What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?

Bruce G. Marcot & Anca M. Hanea

ONLINE FIRST

Computational Statistics

ISSN 0943-4062

Comput Stat
DOI 10.1007/s00180-020-00999-9

Springer

Computational Statistics
https://doi.org/10.1007/s00180-020-00999-9

ORIGINAL PAPER

What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?

Bruce G. Marcot[1] · Anca M. Hanea[2]

Received: 8 November 2019 / Accepted: 4 June 2020

Abstract

Cross-validation using randomized subsets of data—known as k-fold cross-validation—is a powerful means of testing the success rate of models used for classification. However, few if any studies have explored how values of k (number of subsets) affect validation results in models tested with data of known statistical properties. Here, we explore conditions of sample size, model structure, and variable dependence affecting validation outcomes in discrete Bayesian networks (BNs). We created 6 variants of a BN model with known properties of variance and collinearity, along with data sets of n = 50, 500, and 5000 samples, and then tested classification success and evaluated CPU computation time with seven levels of folds (k = 2, 5, 10, 20, n − 5, n − 2, and n − 1). Classification error declined with increasing n, particularly in BN models with high multivariate dependence, and declined with increasing k, generally levelling out at k = 10, although k = 5 sufficed with large samples (n = 5000). Our work supports the common use of k = 10 in the literature, although in some cases k = 5 would suffice with BN models having independent variable structures.

Keywords Model validation · Classification error · randomized subsets · sample size

Bruce G. Marcot and Anca M. Hanea: Co-first authorships.

✉ Bruce G. Marcot
bruce.marcot@usda.gov

Anca M. Hanea
anca.hanea@unimelb.edu.au

[1] U.S. Forest Service, Pacific Northwest Research Station, Portland, OR, USA

[2] Centre of Excellence for Biosecurity Risk Analysis (CEBRA), University of Melbourne, Parkville, VIC 3010, Australia

Published online: 13 June 2020

1 Introduction

One of the more important steps in model building is ensuring the credibility and robustness of validation procedures designed to determine how well a model predicts known outcomes, particularly classifying categories or states of some response variable. As distinguished from calibration—determining the degree of fit of a model to a set of data—validation entails testing a model against an independent data set not used to initially construct and parameterize the model. Such validation procedures can take various forms including bootstrapping, jackknifing, and cross-validation (e.g., Lillegard et al. 2005; Shcheglovitova and Anderson 2013; Arlot and Celisse 2010). In this paper, we focus on the problem of cross-validation because few, if any, studies have determined optimal ways to subset data sets to conduct cross-validation.

1.1 Cross-validation

Cross-validation provides information that model calibration does not. Cross-validation helps reveal the degree to which a model is robust, that is, its accuracy and classification success when applied to new or novel situations. Cross-validation is also key to determining the degree to which a model is overfit. This occurs when calibration error rates are low but cross-validation error rates are high (Last 2006), signalling that a model is well tuned to some initial data or situations but cannot perform well with other data or other situations.

One popular form of model validation uses k-fold[1] cross-validation (Geisser 1975; Arlot and Celisse 2010). In this approach, first a data file is compiled of n cases, each with values of covariates and response variables. The case file is then typically randomized and divided into k equal segments. The first k segment, consisting of n/k cases, is set aside and a model is parameterized with the remaining (n − n/k) cases, then tested against the first segment for rates of classification error, comparing model results to the known outcomes (response variable values) in each case. Next, from the full case file the second k segment is set aside and the model is parameterized with the remaining cases, then tested against the second segment, and so on for all k segments. Values of k can range [2, n − 1], where k = 2 pertains to simply splitting the case-file data set in half, and k = n − 1 refers to the "leave one out" (LOO) approach (e.g., Brady et al. 2010) where the model is parameterized based on the n − 1 cases and then tested against each case individually. The LOO approach, however, can be computationally expensive, often does not provide additional validation benefit over lower values of k (Breiman and Spector 1992), and can result in high variance of model performance and model overfitting (Cawley and Talbot 2007).

[1] K-fold (Anguita et al. 2012) is also referred to as V-fold (Arlot and Celisse 2010) and M-fold (Hobbs and Hooten 2015; M here is in reference to the Markov chain Monte Carlo algorithm). We use K-fold as a synonym for all terms.

Tests of model validation for each k subset "fold" of the data include calculations of rates of model classification accuracy (the complement of model classification error), and bias and variance in error rates. In general, as k varies from 2 to n − 1 (i.e., from few to many fold subsets), bias decreases, variance in error rate of the validation tests increases, and computation time increases (exponentially). Also, bias and model classification error tend to be inversely related. Note that when k = 1 there are no case file subsets, so results pertain to model calibration (degree to which the model fits the given data set) rather than validation (testing against an independent data set).

The question we address here is, what is the best value of k to help ensure optimal evaluation of model validity? Also, to minimize computation time, is there a smallest value of k for which low bias, low variance, and high model accuracy (the complement of low classification error) might stabilize? These questions have been largely ignored in the literature, particularly with discrete Bayesian networks (BNs). Instead, 10-fold is commonly used in the literature (k = 10; e.g., Aguilera et al. 2010; Booms et al. 2010; Zhao and Hasan 2013) but with no particular test of, nor specific rationale given for, this level. Breiman and Spector (1992) used expected squared error as the classification error of simulated analytic models they tested, and found that submodel selection criteria greatly affected validation results, with 5-fold cross-validation providing better outcomes than did the LOO approach.

Ideally, for n cases, the best selection of k would be such that there remains full representation of conditions in both the model and the test data sets. This is not a trivial problem, however, as even large empirical data sets (e.g., "big data") can have small numbers of replications of cases with specific combinations of variables (Hastie et al. 2015) or can be biased by excluding or unnecessarily codifying low values of some variables (Stow et al. 2018). Further, the best selection of k for a given data set also likely depends on a number of attributes of the data such as the degree of collinearity and variability, and attributes of the constructed model such as the degree of data discretization and the presence and form of interaction terms among the covariates.

1.2 Bayesian networks

This study focuses on k-fold cross-validation with discrete BN models. BNs are directed acyclic graphs (DAGs) that essentially represent variables linked by conditional probabilities (e.g., Koski and Noble 2011; Jensen and Nielsen 2007). Outcomes (posterior probabilities) are calculated using Bayes' theorem. BNs are acyclic in that feedback loops—variables linked back to themselves—are excluded. Variables in BNs can be of various forms including continuous numeric values (ratio-scale, interval-scale, equation-derived, or numeric constants) and discrete categories (nominal, cardinal, or ordinal). Typically, continuous variables are discretized into a finite number of exclusive value ranges or states. Here, we focus on BNs that are commonly represented with discrete-state (discretized) numeric variables.

