Alternative methods for interpreting Monte Carlo experiments

Zachary K. Collier, Haobai Zhang, and Olushola Soyoye

University of Delaware, Newark, Delaware, USA

Abstract
Research methodologists typically use descriptive statistics and plots to report the findings of Monte Carlo experiments. But previous literature suggests that Monte Carlo results deserve careful analysis rather than relying on simple descriptive statistics and plots of results, given the complex data conditions in simulation studies. As an alternative, data mining methods can also help readers digest Monte Carlo experiments. Therefore, our paper uses data mining methods to provide two novel contributions. First, we use detailed descriptions and code to illustrate how to use two data mining methods to analyze results from Monte Carlo experiments. Second, we demonstrate how data mining methods can be used in conjunction with interpreting plots, performing analysis of variance tests, and calculating effect sizes. Our study raises the awareness that there are alternative methods to interpretation and serves as a guide to readers for explaining the importance of manipulated conditions in Monte Carlo experiments.

Keywords
Data mining; Mixture model; Monte Carlo simulation

Contact
Zachary K. Collier
collierz@udel.edu
University of Delaware, Newark, Delaware, USA.

Introduction

Monte Carlo simulations allow researchers to investigate the relationship between the performance of model estimators and experimental attributes or conditions with generated data. Because social and behavioral science data do not always exhibit the attributes required for valid applications of a statistical method, Monte Carlo experiments have long been essential to research efforts within measurement and statistics (Robey and Barcikowski 1992; Xu, Fang, and Ying 2020). For example, Meyer's (2010) and Lu et al. (2020) Monte Carlo experiments tested finite mixture models that detect guessing on standardized tests.

The substance of our paper is the application of data mining techniques as an alternative and potentially more useful means to visualize and explain the results of Monte Carlo experiments. Our article is a response to calls for better reporting of Monte Carlo experiments in measurement (and other disciplines) (Hoaglin and Andrews 1975; Halperin 1976; Hauck and Anderson 1984; Boomsma 2013; Sechopoulos et al. 2018).

Often, quantitative researchers find Monte Carlo simulation studies difficult to conceptualize. Additionally, readers of simulation research seek guidance for decision-making about estimation for their specific data. Complicated-to-digest reporting can impact the reach and validity of Monte Carlo experiments. The use of Monte Carlo simulation in social and behavioral research will likely continue to increase as computer processor speed advances and studies become more elaborate.

In the present work, we illustrate classification trees and random forests approaches for analyzing the results of Monte Carlo experiments. These data mining techniques are well-known but are demonstrated in a step-by-step format, with guidelines and a detailed implementation for the purpose of reporting findings of simulation studies. We hope that research methodologists who commonly report works involving Monte Carlo experiments will consider data mining techniques as an alternative and potentially better means to visualize and explain results. To show the benefits and novelty of the proposed approach, we experiment on a popular topic in measurement and statistics, model fit. Then, we present and contrast results among traditional and data mining methods.

Related works

The downside to tradition

According to Boomsma's (2013) guidelines for reporting Monte Carlo simulations, the most widely used approaches to present and evaluate the relationship between outcomes and explanatory variables are basic descriptive and inferential methods. Many Monte Carlo experiments compare estimates with the known or ideal outcomes to obtain the response or outcome variables. In a recent survey of reporting simulation studies in statistical research (Harwell, Kohli, and Peralta-Torres 2018), authors found 99.9% of the simulation studies published in target journals between 1985 and 2012 limited analyses of simulation results to descriptive approaches (i.e., tables and plots). The experimental conditions with multiple levels, such as different levels of sample size or estimation methods, are commonly referred to as explanatory variables. An obvious downside of descriptively reporting the average performance is that it becomes constrained as the number of manipulated experimental factors and conditions increase.

Harwell, Kohli, and Peralta-Torres (2018) surveyed 677 published Monte Carlo studies published in educational/psychological methods and statistical journals (Biometrics, British Journal of Mathematical and Statistical Psychology, Biometrika, Journal of the American Statistical Association, Technometrics, and Psychological Methods), and found that less than 1% used analysis of variance (ANOVA), linear or logistic regression, or some other inferential procedure. In fact, approximately 99% used simple descriptive statistics and/or plots (these percentages are about the same for *Communications in Statistics – Simulation and Computation*).

Although these inferential methods are considered more straightforward and more readily understood compared with descriptive approaches, hindrances still exist when it comes to complex data conditions. Also, when computing and interpreting any effect size, it is important to refer to appropriate sources (Sawilowsky 2009; Yigit and Mendes 2018).

Another shortcoming of many parametric regression-based methods is the user-specification requirement for dealing with possible non-linear relationship between the performance variable and experimental conditions. These simple inferential methods are not capable of capturing all of the possible nonlinear or higher-order interactions automatically without explicit, manual specification by researchers. However, accurately specifying all appropriate interaction terms might be challenging, because this is subjective to researchers' understanding of the domain knowledge which results in possible misspecification of nonlinear patterns. So parametric regression-based methods cannot capture all possible, non-theorized interactions.

Novel applications of well-known methods

Data mining methods are non-parametric, meaning that they do not need to pre-assume the relationship between outcome variables and experimental factors. Such techniques are necessary for complex statistical conditions in Monte Carlo experiments. A central problem in data mining at-large involves choosing the best method for a given application. Using data mining methods to improve the interpretation of information from large amounts of data is well established.

Table 1. A summary of data mining techniques.

Data mining technique	Advantages	Disadvantages	Literatures	Monte Carlo performance outcome
Adaboost	Fast and easy to program	Vulnerable to uniform noise	Ying et al. (2013)	Categorical or Continuous
Artificial neural network	Simplicity; nonlinear; able to obtain high performance accuracy	Requires extensive training; often overfits; difficult to interpret	Rashid (2016)	Categorical or Continuous
Bayesian (belief) networks	Output is explicitly a probability; easy inspection and interpretability	No universally accepted method for construction	Canonne et al. (2016)	Categorical
Decision trees	Handling in nonnumeric data; simplicity; easy to visualize	Tends to overfit	Horning (2013)	Categorical or Continuous
Random forest	Maintains accuracy when large amounts of data are missing	Poor accuracy with continuous outcomes	Horning (2013)	Categorical or Continuous
Support vector machine	Classification without representing the feature space explicitly	Expressing the more complex prior information; analyzing limited samples	Xuegong (2000)	Categorical or Continuous

However, they are not commonly applied in the context of analyzing Monte Carlo experiments in social and behavioral research. Even more so, knowing how to choose the method is limited to experts in other disciplines such as computer science. Table 1 summarizes the advantages and disadvantages of data mining techniques to derive interpretable decisions from Monte Carlo experiments.

The following sections introduce two data mining methods, classification trees and random forest algorithms, in the applied context of analyzing Monte Carlo studies.

