n,\alpha)}{c(n,\alpha+m)}\left|T_0\right|^{\alpha/2}\left|T_m\right|^{-(\alpha+m)/2},$$
$$\text{where}\quad T_m := T_0 + \sum_{j=1}^{m}\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)\left(\mathbf{D}_j - \bar{\mathbf{D}}\right)^T + \left(\frac{\nu m}{\nu+m}\right)\left(\boldsymbol{\mu}_0 - \bar{\mathbf{D}}\right)\left(\boldsymbol{\mu}_0 - \bar{\mathbf{D}}\right)^T.$$

$\mathbf{D}_j$ is the jth column of D, and $\bar{\mathbf{D}} = \left(\bar{D}_1, \ldots, \bar{D}_n\right)$ is a vector of means, where $\bar{D}_i = \sum_{j=1}^{m} D_{i,j}/m$. The marginal likelihood is now denoted the Bayesian Gaussian equivalence (BGe) score:

$$\mathbb{P}_{\text{BGe}}\left(D \mid G_c\right) = \prod_{i=1}^{n}\frac{\mathbb{P}\left(D^{\{X_i,\text{pa}(X_i)\}} \mid G_c\right)}{\mathbb{P}\left(D^{\{\text{pa}(X_i)\}} \mid G_c\right)}.$$

$$\implies \mathbb{P}\left(\text{graph} \mid \text{data}\right) \propto \mathbb{P}_{\text{BGe}}\left(D \mid G_c\right).$$

The above relation is called the BGe score and is only valid under Bayesian Gaussian model assumptions; the next section describes a scoring metric similar to the BGe under Bayesian Dirichlet model conditions.

Bayesian Dirichlet (Dirichlet-Multinomial) model

For the Bayesian Dirichlet model, different model assumptions are imposed and the marginal distribution is dependent on the Dirichlet and Multinomial distributions. Given the value combinations j of its parent nodes in $\{\text{pa}\left(X_i\right)\}$, each domain variable X_i is now Multinomial distributed $\mathcal{M}\left(\theta_{i,j,1}, \ldots, \theta_{i,j,r}\right)$ such that $\sum_{k=1}^{r}\theta_{i,j,k} = 1$, and parameters $\theta_{i,j,1}, \ldots, \theta_{i,j,r}$ are Dirichlet distributed with hyperparameters $\alpha_{i,j,1} > 0, \ldots, \alpha_{i,j,r} > 0$. The joint distribution of the parameters $\theta_{i,j,k}$ is

$$\mathbb{P}\left(\theta_{i,j,1}, \ldots, \theta_{i,j,r}\right) = \frac{\Gamma\left(\sum_{k=1}^{r_i}\alpha_{i,j,k}\right)}{\sum_{k=1}^{r_i}\Gamma\left(\alpha_{i,j,k}\right)} \cdot \prod_{k=1}^{r_i}\theta_{i,j,k}^{\alpha_{i,j,k}-1}.$$

$$\implies \mathbb{P}\left(D \mid G\right) = \int \cdots \int \prod_{i=1}^{n}\left[\underbrace{\mathbb{P}\left(q_i \mid \text{pa}\left(X_i\right)\right)}_{\text{Dirichlet}} \cdot \prod_{j=1}^{m}\underbrace{\mathbb{P}\left(X_i = D_{i,j} \mid \text{pa}\left(X_i\right) = D_{\text{pa}(X_i),j}, q_i\right)}_{\text{Multinomial}}\right]\mathrm{d}q_1 \ldots \mathrm{d}q_n$$
$$= \cdots\cdots\cdots$$
$$= \prod_{i=1}^{n}\Psi_i\left(\text{pa}\left(X_i\right), D^{\{X_i\}}, D^{\{\text{pa}(X_i)\}}\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma\left(\alpha_{i,j}\right)}{\Gamma\left(N_{i,j}+\alpha_{i,j}\right)} \cdot \prod_{k=1}^{r_i}\frac{\Gamma\left(N_{i,j,k}+\alpha_{i,j,k}\right)}{\Gamma\left(\alpha_{i,j,k}\right)}.$$

The nodes are indexed $i \in \{1, \ldots, n\}$, the value combination of $\text{pa}\left(X_i\right)$ are indexed $j \in \{1, \ldots, q_i\}$, the possible realisations of X_i are indexed $k \in \{1, \ldots, r_i\}$, and the number of observations in D for which $X_i = k$ are parent nodes take the jth value combination is denoted $N_{i,j,k}$. The psuedocounts are calculated as $\alpha_{i,j,k} = \alpha/q_i r_i$, where α is given, and $\alpha_{i,j} = \sum_{k=1}^{r_i}\alpha_{i,j,k}$. The marginal likelihood is now denoted the Bayesian Dirichlet equivalence (BDe) score:

$$\mathbb{P}_{\text{BDe}}\left(D \mid G\right) = \underbrace{\mathbb{P}\left(\text{graph}\right)}_{\text{Uniform}} \cdot \int \mathbb{P}\left(D, \theta\left(G\right) \mid G\right)\mathrm{d}\theta\left(G\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma\left(\alpha_{i,j}\right)}{\Gamma\left(N_{i,j}+\alpha_{i,j}\right)} \cdot \prod_{k=1}^{r_i}\frac{\Gamma\left(N_{i,j,k}+\alpha_{i,j,k}\right)}{\Gamma\left(\alpha_{i,j,k}\right)}$$

$$\implies \mathbb{P}\left(\text{graph} \mid \text{data}\right) \propto \mathbb{P}_{\text{BDe}}\left(D \mid G\right).$$

In the following section an algorithm is described which determines the optimal graph structure using the BDe/BGe score by iteratively selecting the graph which maximise this score.

Greedy Search Algorithm (GSA)

Complex graphs tend to have greater likelihoods, but only for a particular parameter, and are therefore penalised as this leads to lower marginal likelihoods. The Greedy Search Algorithm (GSA) reduces the number of $G \in \mathcal{G}$ which are analysed by choosing only those which are neighbouring the starting graph. This is commonly referred to as the maximum minimum hill climbing (MMHC) algorithm and is implemented in `R` using the `hc()` function from the package `bnlearn` (c.f. §6.3.4 [9]; §4.1 [13]; [5]).

Initiate: Start from an arbitrary graph $G \in \mathcal{G}$. Set $G_1 = G$.

Iterate: For $i \in \mathcal{N}$, determine all $N = N(G_i)$ neighbour graphs $G_{i,1}, \ldots, G_{i,N}$ of G_i and compute their scores: $\text{Score}(G_{i,k}) = \mathbb{P}(D \mid G_{i,k}) \cdot \mathbb{P}(G_{i,k})$.

if $\text{Score}(G_i) \geq \text{Score}(G_{i,k})$ for $k \in \{1, \ldots, N\}$, then output G_i.

else Set $G_{i+1} = G_{i,c}$, where $\text{Score}(G_{i,c}) = \max_k \{\text{Score}(G_{i,k})\}$.