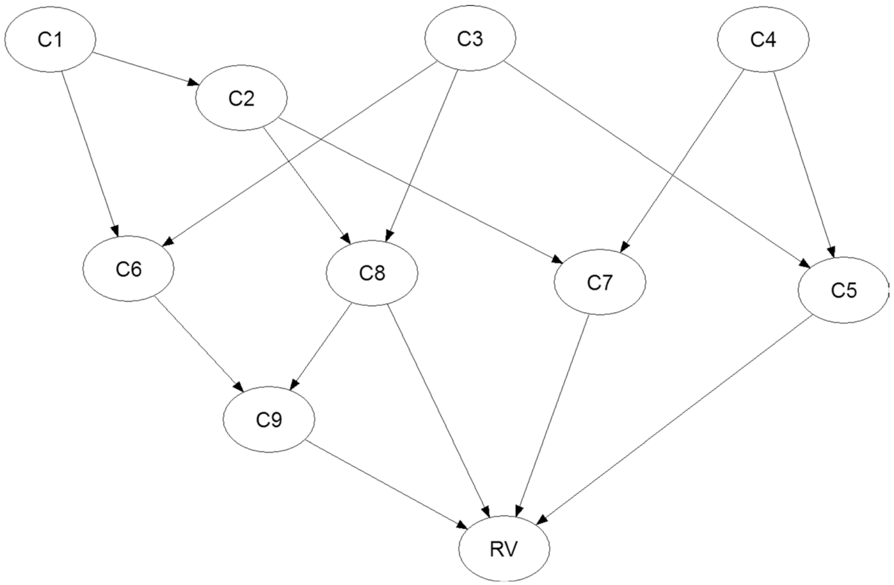

Fig. 1 Bayesian network used in the simulation exercise. The network represents a ten-dimensional joint distribution with nine covariates, called C_1, …, C_9 and one response variable, called RV. There are 15 arcs and the largest set of parents (node RV) has cardinality four

2 General modelling framework

For this project, we developed a BN model from a case file data set that we created with known properties of variance and collinearity (Fig. 1). In this way, we were able to control for the BN structure, parameter values, and uncertainties in the distributions. The BN consists of 9 covariates and one response variable (RV, final outcome node), and 15 arcs (links between variables). Eleven arcs are between covariates, representing correlational, causal, or logical relationships. The BN contains no cut points (nodes which, if removed, would separate the graph). All variables are continuous, and the dependence is specified through direct and conditional rank correlations. Keeping the BN model structure constant, we varied the parameters of the continuous functions in each covariate, devised a series of case file data sets, and tested the efficacy of each model variant with each case file data set using several values of k in conducting k-fold cross validation. Details of model construction are presented further below.

2.1 Variables potentially influencing the efficacy of k-fold cross-validation

How well a model will perform when subjected to cross-validation is likely determined by a host of conditions describing the model structure and complexity, and the extent and diversity of the data set used for the testing (Table 1). We controlled

Table 1 Factors that could affect the results of a cross-validation test of a discrete Bayesian network model and how each was calculated or accounted for in the test model (see text)

Factor	Attribute of the test data set (DS) or the Bayesian network model (BN)	Use or value in the Bayesian network model in this study
Number of cases (n)	DS	n ϵ {50, 500, 5000}
Fraction of cases (k)	DS	k ϵ {2, 5, 10, 20, n − 5, n − 2, n − 1}
Extent (proportion) of missing values in the case file	DS	No missing values
Response variable (RV)	BN	1 response variable with a fixed beta distribution
Covariates (prediction variables, C_i)	BN	9 covariates with various continuous distributions
Number of states s of response variable RV	BN	Discretized into 4 states
Number of states s of covariate i	BN	All variables discretized into 4 states
Total number of variables (model dimension)	BN	10 variables
Size of the CPT for response variable RV	BN	CPT size $=4^5$
Size of the CPT for covariate i	BN	CPT size was 4 (for nodes with no parents), 16 (for nodes with one parent), or 64 (for the nodes with two parents)
Variation in values of response and prediction variables among the cases, var(RV) and var (C_i)	BN	Van Valen coefficient of variation, a multivariate extension of the coefficient of variance, with 3 different values corresponding to low, medium, and high overall dependence
Dependence structure complexity	BN	Determinant of the Spearman rank correlation matrix, with 2 different values corresponding to dependence and near-independence
Degree of explanatory power of the covariates	BN	Not considered (sensitivity analysis not conducted)
Acceptable error rates (threshold values)	BN	Not considered; resulting error rates shown as continuous value outcomes
Threshold levels for defining response variable outcome states	BN	(Default is dominant probability outcome, which varies according to number of states in the response variable, that is, (s|Rvi/100); but one could use an alternate threshold level
Rate of model overfitting (validation accuracy/calibration accuracy)	BN	Not considered

or otherwise accounted for these conditions in our example BN and varied those that pertain to the test data set and values of k. With a total number of cases n, the test values of interest for k lie in the domain [2, n − 1].

2.2 Measures of performance

The main question for this study is to determine if there is a value of k which optimizes BN model validation performance. We measured BN model performance as classification bias and variance. Classification bias is defined as confusion (classification) error: false positives (Type I error), false negatives (Type II error), and total error. In many real-world problems, such as in environmental or natural resource management realms, Type I and II errors carry very different implications for cost and model credibility (Marcot 2007; Pawson et al. 2017), and are generally pertinent to binary outcome states. In our test BN models, the response variable contains 4 states, so we calculated only total error rates. Classification variance is defined as the degree of variation in classification error among the k folds tested.

We refer to BN model validation performance being optimized with low confusion error and low validation variance, with low values of k (thus avoiding unnecessary computational complexity and cost). We used the following measures of BN model performance: (1) log-likelihood loss, also known as negative entropy, which is the negated expected log-likelihood of the test set for the BN fitted from the training set; and (2) classification (prediction) error, which is the misclassification rate for a single node in a discrete network; here, the values of the target node are predicted using only the information present in its local distribution (from its parents). We used two measures of performance because they are not necessarily correlated and can provide complementary insights into model validity (Marcot 2012). Values of classification error range [0,1], where 0 = no error and 1 = complete error. Model prediction accuracy also can be used as a performance measure, but it is completely determined by the classification error (accuracy = 1 – classification error).

In our model parameterization and testing, we used a sequential optimization procedure whereby key parameters were fixed and other parameters were then optimized iteratively (some call this a multilevel inference problem, e.g., Guyon et al. 2010). We plotted the results as the mean model classification error rate as a function of k, where "mean error rate" here refers to averaging error rates across all k classification accuracy analyses for each given value of k.

2.3 Conjectures and hypotheses

We initially hypothesized at least 6 different forms of the relationship between model performance (e.g., classification accuracy) and values of k (Fig. 2a), including a null hypothesis of no effect of k on model performance. Our alternative hypotheses include monotonic and modal relationships, variously suggesting some lowest (asymptotic exponential or sigmoid function, step function) or optimal (modal function) value of k that would provide the highest model performance outcome, or that model performance continues to improve (linear function) all the way through k = n

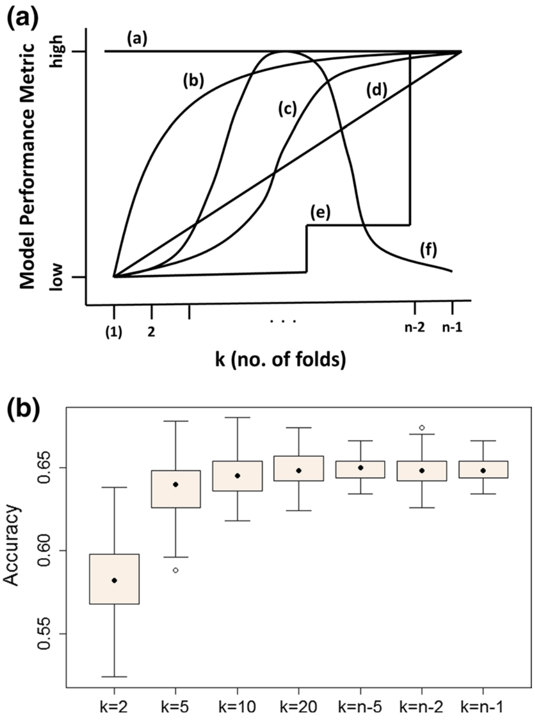

Fig. 2 **a** Hypothetical relationships of model performance (e.g., classification accuracy rates) as a function of values of k, in k-fold cross-validation of discrete Bayesian networks. Forms of relationships: a = null hypothesis of no effect of k on model performance, b = asymptotic exponential, c = asymptotic sigmoid, d = linear, e = step, f = modal. Other forms of relationships are also possible, and the form of the relationship may depend on many factors such as variations in the variable values and the dependency structure of the network. Positions along the y-axis are arbitrary. **b** Results of k-fold cross-validation from Bayesian network models with high dependency and medium variation among variable values, with a simulated case file sample size of 500, for various values of k folds

− 1. We also hypothesized that model classification bias would decrease and classification error variance would increase monotonically, both directly and inversely, with greater values of k and with large samples. Criteria for identifying the best or optimal value of k would vary according to the form of these functions, whether it be an inflection point on a curve, a percentage approach to an asymptote, the threshold of a step function, or a subjective level of acceptable classification accuracy.