Classification trees

Classification and regression trees (CART) are widely used in the data mining field to visualize results with an intuitive tree diagram (Tang et al. 2021). Figure 1a shows a classification tree for a yes/no outcome. For each condition, f, the classification tree uses recursive binary splitting, which segments the categorical predictor into regions and puts each observation into the region with the most occurring cases (James et al. 2013). CART approaches use a greedy technique that automatically searches for nonlinear relations and underlying higher-order interactions among explanatory variables. More importantly, CART can facilitate the intuitive interpretation of our simulation results by visualizing all the situations in a recursive partitioning tree diagram. CART approaches can also be visually displayed in other two- and three-dimensional figures (see Figure 1b and 1c). For more information on decision tree visualization for high-dimensional data, readers are directed to Szücs and Schmidt (2018).

A recent study (Gonzalez et al. 2018) reanalyzed a published Monte Carlo study with classification trees. Gonzalez et al. suggested that CART was capable of providing similar conclusions compared to descriptive and inferential approaches. When the relationship between the predictors and the outcome in Monte Carlo simulations is not in linear functional form, CART may outperform conventional regression approaches in dealing with complex interactions which are not theorized (Gonzalez et al. 2018). CART automatically tests non-theorized interactions without

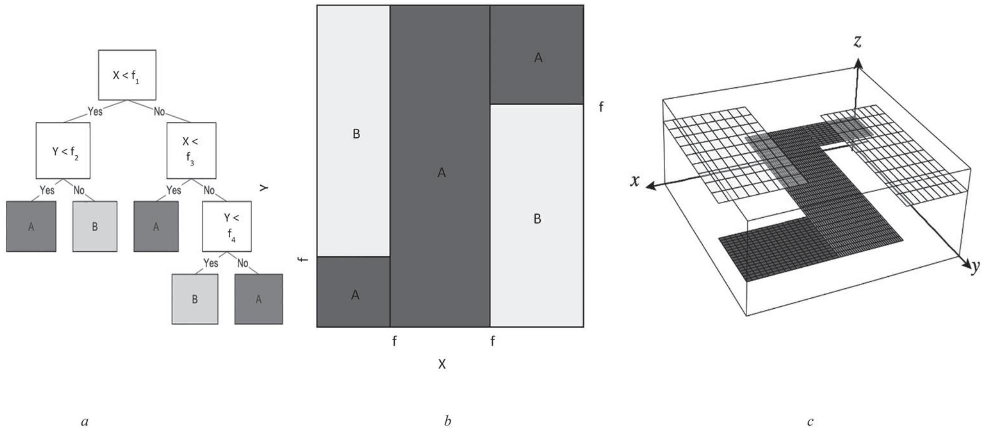

Figure 1. Two-dimensional (2-D) and three-dimensional (3-D) CART visualizations. (a) Top-down decision tree on a 2-D space, which is also shown in (b). Straight lines symbolize decision planes produced by the decision rules. (c) 3-D example with hyper-rectangles that overlap. Overlap oftentimes indicates data points that are not linearly separable (Szücs and Schmidt 2018).

manual specification, which can be subjective to the researchers' knowledge, and visualizes the results more intuitively.

Random Forest

Data miners may avoid CART, because interpreting a large number of splits can be difficult. Additionally, small changes in the data can lead to different initial splits and drastically different trees (Burgette and Reiter 2010). Other approaches, such as random forest, have shown to have better predictive performance compared with classification trees (Hastie, Tibshirani, and Friedman 2009; Chipman, George, and McCulloch 2010). The random forest algorithm uses multiple, uncorrelated CART models. From many "trees" programmers get a "forest." The results and output of the random forest algorithm is a set of variables ranked according to its power to select the conditions that best predict an outcome.

Random forests analyze Monte Carlo simulations by selecting iterations and specific manipulated conditions at random to create multiple classification trees and then averaging the results. When partitioning the data, however, random forests choose a subset of predictors at random. A tuning parameter is the number of variables selected at random as potential variables for partitioning the data. Tuning parameters such as the number of trees can increase the model's accuracy. More trees result in more computationally costly models, implying that the algorithm will take more steps to finish (Li et al. 2021). Later, we will discuss why we chose 501 trees and why this number varies from one analysis to the next.

Once the algorithm produces a large number of trees, each tree decides the most likely outcome on the performance measure. The outcome receiving the most selections across trees is the predicted outcome or classification. We built our random forest models using a method known as "bagging," in which all variables are potential variables for splitting the data (Breiman 1999).

For example, let us assume a sample S of Monte Carlo simulations.

$$S = \begin{matrix} f_{A1} & f_{B1} & f_{C1} & C_1 \\ \vdots & \vdots & \vdots & \vdots \\ f_{AI} & f_{BI} & f_{CI} & C_I \end{matrix} \tag{1}$$

Sample S includes I iterations, and f_{A1}, f_{B1}, and f_{C1} are conditions of our experiment. So f_{A1} is the first condition of the first iteration. Then, we continue up to the Ith iteration. In the final column of our sample, we have C_1 and C_I; which denotes having multiple conditions in our simulation study (Breiman 1999).

We will randomly select subsets S_1, S_2, S_M from our sample S of Monte Carlo simulations.

$$S_1 = \begin{matrix} f_{A12} & f_{B12} & f_{C12} & C_{12} \\ f_{A15} & f_{B15} & f_{C15} & C_{15} \\ \vdots & \vdots & \vdots & \vdots \\ f_{A35} & f_{B35} & f_{C35} & C_{35} \end{matrix} \quad S_2 = \begin{matrix} f_{A2} & f_{B2} & f_{C2} & C_2 \\ f_{A5} & f_{B5} & f_{C5} & C_5 \\ \vdots & \vdots & \vdots & \vdots \\ f_{A26} & f_{B26} & f_{C26} & C_{26} \end{matrix} \quad S_M = \begin{matrix} f_{A7} & f_{B7} & f_{C7} & C_7 \\ f_{A9} & f_{B9} & f_{C9} & C_9 \\ \vdots & \vdots & \vdots & \vdots \\ f_{A13} & f_{B13} & f_{C13} & C_{13} \end{matrix} \tag{2}$$

Let's assume that S_1 randomly selected iterations 12, 15 and 35, which is one third of the data. The remaining data from our simulation is often called, "out-of-bag data" (Breiman 1999). After we create a classification tree for S_1 the out-of-bag data will be used to test the tree and subsequently the entire forest of trees. The average misclassification across trees is known as the "out-of-bag error estimate" (Cutler, Cutler, and Stevens 2012).

Then, we can determine the most important conditions in our simulation by taking the mean decrease in accuracy (MDA). A score near zero indicates a poor relationship between given conditions and accuracy. MDA scores indicate how manipulated conditions are important for accurate cluster enumeration (Cutler, Cutler, and Stevens 2012).