To avoid getting stuck in local optima, initiate from different graphs. An alternative to assuming that each graph is equally likely, i.e. $\mathbb{P}(G) = 1/|\mathcal{G}|$, is

$$\mathbb{P}(G) = Z^{-n} \prod_{i=1}^{n} \binom{n-1}{|\text{pa}(X_i)|}^{-1}, \qquad \text{where} \quad Z := \sum_{j=0}^{n-1} \binom{n-1}{j}^{-1}.$$

$$\implies \text{Score}(G_i) \propto \prod_{n}^{j=1} \Psi_j \left(\text{pa}(X_j \mid G_i), D^{\{X_j\}}, D^{\{\text{pa}(X_j)\}}\right) \binom{n-1}{|\text{pa}(X_j)|}^{-1},$$

where Ψ_j is the Normal-Wishart (BGe) or Multinomial-Dirichlet (BDe). Other conditions imposed to limit $|\mathcal{G}|$:

- Limit the number of parent nodes such that nodes with more than the set threshold has prior probability of zero.
- If all "best graphs" contain a particular edge, initiate from a graph containing this edge and set any graph without this edge to have prior probability zero.
- Similarly, if all "best graphs" do not contain a particular edge, initiate from a graph without this edge and set any graph with this edge to have prior probability zero.

As the correlation coefficient no longer gives direction (only strength of the relationship), we can regulate nodes to determine causation. For example, for a particular set of nodes $\{A, B\}$ prescribe high and low levels (each). If we set A to low levels (inhibit A), and the scatter plot of A v.s. B shows a cluster, then $A \to B$, i.e. $\mathbb{P}(A, B) = \mathbb{P}(A)\mathbb{P}(B \mid A)$. If the scatter plot does not resemble a cluster and instead resembles a line running along the axis of B, then $B \to A$, i.e. $\mathbb{P}(A, B) = \mathbb{P}(B)\mathbb{P}(A \mid B)$. Apply an intervention vector $\mathbf{V} \in \mathbb{R}^m$, where V_j is non-zero if an intervention is applied at observation j. Then $D(i)$ is the matrix D without the columns corresponding to the inventions affecting X_i, and

$$\mathbb{P}_{\text{BGe}}(D \mid G) = \prod_{i=1}^{n} \frac{\mathbb{P}\left(D(i)^{\{X_i, \text{pa}(X_i)\}} \mid G_c\right)}{\mathbb{P}\left(D(i)^{\{\text{pa}(X_i)\}} \mid G_c\right)},$$

$$\text{where} \qquad \mathbb{P}\left(D(i)^{\{S\}} \mid G_c\right) = (2\pi)^{-n_S m(i)/2} \left(\frac{\nu}{\nu + m(i)}\right)^{n_S/2} \frac{c(n_S, \alpha)}{c(n_S, \alpha + m(i))} \left|T_0^{\{S\}}\right|^{\alpha/2} \left|T_{m(i),i}^{\{S\}}\right|^{-(\alpha + m(i))/2},$$

$$T_{m(i),i}^{\{S\}} = T_0^{\{S\}} + \sum_{j=1}^{m(i)} \left(\mathbf{D}_j(i) - \bar{\mathbf{D}}(\mathbf{i})\right)\left(\mathbf{D}_j(i) - \bar{\mathbf{D}}(\mathbf{i})\right)^T + \left(\frac{\nu m(i)}{\nu + m(i)}\right)\left(\boldsymbol{\mu}_0 - \bar{\mathbf{D}}(\mathbf{i})\right)\left(\boldsymbol{\mu}_0 - \bar{\mathbf{D}}(\mathbf{i})\right)^T,$$

$$\text{and} \qquad \bar{D}_i(i) = \frac{\sum_{j=1}^{m(i)} D(i)_{i,j}}{m(i)}.$$

Each node which has been intervened must obtain two "dummy parent nodes", then use the following algorithm to transform the DAG to a CPDAG, and then remove the dummy variables/edges. This yields the CPDAG of the intervened network. Interventions break equivalence classes, and all edges that touch intervened nodes become compelled.

The following algorithm transforms a DAG into a CPDAG.

order nodes: Provide a topological ordering of the nodes, such that a node is proceeded by it's parent nodes in the ordering, and any tie-breakers are broken by numerical/alphabetical sorting of the tied nodes.

order edges: for $i = 0, 1, \dots, K$, where K is the total number of edges. The lowest ordered node with an unordered edge incident into it is denoted X. The highest order node with edge incident into X is denoted Y. Then edge $Y \to X$ is ordered i.

label edges: After ordering the edges, proceed by labelling each (in order) edge as either "compelled" or "reversible". Denote $x \to y$ the lowest order edge without a label.

```
for  all edges $w \to x$ labelled "compelled"
     if   $w$ is not a parent of $y$, label $x \to y$ and every edge incident into $y$ "compelled".
     else label $w \to y$ "compelled".
if   there exists an edge $z \to y$ with $z \neq x$ and $z$ is not a parent of $x$, then label $x \to y$ and all unlabelled edges into y "compelled".
else label $x \to y$ and all unlabelled edges into y "reversible".
```

The GSA is useful if and only if D is a "large" data set (with respect to sample size), otherwise a "model averaging" approach is more appropriate.

Model averaging

The model averaging approach initiates from a given DAG G_i, then $I(G_i) = 1$ if G_i contains a particular directed edge and zero if it doesn't. In a CPDAG, if an edge is bidirectional/undirected then also $I(G_i) = 1$.

$$\mathbb{P}(A \to B \mid D) := \sum_{G \in \mathcal{G}} \mathbb{P}(G \mid D) I(G); \qquad I(G) := \begin{cases} 1, & A \to B \text{ or } A \leftrightarrow B \\ 0, & B \to A \end{cases}$$

$$\implies \text{ estimator } \widehat{\mathbb{P}}(A \to B \mid D) := \frac{\sum_{i=1}^{T} I(G_i)}{T} \xrightarrow[\text{consistency}]{T \to \infty} \mathbb{P}(A \to B \mid D)$$

The Markov Chain Monte Carlo (MCMC) sampling technique is used to generate the graphs $G_1, \dots, G_T$ for some $T \in \mathbb{N}$ (c.f. §6.4 [9]; lecture notes [5]). A Markov Chain (MC) has 1-step memorylessness, that is the probability of realisation at time t depends only on the realisation at time $(t-1)$ and is independent of the realisations at times $\{1, \dots, t-2\}$, such that

$$\mathbb{P}(X_t = x_t \mid X_{t-1} = x_{t-1}, \dots, X_1 = x_1) = \mathbb{P}(X_t = x_t \mid X_{t-1} = x_{t-1}).$$

The MC is called homogeneous if there exists a transition matrix T (transition kernel), such that

$$T_{i,j} = \mathbb{P}(X_t = j \mid X_{t-1} = i) = \mathbb{P}(X_{t-1} = j \mid X_{t-2} = i) = \dots = \mathbb{P}(X_2 = j \mid X_1 = i).$$