Here, we define optimal value(s) of k (k_{op}) as those with more or less stabilized low classification error (high model accuracy) and with the lowest number of k folds, under various model conditions of variable dependence and variation. In a general setting of testing discrete Bayesian networks, calculating k_{op} analytically is likely infeasible. Therefore, we used simulations to derive approximations with simulated data sets tested on a variety of BN models created with specified properties of variable dependence and variation.

3 Analysis methods

Our methods for determining optimal values of k entailed specifying the form of the variables used in our training data sets, specifying the algorithm for parameterizing the BN models from the training data sets and the resulting model structures, and specifying the measures of BN model performance including classification error rates.

3.1 Assumptions chosen for the variables and BN model

The assumptions we applied for choosing the form of the variables in our training data set are as follow:

- No dimensionality reduction, i.e., all variables in the data set are needed.
- No missing values and that any imputations (matrix completion) are already done.
- Parameter values are relearned for each k-fold used.
- For each k, the analysis is repeated 100 times (100 runs per k, where the sequence of the cases is randomized for each run, resulting in different case subsets).
- Retain the overall BN structure for all analyses.

We considered three main factors related to the test data set and 12 factors related to the BN model structure (Table 1). We did not address the potential condition of model overfitting, as this was not a consideration for identifying effects of k for a given model structure and data set.

Many authors use data from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets.html) that provides many data sets with known response variable outcomes. However, we chose to devise our own simulated data sets so that we could control for the statistical distributions from which values of each variable would be derived, for variation in values for each variable, and for correlations among variables. We devised the data set scenarios using measures of overall variation and the degree of dependence among the covariates (described below). In general, in our tests we varied the number of cases (n) from our simulated case files, and the values of k for each case file size. We then repeated cross-validation tests using combinations of values of n and k, and plotted outcomes of our selected model performance metrics.

3.2 Bayesian network structure

BN structures (linkage of variables) and parameters (values of conditional probabilities) can be learned from data by use of a variety of machine-learning algorithms. BN structures can be learned strictly from data using unsupervised algorithms such as naive Bayes structuring (Friedman et al. 1997) and, improving on the naive approach, the tree-augmented network (TAN) algorithm (Aguilera et al. 2010) or with more complex algorithms as well such as constrained-based and score-based algorithms (Murphy 2012). BN parameters can be learned strictly by converting case data into relative frequencies or by other machine-learning algorithms such as expectation maximization (Dempster et al. 1977; Do and Batzoglou 2008), which is a convergent log-likelihood function that adjusts conditional probability values in a specified BN structure to best fit the known outcomes in a case file (Murphy 2012).

Table 2 Continuous distributions used in each covariate (C) and response variable (RV) in the Bayesian network model (Fig. 1)

Covariate	Distribution used
C1	Exponential
C2	Gamma
C2	Weibull
C4	Gama
C5	Beta
C6	Log-normal
C7	Normal
C8	Log uniform
C9	Uniform
RV	Beta

Although machine learning algorithms are available for both learning the structure (the DAG) of a BN and fitting the parameters, there are huge variations between algorithms, especially for small datasets (e.g. when learning a BN on 10 variables from a sample size of 50 cases). Results of different algorithms can vary greatly depending on the class of the algorithm (i.e. constrained-based, score-based or hybrid-class) and the choice of the conditional independence test or of the scoring metric. After a preliminary investigation, in which we learned the structure using one constrained-based algorithm and one score-based algorithm, we decided that the variations in the learned structures would have a very large confounding effect on our simulation experiment. Thus, we decided to fix the DAG and vary only the fit of the parameters (values of the conditional probability tables, CPTs) with the case subsets resulting from varying k.

We created 6 model variants of the general BN structure (Fig. 1) to account for degrees of variation of the values of the variables and of their dependence, that is, model variants corresponding to different parameterizations of the same network structure. The parametric families of the marginal distributions remained unchanged across the 6 model variants, but their parameters varied (see Table 2). The different parametrisations correspond to different combinations of marginal variability and overall dependence as detailed below.

The BN model we devised is a moderately large model (10 nodes, 15 links) with a resonable number of interconnections more or less mimicking the size of real-world BN models found in publications. In particular, we considered its similarity to one of the examples from Pourret et al. (2008). All variables have continuous parametric marginal distributions (exponential, Weibull, gamma, beta, lognormal, normal, uniform, log-uniform), and their dependence was specified through Spearman's rank correlations and conditional correlations. The marginal distribution families remained unchanged in each of the 6 variants of the BN model. However, their parameters changed to allow for various degrees of overall variation. The only marginal distribution whose parameters remained unchanged across all BN model variants was the distribution of the response variable, RV ~ Beta (20, 5).

Table 3 Variants of Bayesian network models as defined by summary measures of dependence (the determinant D of the rank correlation matrix) and the coefficient of variation (CoV), of the values of the model covariates (model affector variables)

Bayesian network model variants	D	CoV
Independence and low variation	0.66	~ 10^{-2}
Independence and medium variation	0.66	~ 10^{-1}
Independence and high variation	0.66	~ 10
Dependence and low variation	10^{-6}	~ 10^{-2}
Dependence and medium variation	10^{-6}	~ 10^{-1}
Dependence and high variation	10^{-6}	~ 10

D takes values between 0 (complete linear dependence) and 1 (complete independence), where 0.66 signifies very low average dependence (thus denoted here as independence). CoV takes values ≥ 0; where 10^{-2} denotes low variation, 10^{-1} medium variation, and 10 high variation

3.3 Measures of BN model performance

We used the Van Valen coefficient of variation (CoV; defined by Eq. 2 of Adelin and Zhang 2010) as the overall measure of variation. This multivariate extension of the univariate coefficient of variation does not account for the dependence between variables. CoV takes values larger than zero with smaller values indicating less average variation. We considered three different values of this coefficient: 10^{-2} that indicated very little variation, 10^{-1} that indicates medium variation, and 10 that indicates large variation (Table 3). In our analysis, the degree of dependence among the covariates is indicated separately by the determinant (D) of the rank correlation matrix, which measures linear dependence between a monotonic transformation of original margins. An important reason to choose D as a summary measure of dependence is that D factorises on the arcs of a BN, and thus can be easily controlled for in a simulation exercise. D takes values between zero and one, with one indicating complete linear independence and zero corresponding to complete linear dependence.

We chose two distinct values for D (10^{-6} and 0.66) to differentiate between two degrees of overall dependence in the multivariate distribution (Table 3). The two values were chosen on the following basis. The distribution of the determinant of a random 10 × 10 correlation matrix is very skewed toward 0. There are many more 10-dimensional distributions which exhibit at least one linear dependence between the 10 variables (actually 2 perfectly correlated variables are enough for the determinant to be 0) than there are multivariate distributions where all pairwise correlations are 0, which is the only case for the determinant to be 1 (e.g., see Hanea and Nane 2018). That is, the relationship between values of D and the degree of independence is highly nonlinear, so we chose 0.66 as a compromise to represent a relatively high degree of independence without forcing total independence which would be unlikely in real-world research data sets.