Beyond using the typical ANOVA approach to analyze Monte Carlo simulation outcomes, we propose to apply the CART and random forest approaches. Using a Monte Carlo simulation that explores model fit in finite mixture modeling, the remainder of this paper compares the results of CART and random forest approaches to traditional methodologies. The model fit or performance metrics used for comparisons were the Bayesian information criterion (BIC; Schwarz 1978) and relative entropy (Kim et al. 2016). We chose these performance measures because many readers of this article apply them in isolation and will understand that each is sensitive to the conditions manipulated in our study. These considerations make for an ideal demonstration of how data mining techniques can aid in the interpretation of simulation complexities.

Methodology

Our demonstration evaluates cross-validation techniques and measures of model fit for detecting the number of clusters in latent profile analysis (LPA), a finite mixture model increasing in popularity within social and behavioral research (Bravo, Pearson, and Kelley 2018; De Clercq et al. 2019; Shim et al. 2020). A latent categorical variable has continuous indicators for LPA. Each latent profile p reveals itself through responses to a set of locally independent indicators $x_v(v = 1, \ldots, n)$. The LPA density function $\mathrm{f}(x_v\theta) =$ is given by

$$f(x_v|\theta) = \sum_{1}^{P} \pi_p f_p(x_v|\theta_p). \tag{3}$$

The probability of belonging to p is π_p and $f_p(x_v|\theta_p)$ is a profile-specific normal density function with profile-specific mean vector and covariance matrix $\theta_p = (\mu_p, \Sigma_p)$ (Collier, Zhang, and Johnson 2021).

We simulated LPA data using R version 3.6.0 (R Core Team 2019). Models were fit using Mplus 8.1 software. Overall, three factors were manipulated: sample size (N = 250, 500, 1,000, and 2,000), class separation (Mahalanobis distance (MD) = .5, .8, 1.2, and 2), and validation method (hold-out, k-fold, j x k-fold, and bootstrap). Fixed factors (i.e., the true number of latent classes, number of indicators, and the number of replications) were determined based on extensive review of previous simulation studies evaluating model fit for finite mixture models (Lo, Mendell, and Rubin 2001; Nylund, Asparouhov, and Muthén 2007; Tofighi and Enders 2008; He and Fan 2019). Fixed factors simulated in the population models were three latent profiles with five indicators. We simulated one thousand datasets for each combination of conditions manipulated. We generated a total of 64,000 datasets in this project and fitted each dataset to two, three, four, and five-class models.

Cross-validating latent profile analyses

Masyn (2013) explained a generic cross-validation approach, which we will refer to as "hold-out":

Step 1. Randomly select two subsamples with equal numbers of observations, a "calibration" dataset and a "validation" dataset.
Step 2. Using the calibration dataset from the previous step, compare model fit indices for $P+1$ profile models until the final model is chosen.
Step 3. Store the last model parameters estimated in Step 2.
Step 4. Fit the model to the validation dataset with the parameters from Step 3.
Step 5. If the parameters estimated with the calibration dataset fit the validation dataset, then the P profile model is supported (Collins et al. 1994; Collier, Zhang, and Johnson 2021).
Step 6. Fit the P profile model to the validation dataset
Step 7. Compare the model parameters from Step 6 with the model parameters from Step 3 using a nested-model likelihood ratio test. If there is not a significant decrement in fit for the fixed-parameter model, P latent profiles can be considered stable across the two subsamples. Hold-out is the simplest method of cross-validation and is prone to sample bias, because it requires random sampling of the data points for each sample. That may not be representative, because changing the splitting pattern may change the result (Kim 2009). We compare variates of hold-out cross-validation.

k-fold

Grimm, Mazza, and Davoudzadeh (2017) proposed using k-fold cross-validation as an additional step to validate the number of clusters in a mixture model. The main difference in this approach is that in Step 1, the researcher divides the original dataset into any number of k partitions. For example, if a dataset has 200 data points and $k=10$, there would be 20 observations in each split of data. Steps 2–7 are completed k times such that each partition is part of the calibration dataset $k-1$ times and part of the validation dataset one time. Grimm et al.'s recommendations were limited in that they derived from a single empirical study. Thus, the researchers had no way to determine if cross-validation selected the correct number of classes for the growth mixture model.

j x k-fold

Previous literature argues that variability of estimates is more important than bias and advocates for fewer k partitions, as in k-fold cross-validation. In j x k-fold cross-validation, j independent k-fold cross-validations are used to assess performance. So, in summary, j x k-fold is repeated k-fold cross-validation. The technique averages the k-fold estimate from j different partition choices. Although not tested in finite mixture models, empirical studies show that repeated cross-validation reduces variability of estimates (Chen et al. 2012; Vanwinckelen and Blockeel 2015; Jiang and Wang 2017), especially for smaller datasets (Rodríguez et al. 2010).

Bootstrap

Tibshirani and Efron (1993) introduced and fully described the bootstrap method. Given a dataset of size n, researchers in machine learning fields created a bootstrap sample by sampling n instances from the data with replacement (Kohavi 1995). Unlike "Step 1" for cross-validation techniques, bootstrapping draws with replacement. A bootstrapped dataset may contain multiple of the same cases, therefore, altogether omitting other cases. The remaining steps, two-seven, are the same for bootstrap methods.

Analysis

We evaluated the performance of cross-validation and bootstrap methods for selecting the accurate number of latent profiles in LPA. So, our main outcome variable was a binary indicator of replications that correctly enumerated the three-profile model based on two measures of model fit, the Bayesian information criterion (BIC; Schwarz 1978) and relative entropy (Kim et al. 2016). Schwarz (1978) proposed BIC based on Bayesian arguments:

$$BIC(P) = -2LogLiklihood(P) + k(\ln(N)) \quad . \tag{4}$$

where BIC has P latent profiles and includes the effect of the sensitivity of the likelihood function as a function of the k parameters.

Entropy is a type of statistic that measures latent cluster separation:

$$E_c = 1 - \frac{\Sigma_i \Sigma_p(-\widehat{prob}_{ip} \; ln \; \widehat{prob}_{ip})}{n\ln p} \tag{5}$$

where n denotes the number of observations and $\widehat{Prob}_{ip}$ is the conditional probability of an individual i belonging in profile p (Kim et al. 2016). Entropy with values approaching 1 indicates clear separation of profiles (Celeux and Soromenho 1996). A general rule of thumb for acceptable values of entropy is $\geq$.80. Lower values of entropy have significant effects on interpretability and fit of mixture models (Bakk et al. 2013, 2014; Lubke and Muthén 2007; Park et al. 2010; Vermunt 2010).