T is called stochastic if the row sums or column sums total to one, and doubly stochastic if both the row and column sums total to one. The initial probabilities $\{\mathbb{P}(X_1 = i) \mid i \in S\}$ together with the transition matrix T fully define the distribution of the homogeneous MC, where S is the state space. Consider for some $j \in S$ the following:

$$\begin{aligned}
\mathbb{P}(X_2 = j) &= \sum_{i \in S} \mathbb{P}(X_2 = j \mid X_1 = i) \cdot \mathbb{P}(X_1 = i) = \sum_{i \in S} T_{i,j} \cdot \mathbb{P}(X_1 = i) \\
\mathbb{P}(X_3 = j) &= \sum_{k \in S} \mathbb{P}(X_3 = j \mid X_2 = k) \cdot \mathbb{P}(X_2 = k) = \sum_{k \in S} T_{k,j} \cdot \sum_{i \in S} T_{i,j} \cdot \mathbb{P}(X_1 = i) = \sum_{i \in S} T_{i,j}^2 \cdot \mathbb{P}(X_1 = i) \\
&\vdots \\
\implies \mathbb{P}(X_t = j) &= \sum_{i \in S} T_{i,j}^t \, \mathbb{P}(X_1 = i) \xrightarrow{t \to \infty} \lim_{t \to \infty} \mathbb{P}(X_t = j) =: \pi_j, \qquad j \in S.
\end{aligned}$$

The stationary distribution $\boldsymbol{\pi} := (\pi_1, \dots, \pi_k)$, i.e. the limiting probability distributions of one-move transitions, can be determined by solving $\boldsymbol{\pi} T = \boldsymbol{\pi}$ or equivalently $\mathcal{N}(\boldsymbol{\pi}(T - \mathbb{I}))$, for some discrete state space $S = \{1 \dots, k\}$. In the context of Bayesian Networks, we have that $\pi_i = \mathbb{P}(G_i \mid D)$, where the G_i are generated by the following MCMC algorithm.

Initialise: Start with an arbitrary DAG G; set $G_1 = G$.

Iterate: For $t = 1, 2, 3, \dots, T$, a new DAG G^* is proposed with probability $Q(G, G^*) = 1/|N(G)|$ if $G^* \in N(G)$ or zero otherwise, where $N(G)$ is the set of neighbour graphs (reachable from G with 1-edge moves).

Then $Q(G, G^*) > 0 \iff Q(G^*, G) > 0$.

The new DAG G^* is accepted with probability $A(G, G^*)$, which are determined in order to satisfy the equation of detailed balance:

$$\frac{T(G, G^*)}{T(G^*, G)} = \frac{\mathbb{P}(G^* \mid D)}{\mathbb{P}(G \mid D)} \implies A(G, G^*) = \min\left\{1, \frac{\mathbb{P}(G^* \mid D)}{\mathbb{P}(G \mid D)} \cdot \frac{|N(G)|}{|N(G^*)|}\right\}.$$

Draw $p \in \mathcal{U}\mathrm{ni}(0,1)$ and accept G^* if $p \le A(G, G^*)$, then $G_2 = G^*$; reject if $p > A(G, G^*)$, then $G_2 = G$.

Burn-in phase: Generate $G_1, \dots, G_T$ and discard the first 10 000 or so. This is done in order to reach the stationary distribution $\boldsymbol{\pi}$.

Thinning: As neighbour graphs are too similar (auto-correlation), select only every nth realisation, e.g. $n = 10, 100, 1000$.

This method is implemented with the `gs()` function in the package `bnlearn`, and visualisations of the graphs generated by `gs()` or `hc()` can be implemented in `R` using the packages `Rgraphviz` and `lattice`, which utilises the `strength.plot()` function to illustrate arc direction and strength (with respect to conditional dependency).[1] The following section demonstrates the use of both functions.

[1] These functions have been programmed to an online webtool which allows the user to demonstrate the different methods for sample or user-uploaded data [4].

Application

Method

Inputting the data in `R` into a data frame requires some knowledge of the `merge` function, which merges columns by a reference column containing the person identifiers and omits rows which are not common between the merging data frames. After merging the data frames, omitted rows are included by inserting `NA`s in the respective columns to ensure the included row has the correct dimension. Furthermore any rows with only `NA`s are removed, as these represent students who enrolled for the course but did not sit any exams and are redundant in the analysis. This constitutes the master data frame, containing all exam grades (first and resit) and from all available years.

There is evidence in the data of several students attempting the exam of any particular course over many years. Therefore the yearly exam grade data sets contain unwanted error noise due to correlation, and we can represent the grades as a fixed or random effects model:

$$y_{it} = X_{it}\beta + \alpha_i + u_{it},$$
$$\text{students} \quad i = 1, \ldots, n, \qquad \text{time periods} \quad t = 1, \ldots, T,$$

where y_{it} is the grade of student i at time t for some course, X_{it} is the row-vector of previously recorded course grades, β is the column vector of parameters, α_i is the unobserved individual effects parameter, and u_{it} is the error term. Consequently, the data is split by starting year cohorts into cohorts 2016-17 and 2017-18, and only the maximum grade over all years is considered for the analysis to reduce the correlation due to students taking the exam, for any course, more than once. The split into cohorts is achieved by recording the year for which each student number first appears as the respective starting year in that a student whose first grade appears in 2016-17 is sorted to this cohort, and any students present in 2015-16 are omitted. Similarly, any student numbers present in 2016-17 data, or prior, are omitted from the 2017-18 cohort. Finally, any students whose first grade entry appears in 2018-19 or later are omitted from the 2017-18 cohort.

Next, the maximum of the first and resit exams is selected as the final exam grade per year (per student), and this constitutes the final grade data frame and contains many missing values (MNAR). Courses taken at the beginning of year one have fewer missing values than later courses, which may be attributed to students abandoning their studies or that these elementary mathematics courses are offered to students from other programmes. The set of first-year courses are Calculus 1, Linear Algebra1, Calculus 2, Computer-Aided Problem Solving (CAPS), Linear Algebra 2, Analysis and Probability Theory, and the set of second-year courses are Ordinary Differential Equations (ODE), Statistical Reasoning, Statistics, Complex Analysis and Numerical Mathematics 1.

