INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Reduced Modeling Approach for Making Predictions with Incomplete Data Having Blockwise Missing Patterns

Karthik Srinivasan; , Faiz Currim, Sudha Ram

To cite this article:
Karthik Srinivasan; , Faiz Currim, Sudha Ram (2025) A Reduced Modeling Approach for Making Predictions with Incomplete Data Having Blockwise Missing Patterns. INFORMS Journal on Data Science 4(1):85-99. https://doi.org/10.1287/ijds.2022.9016

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Vol. 4, No. 1, January–March 2025, pp. 85–99
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

A Reduced Modeling Approach for Making Predictions with Incomplete Data Having Blockwise Missing Patterns

Karthik Srinivasan,[a,*] Faiz Currim,[b] Sudha Ram[b]

[a] School of Business, University of Kansas, Lawrence, Kansas 66045; [b] Department of MIS, Eller College of Management, University of Arizona, Tucson, Arizona 85721

*Corresponding author

Contact: karthiks@ku.edu, https://orcid.org/0000-0002-1608-6190 (KS); currim@arizona.edu, https://orcid.org/0000-0002-5025-811X (FC); sram@arizona.edu, https://orcid.org/0000-0001-6053-1311 (SR)

Received: May 27, 2022
Revised: May 29, 2023; February 7, 2024; August 7, 2024
Accepted: September 18, 2024
Published Online in Articles in Advance: November 26, 2024

https://doi.org/10.1287/ijds.2022.9016

Abstract. Incomplete data with blockwise missing patterns are commonly encountered in analytics, and solutions typically entail listwise deletion or imputation. However, as the proportion of missing values in input features increases, listwise or columnwise deletion leads to information loss, whereas imputation diminishes the integrity of the training data set. We present the blockwise reduced modeling (BRM) method for analyzing blockwise missing patterns, which adapts and improves on the notion of reduced modeling proposed by Friedman, Kohavi, and Yun in 1996 as lazy decision trees. In contrast to the original idea of reduced modeling of delaying model induction until a prediction is required, our method is significantly faster because it exploits the blockwise missing patterns to pretrain ensemble models that require minimum imputation of data. Models are pretrained over the overlapping subsets of an incomplete data set that contain only populated values. During prediction, each test instance is mapped to one of these models based on its feature-missing pattern. BRM can be applied to any supervised learning model for tabular data. We benchmark the predictive performance of BRM using simulations of blockwise missing patterns on three complete data sets from public repositories. Thereafter, we evaluate its utility on three data sets with actual blockwise missing patterns. We demonstrate that BRM is superior to most existing benchmarks in terms of predictive performance for linear and nonlinear models. It also scales well and is more reliable than existing benchmarks for making predictions with blockwise missing pattern data.

History: Maytal Saar-Tsechansky served as the senior editor for this article.
Data Ethics & Reproducibility Note: The code capsule is available on Code Ocean at https://codeocean.com/capsule/0274716/tree and in the e-Companion to this article (available at https://doi.org/10.1287/ijds.2022.9016).

Keywords: reduced modeling • blockwise missing patterns • incomplete data • predictive modeling • complex data analysis

1. Introduction

It is well known that incomplete data or the presence of missing values in a data set can pose challenges for data analysis. Not addressing the missing values issue can lead to biased parameter estimation, faulty conclusions about predictor-response effects, and deterioration in the predictive performance of models (Schafer and Graham 2002). Irrespective of the type of data and problem domain, missing values have been studied for several decades. Prior studies have primarily focused on problems with a small (<5%) to moderate (<20%) proportion of missingness in the input feature space (Rubin 1976, van Buuren 2018). Practical applications in data science often grapple with moderate to high missingness (>25%) that may be systematically missing for several reasons. For instance, multisource data integration or performing outer joins in relational databases could generate an incomplete resultant data set with a high proportion of missingness in some critical input features. Figure 1 illustrates blockwise missing values generated by doing an outer join from two source tables. Another typical example of such incomplete data are from surveys that accommodate follow-up questions or record items with optional nonresponses. In contrast to randomly missing values within features, these incomplete data sets with moderate to high missingness have spurts of data populated followed by missingness (i.e., blockwise missingness), as shown in Table 1. Although literature is abundant in analyzing randomly occurring missing values (Rubin 1976, van Buuren 2018), there needs to be more literature focusing on incomplete data with blockwise missing patterns, a recent occurrence in data science applications. Blockwise missingness is a phenomenon where missing values are not randomly distributed across the data set but occur in blocks or contiguous groups of observations and variables. Blockwise missing patterns may occur due to tractable or intractable reasons including systematic measurement errors,

Figure 1. Outer Joins in Relational Databases Resulting in Data Sets with Blockwise Missing Patterns

Patient ID	Hemoglobin levels	Mean Heart rate
101	300	70
102	350	110
103	200	120
104	345	90
105	320	80

⟗

Patient ID	Mean sitting bout (%)	Caffeine consumption/hour
103	80.0	0.2
104	74.5	1.4
105	88.5	0.1
106	81.1	
107	63.2	0.8

Patient ID	Hemoglobin levels	Mean Heart rate	Mean sitting bout (%)	Caffeine consumption/hour
101	300	70		
102	350	110		
103	200	120	80.0	0.2
104	345	90	74.5	1.4
105	320	80	88.5	0.1
106			81.1	
107			63.2	0.8

study design, data entry errors, systematic survey nonresponse, or data fusion challenges.

There are three general approaches for processing and analyzing incomplete data: (i) pruning, which is reviewing and systematically removing sparse rows/columns; (ii) choice of special models, which is selecting models that can inherently handle missing values such as classification and regression trees (Breiman et al. 1984), Bayesian networks with model selection using the expectation maximization method (Friedman 1997), and so on; and (iii) treatment, which is using imputation methods to replace missing values in a data set with estimated values (Schafer and Graham 2002). The first approach is practical when the ratio of missing values to populated values in the data set is low, and the pruned data set is still representative of the initial data set. The second approach precludes the use of several commonly adopted models, such as regression models, ensemble learners, deep neural networks, and so on, that require complete data. In the third approach, incomplete data are transformed into a complete data set by imputing values using machine learning or statistical methods. This approach also has limitations if the proportion of missing values is significant.

Table 1. Example of Incomplete Data with Blockwise Missing Patterns

Observation	Feature 1	Feature 2	Feature 3	Feature 4
1		56.09	0.35	
2		56.10	0.39	
3		56.10	0.43	
4	55.50			755.38
5	45.43			655.70
6	43.54			
7	44.93			
8	45.24			
9	50.33			
10	42.21	58.33	0.52	664.20
11	56.99	55.82	0.47	709.50

An alternative (fourth) approach proposed for predictive modeling with incomplete data is to employ only those features in model training that are known for a given test data instance, a technique termed reduced modeling or lazy decision tree (Friedman et al. 1996, Schuurmans and Greiner 1997, Saar-Tsechansky and Provost 2007). That is, for each pattern of missing features observed in the test instances, a different model is trained over the training data. The classical implementation of this approach is unpopular in practice due to the prohibitive cost of storing all possible models or computing models at run time. However, we posit that reduced modeling can be exploited for the problem of analyzing incomplete data with blockwise missing patterns as the first three approaches, pruning, choice of special models, and treatment, may be wasteful, restrictive in terms of model formulation, or presenting the risk of compromising the integrity of the training data, respectively. We propose a new method called blockwise reduced modeling (BRM) for handling incomplete data with blockwise missing patterns, which minimizes information loss and precludes the need for imputing each missing value. BRM has two phases: handling incomplete training data and making predictions for test instances. In the first phase, our algorithm automatically groups the training data into overlapping subsets with different combinations of features that contain only populated values. We

then train candidate models over each subset. In the second phase, our algorithm assigns each instance of the test data set to one of the candidate models using a similarity scoring mechanism.