Analysis of variance

In order to explain the effects of the manipulated conditions, we used a mixed-design analysis of variance (ANOVA). Then, we calculated generalized eta squared ($G\ \eta_2$; Olejnik and Algina 2003) measures of effect size, which sorted the manipulated conditions concerning the magnitude of their effect on model fit indices. Two ANOVAs were fit for BIC and entropy. We used ANOVAs to differentiate the contributions from the manipulated factors of each study and possible interactions among them. In these mixed-design ANOVAs, the between-dataset factors were sample size, class separation, and the number of tested classes. The within-dataset factor was validation methods, with four levels: hold-out, k-fold, j x k-fold, and bootstrap. Effects with $G\ \eta^2$ above .001 were examined because they were large enough to qualify for interpretation (Olejnik and Algina 2003).

Classification trees

We used the rpart package in R-3.6.1 to visually represent the most likely conditions for selecting the correct number of latent profiles under all 64 conditions with classification trees. Below we provide pseudocode for our implementation of the CART algorithm in the supplementary material. The pseudocode is readable to methodologists who may not know Fortran, R, or other programming languages.

1. Randomly select 80% of the simulation for training and 20% for testing.
2. Select the most predictive manipulated conditions of the training dataset at the top or root of the classification tree.
3. Split the training set into equal subsets for each condition.
4. Repeat steps one and two on each subset until you find leaf nodes in all the branches of the classification tree.
5. Take the simulations for testing and use the trained classification tree model to predict the correct number of latent profiles.

To evaluate the performance of this data mining implementation, we calculated error rates by using a misspecification table and adding the incorrect number of classifications divided by the total number of observations in the test dataset.

Random Forest

We used the random forest algorithm to extract from the Monte Carlo simulations the conditions that were most relevant for selecting the correct number of latent profiles. We estimated the importance of manipulated conditions by determining the percentage increase in prediction error, arising from the exclusion of a manipulated condition. Thus, conditions with large values contribute most to the prediction accuracy in our simulation study (Hardman, Paucar-Caceres, and Fielding 2013). As with the classification tree analyses, the random forest algorithm predicted binary indicators of whether or not the performance measures matched with the correct number of latent profiles. We employed the *randomForest* package (Liaw and Wiener 2002) to implement Breiman's (2001) random forest algorithm for classification as follows (R code is available in the supplementary material):

1. Randomly select 80% of the simulation for training and 20% for testing.
2. Randomly select c manipulated conditions from the total C conditions (i.e., each combination of the manipulated factors) and i iterations from the total of I iterations; where $c < C$ and $i < I$
3. Among the c conditions, calculate the node "d" using the best split (same as step three in the classification tree algorithm).
4. Split the root node into subsequent (i.e., daughter) nodes using the best split.
5. Repeat steps one through three.
6. Build forest by repeating step 2 through 5 for 501 times to create 501 trees.

Step 5 demonstrates the bagging approach described earlier, where each tree in our model is independent and constructed using a bootstrapped sample of the full dataset of Monte Carlo simulations. The *randomForest* package uses 500 trees by default, but we chose to make the number of trees odd to ensure a "winning" decision or classification. Other numbers of trees (e.g., 100 or 250) may yield more accurate results depending on the conditions of the data inputted into the algorithm (Latinne, Debeir, and Decaestecker 2001). To predict whether or not the performance measures will select the correct number of classes using the algorithm above, we did the following (R code is available in supplementary material):

1. Take 20% of the test simulations and apply the rules of each randomly created classification tree to predict the correct number of classes.

Table 2. Descriptive statistics of class enumeration.

Performance measure	Correct enumeration	Bootstrap	Hold-out	j x k-fold	k-fold
Entropy	No	70.80%	71.10%	71.60%	34.40%
	Yes	29.20%	28.90%	28.40%	65.60%
BIC	No	88.60%	58.70%	58.00%	30.60%
	Yes	11.40%	41.30%	42.00%	69.40%

2. Calculate the decision for each prediction.
3. Determine the most reoccurring decision. Use this decision as the final prediction.
4. Calculate the MDA score.

Comparing the interpretability of results

Here, we present the results of our simulation study. As stated previously, cross-tables of manipulated conditions and outcome variables are a commonly used form of presenting relationships descriptively. Table 2 shows descriptive statistics of class enumeration for each measure of model fit (BIC and entropy) and validation technique (hold-out, k-fold, j x k-fold, and bootstrap).

Descriptive statistics

K-fold cross-validation yielded the highest percentage of correct class enumeration using BIC (69.40%) and entropy (65.60%). All other validation methods performed similarly in terms of entropy, ranging between 28.40% and 29.20% accuracy. For BIC, the bootstrap method had the lowest percentage (11.40%) of accurate class enumeration.

ANOVA results

In this section, we provide the effect of the manipulated conditions as detected with ANOVA and the $G\ \eta^2$ effect size measure.

Bayesian information criterion

We detected significant effects of sample size ($G\ \eta^2 = .99$), MD ($G\ \eta^2 = .80$), the number of tested classes ($G\ \eta^2 = .53$), and validation method ($G\ \eta^2 = .99$) on BIC. All possible combinations of interactions between manipulated conditions had significant effects. The interaction with the largest effect was between sample size and the validation method ($G\ \eta^2 = .99$). Figure 2 displays the main effects and interactions for sample size and methods of validation on BIC.

Entropy

There was a significant main effect of each of our manipulated conditions. The interaction with the greatest effect was between sample size and validation method ($G\ \eta^2 = .44$), Figure 3 shows the main effects of sample size ($G\ \eta^2 = .77$), validation method ($G\ \eta^2 = .81$), and MD ($G\ \eta^2 = .64$) on entropy. Starting from the left, Figure 3 shows the negative relationship between sample size and entropy. The second plot shows similar entropy values across the bootstrap, hold-out and j x k-fold methods. The k-fold method averaged higher values of entropy compared with other validation techniques. The third plot from the left shows the positive relationship between MD and entropy.

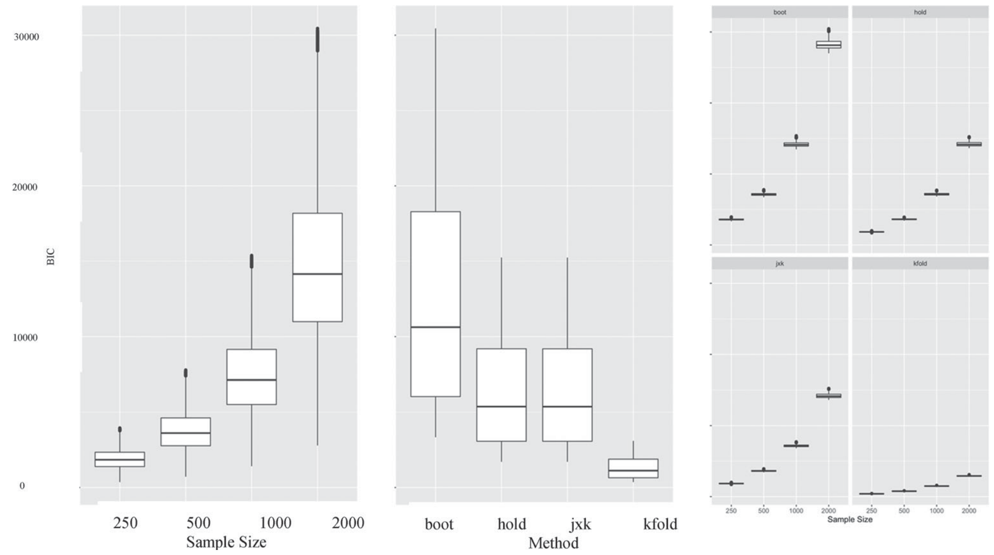

Figure 2. Main effects and two-way interaction effects of sample size and validation method on BIC.