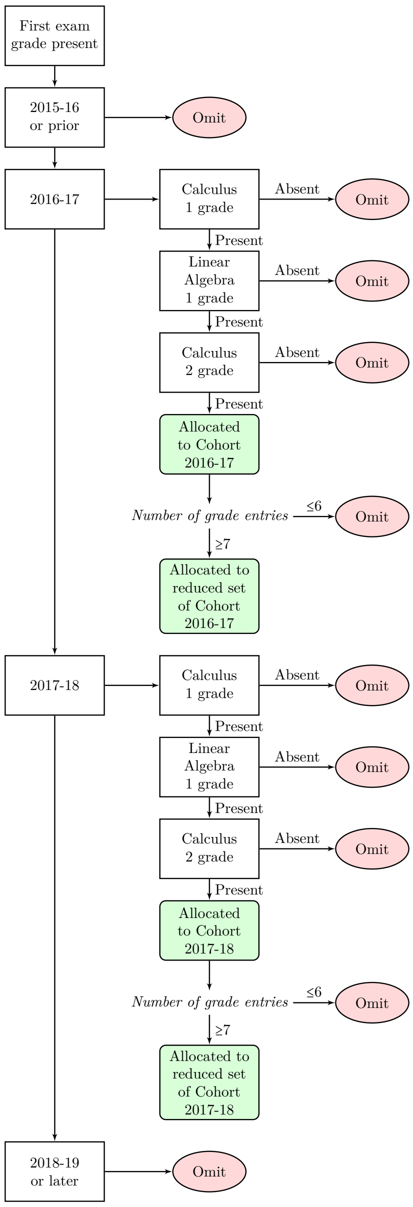
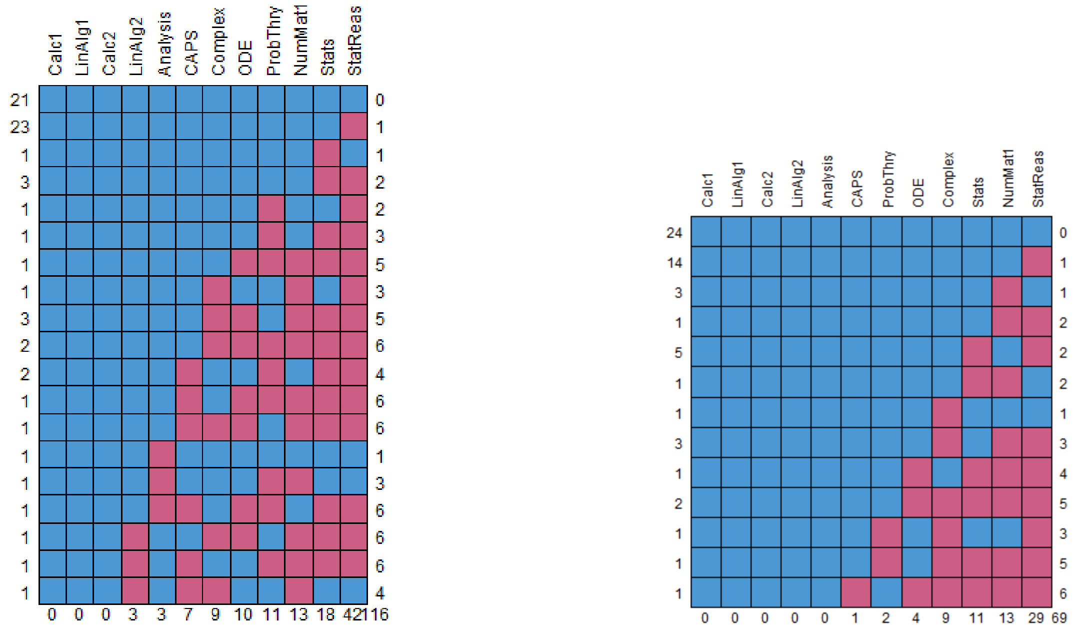


Figure 2: Decision tree for sorting students into their respective cohorts.

In order to reduce the number of missing values, students who achieve a grade in Calculus 1 and 2, and Linear Algebra 1 are omitted. The maximum grade per student, per course, over all years is calculated and the student must have a grade entered for these three courses to not be omitted.

The resulting cohort data frames contain comparatively less missing values, however both contain over 40% MNAR, and further reduction is required. The second reduction function omits any students with less than seven grade entries, in order to isolate the students who achieved a positive Binding Study Advice (BSA). The missing data patterns indicate that there remain few students with incomplete grade lists, and the percentage of MNAR now lies comfortably between 9-15% for either cohort.



(a) Missing data patterns for cohort 2016-17 after the second reduction. There are 21 complete cases, and 25 cases in which only one grade entry is missing.

(b) Missing data patterns for cohort 2017-18 after the second reduction. There are 24 complete cases, and 18 cases in which only one grade entry is missing.

Figure 3: Missing data patterns for cohorts 2016-17 after the second reduction described in Fig. 2.

Initial analysis

Prior to imputation of the missing data, the univariate and multivariate statistics are computed to determine appropriate imputation methods and predictor matrices. The distributions of the course grade data exhibit bimodal behaviour, with the lower peak density hovering around a grade of 4 and the higher peak density around 7. Indeed the assumption that the course grade data is normal has been violated (Shapiro-Wilk test statistic W; p-value < 0.05), and similar observations can be made from the QQ-plots of residuals. Therefore analysis of each imputation method is conducted to determine the most appropriate method for each course.

To refine the prediction process, the partial regression correlation coefficients are recorded in Table 4a with small, medium or large significance indicated (c.f. [2] §9.2.2). The coefficients are calculated using Spearman's ρ technique, in that the data is transformed to ranked data and then the Pearson's correlation coefficient is computed on the residuals of the ranked data, after the effects of all other variables have been regressed out. As the grades are rounded between 1 and 10 to the nearest 0.5, the data are semi-continuous and additionally contains tied ranks. Therefore Spearman's method is more suitable than Kendall's or Pearson's method [8].

With reference to the graphs in Fig. 1, it is known that an edge exists between two nodes if the variables indicated by the nodes are correlated. In order to preserve the conditional dependency, those cells containing medium or large significance in Table 4 are selected to be predictors with a "1" entered in each respective cell for the predictor matrix.

*Table 4: Partial regression correlation coefficients pr_j with significance of effect size given by asterisks (small * ($pr_j^2 \leq 0.024$); medium ** ($pr_j^2 \leq 0.1176$); large *** ($pr_j^2 \leq 0.538$)).*

(a) Partial regression correlation coefficients for cohort 2016-17.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	0.155*										
Calc2	0.122*	0.057*									
CAPS	0.053*	-0.146*	0.291*								
LinAlg2	-0.267*	0.184*	0.326*	0.183*							
Analysis	0.08*	0.137*	-0.225*	-0.101*	0.414**						
ProbThry	-0.032*	0.23*	0.025*	-0.184*	0.356*	-0.246*					
ODE	-0.288*	0.125*	0.235*	0.025*	-0.225*	0.389**	-0.165*				
StatReas	0.186*	-0.352*	0.102*	0.032*	0.026*	0.134*	0.106*	0.113*			
Stats	0.11*	0.082*	-0.217*	0.014*	0.375**	-0.364**	-0.067*	0.314*	0.102*		
Complex	0.156*	0.301*	0.007*	0.095*	-0.247*	-0.052*	0.074*	0.078*	0.187*	0.287*	
NumMat1	0.345*	-0.472**	-0.016*	-0.124*	0.461**	-0.058*	-0.022*	0.34*	-0.271*	-0.241*	0.406**

(b) Partial regression correlation coefficients for cohort 2017-18.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	-0.414**										
Calc2	0.531***	0.515***									
CAPS	-0.218*	-0.213*	0.563***								
LinAlg2	0.213*	0.126*	-0.092*	0.056*							
Analysis	0.032*	-0.029*	0.016*	0.21*	0.315*						
ProbThry	0.27*	0.509**	-0.318*	0.073*	-0.217*	0.269*					
ODE	0.388**	0.639***	-0.475**	0.145*	-0.295*	0.233*	-0.722***				
StatReas	-0.074*	-0.195*	0.277*	-0.468**	0.051*	0.065*	0.101*	0.138*			
Stats	-0.025*	-0.049*	-0.094*	0.349*	0.164*	-0.063*	0.265*	0.27*	0.157*		
Complex	0.528***	0.256*	-0.3*	0.205*	0.181*	-0.337*	0.029*	-0.02*	0.163*	-0.06*	
NumMat1	-0.397**	-0.305*	0.177*	0.067*	0.189*	0.027*	0.329*	0.304*	0.123*	-0.036*	0.189*