BRM is an improvement over the classical reduced modeling approach in the following ways. First, it is scalable as training is done in advance instead of at run time. Second, BRM can be used to train any type of model (e.g., linear regression, logistic regression, tree-based models, neural networks, etc.) for supervised learning (i.e., classification and regression) for tabular data.

We evaluate BRM using simulations and real data applications. We use three complete data sets from public repositories (BIKE, Capital bikeshare system; HOUSE, King County House sales; ADULT, census income) for simulating blockwise missing patterns and comparing the prediction performance of BRM and benchmarks. After that, we employ three data sets with actual blockwise missing patterns (ADNI, Alzheimer's Disease Neuroimaging Initiative; WB2, Wellbuilt for Wellbeing; COVID, Facebook COVID-19 symptoms) to demonstrate the proof of the utility of BRM for predictive modeling using data with actual blockwise missing patterns occurring in different problem contexts. Our analysis shows that BRM has better predictive performance than existing methods across data sets.

The remainder of this paper is structured as follows. In Section 2, we discuss related work and describe the BRM method in Section 3. Experiments and findings are presented in Section 4. Sections 5 and 6 contain the discussion and conclusions, respectively.

2. Research Background

Since Rubin's framework for describing and analyzing missing data (Rubin 1976), numerous methods have been proposed to analyze missing values. Broadly, classical methods for handling missing values in tabular data can be classified into three approaches: pruning, using special models, and treatment or imputation. The first approach is to prune or discard observations or features containing missing values. This approach may lead to a loss of information and a biased training sample (van Buuren 2018). The second approach is to select models that do not require complete data for training. Data mining models such as classification and regression trees (CART) handle missing values during the creation of classification trees using surrogate splits (Breiman et al. 1984); that is, when considering a predictor for a split, only the observations for which that predictor is not missing are used for training. These models are known to be biased toward features with fewer missing values in the data set. Hence, the models can be misleading when comparing feature importance in the presence of incomplete data. The second approach also precludes the use of several commonly adopted models, such as generalized linear models, ensemble models, neural networks, and so on, that require complete data.

The third (treatment) approach involves imputation or replacing a missing value with a representative value using univariate or multivariate estimation methods. This is the most common class of methods to generate complete data from incomplete data (Little and Rubin 2002). There are several popular univariate and multivariate imputation methods, including expectation-maximization (Dempster et al. 1977), proximity matrix (Liaw and Wiener 2002), predictive mean matching (Batista and Monard 2003), multiple imputation (Rubin 2004), soft impute (Mazumder et al. 2010), and missForest (Stekhoven and Buhlmann 2012). These imputation methods can be further classified as single value imputation (SVI) and multiple imputation (MI) methods depending on whether a single replacement or a distribution of replacement values is generated. Simple imputation methods such as mean/median value replacement can understate variability in the imputed features, whereas MI methods require model averaging techniques for estimating coefficients and making predictions (van Buuren 2018).

Algorithmic imputation methods such as expectation maximization or multiple imputation pose other problems. They could distort the data by oversynthesizing artificial replacements when the missing value proportion is high (van Buuren 2018). As observed in our analyses, they also become computationally infeasible for moderate to large-sized data sets. Researchers have explored generative modeling approaches such as generative adversarial networks (GANs) and variational autoencoders (VAEs) for training deep neural networks for analyzing nontabular data such as images and text (Smieja et al. 2018, Ipsen et al. 2020). Although there have been attempts to extend these methods to tabular data sets (Yoon et al. 2018, Mattei and Freiisen 2019), their implementation still poses challenges such as sensitivity to the tuning of multiple hyperparameters of the generative models (Camino et al. 2020). Most existing methods with software package implementations commonly assume the observations in the data set to be independent and identically distributed and do not consider incomplete data such as sparse, multisource, or nontabular data sets. No single approach or method is optimal for all types of data and models, and their relative performance is context specific (Woźnica and Biecek 2020).

In contrast to randomly distributed missing values, blockwise missing patterns have contiguous rows of one or more columns with no values populated due to known or unknown reasons. Blockwise missingness is a more recent data science phenomenon owing to

complex data sourcing, integration, and preprocessing needs and challenges of contemporary big data applications. The presence of blockwise patterns can be easily identified by eyeballing sample data. They occur as spurts of cells with missing values across multiple rows in one or more columns as opposed to the commonly observed random pattern of missingness. The example data shown in Table 1 is a representative sample of an incomplete data set containing blockwise missing patterns with 11 observations and four features. We see approximately 27%, 55%, 55%, and 64% of values missing for Feature 1, Feature 2, Feature 3, and Feature 4, respectively. Let's consider this example to compare existing methods for processing incomplete data. Pruning rows with missing values (i.e., listwise deletion) will leave us with only 2 of 11 observations (i.e., rows 10 and 11). Columnwise deletion is also not a feasible option with multiple columns containing blockwise missing patterns (and the exclusion of multiple features would bias the model). Naïve treatment approaches such as mean value substitution will understate the variability as three or more observations in each feature will have the same values. On the other hand, sophisticated imputation methods assign values for missing cells using iterative procedures, generating a resultant data set that may be very different from the original data. Therefore, traditional approaches are not ideal for processing incomplete data with blockwise missing patterns, especially as the data set size and the extent of missing values increase.

To better understand the prevalence of blockwise missing data, we looked at data sets shared publicly by Kaggle.com. In their data sets search engine, we applied a size filter between 200 KB and 10 MB and downloaded 240 cross-sectional data sets in .csv format. Within these data sets, 41 data sets had missing values (17%), of which 13 data sets had more than 5% of cells missing. These 13 data sets, on closer observation, had spurts of contiguous data missing for multiple variables and therefore had blockwise missing patterns. Their metadata indicated integration from heterogeneous raw sources. This exercise showed that blockwise missing patterns are more common during cases of integration from heterogeneous data sources, as also reported in other studies focusing on blockwise missing value handling (Yuan et al. 2012, Yu et al. 2020, Xue and Qu 2021).

There is growing interest in analyzing incomplete data with blockwise missing patterns. Yuan et al. (2012) developed a method for selecting relevant features in a high-dimensional medical incomplete data set with blockwise patterns. The method was further extended by Xiang et al. (2013) to perform simultaneous feature-level and source-level analyses. Srinivasan et al. (2016) proposed an ensemble-based method that combines predictions from models trained over nonoverlapping subsets of original data, where test instances are still handled using traditional imputation approaches. Xue and Qu (2021) iteratively impute missing blocks from the collection of other data subsets containing values of features in the missing block, followed by combining the imputations using a weighting function. Yu et al. (2020) do not attempt imputing missing blocks but focus on estimating the covariance matrix of blockwise missing predictors to get improved coefficient estimates. Methods have also been proposed for dimension reduction (Zhu et al. 2020) and matrix completion in an unsupervised setting (Zhou et al. 2023). Table 2 lists existing methods literature for analyzing blockwise missing patterns. Previous methods use a sparse learning framework to address the feature selection problem and are not designed for general-purpose explanatory or predictive modeling. Except for one research effort (Srinivasan et al. 2016), all other methods assume the blockwise missing patterns to be predetermined. All existing methods depend on a complete test data set primarily achieved by imputing the raw test instances.