For each combination of values of D and CoV, we built a new model having the same structure but with different parametrization. We parameterized each of the 6 BN model variants by using sample sizes of 50, 500 and 5000, resulting in 18 different synthetic datasets. The models were constructed and sampled using the software Uninet (https://lighttwist-software.com/uninet/; Cooke et al. 2007).

The data sets were read into the software R and discretized before building the BNs and fitting their parameters. All variables (for all models) were discretized into four states using Haremink's Algorithm (Hartemink 2001). Other discretization techniques were trialled, and Hartemink's Algorithm was the only one that recovered the correlation structure within a maximum of 10^{-2} absolute difference when compared to the original correlation structure. Other choices for the number of states (two, three, and five) did not perform as well in terms of recovering the correlation structure. No dynamic discretisation was possible using the method chosen, so this is one potential limitation of the algorithm in Hartemink (2001); all covariates and the response variable are discretized using the same fixed number of states.

For each n, each k, each fold, and each repetition, we fitted the parameters using the maximum likelihood estimation procedure implemented in the R package bnlearn (Scutari 2010). We used two of the performance measures mentioned previously (also available in the R package): classification (prediction) error (pred) and log-likelihood loss (logl) to evaluate prediction power. Because the variables are not binary we could not parse the classification error into Type I and Type II errors. Instead we evaluated the accuracy and the amount of variation when using multiple runs (with100 repetitions).

We also evaluated the degree to which classification error varied across the sets of repeated, replicated runs for each k fold value by calculating the running standard error SE of classification error across the increasing number of replicate runs. For this, we used BN models with high multivariate dependence and medium variation among variable values with a simulated case file sample size of n = 500.

We tracked computer central processing unit (CPU) computation times for all cross-validation analyses under all combinations of k-fold values, model variants, and database sizes. We used a laptop computer with an Intel(R) Core(TM) i7-8550U CPU processor operating at 1.80 GHz, with 16.0 GB installed memory (RAM), and a 64-bit Windows 10 (version 1809) operating system. We used the R package *bnlear* (R version 3.5) and tracked the time to run the function *bn.cv* which performed the k-fold cross-validations. CPU time was determined with the R function *system.time* which produced *elapsed time* that tracked the duration of CPU seconds charged to each cross-validation analysis separately for classification error and log-likelihood loss calculations.

4 Results

Our results consisted of 126 combinations of the 6 BN model variants (two levels of multivariate dependence and three levels of generalised marginal variation), the three case-file sample sizes (n = 50, 500, and 5000), and the seven levels of folds (k = 2, 5, 10, 20, n − 5, n − 2, and n − 1). For each of the 126 combinations, and for each of the 100 replicate runs by which we reshuffled the case file entries, we produced confusion matrices that are tables enumerating Type I, Type II, and total model classification error. With the case file sample size n = 5000, log-likelihood loss was calculable only for four values of k (i.e., 2, 5, 10, 20), where the remaining values of k (i.e., n − 5, n − 2, and n − 1) resulted in incalculable (infinite) values of log-likelihood loss. Due to very long computational time for n = 5000, the confusion matrices for k is n − 5, n − 2, and n − 1 were not considered either. Thus, in total, we produced 10,800 confusion matrices: 4200 confusion matrices (for 7 levels of k and 6 BN model variants) each for simulated case files of n = 50 and n = 500 and each repeat from 100, and 2400 confusion matrices (for 4 levels of k and 6 BN model variants) for simulated case files of n = 5000, and repeated 100 times each.

We present here selected findings for major subsets of these combinations that best exemplify overall patterns, showing the influence of case file sample sizes n, BN model variants of multivariate dependence and marginal variation, and numbers of folds k, on model classification error, model accuracy, and log-likelihood loss. We also interpret our findings in the context of our hypothesized relationships of model performance to values of k, and we summarize values of k that seem to satisfy optimality criteria of performance outcomes.

4.1 Influence of case file size n

Among all BN model variants, all case file sizes n, and all k folds, classification error ranged ~ 0.27 to ~ 0.72 (Table 4). For all values of k and all BN model variants, classification error generally declined with increasing case file size n, with a greater decline between n = 50 and 500, than between n = 500 and 5000 (Table 4; Figs. 3, 4, 5, 6). Classification error was highly statistically correlated with case file size n among all BN model variants and k values (df = 143, F = 48.62, $p < 0.001$), among BN models with high variable dependence and low variable dependence (both df = 71, F = 23.96, $p < 0.001$), and among BN models with high, medium, and low variable variation (all df = 47, F = 15.75, $p < 0.001$).

4.2 Influence of multivariate dependence

Declines in classification error with increasing case file size n were more prominent in BN models with high multivariate dependence than for BN models with low multivariate dependence. The most precipitous drops in classification error occurred between case file sizes n = 500 and n = 5000 in high dependence models (Figs. 3, 4, 5, 6). This held true with all values of k tested.

4.3 Influence of marginal variation

The influence of the marginal variation on classification error seems less dramatic than that of the multivariate dependence. For n = 50, variations in the level of marginal variability did not affect the classification error. The same holds for larger n (i.e. 500 and 5000) conditional on a given dependence structure. The only exception to the lack of influence of the variability on the classification error is with very large case files (n = 5000) and the high multivariate dependency model variant, where low

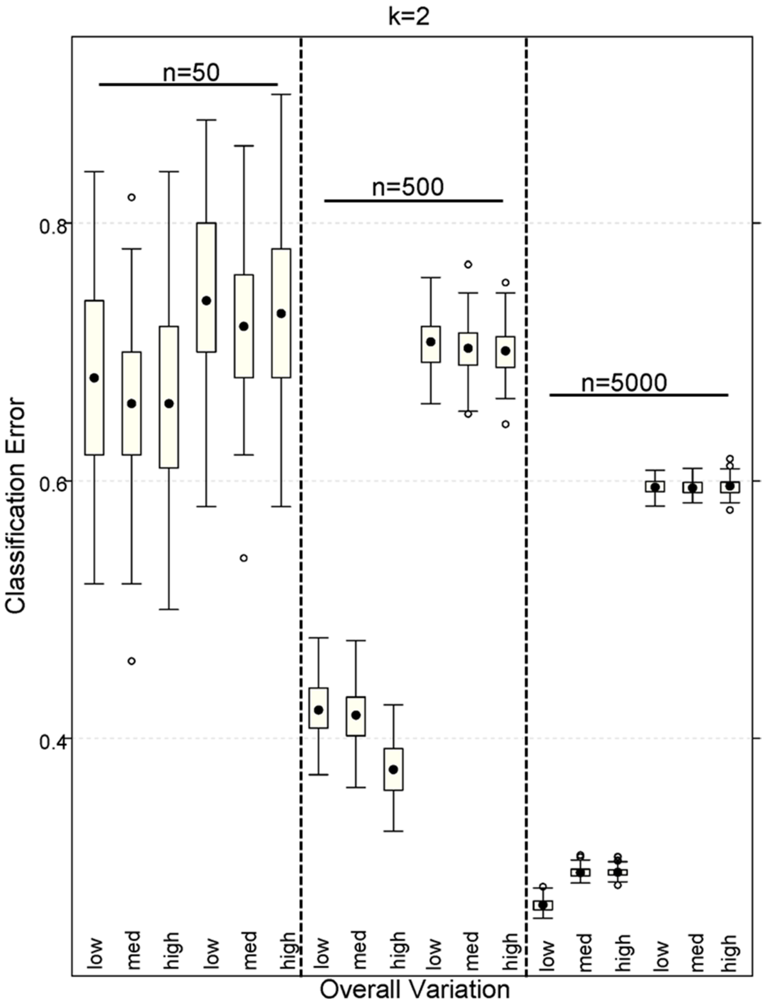

Fig. 3 Classification (prediction) error for all Bayesian network models, for k = 2 folds, for three case file sizes n. Boxplots correspond to 100 replications for each model. In each set of 6 boxplots, the first three are for Bayesian network models with high model variable dependency, and the second three are for Bayesian network models with low variable dependency

marginal variance results in a significantly lower classification error (see Figs. 3, 4, 5, 6).