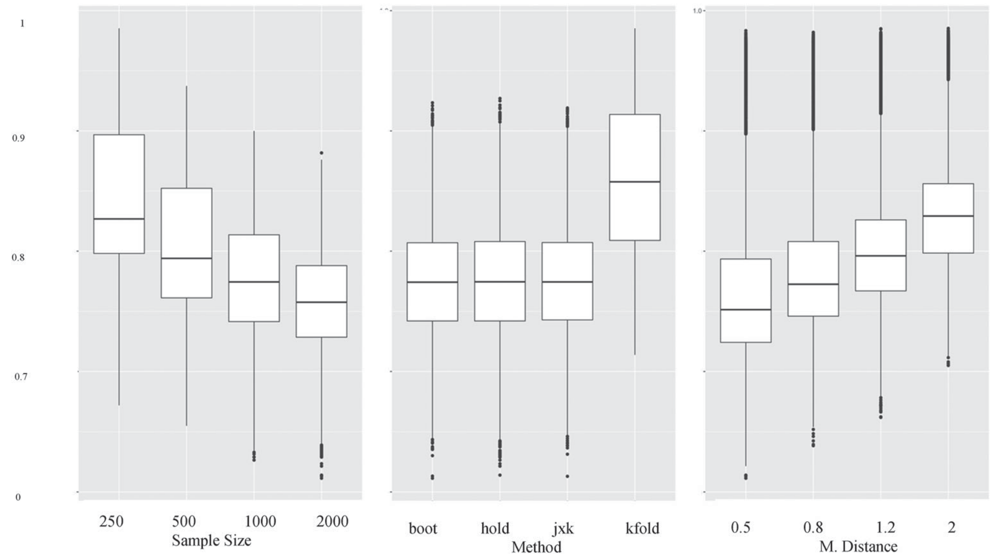

Figure 3. Main effects of sample size, validation method, and MD on entropy.

Classification tree results

Figure 4 presents classification trees for BIC and entropy. Each tree in Figure 4 begins with a predictor variable that has the greatest power to predict the outcome (i.e., a yes/no indicator of correct profile enumeration). After the first split, the process continues with each subgroup treated as an independent group for further splitting.

Bayesian information criterion

Our classification error was 12.59% on our test dataset, which represents the fraction of the misclassification. Implementing the bootstrap method for model validation was the strongest

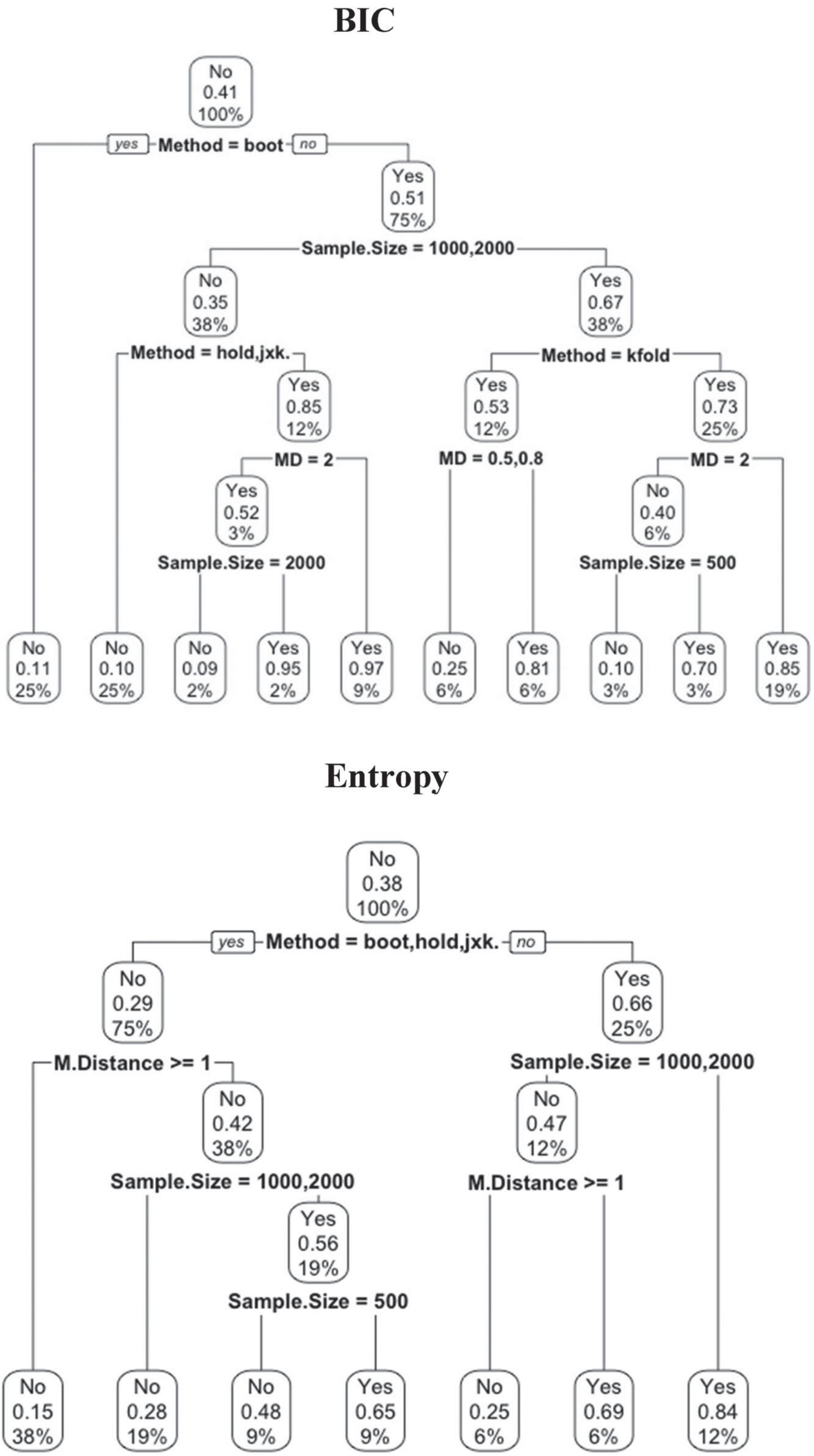

Figure 4. A decision tree classifier of the LPA Monte Carlo simulation results. Note: Each line leads to a node at which tests are applied to split the data into successively smaller groups recursively. The labels (Yes, No) refer to most likely cases of selecting the correct number of profiles. Nodes display the proportion of labels and the percentage of the correct classification from the dataset at that node.