To summarize, most methods require manual annotation of missing blocks, do not scale well for larger data sets, and are restrained to specific model families such as generalized linear models. Therefore, there is an imminent need for new methods to handle large incomplete data sets with blockwise missing patterns that not only minimize inference errors but are also fast, efficient, and scalable.

The concept of reduced modeling was first introduced by Friedman et al. (1996) in their lazy classification trees method. The basic idea was to build individual predictive models for features with values populated in a test instance. This approach required training a different reduced candidate predictive model for every test instance. Because this was computationally intensive and time consuming, a hybrid method that combined reduced modeling and imputation was proposed (Saar-Tsechansky and Provost 2007). In this method, a separate model is trained for each input feature so that all other input features are imputed and included in the model. For prediction, for each test instance, if a feature is missing, the corresponding model is chosen with the particular feature excluded. In terms of more than one missing value in the test instance, imputation is randomly carried out for all except one input, followed by the simple feature-matching procedure. This hybrid method leads to pretraining for a number of models equal to the number of features in the data set and was shown to perform better than different types of imputation for classification problems (Saar-Tsechansky and Provost 2007). The classical and hybrid methods of reduced modeling are not explicitly designed for analyzing blockwise missing patterns.

Table 2. Summary of Methods for Analyzing Blockwise Missing Patterns

Studies	Focus	Notes
(Yuan et al. 2012, Xiang et al. 2013)	Feature selection in a high-dimensional setting	Assume blockwise patterns to be predetermined. Impute missing values in test data. Use grouped lasso as the underlying feature selection mechanism. No sensitivity studies were conducted.
(Srinivasan et al. 2016)	Predictive modeling with minimum imputation for blockwise missing data	Impute missing values in test data. Propose a meta-learner trained over multiple imputation results to make predictions using a multisource sensor data set. No sensitivity studies were conducted.
(Xue and Qu 2021)	Impute missing values in blockwise missing data	Assume blockwise patterns to be predetermined. Develop a weighted averaging approach of coefficient estimation using iterative sampling of nonmissing data. Impute missing values in training and test data. Perform sensitivity analysis on synthetic data.
(Yu et al. 2020)	Improved coefficient estimates of the linear model over blockwise missing data	Assume blockwise patterns to be predetermined. Iteratively update the covariance matrix using the blockwise missing structure. Impute missing values in test data. Perform sensitivity analysis on synthetic data.
(Zhu et al. 2020)	Principal component analysis on blockwise missing data	Assume blockwise patterns to be predetermined. Unsupervised learning for dimensionality reduction. Report challenges to scale up for more than two sources of data.
(Zhou et al. 2023)	Input data matrix completion of blockwise missing data	Assume blockwise patterns to be predetermined. An unsupervised approach to complete input matrix information. Perform sensitivity analysis on synthetic data.

3. Our Method

The prospect of minimizing mechanical imputation of training and test samples is appealing as it diminishes the human error of choosing suboptimal replacements for missing values. Saar-Tsechansky and Provost (2007) show using experiments that imputing variables at prediction increases error variance irrespective of whether it is an irrelevant or essential predictor. Furthermore, because imputation tries to approximate the joint distribution of the input vectors implicitly, there are increased risks of corrupting the training data and overlooking important fully populated features during the training process. Reduced modeling has lower variance and generalization error as it helps minimize the imputation of training and test instances (Saar-Tsechansky and Provost 2007). We considered the reduced modeling approach suitable for blockwise missing patterns as high error variance is a crucial challenge for analyzing such incomplete data (Yuan et al. 2012, Yu et al. 2020, Xue and Qu 2021).

Our proposed method (BRM) is a scalable improvement of the classical reduced modeling approach that specifically addresses the problem of analyzing incomplete data with blockwise missing patterns. It optimally utilizes information contained in incomplete data with blockwise missing patterns to improve prediction performance. BRM detects the number of missing block patterns and generates subsets of nonmissing data blocks that can overlap but contain minimal imputable missing values. Once these nonmissing subsets of the data have been identified, models are pretrained over each of these subsets. Predictions are made by matching the test instance vector to the model with the most similar set of input features. After that, a scoring mechanism is proposed for comparing the feature importance of the trained model inputs.

3.1. Problem Formulation

Incomplete data sets with blockwise missing patterns have contiguous observations with one or more columns missing. Let matrix $X \in \mathbb{R}^{n \times p}$ denote the input data matrix such that $x_{i,j}$ represent the value of the jth variable in the ith observation for $i \in \mathbb{N}_{\leq n}$, $j \in \mathbb{N}_{\leq p}$. Let the order of the columns $x_{\cdot,j}$ be inconsequential, such that columns with similar missing profiles can be placed next to each other. We assume observations $x_{i,\cdot}$ are independent but need not be identically distributed, allowing their shuffling and allocation into subsets. In simpler words, there is no adjacency requirement of rows and columns in the data set for our method. We assume that the univariate outcome $y = (y_1, y_2, \ldots, y_n)$ does not contain any missing values.

Let X be grouped into K subsets $\{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\}$ of training observations such that $X^{(k)} = \{x^{(k)} | x^{(k)} \subset X, k \in \mathbb{N}_{\leq K}\}$, such that within each input subset $X^{(k)}$, a set of features $p^{(k)}_{missing}$ contains no information; the remaining features $p^{(k)}$ may have randomly missing values. Note that $\{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\}$ may contain a subset $X^{(l)}$ with $p^{(l)}_{missing} = \emptyset$. Furthermore, each training subset $\{X^{(k)}, y^{(k)}\}$ contains an exclusive set of observations,

that is, $\{x^{(k)}, y^{(k)}\} \cap \{x^{(m)}, y^{(m)}\} = \emptyset$, for $m \neq k$. Also, each observation in the data set needs to belong to at least one subset, that is, $\{X, y\} = \cup_{k=1}^{K} \{X^{(k)}, y^{(k)}\}$. The cells in each input subset $X^{(k)}$ corresponding to completely missing features $p^{(k)}_{missing}$ are blockwise missing values. The input matrix X has K blockwise missing patterns, which may include a "nothing missing" pattern, corresponding to subset $X^{(l)}$ with all features containing information.

The predictive model is trained over incomplete data with blockwise missing patterns by not imputing the blockwise missing values and only imputing randomly missing values, thus minimizing overall imputation of the training data.

The rest of this section describes the training and prediction phases of BRM.

3.2. Training

In the training stage, we first characterize the blockwise missing patterns and then create training subsets with complete data for training. We present the training stage of BRM in a sequence of steps: (i) capturing blockwise missing patterns, (ii) creating training subsets, (iii) training candidate reduced models, and (iv) computing feature importance scores for the overall training process.