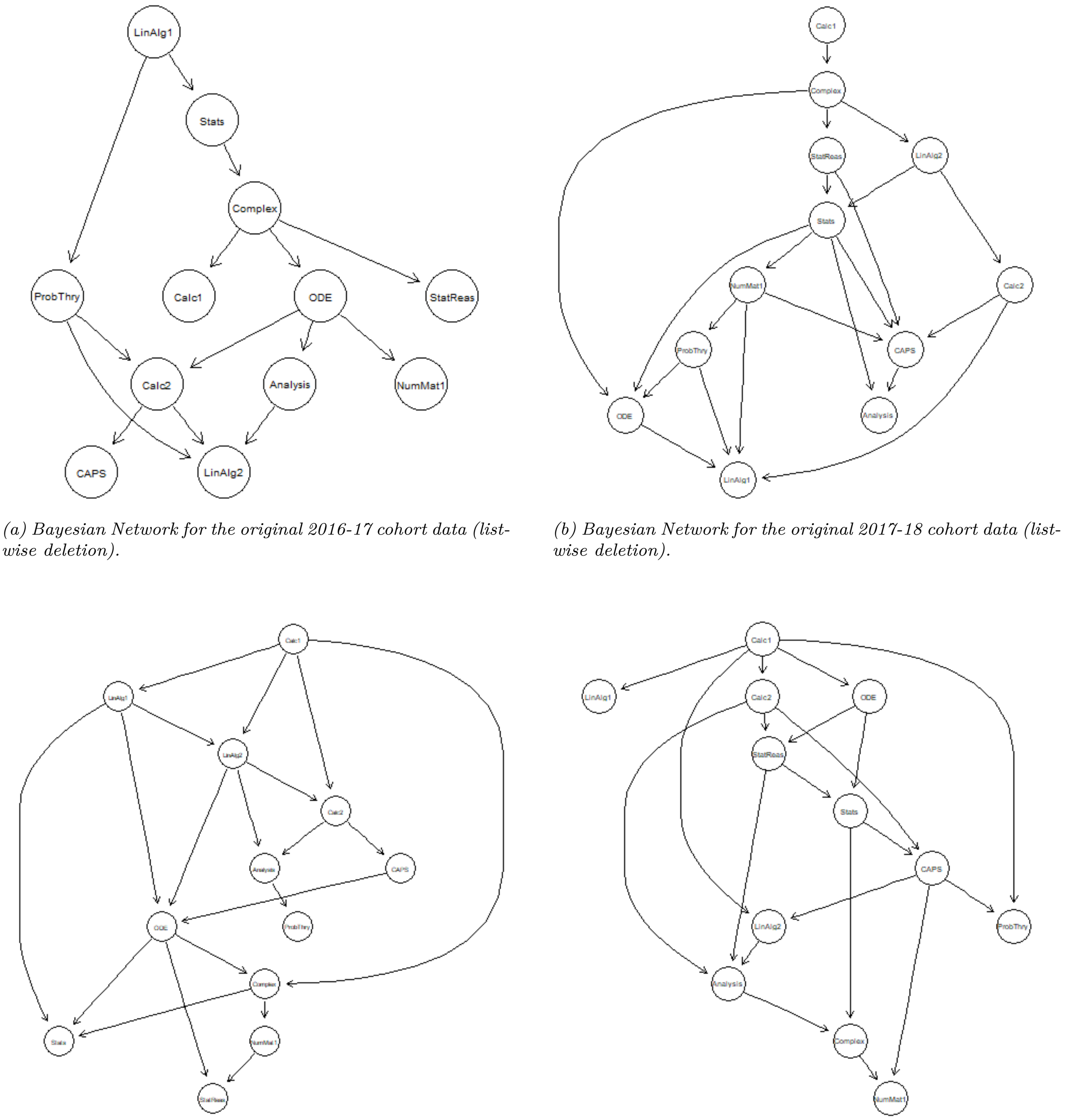
MICE

Referring to Fig. A.1, the flux plots display obvious differences between the cohorts. For the 2016-17 flux plot, the incomplete variables `Analysis`, `LinAlg2` and `CAPS` will be most informative in the imputation of the other incomplete variables. Whereas for cohort 2017-18, additionally `ODE` and `ProbThry` are useful. The completeness of `Calc1`, `LinAlg1` and `Calc2` for both cohorts asserts their usefulness in the imputation of incomplete variables which have significant partial correlation. The initial runs of `mice` on either cohort are implemented with `m = 5` imputations, a global `midastouch` or `pmm` method and without a predictor matrix input. The subsequent runs of `mice` are implemented with a predictor matrix which is dependent on the findings of the partial correlation and flux plot analyses. For example, no predictors are chosen for variable `Stats` in cohort 2017-18 when the decision is solely based on the significance of partial regression coefficients (c.f. Table 4b). However there are small partial correlations between `Stats` and `CAPS`, `ProbThry` and `ODE`, and the fluxes of `CAPS`, `ProbThry` and `ODE` nearly sum to one. Therefore we may manually alter the predictor matrix for cohort 2017-18 so that `CAPS` and `ODE` are used in the prediction of `Stats`. The final predictor matrices and post-MICE partial correlation tables are supplied in Appendix A.

Bayesian Network

After imputing the cohort data sets using the PMM and Midastouch methods, and using the altered predictor matrices, the GSA is employed using `hc()` and the outputs are presented in Fig. 4. With regards to Figs. 4a and 4b, the conditional probabilities between the (original) cohort data sets are dissimilar, for example the direction of the arc between `Calc1` and `Complex` is reversed. Moreover the graph skeletons (CPDAG with undirected edges) are dissimilar. Therefore it cannot be concluded that individual effects of the data set have been adequately accounted for using the selection criteria outlined in Fig. 2.

Comparisons of Figs. 4a, 4c and 4e show that there are also different conditional dependencies and different skeletons. It can be anticipated that there are minor differences between the original and imputed data Bayesian Networks, with the best method displaying the highest similarity to the original data Bayesian Network. An examination of Figs. 4b, 4d and 4f yields interesting results: the skeletons for PMM and Midastouch imputed data are similar, and the CPDAG for the Midastouch imputed data resembles closely the time-linear organisation of the courses in the programme. The differences in the three Bayesian Networks (within each cohort) can be attributed to the use of a global imputation method. A potential solution to this problem could be to permutate the two imputation methods across all variables and determine the vector of imputation methods, e.g. `c("pmm", "midastouch", "pmm")`, which minimises the difference between the original and the imputed data set graphs.