predictor of whether or not the BIC selected the correct number of profiles. The bootstrap method correctly identified the number of profiles for 25% of the simulated datasets. The probability of selecting the correct number of profiles with the bootstrap method was 0.11, which pertained to 25% of our Monte Carlo simulation. The k-fold method had the highest probability (0.97) of selecting the correct number of profiles when the sample size was either 1,000 or 2,000, and when MD was 0.5, 0.8. or 1.2. When sample size was 2,000 and MD was 2, the k-fold method correctly identified the number of profiles for 2% of the simulation.

Entropy

The classification error was 25.02% in the test dataset. Entropy most likely (probability = 0.84) selected the correct number of latent profiles with the k-fold method and when sample size was 250 and 500. Entropy least likely (probability = 0.15) identified the correct number of profiles when we implemented the other methods and when MD was 1.2 or 2. Entropy was accurate for 27% of our simulation study.

Random Forest results

Bayesian information criterion

The out-of-bag error rate was 14.17%. Therefore, accuracy was 85.83% for the number of correctly predicted outcomes from our out-of-bag data. Each of our manipulated conditions impacted the selection of the correct number of latent profiles, indicated by the mean difference in accuracy (MDA) values higher than zero. The rankings were (1) Method with MDA = 2989.47, (2) Sample size with MDA = 1580.69, and (3) MD with MDA = 470.94.

Entropy

The error rate was 24%. Therefore, our random forest model correctly predicted out-of-bag data, with 76% accuracy. Same as with BIC, the cross-validation method, sample size, and MD impacted the selection of the correct number of latent profiles. The ranking of importance was (1) Method with MDA = 1035.98, (2) Sample size with MDA = 977.39, and (3) MD with MDA = 904.69. It was not feasible to include 501 classification trees from our random forest in this paper. However, at the end of this article we provided code to create the model on GitHub (Jgozal 2018). A random forest plot would show “Method” as the root node, indicating the most important variable among manipulated conditions.

Discussion

We expected similarities across the results of conventional and data mining approaches because previous literature has well-established the high sensitivity of model fit for finite mixture models (Henson, Reise, and Kim 2007; Masyn 2013; Grimm et al. 2016; He and Fan 2019). As expected, the Monte Carlo experiment favored the k-fold cross-validation method, and each condition affected the model fit.

Similarities between approaches are most clearly visible when comparing results with BIC. Starting at the root node of the CART analysis, the choice of the cross-validation method and sample size are the strongest predictors. For the same conditions using random forest and ANOVA, the MDA and $G\ \eta^2$ had the highest values. Similarities in the results are worth noting because these data mining approaches can be used as an alternative to traditional methods to confirm theory and aid interpretation.

In our simulation, ANOVA and effect size measures informed us of significant interactions. However, we needed to refer to tables of descriptive statistics and box-and-whisker plots to better under each interaction. The data mining approaches innately provided us with visualizations of each interaction. Figure 4 is an example of how classification trees can explicitly and visually guide applied researchers based on results from a Monte Carlo simulation. Under more complex simulation conditions, our trees may have had a larger number of nodes and thus would have been harder to comprehend all of the splits that denote interactions. In such a case, we recommend users perform one of the following methods:

1. Terminate the growth of the tree at a point that is effectively interpretable by monitoring the error rate.
2. After the tree is at its largest, cut (i.e., prune) it to a more interpretable size.

Our classification error rates provided us with a means to evaluate the performance of our data mining algorithms. Although our error rates were relatively low and suggest adequate performance, other measures of error and accuracy exist and provide more insight into model performance. Readers interested in other evaluation methods are encouraged to refer to Tan, Steinbach, and Kumar (2016) as a resource.

Conclusion

Because of the massive size of data in most Monte Carlo simulations and the need to procure digestible results for applied researchers, data mining is an appropriate family of models. We used classification trees and random forest techniques in our example study, but other data mining approaches are available (e.g., neural networks and Bayesian additive regression trees). For example, neural networks have more tuning parameters compared to CART. All the tuning parameters (also known as "hyperparameters") can adjust to regulate how neural networks learn between the input and output (Collier and Leite 2020). Further investigation should consider such methods because of their potential for improved predictive performance. However, methodologists should be aware that these approaches are typically more computationally intensive (Collier, Zhang, and Liu 2022).

As an alternative to the two-step approach of performing an ANOVA and calculating effect sizes, researchers could use classification trees because they are nonparametric, and results can be presented in a more intuitively interpretable way. Or researchers could employ random forest to increase the accuracy of classification trees and measure the importance of manipulated conditions. Moreover, data mining can be applied to all Monte Carlo simulation studies because it can improve interpretation and is more robust to assumption violations. Although researchers can generate an orthogonal design with Monte Carlo simulations and avoid multicollinearity issues, one limitation of using CART is that any deviations from the orthogonality that may cause structural zeroes involved would result in producing biased estimates.

Also, worth noting is the difference between significance and practically meaningful findings. Monte Carlo studies typically require hundreds of replications per condition, which produces significant p values in traditional regression-based methods (Gonzalez et al. 2018). This is precisely why researchers use effect sizes like $G\ \eta^2$ post hoc to summarize Monte Carlo results. As an alternative, researchers could implement data mining approaches, which require large numbers of replications to obtain accurate results.

In response to calls for rigorous analyses and better ways of presenting results from Monte Carlo experiments (Harwell, Kohli, and Peralta-Torres 2018; Hoaglin and Andrews 1975), our present study introduced data mining-based approaches to analyze the results from simulation experiments in the LPA example. We showed that the results analyzed with the new methods are

Accepted Manuscript
Version of record at: https://doi.org/10.1080/03610918.2022.2082474

as informative as the traditional methods. More importantly, going beyond advocating the recommendations of Hoaglin and Andrews (1975), we addressed the importance of presenting results in a more intuitively interpretable and visualized way for simulation studies. We encourage methodologists to try these new approaches in analyzing their simulation results for the following reason. With the visualization of the results, simulation studies may become more friendly to applied researchers who hope to seek answers for the best methodological solution to their empirical research questions. For example, when applied researchers who conduct LPA with empirical data run into the issue of disagreement of model fit indices, they can easily match the branch in this study's classification tree with their data condition. In this way, they can get support from simulation studies in a more intuitive way. Applied researchers would struggle less to find the most useful information in numerous descriptive tables and complex results from mixed-model factorial ANOVA given the complex data conditions simulated in most Monte Carlo studies. In the present and future studies, we hope to build a bridge between simulation studies and empirical studies that use the same models. With our proposed approaches, the simulation work from methodologists will attract a broader audience in the applied field. So that methodologists can better deliver their expertise and recommendations in simulations into practical use. For applied researchers, they can obtain information from simulation studies that benefit their research in an easier accessible way.