3.2.1. Capturing Blockwise Missing Patterns. Prior studies assume the blockwise pattern as given information in their problem formulation (Yuan et al. 2012, Yu et al. 2020, Xue and Qu 2021). However, blockwise pattern determination may be nontrivial, as shown in cases such as multisource sensor data fusion, presenting issues including sensor value thresholds, disconnections, and postprocessing challenges (Srinivasan et al. 2016). Therefore, a key component of our method is the automated characterization of blockwise missing patterns. For this, we use clustering as an unsupervised learning method to detect subsets of training data with identical missing patterns. Similar to the mask vector defined in Yoon et al. (2018, p. 2), we define a missing values mask matrix D corresponding to X such that $d_{i,j} = 1$ if $x_{i,j}$ is missing, and $d_{i,j} = 0$ otherwise. That is, a cell in the mask matrix contains one if there is a missing value in corresponding cell in the input data matrix and zero otherwise. Using a standard clustering method such as the K-means algorithm (Han et al. 2022) over the missing value mask matrix D, we identify subsets of data with different combinations of the features on which we can fit independent models. For ease of representation of training subsets, we extend the standard matrix cell value notation $x_{i,j}$ to express a training subset $X^{(k)}$ with $r^{(k)}$ observations and $p^{(k)}$ nonmissing features as $x_{\{r^{(k)}\},\{p^{(k)}\}}$.

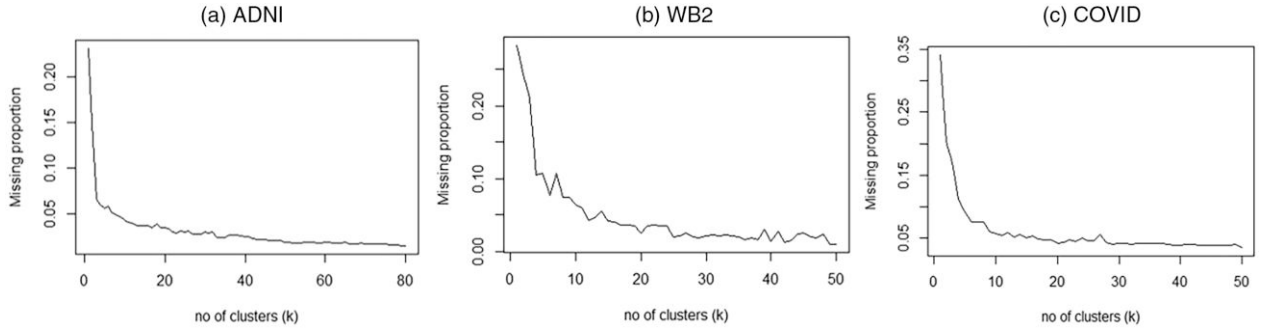
Even after splitting the training data into subsets, residual missing values can still be observed within the subsets, which need to be imputed before training models. This is because real-world data sets often contain a combination of blockwise missing and randomly missing patterns. Therefore, for each subset, we discard features with more than α proportion of missing values, where α is a nuisance parameter that accounts for stochasticity in the missing values. We introduce this parameter to allow the user to control overfitting missing values with microsized blocks. We tested for different values of α across data sets. The sensitivity of prediction performance to changes in α is shown in the online appendices. We recommend the nuisance parameter α to be set at 0.05. Note that α is unrelated to the total proportion of missing values in the data set.

When using K-means clustering, we run the algorithm t number of times and select the configuration with optimal internal cohesion and external separation jointly computed by the ratio (Total within-cluster sum of squares)/(Total sum of squares) (Han et al. 2022). To identify the number of clusters, we plot the proportion of missing values remaining across training subsets (i.e., proportion missing) as a function of the number of clusters K. For incomplete data with blockwise missing patterns, we observed this plot to resemble elbow plots as shown in Figure 2, (a)–(c).

The value of K can be automatically determined using the elbow plot as follows. The optimal choice of K corresponds to the point in the plot that is farthest from the line connecting the end points of the plot. The left-most and right-most points of the curve are connected using a straight line. The x axis of the point that is most distant from this line is K. For example, the optimal K for elbow plots in Figure 2, (a)–(c), are 4, 12, and 6, respectively (later reported in Table 3). Although the number of blockwise missing patterns can vary up to the number of features in data, the value of K is often smaller for real-world data sets, and hence we set $K_{max} = 50$ for generating the elbow plot for data sets with more than 50,000 observations. Alternatively, users may input the value of K manually after referring to the elbow plot (not recommended).

3.2.2. Creating Training Subsets. The subsets of training data created using the method described earlier are nonoverlapping as each observation in the training data belongs to only one of the subsets determined by the clustering algorithm. However, the training data subsets need not be nonoverlapping, and tuples with populated values for specific feature sets can be repeated across multiple subsets. Therefore, we propose creating overlapping subsets that build on the nonoverlapping subsets to utilize the maximum possible number of observations for training candidate models in the subsequent step of the training phase. For each nonoverlapping subset identified previously, we define a partial order of subsets such that the

Figure 2. Missing Value Proportion as a Function of the Number of Training Subsets for ADNI, WB2, and COVID Data Sets



features in the lower order are subsets of features in the higher-order subsets.

That is, for $\{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\} = \{x_{\{r^{(1)}\},\{p^{(1)}\}}, x_{\{r^{(2)}\},\{p^{(2)}\}}, \ldots, x_{\{r^{(K)}\},\{p^{(K)}\}}\}$, we define a partial order for each $X^{(k)}$ such that $X^{(k)} \leq X^{(m)}$ if $p^{(k)} \leq p^{(m)}$ (i.e., input feature set of $X^{(k)}$ is a subset of input feature set of $X^{(m)}$). It is easy to note that the law of transitivity applies such that if $X^{(k)} \leq X^{(m)}$ and $X^{(m)} \leq X^{(h)}$, then $X^{(k)} \leq X^{(h)}$.

The data contained in the subsets that are higher in the partial order can be included in the lower ordered subsets as they contain elements equal to or greater than the set of features lower in the partial ordering. Let $\tilde{X}^{(k)} = x_{\{\tilde{r}^{(k)}\},\{p^{(k)}\}}$ represent the superset of all training subsets that are at a higher order than $X^{(k)}$. We can generate overlapping training subsets $\{X^{(k)}_{OV}, y^{(k)}_{OV}\} = \{x_{\{r^{(k)}_{OV}\},\{p^{(k)}\}}, y^{(k)}_{OV}\}$ such that $r^{(k)}_{OV} = r^{(k)} \cup \tilde{r}^{(k)}$. Note that number of input features $p^{(k)}$ remains the same for nonoverlapping subset $X^{(k)}$ and the corresponding overlapping subset $X^{(k)}_{OV}$. Here onward, we use the terms overlapping subsets and training subsets interchangeably.

The intuition behind creating the overlapping subsets is that clustering only partitions the data horizontally (i.e., grouping rows). Even though clustering may be used to create supersets of columns with similar missing patterns, the grouping of rows precludes simultaneous exploration of rows and columns with contiguous missingness. The walk-through of BRM in the online appendices shows an illustration supporting this intuition.

3.2.3. Training Candidate Reduced Models. Using the previous steps, the training data are divided into multiple overlapping subsets. The remaining missing values within each subset are imputed using a fast SVI method such as the hot-deck imputation method (Kowarik and Templ 2016) to generate complete data for training models. For each subset, we train an independent candidate reduced model such that columns of the subsets are input features of the candidate models.