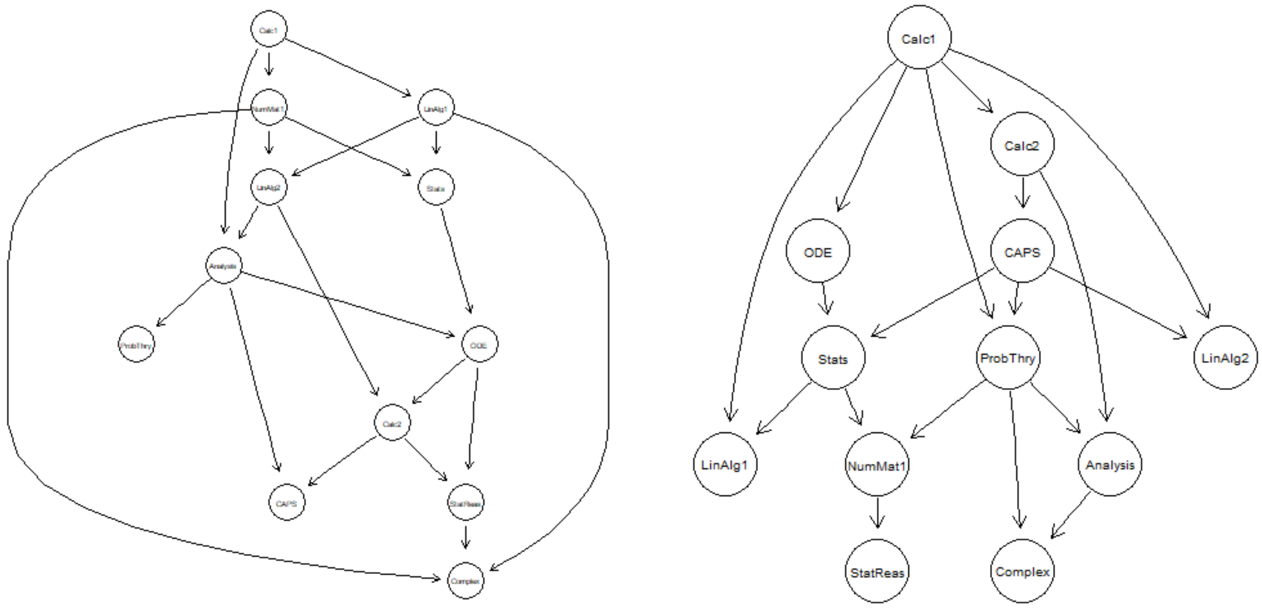


(a) Bayesian Network for the original 2016-17 cohort data (list-wise deletion).

(b) Bayesian Network for the original 2017-18 cohort data (list-wise deletion).

(c) Bayesian Network for the PMM imputed 2016-17 cohort data.

(d) Bayesian Network for the PMM imputed 2017-18 cohort data.



(e) Bayesian Network for the Midastouch imputed 2016-17 cohort data.

(f) Bayesian Network for the Midastouch imputed 2017-18 cohort data.

Figure 4: Bayesian Networks for the cohort 2016-17 and 2017-18 data sets, and the post-MICE data sets using methods PMM and Midastouch.

Discussion

The results of the previous section illustrate the requirement in science for repeatability and testability when constructing a global solution. The statistical pipeline as currently described does not ensure accurate interpretation of the course data results to which it is applied, and improvements are proposed for the selection criteria (Fig. 2) and MICE implementation.

It may be prudent to further reduce the sample size to students who have received passing grades in all first-year courses to ensure a unimodal distribution of these variables. Furthermore, any students of this new subset with a failing grade in second-year courses could be omitted, or alternatively their value could also be imputed. A suitable transformation of these first-year courses could remodel the variables to satisfy the Gaussian model assumption. Imputation under the Gaussian model may yield Bayesian Networks with a more accurate demonstration of the population variable interactions. Moreover, this may yield similar results between cohorts.

Efforts to permutate the PMM and Midastouch methods across all variables resulted in R crashing, which may be attributed to approach to coding scripts being somewhat ad-hoc. Future research could be conducted on the validity of a function with permutates appropriate imputation methods and confirms said validity by making graph comparisons to the graph of the original data set.

Future corrections to the statistical pipeline can also be proposed in analysing the differences in student grades for an online learning environment. Currently the COVID-19 pandemic has imposed a demand for a transition from on-campus to online learning, and the sudden change has resulted in an abnormal examination environment. Indeed the examination procedures are course dependent and not yet uniformly prescribed across programmes or faculties. An appropriate use of the corrected statistical pipeline may allow for differences in grade distributions to be easily identified, such as lower mean and mode values, or different (co)variances in the sample. Thus allowing for any corrections to the raw data to ensure that students' grades are not (negatively) affected by the change in examination environment.

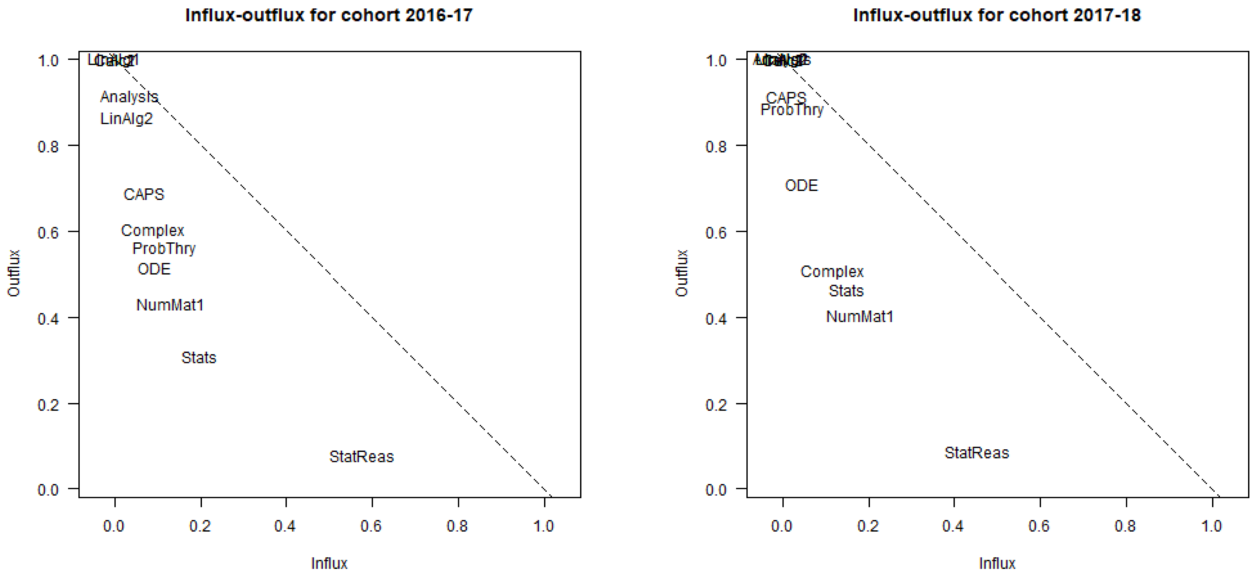
Appendix A

MICE analysis

The missing data patterns allow the computation of influx and outflux which provide insight into appropriate predictor matrices (c.f. §4.1.3 [1]). The influx is the ratio of the sum of variable pairs (X_i, X_k), with X_i missing and X_k observed, to the total number of observations. The outflux is analogously defined as the ratio of the sum of variable pairs (X_i, X_k), with X_k missing and X_i observed, to the total number of observations.

For two variables with the same influx, the variable with higher outflux is more useful for imputing the other variables as it is better connected to the data. Analogously, for two variables with the same outflux, the variable with the higher influx is more useful for imputation. Therefore variables which lie on or near the $y = 1 - x$ line are most useful for imputing the other variables.