We provided pseudocode to clarify the flow and for loops in our data mining algorithms. For readers interested in adapting our techniques to their own Monte Carlo simulations, please adapt our R code on GitHub (https://github.com/collierlaboratory/Data-Mining-for-Reporting-the-Results-of-Monte-Carlo-Simulations).

ORCID

Zachary K. Collier http://orcid.org/0000-0003-2526-5120

References

Bakk, Z., D. L. Oberski, and J. K. Vermunt. 2014. Relating latent class assignments to external variables: Standard errors for correct inference. *Political Analysis* 22 (4). doi:10.1093/pan/mpu003.

Bakk, Z., F. B. Tekle, and J. K. Vermunt. 2013. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. *Sociological Methodology* 43 (1): 272–311. doi:10.1177/0081175012470644.

Boomsma, A. 2013. Reporting Monte Carlo studies in structural equation modeling. *Structural Equation Modeling* 20 (3):518–40. doi:10.1080/10705511.2013.797839.

Bravo, A. J., M. R. Pearson, and M. L. Kelley. 2018. Mindfulness and psychological health outcomes: A latent profile analysis among military personnel and college students. *Mindfulness* 9 (1):258–70. doi:10.1007/s12671-017-0771-5.

Breiman, L. 1999. Random forests. *UC Berkeley TR567*.

Breiman, L. 2001. Random forests. *Machine Learning* 45 (1):5–32. doi:10.1023/A:1010933404324.

Burgette, L. F., and J. P. Reiter. 2010. Multiple imputation for missing data via sequential regression trees. *American Journal of epidemiology* 172 (9):1070–6. doi:10.1093/aje/kwq260.

Canonne, C., I. Diakonikolas, D. Kane, and A. Stewart. 2016. Testing bayesian networks. *arXiv preprint arXiv: 1612.03156*.

Chen, C., Y. Wang, Y. Chang, and K. Ricanek. 2012. Sensitivity analysis with cross-validation for feature selection and manifold learning. In *International symposium on neural networks*, 458–67. Berlin, Heidelberg: Springer.

Chipman, H. A., E. I. George, and R. E. McCulloch. 2010. BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 4 (1):266–98. doi:10.1214/09-AOAS285.

Celeux, G., and G. Soromenho. 1996. An entropy criterion for assessing the number of clusters in a mixture model. *Journal of Classification* 13 (2):195–212. doi:10.1007/BF01246098.

Collier, Z. K., and W. L. Leite. 2020. A tutorial on artificial neural networks in propensity score analysis. *The Journal of Experimental Education*: 1–18.

Collier, Z. K., H. Zhang, and B. Johnson. 2021. Finite mixture modeling for program evaluation: Resampling and pre-processing approaches. *Evaluation Review* 45 (6):309–33. doi:10.1177/0193841X211065619.

Collier, Z. K., H. Zhang, and L. Liu. 2022. Explained: Artificial intelligence for propensity score estimation in multilevel educational settings. *Practical Assessment, Research, and Evaluation* 27 (1):3.

Collins, L. M., J. W. Graham, J. D. Long, and W. B. Hansen. 1994. Crossvalidation of latent class models of early substance use onset. *Multivariate Behavioral Research* 29 (2):165–83. doi:10.1207/s15327906mbr2902_3.

Cutler, A., D. R. Cutler, and J. R. Stevens. 2012. Random forests. In *Ensemble machine learning*, 157–75. Boston, MA: Springer.

De Clercq, M., B. Galand, and M. Frenay. 2019. One goal, different pathways: Capturing diversity in processes leading to first-year students' achievement. *Learning and Individual Differences* 81: 101908.

Grimm, K. J., N. Ram, and R. Estabrook. 2016. *Growth modeling: Structural equation and multilevel modeling approaches*. Guilford Publications.

Grimm, K. J., G. L. Mazza, and P. Davoudzadeh. 2017. Model selection in finite mixture models: A k-fold cross-validation approach. *Structural Equation Modeling* 24 (2):246–56. doi:10.1080/10705511.2016.1250638.

Gonzalez, O., H. P. O'Rourke, I. C. Wurpts, and K. J. Grimm. 2018. Analyzing Monte Carlo simulation studies with classification and regression trees. *Structural Equation Modeling* 25 (3):403–13. doi:10.1080/10705511.2017.1369353.

Halperin, S. 1976. Design of Monte Carlo Studies.

Hardman, J., A. Paucar-Caceres, and A. Fielding. 2013. Predicting students' progression in higher education by using the random forest algorithm. *Systems Research and Behavioral Science* 30 (2):194–203. doi:10.1002/sres.2130.

Harwell, M., N. Kohli, and Y. Peralta-Torres. 2018. A survey of reporting practices of computer simulation studies in statistical research. *The American Statistician* 72 (4):321–7. doi:10.1080/00031305.2017.1342692.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman. 2009. *The elements of statistical learning: Data mining, inference, and prediction* Vol. 2, pp. 1–758. New York: springer.

Hauck, W. W., and S. Anderson. 1984. A survey regarding the reporting of simulation studies. *The American Statistician* 38(3): 214–6.

He, J., and X. Fan. 2019. Evaluating the performance of the k-fold cross-validation approach for model selection in growth mixture modeling. *Structural Equation Modeling: A Multidisciplinary Journal* 26 (1): 66–79. doi:10.1080/10705511.2018.1500140.

Henson, J. M., S. P. Reise, and K. H. Kim. 2007. Detecting mixtures from structural model differences using latent variable mixture modeling: A comparison of relative model fit statistics. *Structural Equation Modeling* 14 (2): 202–26. doi:10.1080/10705510709336744.

Hoaglin, D. C., and D. F. Andrews. 1975. The reporting of computation-based results in statistics. *The American Statistician* 29 (3): 122–6.

Horning, N. 2013. Introduction to decision trees and random forests. *American Museum of Natural History* 2: 1–27.

James, G., D. Witten, T. Hastie, and R. Tibshirani, 2013. *An introduction to statistical learning*. Vol. 112, p. 18. New York: Springer.