Table 3. Summary of BIKE, HOUSE, and ADULT Data Sets

Data set	Size (row, columns)	Problem type
BIKE	(17,379, 11)	Regression
HOUSE	(21,597, 9)	Regression
ADULT	(32,561, 13)	Classification

3.2.4. Computing Global Feature Importance Scores. Feature importance scores are useful for determining the relative importance of features in prediction models. Most data mining methods implemented in packages provide feature importance as a function of the model class. Although prediction models including BRM can run without calculating feature importance, users frequently want to know these scores. Hence, we generate global feature importance scores for the BRM method irrespective of the underlying models used over training subsets. We adopt a weighted model averaging strategy to provide a cumulative weighted score for each feature based on their corresponding feature importance in candidate models. The conventional model averaging technique used in multiple imputation averages the values of the parameter estimates across the missing value samples generated in multiple imputation methods to obtain single-point estimates (Rubin 2004, van Buuren 2018). Instead, we combine results from data sets with different dimensions as shown in the following equation:

$$B_p = \frac{\sum_{i=1}^{k} B_{p,i} \cdot W_i}{\sum_{i=1}^{k} W_i}, W_i = n_i/\xi_i. \quad (1)$$

In Equation (1), B_p is the (global) importance of the pth feature across K candidate models, $B_{p,i}$ is the feature importance of the pth feature in the ith model, W_i is the weighting factor for the ith model given by n_i/ξ_i, where n_i is the number of rows of the ith training subset, and ξ_i is the training error of the ith model. The weighting strategy accounts for the size of the training data set and training error of candidate reduced models. This ensures that candidates with better fit contribute more to the score. Accounting for the size of the training data penalizes the local influence of inputs on

the outcome captured by the smaller subsets. Akin to popular weighted model averaging estimators (Hansen and Racine 2012, Chen and Xie 2014), the properties and cautions related to making causal interpretations (Banner and Higgs 2017) also apply to the proposed global feature importance scoring method. As an alternative to the proposed global feature importance scores, model-agnostic feature attribution methods such as Shapley additive explanations (SHAP) (Lundberg and Lee 2017) or accumulated local effects (ALE) (Apley and Zhu 2020) may also be used to describe feature importance solely based on predictions generated using BRM.

3.3. Making Predictions

We assume blockwise missing patterns are similar in the training and test samples. In line with the reduced modeling methodology, each test instance is matched with a suitable model for making a prediction. We use the Jaccard similarity index to compare the feature vector of a test instance and assign it to the nearest candidate reduced model. That is, the candidate model with the input feature set that has a maximal overlap with the test input feature vector is selected for making predictions. In case of a tie in terms of similarity scores between two or more models, we select the model with a higher number of input features to avoid the imputation of missing values in the test instance. These steps are repeated for each test instance.

Algorithm 1 summarizes the training and prediction phases of BRM method. We provide a walk-through of the steps of BRM using the simple example presented in Table 1, as well as a brief discussion on the computational complexity of BRM in the online appendices.

Algorithm 1 (BRM Method)

Input: Training data $\{X_{\text{train}}, y_{\text{train}}\}$ and test instances $x_t \in X_{\text{test}}$ with blockwise missing patterns.

Initialize: Missing value mask matrix D_{train} corresponding to missing value pattern in X_{train}, mask vector d_{test} corresponding to missing value pattern in test instance x_t, random missingness threshold $\alpha = 0.05$, maximum blockwise pattern threshold $K_{\max}$.

Training:

1. For K in 1 to $K_{\max}$, do:
   (a) Divide X_{train} into K subsets $\{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\}$ using results from clustering of D_{train}.
   (b) Discard features in each subset with more than α missing values.
   (c) Record the proportion of missing values across subsets.
2. Choose optimal $K \in \{1, 2, \ldots, K_{\max}\}$ using the procedure described in Section 3.2.1.
3. For k in 1 to K, do:
   (a) Determine the partial order $X^{(k)} \leq X^{(m)}$, $m \neq k$ such that $p^{(k)} \leq p^{(m)}$ (i.e., input feature set of $X^{(k)}$ is a subset of input feature set of $X^{(m)}$).
   (b) Initialize overlapping subset as $X_{OV}^{(k)} = X^{(k)}$.
   (c) Assign additional observations to overlapping subset $X_{OV}^{(k)}$ from other nonoverlapping subsets $\{X^{(1)}, X^{(2)}, \ldots, X^{(K)}\} \backslash X^{(k)}$ such that observations in $X^{(m)}$ are included in $X_{OV}^{(k)}$ if $X^{(k)} \leq X^{(m)}$.
   (d) Impute residual missing values in $X_{OV}^{(k)}$ using a fast SVI method (e.g., hot-deck imputation).
   (e) Train candidate model M_k over $\{X_{OV}^{(k)}, y_{OV}^{(k)}\}$.
4. Generate global feature importance scores as described in Equation (1).

Prediction:

For $x_t \in X_{\text{test}}$, do:

(a) Compute Jaccard similarity index J_i of d_{test} with feature indicator vectors of $\{X_{OV}^{(1)}, X_{OV}^{(2)}, \ldots, X_{OV}^{(K)}\}$. Select model $M_{\text{best}} = arg\,min_{M_i \in \{M_1, M_2, \ldots, M_k\}} J_i$

(b) Predict outcome $y_t \in y_{\text{test}}$ for input x_t using candidate model M_{best}.

Output: Predictions y_{test} for X_{test} and global feature importance scores.

4. Analysis

In this section, we first enumerate the benchmarks and provide implementation details. Thereafter, we describe a simulation study followed by experiments on incomplete data sets with actual blockwise missing patterns.

4.1. Benchmarks

We have considered the following comprehensive set of competing benchmarks representing various missing value handling approaches discussed earlier:

1. **LISTWISE:** Deletion of rows/records with at least one missing value.
2. **COLUMNWISE:** Deletion of columns/features with more than 50% missing values.
3. **CART:** Classification and regression tree as a representative of models that can implicitly handle missing values (Breiman et al. 1984).
4. **SI**: The soft-impute method that uses spectral regularization as a representative of single value imputation methods. (Mazumder et al. 2010).
5. **MI:** Multivariate imputation by chained equations (MICE) (Van Buuren and Groothuis-Oudshoorn 2011) combined with meta-learners (Srinivasan et al. 2016) as a representative of multiple imputation methods.
6. **GAIN:** Generative adversarial imputation network method (Yoon et al. 2018) as a representative of imputation methods using GANs and variational autoencoders (VAEs).
7. **REFE:** Reduced-Feature Ensemble, a hybrid method combining reduced modeling and SVI imputation (Saar-Tsechansky and Provost 2007).
8. **IMSF:** Incomplete multisource feature learning method for blockwise missing patterns (Yuan et al. 2012).

9. **DISCOM:** Direct sparse regression procedure using covariance from multimodality data method representing regression coefficient estimation approaches for analyzing blockwise missing patterns (Yu et al. 2020).

10. **CART_I:** Classification and regression tree with a missing value indicator matrix included in an augmented input feature set.

11. **LAZY:** The original reduced modeling method implemented as lazy decision tree (Friedman et al. 1996).

In addition, we include an alternative version of our method BRM_NOV, which includes all the steps of BRM, but it trains models on nonoverlapping subsets instead of overlapping subsets.