The flux plots below display which variables are most useful in the imputation of other incomplete variables for the cohorts 2016-17 and 2017-18 data. It is useful to restrict the set of potential predictors to those which lie on or near to the dotted line. For instance, for both cohorts, if the variable which is being imputed is not significantly (partially) correlated to either `Complex`, `Stats`, or `NumMat1`, it may be beneficial to exclude them from the set of predictor variables.



(a) Flux plot of cohort 2016-17 data. `Calc1`*,* `LinAlg1` *and* `Calc2` *are complete, and have outflux equal to one and influx equal to zero. These three variables as well as* `Analysis` *and* `LinAlg2` *are the most useful variables, in terms of missingness, to impute the other incomplete variables.*

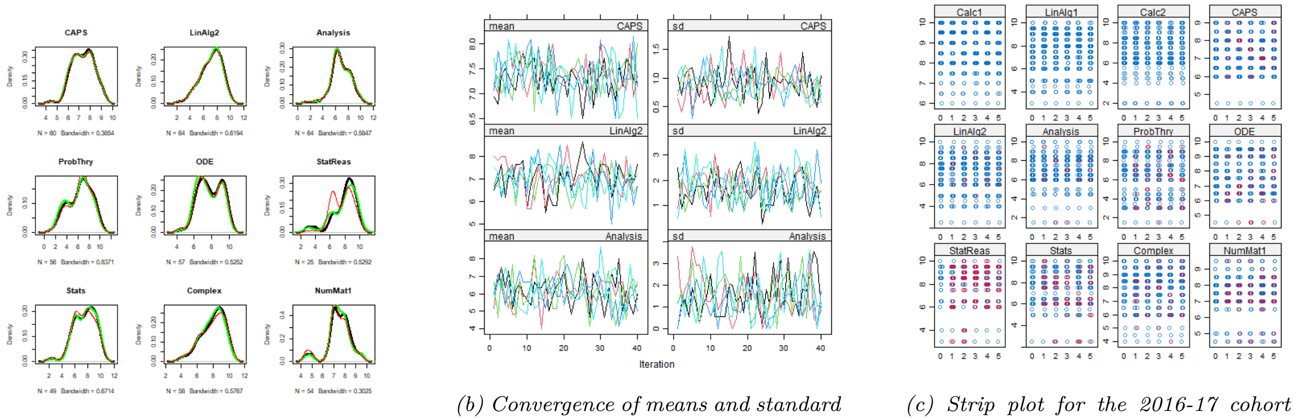
(b) Flux plot of cohort 2017-18 data. `Calc1`*,* `LinAlg1`*,* `Calc2`*,* `Analysis` *and* `LinAlg2` *are complete, and have outflux equal to one and influx equal to zero. These five variables as well as* `CAPS` *and* `ProbThry` *are the most useful variables, in terms of missingness, to impute the other incomplete variables.*

Figure A.1: Flux plots of cohort data sets, where complete variables have outflux equal to one and influx equal to zero. Variables which are most useful, in terms of missingness, for the imputation of other variables are displayed close to the $y = 1 - x$ dotted line.

After carefully considering the partial correlations and flux plots, the following predictor matrices are generated and utilised in the second run of MICE.

```
> pred_cohort1617
         Calc1 LinAlg1 Calc2 CAPS LinAlg2 Analysis ProbThry ODE StatReas Stats Complex NumMat1
Calc1        0       0     0    0       0        0        0   0        0     0       0       0
LinAlg1      0       0     0    0       0        0        0   0        0     0       0       1
Calc2        0       0     0    0       0        0        0   0        0     0       0       0
CAPS         0       0     1    0       0        0        0   0        0     0       0       1
LinAlg2      0       0     0    0       0        1        0   0        0     1       0       1
Analysis     0       0     0    0       1        0        0   1        0     1       0       0
ProbThry     0       1     0    0       1        1        0   1        0     0       0       0
ODE          0       0     1    0       1        1        0   0        0     1       0       0
StatReas     0       1     0    0       0        0        0   1        0     0       1       1
Stats        0       0     0    0       1        1        0   0        0     0       0       0
Complex      0       1     0    0       1        0        0   0        0     1       0       1
NumMat1      0       1     0    0       1        0        0   0        0     0       1       0
> pred_cohort1718
         Calc1 LinAlg1 Calc2 CAPS LinAlg2 Analysis ProbThry ODE StatReas Stats Complex NumMat1
Calc1        0       1     1    0       0        0        0   1        0     0       1       1
LinAlg1      1       0     1    0       0        0        1   1        0     0       0       0
Calc2        1       1     0    1       0        0        0   1        0     0       0       0
CAPS         0       0     1    0       0        0        0   0        1     0       0       0
LinAlg2      0       0     0    0       0        0        0   0        0     0       0       0
Analysis     0       0     0    0       0        0        0   0        0     0       0       0
ProbThry     0       1     0    0       0        0        0   1        0     0       0       0
ODE          1       1     1    0       0        0        1   0        0     0       0       0
StatReas     0       1     1    1       0        0        0   0        0     1       1       0
Stats        0       0     0    1       0        0        1   1        0     0       0       0
Complex      1       1     1    0       0        1        0   0        0     0       0       0
NumMat1      1       1     0    0       0        0        1   1        0     0       0       0
```

Convergence and fit of the imputations may be studied using the `stripplot()`, `plot(mice.mids())` and `densityplot()` commands available from the package `mice` [15].



(a) Comparison of original (black line) and imputed data distributions using the PMM (red line) and Midastouch (green line) methods.

(b) Convergence of means and standard deviations of the Midastouch imputed data sets. The absence of a distinct pattern indicates convergence of the iterations.

(c) Strip plot for the 2016-17 cohort imputed data using the Midastouch method; the blue coloured points are the original data values and the red points are the imputed values.

Figure A.2: Examples of the use of the `stripplot()`, `plot(mice.mids())` and `densityplot()` commands for imputation convergence analysis.

We are primarily concerned with the effect of the imputations on the partial correlations, as this will in turn effect the conditional (in)dependencies. The underlying null hypothesis is that the partial correlations for the imputed data set do not significantly differ from the partial correlations for the original data set.

Table A.1: Partial correlation coefficients for the imputed data cohorts 2016-17 and 2017-18 for Midastouch and PMM methods.