4.4 Influence of number of folds k

Classification error dropped steeply between number of folds of k = 2 and k = 5, then less steeply, levelling out for k ≥ 10. A typical example (Fig. 7) is with BN models with high multivariate dependence and medium overall variation, with a case file sample n = 500. Also, the degree of variation (height of the box whiskers in

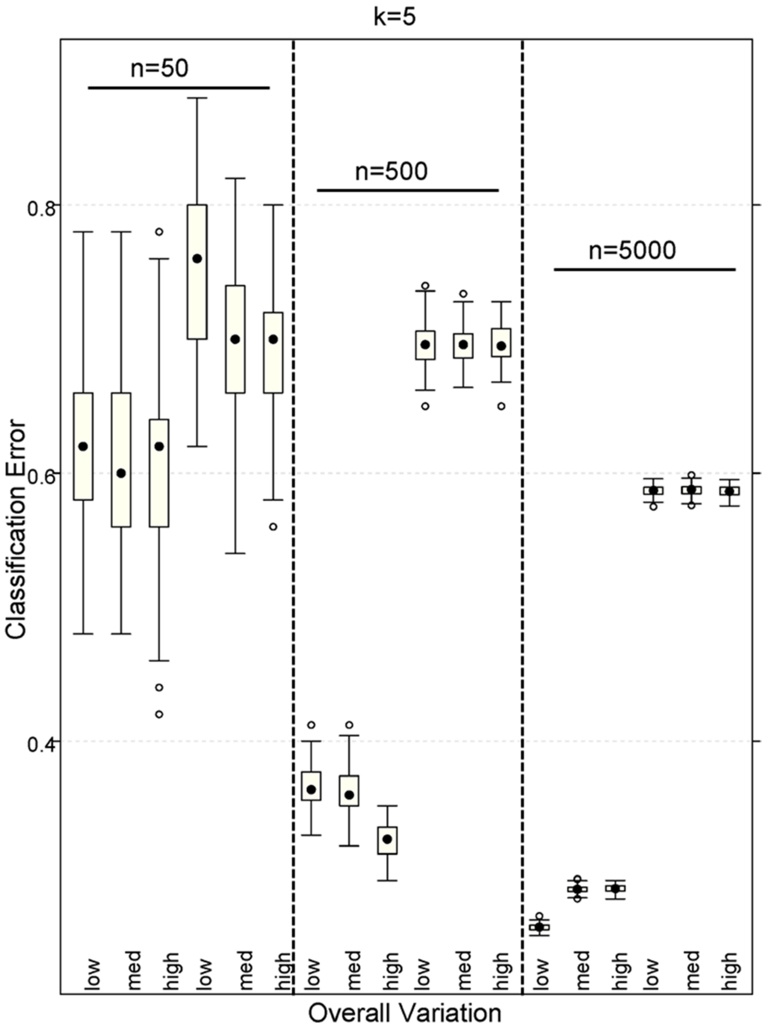

Fig. 4 Classification (prediction) error for all Bayesian network models, for k = 5 folds, for three case file sizes n. Boxplots correspond to 100 replications for each model. In each set of 6 boxplots, the first three are for Bayesian network models with high model variable dependency, and the second three are for Bayesian network models with low variable dependency

Figs. 7, 8) in classification error declined with increasing values of k, again more or less stabilizing with k ≥ 10 to 20. Classification error was highly statistically correlated with number of folds k among all BN model variants and k values (df = 143, F = 114.86, $p < 0.001$), among BN models with high multivariate dependence (df = 71, F = 58.07, $p < 0.001$) and low multivariate dependence (df = 71, F = 55.22, $p < 0.001$), and among BN models with high (df = 47, F = 37.23, $p < 0.001$), medium (df = 47, F = 37.23, $p < 0.001$), and low (df = 47, F = 37.16, $p < 0.001$) variable variation.

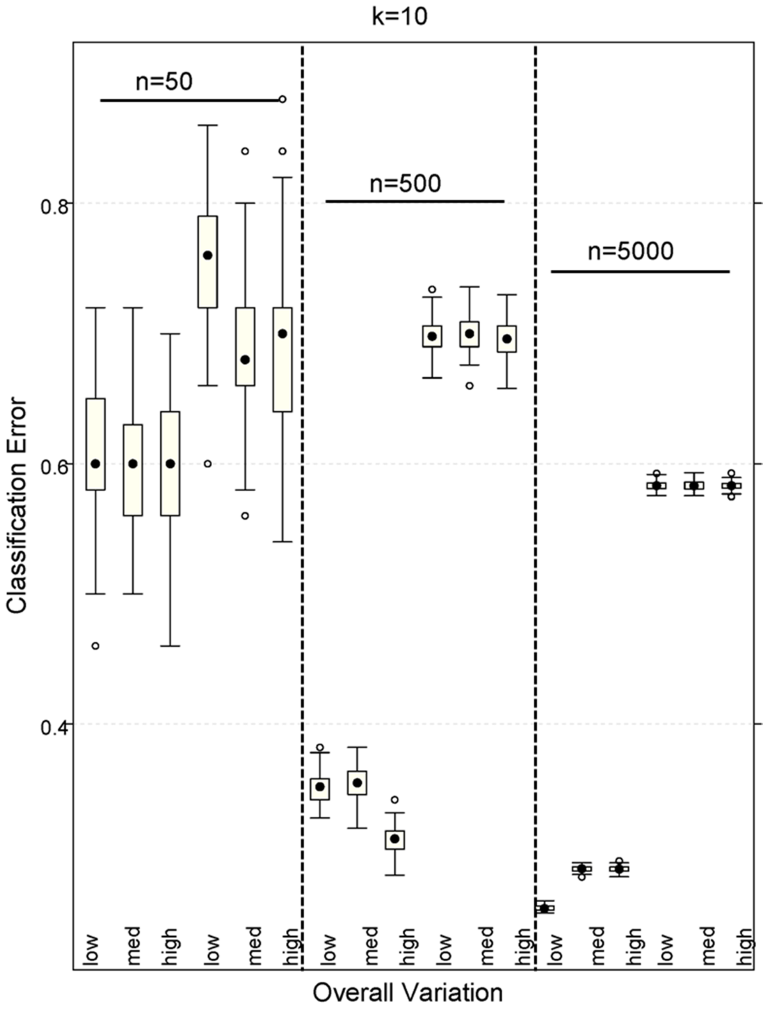

Fig. 5 Classification (prediction) error for all Bayesian network models, for k = 10 folds, for three case file sizes n. Boxplots correspond to 100 replications for each model. In each set of 6 boxplots, the first three are for Bayesian network models with high model variable dependency, and the second three are for Bayesian network models with low variable dependency

The variation (SE) in classification error rates declined, as expected, with increasing numbers of replicated runs, more or less levelling out past about 30 runs (Fig. 9) The degree of variation started and remained greater with smaller values of k, and took a higher number of replicate runs to achieve an equivalent levelling as with runs with larger values of k.