4.2. Implementation Details

A 75:25 split was used for training and test samples for evaluations. We measured the prediction performance of the models using various metrics for the regression and classification data sets. In the paper, we report root mean square error (RMSE) and symmetric mean absolute percentage error (SMAPE) for continuous outcomes and area under the curve (AUC) and F1 score for discrete outcomes (Han et al. 2022), as they best summarized the relative performance of all models across data sets. The benchmarks in the following tables and figures have been ordered in terms of the predictive performance scores (F1 for classification and SMAPE for regression) for ease of inference. Benchmarks not reported in a table or figure failed to execute for the corresponding data set or simulation. From the error logs, we observed execution failures to be related to memory overflow for methods such as MI, GAIN, invalid training data postpruning for methods such as LISTWISE and COLUMNWISE, a singular matrix or lack of convergence in a fixed set of iterations for DISCOM and IMSF, or an invalid feature matrix for one or more training subsets of REFE and LAZY methods. For simulations, execution failures were observed across many methods in cases with a high proportion of missing values (i.e., >0.7). All models were run using the R programming software on a server with Windows OS, 64-GB RAM memory, and 2.00-GHz 24-core 13th generation 64-bit Intel i9 processor.

Because the original methods IMSF and DISCOM assume the blockwise missing pattern to be predetermined, we used the clustering procedure of BRM to determine the missing patterns in those cases. We identified the blockwise imputation method proposed by Xue and Qu (2021) as a potential benchmark but were not able to replicate their code shared in supplementary files of Xue and Qu (2021) due to the need for hardcoding of multiple input feature sets used for imputing each block. Also, we have not considered blockwise pattern analysis methods with a focus different from making predictions such as dimension reduction (Zhu et al. 2020) or unsupervised learning (Zhou et al. 2023). Based on results reported in prior studies that use penalized parameter estimation modeling approaches (Yu et al. 2020, Xue and Qu 2021), we believe that DISCOM (Yu et al. 2020) and IMSF (Yu et al. 2020) can be considered as competing benchmarks representing these studies. Because GAN-based and VAE-based methods are comparable for tabular data (Camino et al. 2020), we considered GAIN (Yoon et al. 2018), a popular GAN-based method as the representative benchmark for generative approaches to imputation. For MI, we generated multiple imputation data sets, followed by training models on each of these data sets. The outputs of these models were pooled using a meta-learner such as Randomforest that gave the final ensemble prediction (Srinivasan et al. 2016).

The *rpart*, *randomForest*, and *mice* packages in R with default parameters were used for fitting CART and MI models. DISCOM and IMSF were used for feature selection and applied to linear models as designed. Codes for DISCOM and IMSF were adopted from the supplementary section of Yu et al. (2020), and the *reticulate* package (Ushey et al. 2018) was used to run the python implementation of GAIN. CART and SI were trained using *rpart* and *softimpute* packages (Hastie et al. 2015).

BRM is independent of the training model. We considered a wide variety of nonlinear models for our experiments including gradient boosting machine (GBM), multivariate adaptive regression splines (MARS), K-nearest neighbor classifier (KNN), Randomforest, neural networks, ridge regression, linear model trees, and bagged trees, and found that GBM gave the best prediction results in general. Therefore, we use GBM as the underlying nonlinear model for the comparison of methods across all data sets. For linear modeling, we used generalized linear models (GLM) with different error distributions based on the respective outcomes of the data sets. For all data sets, the number of trees (i.e., n.trees) parameter for GBM was set to 500. To improve the model fit of GBM over COVID and WB2 data sets, shrinkage and interaction depth were set to 0.05 and 4, respectively. Except for CART, CART_I, and the nonlinear modeling with LAZY, all other benchmarks were trained using the same set of models for the same data set. This is because CART models inherently handle missing values, and the LAZY implementation with GBM or any other ensemble learner took more than 24 hours to train for each simulation. Therefore, we used the same linear model for LAZY but implemented LAZY with CART for the nonlinear modeling case.

4.3. Simulation Study

In the simulation study, we synthesize different conditions of blockwise missing patterns and then evaluate BRM's performance against known benchmarks. This

allows us to see performance degradation as the relative proportion of missing values is increased. We use the following three publicly available complete data sets for our simulations.

BIKE: The Capital bike-sharing data set is archived in the University of California Irvine (UCI) data repository and contains hourly counts of bicycles rented during 2011–2012 in Washington, DC (Fanaee-T and Gama 2013). The task is to predict the hourly bike rental rate based on input conditions including time and weather conditions.

HOUSE: The King County house sales data set contains historic data of houses sold between May 2014 and May 2015 in King County, Washington. The data set is archived in the public data repository hosted by Kaggle.com. The outcome variable for the prediction task is house prices in dollars.

ADULT: The census income data set is archived in the UCI data repository and contains information from the 1994 census database. The prediction task is to determine whether a person makes more than $50,000 per year.

Table 3 summarizes the three complete data sets used for the simulation study. For each of the three complete data sets, we vary the proportion of blockwise missingness from a factor of 0.1 to 0.9 per feature. In addition, for the BIKE data set, we also benchmark performance after varying the data size and the number of blocks. For the BIKE data set, we also examine global importance scores as a function of missing proportion. For all simulations, values from groups of features were removed simultaneously to result in four blockwise missing patterns in the data set. In addition to synthesizing blockwise patterns of varying magnitude, we randomly sampled and deleted 5% of the observations from these features to add a stochastic element to the missing data.

Because the outcome represents count in the BIKE data set, we used a negative binomial as a link function for the linear model (i.e., NB-GLM) and a Poisson error distribution for GBM as the nonlinear model. For HOUSE and ADULT data sets, we use logistic regression (i.e., GLM with binomial distribution) as the linear model and GBM with Bernoulli distribution as the nonlinear model.

Figures 3–5 show the change in prediction performance of BRM and benchmarks when the blockwise missing proportion is varied from 0.1 to 0.9 for select input features of BIKE, HOUSE, and ADULT data sets, respectively. The simulation studies show that BRM and BRM_NOV have superior prediction performance than state-of-the-art methods in general.

Figure 3 shows that the classical reduced modeling method LAZY has better prediction performance than other benchmarks but does not execute for missing proportions 0.8 and 0.9. SI and COLUMNWISE perform worse than most benchmarks for the BIKE data set.

Figure 3. (Color online) Prediction Performance of BRM and Benchmarks as a Function of Proportion Missing Values in Features in the BIKE Data Set