Jgozal. 2018. How to actually plot a sample tree from randomForest::getTree()? URL (version: 2018-01-11). https://stats.stackexchange.com/q/241684

Jiang, G., and W. Wang. 2017. Error estimation based on variance analysis of k-fold cross-validation. *Pattern Recognition* 69:94–106. doi:10.1016/j.patcog.2017.03.025.

Kim, J. H. 2009. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis* 53 (11):3735–45. doi:10.1016/j.csda.2009.04.009.

Kim, S. H., C. Jung, and Y. J. Lee. 2016. An entropy-based analytic model for the privacy-preserving in open data. In 2016 IEEE International Conference on Big Data (Big Data), IEEE, pp. 3676–84.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI* 14 (2):1137–45.

Latinne, P., O. Debeir, and C. Decaestecker. 2001. Limiting the number of trees in random forests. In *International workshop on multiple classifier systems*, 178–87. Berlin, Heidelberg: Springer.

Li, R., H. Wang, Y. Zhao, J. Su, and W. Tu. 2021. Robust estimation of heterogeneous treatment effects: an algorithm-based approach. *Communications in Statistics-Simulation and Computation*: 1–18.

Liaw, A., and M. Wiener. 2002. Classification and regression by randomForest. *R News* 2 (3):18–22.

Lo, Y., N. R. Mendell, and D. B. Rubin. 2001. Testing the number of components in a normal mixture. *Biometrika* 88 (3):767–78. doi:10.1093/biomet/88.3.767.

Lu, J., C. Wang, J. Zhang, and J. Tao. 2020. A mixture model for responses and response times with a higher-order ability structure to detect rapid guessing behaviour. *The British Journal of Mathematical and Statistical Psychology* 73 (2):261–88. doi:10.1111/bmsp.12175.

Lubke, G., and B. O. Muthén. 2007. Performance of factor mixture models as a function of model size, covariate effects, and class-specific parameters. *Structural Equation Modeling: A Multidisciplinary Journal* 14 (1): 26–47. doi:10.1080/10705510709336735.

Masyn, K. E. 2013. Latent class analysis and finite mixture modeling. In T. D. Little (Ed.), *Oxford library of psychology. The Oxford handbook of quantitative methods: Statistical analysis*, 551–611. Oxford University Press.

McLachlan, G. J., and D. Peel. 2000. *Finite mixture models*. New York, NY: Wiley.

Meyer, J. P. 2010. A mixture Rasch model with item response time components. *Applied Psychological Measurement* 34 (7): 521–8. doi:10.1177/0146621609355451.

Nylund, K. L., T. Asparouhov, and B. O. Muthén. 2007. Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. *Structural Equation Modeling* 14 (4):535–69. doi:10.1080/10705510701575396.

Olejnik, S., and J. Algina. 2003. Generalized eta and omega squared statistics: Measures of effect size for some common research designs. *Psychological methods* 8 (4):434–47. doi:10.1037/1082-989X.8.4.434.

Park, B. J., D. Lord, and J. D. Hart. 2010. Bias properties of Bayesian statistics in finite mixture of negative binomial regression models in crash data analysis. *Accident Analysis & Prevention* 42 (2): 741–9. doi:10.1016/j.aap.2009.11.002.

Rashid, T. 2016. *Make your own neural network*, p. 222. CreateSpace Independent Publishing Platform.

Robey, R. R., and R. S. Barcikowski. 1992. Type I error and the number of iterations in Monte Carlo studies of robustness. *British Journal of Mathematical and Statistical Psychology* 45 (2):283–8. doi:10.1111/j.2044-8317.1992.tb00993.x.

Rodríguez, J. D., A. Pérez, and J. A. Lozano. 2010. Sensitivity analysis of kappa-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine intelligence* 32 (3):569–75. doi:10.1109/TPAMI.2009.187.

Sawilowsky, S. S. 2009. New effect size rules of thumb. *Journal of Modern Applied Statistical Methods* 8 (2):597–9. doi:10.22237/jmasm/1257035100.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 461–4.

Sechopoulos, I., D. Rogers, M. Bazalova-Carter, W. E. Bolch, E. C. Heath, M. F. McNitt-Gray, J. Sempau, and J. F. Williamson. 2018. RECORDS: Improved Reporting of montE CarlO RaDiation transport Studies: Report of the AAPM Research Committee Task Group 268. *Medical Physics* 45 (1):e1–e5. doi:10.1002/mp.12702.

Shim, E. J., D. Jeong, H. G. Moon, D. Y. Noh, S. Y. Jung, E. Lee, Z. Kim, H. J. Youn, J. Cho, and J. E. Lee. 2020. Profiles of depressive symptoms and the association with anxiety and quality of life in breast cancer survivors: A latent profile analysis. *Quality of Life Research* 29 (2):421–9. doi:10.1007/s11136-019-02330-6.

Szücs, D., and F. Schmidt. 2018. *Decision tree visualization for high-dimensional numerical data*. 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), Valencia, IEEE, 190–5, doi:10.1109/SNAMS.2018.8554961.

Tan, P. N., M. Steinbach, and V. Kumar. 2016. *Introduction to data mining*. Pearson Education India.

Tang, J., S. Lewis, L. Cutler, R. Hallam, and Z. Collier. 2021. Characteristics of home-based child care providers who offer non-standard hour care. *Early Childhood Research Quarterly* 55:284–94. doi:10.1016/j.ecresq.2020.12.005.

Team, R. C. 2019. *R: A language and environment for statistical computing*. Vienna, Austria.

Tibshirani, R. J., and B. Efron. 1993. An introduction to the bootstrap. *Monographs on Statistics and Applied Probability* 57: 1–436.

Tofighi, D., and C. K. Enders. 2008. Identifying the correct number of classes in growth mixture models. *Advances in Latent Variable Mixture Models* 2007 (1): 317.

Vanwinckelen, G., and H. Blockeel. 2015. *Look before you leap: Some insights into learner evaluation with cross-validation*. Proceedings of the Workshop on Statistically Sound Data Mining at ECML/PKDD, in PMLR 47:3-20

Vapnik, V., and O. Chapelle. 2000. Bounds on error expectation for support vector machines. *Neural Computation*, 12 (9):2013–36.

Vermunt, J. K. 2010. Latent class modeling with covariates: two improved three-step approaches. *Political Analysis* 18 (4): 450–69. doi:10.1093/pan/mpq025.

Xu, H., G. Fang, and Z. Ying. 2020. A latent topic model with Markov transition for process data. *The British Journal of Mathematical and Statistical psychology* 73 (3):474–505. doi:10.1111/bmsp.12197.

Xuegong, Z. 2000. Introduction to statistical learning theory and support vector machines. *Acta Automatica Sinica* 26 (1).

Yigit, S., and M. Mendes. 2018. Which effect size measure is appropriate for one-way and two-way ANOVA models? A Monte Carlo simulation study. *Revstat Statistical Journal* 16:295–313.