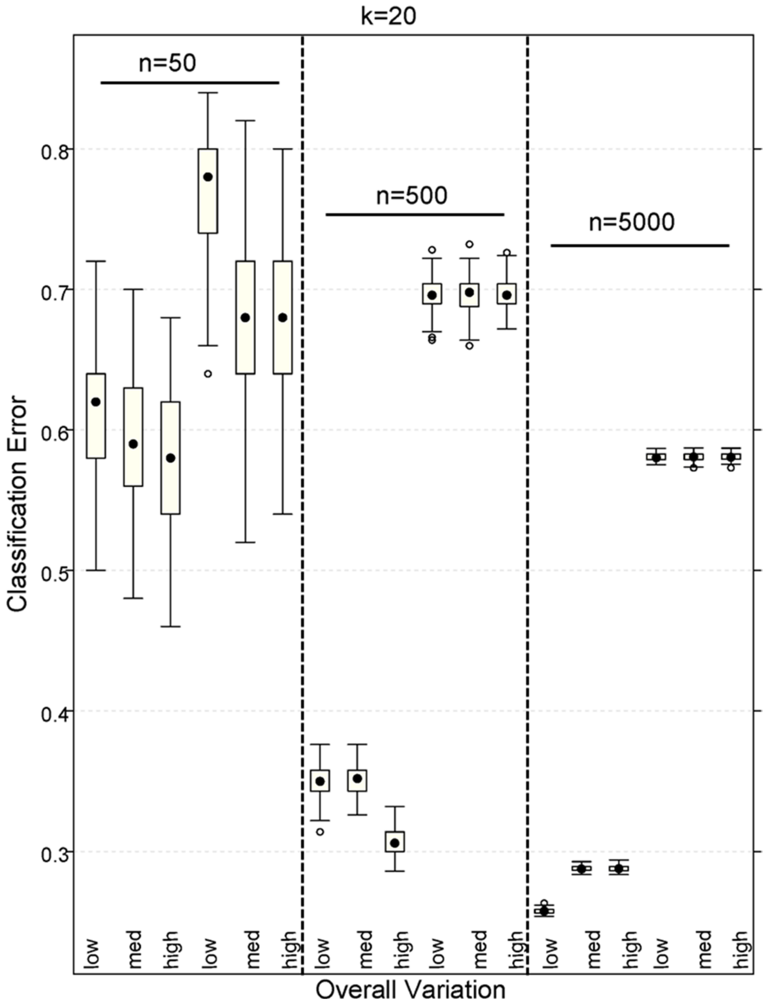

Fig. 6 Classification (prediction) error for all Bayesian network models, for k = 20 folds, for three case file sizes n. Boxplots correspond to 100 replications for each model. In each set of 6 boxplots, the first three are for Bayesian network models with high model variable dependency, and the second three are for Bayesian network models with low variable dependency

4.5 Overall performance relationships

We plotted a typical example of classification accuracy (the converse of classification error) from BN models with high multivariate dependence and medium overall marginal variation, for simulated case file sample size of n = 500, across the seven k-fold values (Fig. 2b) to compare with our initial hypotheses (Fig. 2a). Results suggest that the effect of increasing the number of k folds on classification accuracy best fits the asymptotic exponential hypothesis (curve b in Fig. 2a), with an initial surge in accuracy with low numbers of k-folds, followed by progressively decreasing

Table 4 Classification error, ranging [0,1], for 3 case file sample sizes n and 7 values of k folds, across 6 Bayesian network (BN) model variants

		BN model variant					
n	k	Dep, low variation	Dep, medium variation	Dep, high variation	Indep, low variation	Indep, medium variation	Indep, high variation
5000	2	~0.27	~0.29	~0.29	~0.6	0.6	0.6
5000	≥5	~0.2	~0.29	~0.29	~0.58	~0.58	~0.58
500	2	~0.42	~0.42	~0.38	~0.7	~0.7	~0.7
500	5	~0.37	~0.36	~0.33	~0.7	~0.7	~0.7
500	≥10	~0.35	~0.35	~0.3	~0.7	~0.7	~0.7
50	2	~0.68	~0.66	~0.65	~0.75	~0.72	~0.72
50	5	~0.63	~0.61	~0.6	~0.74	~0.7	~0.7
50	≥10	~0.6	~0.6	~0.6	~0.76	~0.68	~0.68

BN model variants are denoted by their degree of multivariate dependence (Dep = high dependence, Indep = independence) and degree of covariates variability (low, medium, high) (see Table 3)

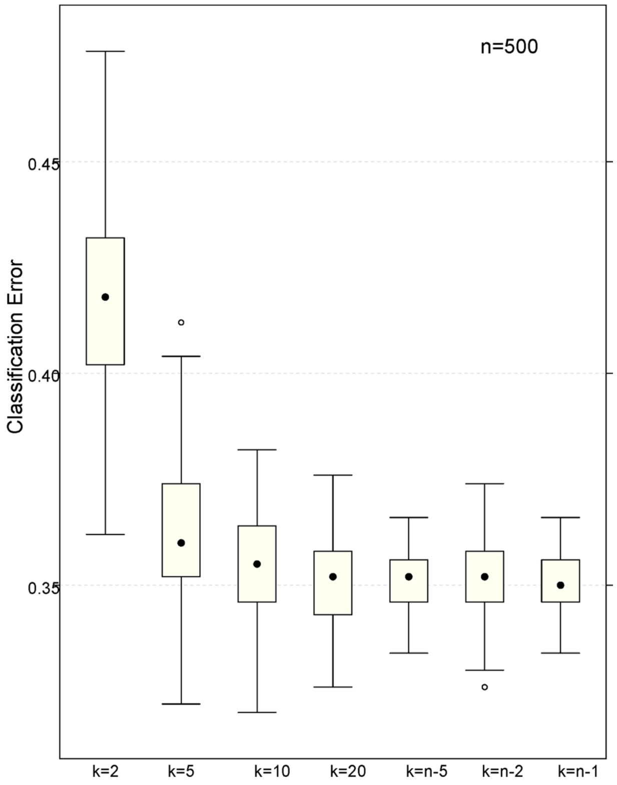

Fig. 7 Classification (prediction) error for the Bayesian network model with high variable dependence and medium overall variation, with data set of size n = 500. Each boxplot corresponds to one value of k with 100 replications

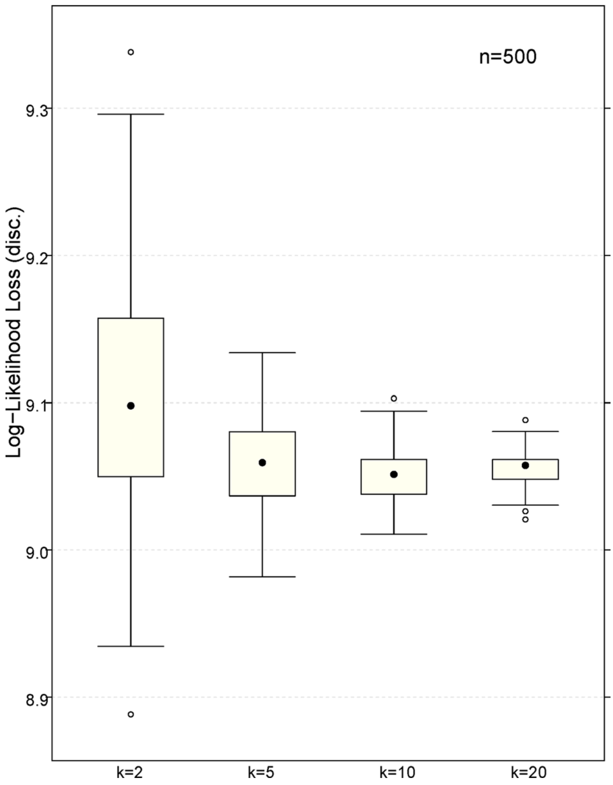

Fig. 8 Log-likelihood loss for the model with high dependence and medium overall variation, with data set of size n = 500. Each boxplot corresponds to one value of k with 100 replications. Likelihood values are not finite (do not converge) for k ϵ{{n − 1, n − 2, n − 5}, so those presented here are only for k ϵ {2, 5,10, 20}

gains on accuracy with greater numbers of k-folds, with means generally stabilizing with k ≥ 20 although with little statistical difference past k = 10.