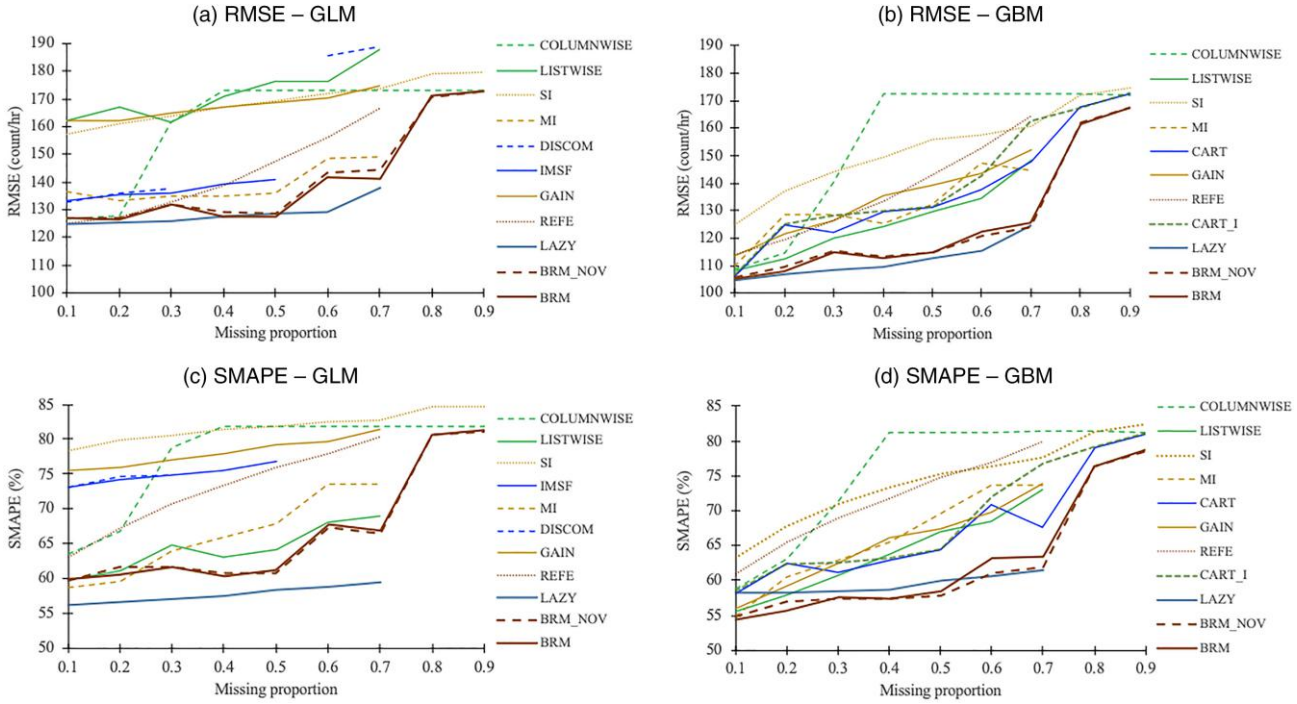


Figure 4 shows that BRM and BRM_NOV perform better than the LAZY method in the case of nonlinear modeling, but their performance drops slightly at blockwise missing proportion 0.4 due to unknown reasons. LISTWISE, COLUMNWISE, DISCOM, and IMSF perform worse than others for the HOUSE data set.

Figure 5 shows that BRM, BRM_NOV, LAZY, and LISTWISE methods have similar prediction performance until missing proportion 0.5. Beyond 0.5, BRM and BRM_NOV perform better than other benchmarks.

Running times of BRM and alternative methods were compared for linear models trained on synthetic data sets of varying sizes as shown in Figure 6. Running time is represented on a linear scale, whereas the number of blockwise missing patterns is in a logarithmic scale. The missing proportion of inputs was set to 0.5. We did not consider CART, CART_I, DISCOM, or IMSF for comparison as we were more interested in understanding the time cost in training multiple candidate models through the BRM method compared with (i) ignoring missing values (i.e., LISTWISE, COLUMNWISE), (ii) imputation methods (SI, GAIN, MI), (iii) the hybrid reduced modeling method REFE, and (iv) the classical reduced modeling method LAZY. We were able to train BRM in a reasonable time even for a million records, and the computing time is between simple imputation, pruning, and more sophisticated methods. Note that the classical reduced modeling method LAZY is more than 100 times slower than all other methods except MI for any data size. Therefore, even though LAZY performs slightly better than BRM in certain situations, its computation complexity makes it infeasible for large data sets.

4.4. Evaluation on Incomplete Data Sets

To test the utility of our method on incomplete data sets with actual blockwise missing patterns, we analyze the following three data sets belonging to different problem domains.

ADNI: The Alzheimer's Disease Neuroimaging Initiative is a data archive managed by the Laboratory of Neuroimaging at the University of South California. It contains blockwise missing patterns as it combines data from multiple sources including serial magnetic resonance imaging (MRI), positron emission tomography (PET), cerebrospinal fluid (CSF) measurements, and neuropsychological assessments. We consider the ADNIMERGE data set, a curated version of ADNI containing coded information of the most relevant inputs to Alzheimer's disease modeling. We focus on the problem of classifying cognitively normal and mild cognitive impairment subjects and discard observations labeled as dementia.

WB2: The Wellbuilt for Wellbeing project (GSA 2021) was a 16-month multiphase field study funded by the U.S. General Services Administration to better understand the influence of the office environment on human health, comfort, and performance. The data set

Figure 4. (Color online) Prediction Performance of BRM and Benchmarks as a Function of Proportion Missing Values in Features in the HOUSE Data Set