(a) Partial correlation coefficients for cohort 2016-17 with the Midastouch imputation method.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	0.177*										
Calc2	0.171*	0.105*									
CAPS	0.071*	-0.137*	0.31*								
LinAlg2	0.064*	0.273*	0.292*	-0.124*							
Analysis	0.157*	0.211*	-0.218*	0.246*	0.295*						
ProbThry	-0.2*	0.058*	0.036*	0.105*	0.06*	0.285*					
ODE	-0.114*	0.072*	0.199*	0.095*	-0.03*	0.388**	-0.191*				
StatReas	0.04*	-0.321*	0.282*	-0.175*	-0.11*	0.261*	0.101*	0.106*			
Stats	0.137*	0.425**	-0.055*	0.086*	-0.068*	-0.275*	0.119*	0.318*	0.195*		
Complex	0.031*	0.323*	-0.143*	0.12*	0.094*	-0.122*	-0.109*	0.008*	0.499**	-0.114*	
NumMat1	0.061*	-0.524***	0.172*	-0.096*	0.131*	0.137*	0.095*	0.118*	-0.423**	0.237*	0.528***

(b) Partial correlation coefficients for cohort 2016-17 with the PMM imputation method.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	0.255*										
Calc2	0.134*	0.064*									
CAPS	0.128*	-0.05*	0.274*								
LinAlg2	0.039*	0.279*	0.246*	-0.126*							
Analysis	0.163*	-0.006*	0.039*	0.137*	0.275*						
ProbThry	-0.163*	0.032*	0.085*	0.111*	0.043*	0.261*					
ODE	-0.083*	0.106*	0.277*	-0.02*	-0.023*	0.252*	-0.121*				
StatReas	0.016*	0.103*	-0.243*	0.212*	0.095*	-0.176*	-0.086*	0.517***			
Stats	0.002*	0.246*	-0.079*	0.039*	0.14*	-0.084*	0.05*	0.156*	0.07*		
Complex	0.176*	0.014*	0.098*	-0.001*	-0.052*	0.145*	-0.052*	-0.051*	0.194*	0.208*	
NumMat1	0.085*	-0.305*	-0.069*	0.114*	0.189*	-0.091*	-0.012*	0.33*	-0.335*	0.056*	0.313*

(c) Partial correlation coefficients for cohort 2017-18 with the Midastouch imputation method.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	0.042*										
Calc2	0.45**	0.108*									
CAPS	-0.256*	0.092*	0.416**								
LinAlg2	0.325*	0.178*	-0.153*	0.297*							
Analysis	-0.061*	-0.054*	0.311*	-0.083*	0.174*						
ProbThry	0.183*	0.144*	-0.15*	0.148*	-0.001*	0.152*					
ODE	0.556***	0.192*	-0.233*	0.002*	-0.145*	0.113*	-0.117*				
StatReas	0.143*	0.229*	-0.072*	0.002*	-0.2*	0.087*	-0.259*	-0.216*			
Stats	-0.004*	0.139*	-0.009*	0.244*	-0.035*	0.015*	0.043*	0.289*	0.189*		
Complex	0.172*	0.047*	-0.101*	0.144*	-0.063*	0.303*	0.104*	-0.094*	0.054*	-0.155*	
NumMat1	-0.096*	-0.382**	0.139*	0.075*	0.187*	-0.004*	0.454**	0.277*	0.414**	0.095*	0.086*

(d) Partial correlation coefficients for cohort 2017-18 with the PMM imputation method.

	Calc1	LinAlg1	Calc2	CAPS	LinAlg2	Analysis	ProbThry	ODE	StatReas	Stats	Complex
Calc1											
LinAlg1	0.151*										
Calc2	0.421**	0.059*									
CAPS	-0.263*	0.055*	0.45**								
LinAlg2	0.381**	0.083*	-0.151*	0.292*							
Analysis	-0.027*	-0.064*	0.205*	0.083*	0.168*						
ProbThry	0.198*	-0.017*	-0.083*	0.137*	0.101*	0.187*					
ODE	0.261*	0.091*	-0.133*	0.075*	-0.027*	0.12*	-0.098*				
StatReas	-0.084*	0.077*	0.1*	-0.059*	0.01*	0.227*	0.037*	0.346*			
Stats	0.113*	0.032*	-0.148*	0.313*	-0.158*	-0.273*	0.199*	0.263*	0.182*		
Complex	0.261*	-0.045*	-0.005*	-0.136*	-0.158*	0.35*	-0.069*	-0.1*	0.125*	0.205*	
NumMat1	-0.161*	0.027*	0.09*	0.127*	0.233*	-0.029*	0.062*	0.031*	-0.034*	0.202*	0.267*

Bibliography

[1] Stef van Buuren. *Flexible Imputation of Missing Data.* 2nd ed. Chapman, Hall, and CRC Press, 2018. URL: https://stefvanbuuren.name/fimd/.

[2] Jacob Cohen. *Statistical power analysis for the behavioral sciences.* 2nd ed. Lawrence Erlbaum Associates, 1988.

[3] A. Dasgupta et al. “Correlation in a Bayesian Framework”. In: *The Canadian Journal of Statistics* 28.4 (2000), pp. 675–687. ISSN: 03195724. URL: http://www.jstor.org/stable/3315910.

[4] P. Govan. *Bayesian Network Modeling and Analysis.* 2020. URL: https://paulgovan.shinyapps.io/BayesianNetwork/.

[5] M. Grzegorczyk. *Statistical Genomics [WISG-09].* 2020.

[6] Hirotaka Itoh, Keisuke Itoh, and Kenji Funahashi. “Forecasting Students’ Grades Using Bayesian Network Models and an Evaluation of Their Usefulness”. In: *The Journal of Information and Systems in Education* 11.1 (2012), pp. 32–41. DOI: 10.12937/ejsise.11.32.

[7] Hirotaka Itoh, Keisuke Itoh, and Kenji Funahashi. “Forecasting Students’ Future Academic Records Using Past Attendance Recording Data and Grade Data”. In: *Procedia Computer Science* 22 (2013), pp. 921–927. DOI: 10.1016/j.procs.2013.09.175.

[8] M.G. Kendall. “The treatment of ties in ranking problems”. In: *Biometrika* 33.3 (Nov. 1945), pp. 239–251. ISSN: 0006-3444. DOI: 10.1093/biomet/33.3.239. eprint: https://academic.oup.com/biomet/article-pdf/33/3/239/573257/33-3-239.pdf. URL: https://doi.org/10.1093/biomet/33.3.239.

[9] Koski, T. and Noble, J.M. *Bayesian Networks: An Introduction.* Wiley series in probability and statistics. Chichester, West Sussex, UK: John Wiley & Sons, Ltd., 2009. ISBN: 978-0-47074-304-1.

[10] J. R. Leslie, M. A. Stephens, and S. Fotopoulos. “Asymptotic Distribution of the Shapiro-Wilk W for Testing for Normality”. In: *The Annals of Statistics* 14.4 (1986), pp. 1497–1506. DOI: 10.1214/aos/1176350172. URL: http://www.jstor.com/stable/2241484.

[11] K.J. Levy and S.C. Narula. “Testing Hypotheses concerning Partial Correlations: Some Methods and Discussion”. In: *International Statistical Review* 46.2 (1978), pp. 215–218. ISSN: 03067734, 17515823. URL: http://www.jstor.org/stable/1402814.

[12] D. Makowski et al. “Methods and Algorithms for Correlation Analysis in R”. In: *CRAN* (2020). URL: https://github.com/easystats/correlation.

[13] M. Scutari. “Learning Bayesian Networks with the bnlearn R Package”. In: *Journal of Statistical Software* 35.3 (2010), pp. 1–22. ISSN: 1548-7660. DOI: 10.18637/jss.v035.i03. URL: https://www.jstatsoft.org/v035/i03.

[14] Stef van Buuren and Karin Groothuis-Oudshoorn. “mice: Multivariate Imputation by Chained Equations in R”. In: *Journal of Statistical Software* 45.3 (2011). DOI: 10.18637/jss.v045.i03.

[15] G. Vink and S. van Buuren. *miceVignettes | A detailed course on solving realistic inference problems with mice.* 2018. URL: http://www.gerkovink.com/miceVignettes/.