4.6 Optimal values of k

Finally, we interpreted the above results to determine optimal values of k (k_{op}) by BN model variant and case file sample size n (Table 5). We define optimal values of k as the minimum number of folds (to avoid undue computational complexity) for which the classification error rate is stable (the decrease in error with increasing number of folds is less than 10^{-3}).

With very large case files (n = 5000), k_{op} = 5 folds seemed sufficient across all BN model variants. With smaller case files, however, the BN model variant played a role in determining k_{op}: BN models with higher multivariate dependence tended to warrant higher numbers of folds, i.e., k_{op} = 10 in most cases but the

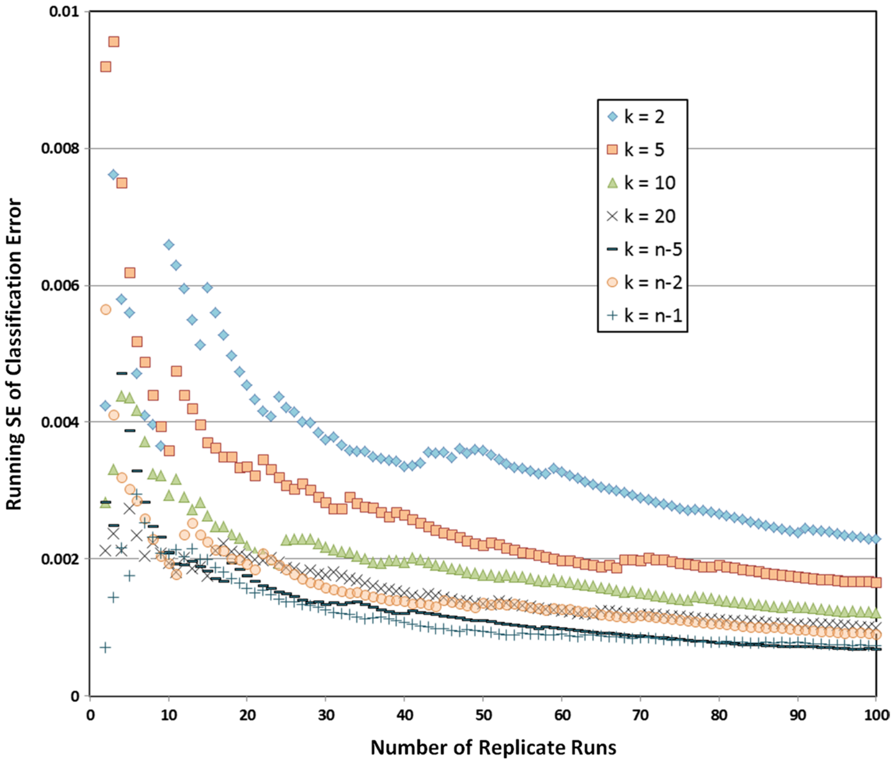

Fig. 9 Variability (running standard error, SE) of classification error as a function of number of replicate runs of k-fold cross-validation. Results are from Bayesian network models with high dependency and medium variation among variable values, with a simulated case file sample size of n = 500, for 7 values of k

Table 5 Optimal values of k (k_{op}), by Bayesian network (BN) model variant (see Tables 3, 4)

n	BN model variant					
	Dep, low variation	Dep, medium variation	Dep, high variation	Indep, low variation	Indep, medium variation	Indep, high variation
5000	$k_{op} = 5$	$k_{op} = 5$	$k_{op} = 5$	$k_{op} = 5$	$k_{op} = 5$	$k_{op} = 5$
500	$k_{op} = 10$	$k_{op} = 10$	$k_{op} = 10$	$k_{op} = 2$	$k_{op} = 2$	$k_{op} = 2$
50	$k_{op} = 10$	$k_{op} = 10$	$k_{op} = 5$	$k_{op} = 10$	$k_{op} = 10$	$k_{op} = 10$

ones mentioned further. For n = 500 and BN models with high independence of variables, regardless of variation in variable values, $k_{op} = 2$ folds seem to suffice. For a very low but possibly realistic sample size (n = 50) the optimal k was 10 except for when there is high marginal variation and high multivariate dependence, when five folds are enough.

4.7 Computation time

CPU elapsed times for calculating classification error and log-likelihood loss were highly statistically correlated (Pearson r = 0.971, $p < 0.001$, n = 108 cross-validation combinations of k folds, data set sizes n, and model structures of variance and dependence) and nearly identical in values. Using the computer described in Methods, CPU time for computing classification error across all values of n and all $k \leq 20$ averaged 4.0 s (SD 2.9, minimum 0.8, maximum 11.1). Computation time differed at most by being 8 s longer with k = 20 compared with k = 2.

However, computation time greatly increased with larger size data sets. With n = 50, CPU time averaged 18.6 s (SD 0.8, minimum 17.3, maximum 20.1); and with n = 500, CPU time averaged 257.4 s (SD 105.4, minimum 189.5, maximum 480.4). Again, this was across all values of k = n − 5, n − 2, and n − 1 (LOO). The longest computation time we encountered was the maximum time for calculating log-likelihood loss with n = 500 and k = 499 (that is, k = n − 1, LOO), requiring 597.8 s, which was about 150 times longer than the average computation time required with $k \leq 20$. Calculations of model performance for any values of k = n − 5, n − 2, or n − 1 with the largest data set of n = 5000 took such an inordinate amount of time and CPU cycles that we terminated runs before completion, and thus do not present results of these combinations.

5 Discussion

Several unexpected relationships emerged from our analysis. For one, we found that independent of the number of data cases n, model performance measures were similar for all values of k when the variables' dependence structure is closer to independence, but they differed with a greater degree of dependence and when n was 50 or 500. For another, when n was 50, we found that k_{op} was 5 or 10, depending on the variability; and when n was 500, k_{op} was either 2 or 10 (Table 5). Both measures took surprisingly different values for the dependent versus the independent cases.

Note that k_{op} was greater with BN models with high dependency among predictor variables, but this dependency was highest with intermediate values of sample size n (Table 5). Although this seems like an anomaly, it may likely be due to spurious correlation, explained in that dependence is more easily represented and modelled more accurately in the folds, whereas independence is more difficult to detect because of spurious correlations that each fold might erroneously exhibit. Such spurious correlations occurring with folds may complicate our originally hypothesis that bias declines (and variance increases) monotonically with increasing k.

Cross-validation of a discrete Bayesian network is generally conducted along the lines of our work here, where a given model structure with a fixed set of probability parameters is tested against a series of k folds of cases (e.g., Constantinuo et al. 2016; Forio et al. 2015; Hammond and Ellis 2002). However, it is also possible to conduct cross-validation by reparameterizing the probability values in the model, or even restructuring the model, for each successive fold. These approaches would add further complication in the validation process and particularly in parsing out optimal values of k. Our results may need to be revisited should such validation procedures include model reparameterization and restructuring with each fold.

Our results pertain to overall classification error, but our general approach could be repeated with binary output models to also track Type I and II error rates (false positives and false negatives) are of interest. Further, it would be possible to calculate Type I and II error rates for models with >2 output states by sequentially focusing on error rates of each state and considering error rates of all other states combined, thereby collapsing the output to a series of binary states. We have not conducted that analysis here but with our balanced sets of simulated cases, drawn from statistical distributions, such results would not have changed outcomes of optimal values of k. They may differ in models where variables do not follow generalized statistical distributions. Also, our findings likely hold for very simple models but are unclear for very complex models with many more variables, more linkages among variables, and greater model depth.

We confirmed our assumption and previous assertions that computation time increased at least exponentially, becoming intractable, with the higher values of $k=n-5$, $n-2$, and especially $n-1$, and particularly with large data sets consisting on the order of our test set of $n=5000$ cases. This has important implications for validation of "big data" sets which have become popular in developing and training prediction models (LaDeau et al. 2017; Marcot and Penman 2019). Fortunately, we have demonstrated that validation results with $k\leq 20$ produce fully acceptable results and require far less computation time.

6 Conclusions

Most uses of $k=10$ in the literature (as reviewed above) can be supported by our findings, but in many cases $k=5$ would suffice with BN models with independent variable structures regardless of variation in variable values, particularly where saving time and computational complexity is an issue.

Our findings are likely to be generally robust for BN models with variables exhibiting high independence or high dependence, and various levels of variation in variable values. However, clearly, many forms of BN models can be created, in which cases when they deviate from our contrived examples it may be prudent to slightly increase the number of k folds over the optimal values we present here.

Acknowledgements We thank Clint Epps, Julie Heinrichs, and an anonymous reviewer for helpful comments on the manuscript. Marcot acknowledges support from U.S. Forest Service, Pacific Northwest Research Station, and University of Melbourne, Australia. Mention of commercial or other products does not necessarily imply endorsement by the U.S. Government.

References

Adelin AA, Zhang L (2010) A novel definition of the multivariate coefficient of variation. Biomet J 52(5):667–675

Aguilera PA, Fernández A, Reche F, Rumi R (2010) Hybrid Bayesian network classifiers: application to species distribution models. Environ Mod Softw 25:1630–1639

Anguita D, Ghelardoni L, Ghio A, Oneto L, Ridella S (2012) The 'K' in K-fold cross validation. In: Proceedings, ESANN 2012, European symposium on artificial neural networks, computational intelligence and Mmachine learning. Bruges (Belgium), 25–27 Apr 2012, i6doc.com publ. http://www.i6doc.com/en/livre/?GCOI=28001100967420

Arlot S, Celisse A (2010) A survey of cross-validation procedures for model selection. Stat Surv 4:40–79

Booms TL, Huettmann F, Schempf PF (2010) Gyrfalcon nest distribution in Alaska based on a predictive GIS model. Polar Biol 33:347–358

Brady TJ, Monleon VJ, Gray AN (2010) Calibrating vascular plant abundance for detecting future climate changes in Oregon and Washington, USA. Ecol Ind 10:657–667

Breiman L, Spector P (1992) Submodel selection and evaluation in regression: the X-random case. Int Stat Rev 291–319

Cawley GC, Talbot NLC (2007) Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. J Mach Learn Res 8:841–861

Constantinuo AC, Fenton N, Marsh W, Radlinski L (2016) From complex questionnaire and interviewing data to intelligent Bayesian network models for medical decision support. Artif Intell Med 67:75–93

Cooke RM, Kurowicka D, Hanea AM, Morales O, Ababei DA, Ale B, Roelen A (2007) Continuous/discrete non parametric Bayesian belief nets with UNICORN and UNINET. In: Proceedings of Mathematical Methods in Reliability MMR, 1–4 July 2007, Glasgow, UK

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc 39(Series B):1–38

Do CB, Batzoglou S (2008) What is the expectation maximization algorithm? Nat Biotechnol 26:897–899

Forio MAE, Landuyt D, Bennetsen E, Lock K, Nguyen THT, Ambarita MND, Musonge PLS, Boets P, Everaert G, Dominguez-Granda L, Goethals PLM (2015) Bayesian belief network models to analyse and predict ecological water quality in rivers. Ecol Model 312:222–238

Friedman N, Geiger D, Goldszmidt M (1997) Bayesian network classifiers. Mach Learn 29:131–163

Geisser S (1975) The predictive sample reuse method with applications. J Amer Stat Assoc 70:320–328

Guyon I, Saffari A, Dror G, Cawley G (2010) Model selection: beyond the Bayesian-Frequentist divide. J Mach Learn Res 11:61–87

Hammond TR, Ellis JR (2002) A meta-assessment for elasmobranchs based on dietary data and Bayesian networks. Ecol Ind 1:197–211

Hanea AM, Nane GF (2018) The asymptotic distribution of the determinant of a random correlation matrix. Stat Neerl 72:14–33

Hartemink AJ (2001) Principled computational methods for the validation and discovery of genetic regulatory networks. PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA

Hastie T, Tibshirani R, Wainwright M (2015) Statistical learning with sparsity: the Lasso and generalizations. Monographs on statistics and applied probability 143. CRC Press, Chapman

Hobbs NT, Hooten MB (2015) Bayesian models: a statistical primer for ecologists. Princeton University Press, Princeton

Jensen FV, Nielsen TD (2007) Bayesian networks and decision graphs, 2nd edn. Springer, New York

Koski T, Noble J (2011) Bayesian networks: an introduction. Wiley, London

LaDeau SL, Han BA, Rosi-Marshall EJ, Weathers KC (2017) The next decade of big data in ecosystem science. Ecosystems 20:274–283

Last M (2006) The uncertainty principle of cross-validation. In: 2006 IEEE International conference on granular computing, 10–12 May 2006, pp 275–208

Lillegard M, Engen S, Saether BE (2005) Bootstrap methods for estimating spatial synchrony of fluctuating populations. Oikos 109:342–350

Marcot BG (2007) Étude de cas n°5: gestion de ressources naturelles et analyses de risques (Natural resource assessment and risk management). In: Naim P, Wuillemin P-H, Leray P, Pourret O, Becker A (eds) Réseaux Bayésiens (Bayesian networks; in French). Eyrolles, Paris, pp 293–315

Marcot BG (2012) Metrics for evaluating performance and uncertainty of Bayesian network models. Ecol Mod 230:50–62

Marcot BG, Penman TD (2019) Advances in Bayesian network modelling: integration of modelling technologies. Environ Model softw 111:386–393

Murphy KP (2012) Machine learning: a probabilistic perspective. The MIT Press, Cambridge

Pawson SM, Marcot BG, Woodberry O (2017) Predicting forest insect flight activity: a Bayesian network approach. PLoS ONE 12:e0183464

Pourret O, Naïm P, Marcot BG (eds) (2008) Bayesian belief networks: a practical guide to applications. Wiley, West Sussex

Scutari M (2010) Learning Bayesian networks with the bnlearn R package. J Stat Softw 35(3):1–22

Shcheglovitova M, Anderson RP (2013) Estimating optimal complexity for ecological niche models: a jackknife approach for species with small sample sizes. Ecol Mod 269:9–17

Stow CA, Webster KE, Wagner T, Lottig N, Soranno PA, Cha Y (2018) Small values in big data: the continuing need for appropriate metadata. Eco Inform 45:26–30

Van Valen L (2005) The statistics of variation. In: Hallgrímsson B, Hall BK (eds) Variation. Elsevier, Amsterdam, pp 29–47

Zhao Y, Hasan YA (2013) Machine learning algorithms for predicting roadside fine particulate matter concentration level in Hong Kong Central. Comput Ecol Softw 3:61–73