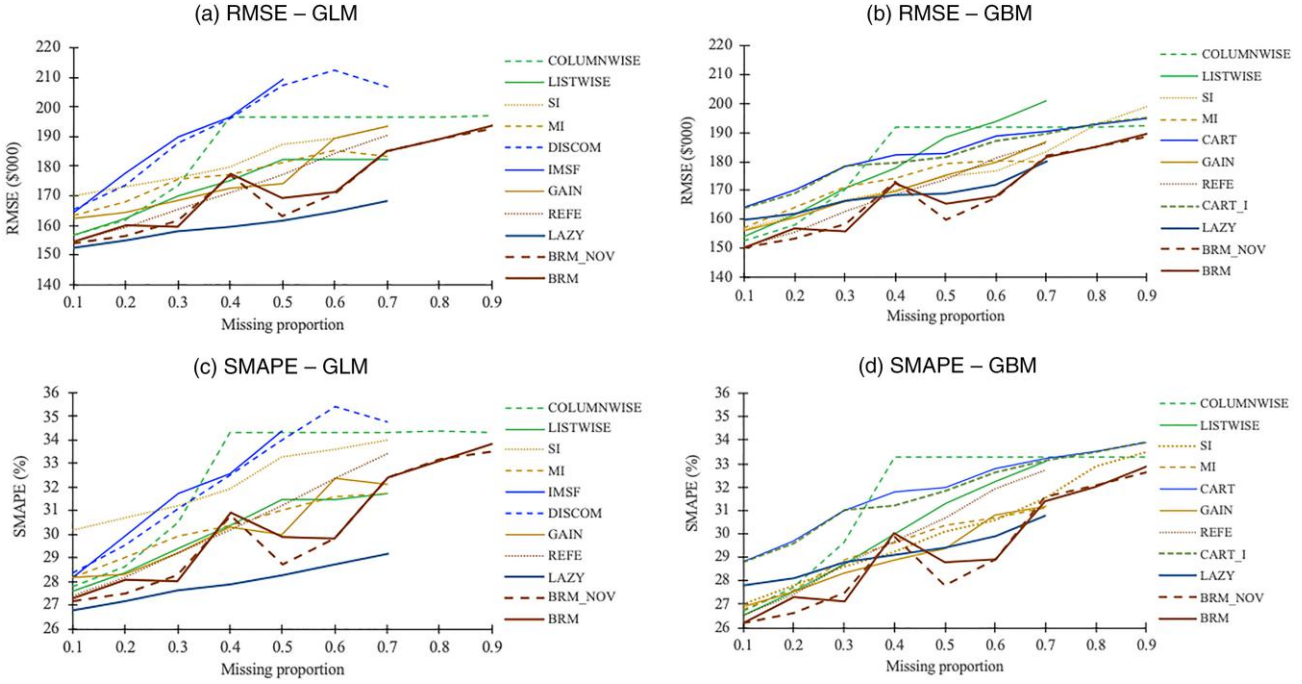
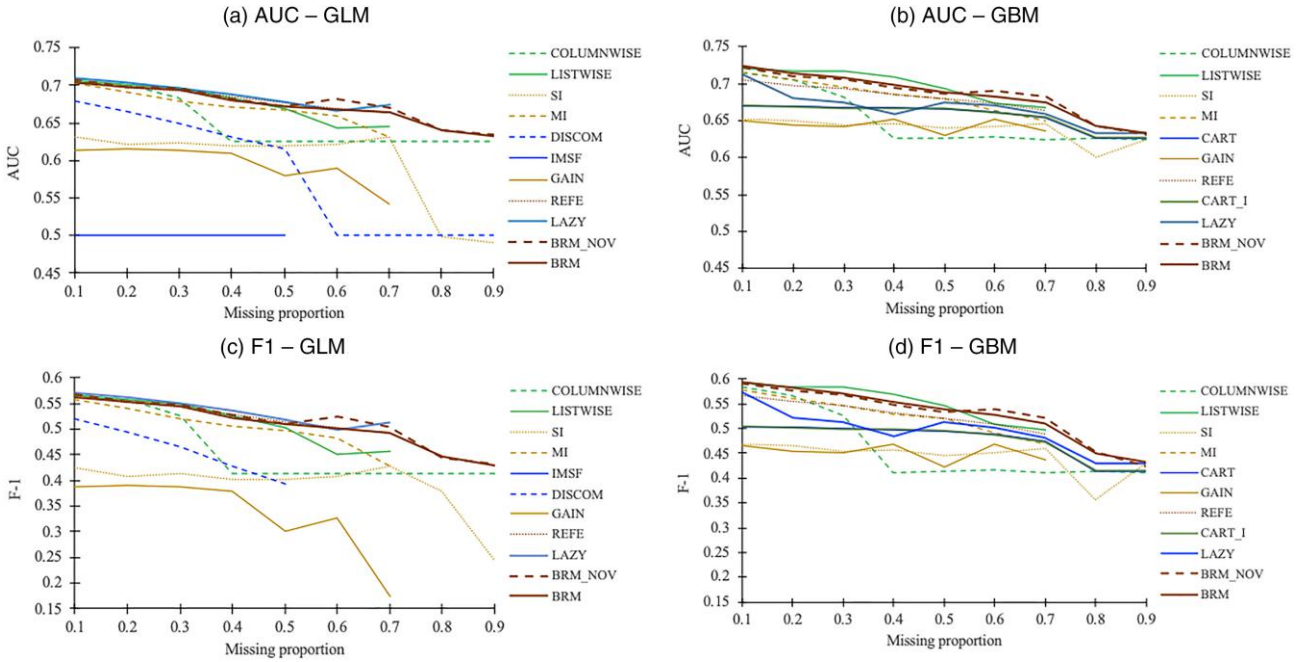


Figure 5. (Color online) Prediction Performance of BRM and Benchmarks as a Function of Proportion Missing Values in Features in the ADULT Data Set



contains blockwise missing patterns as it combines data from multiple sources including wearable environment sensors, wall-mounted environmental sensors, heart and physical activity monitors, experience sampling mobile surveys, location descriptions, and demographic information. We focus on the problem of predicting heart rate variability using indoor environment and work-related factors (Srinivasan et al. 2023).

COVID: The Delphi group of Carnegie Mellon University and Facebook conducted a large-scale survey to monitor the spread and impact of the COVID-19 pandemic in the United States (Salomon et al. 2021). Multiple questions in the survey are optional, and there are also follow-up questions. Both these factors lead to blockwise missing patterns. We choose the problem of predicting COVID-19–positive reporting of survey respondents. We consider a representative subset of all the survey responses and input features in our data set.

Figure 6. (Color online) Computing Time of BRM and Benchmarks as a Function of Data Size

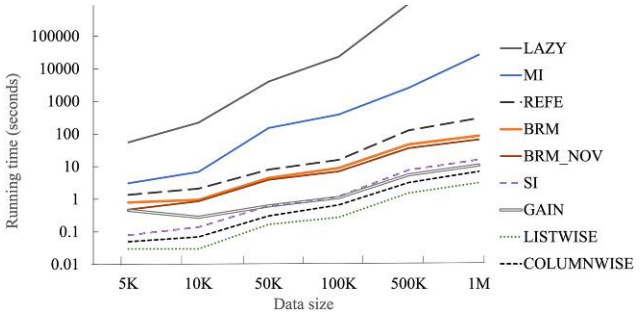


Table 4 summarizes the incomplete data sets used in this study and the number of blockwise patterns identified using BRM. Figure 7 shows grid visualizations of the missing value indicator matrix of ADNI, WB2, and COVID data sets. The darker cells are missing values across input features clustered in the y axis. The number of missing blocks identified for each data set in Table 4 can be visually verified using Figure 7. We can also observe multiple features in each data set with a 50% or higher proportion of missing values. The large proportion of missing values for certain features of the WB2 data set is because, although sensor data were measured at a finer grain (e.g., per minute, per second), experience sampling survey responses were recorded from participants only once per hour or so.

Tables 5–7 show the prediction performance comparison of linear (i.e., GLM) and nonlinear models (i.e., GBM) trained using BRM and benchmark methods across ADNI, WB2, and COVID data sets, respectively. The best performing metrics are highlighted in bold.

Table 5 shows the performance of benchmark methods and BRM for the ADNI data set. DISCOM, MI, and LISTWISE did not execute for the ADNI data set, as there were no complete observations. BRM has a higher AUC and F1 score than other methods for linear and nonlinear model configurations.

Table 6 shows the performance of benchmark methods and BRM for the WB2 data set. Because this data set was larger with blockwise patterns across multiple

Table 4. Summary of ADNI, WB2, and COVID Data Sets

Data set	Size (row, columns)	Overall missing (%)	Problem type	Accessibility	No. of missing blocks
ADNI	(8,649, 81)	23	Classification	Via DUA	4
WB2	(31,553, 99)	36	Regression	Not public	12
COVID	(73,536, 71)	34	Classification	Via DUA	6

features, the following benchmarks did not run: LISTWISE, IMSF, DISCOM, and CART. Although classical CART did not run, CART with missing value indicators as inputs, CART_I, successfully trained over the data set. BRM and BRM_NOV have the lowest prediction errors, whereas MI performs the worst for this data set.

Table 7 shows the performance of benchmark methods and BRM for the COVID data set. The classical reduced modeling method LAZY failed to train over the COVID data set. Although BRM and BRM_NOV outperformed benchmarks for linear models, simpler methods such as SI and CART methods have a higher AUC and F1 score for nonlinear models. This could be attributed to the collinearity between features with different blockwise missing patterns. Also, BRM_NOV slightly outperforms BRM for this data set.

5. Discussion

There are several data science applications where we encounter blockwise missing patterns in the data, including participant nonresponses in surveys, asynchronous data capture from Internet of Things (IoT) sensors, flat tables obtained using multiple outer joins, partial destruction of data, loss of data during transmission, and so on. To optimize the modeling of blockwise missing patterns, it is essential to devise a method that minimizes imputation at the training and prediction phases. Identifying overlapping subsets containing only populated values ensures minimal imputation in the training phase of BRM. In the prediction phase, BRM does not require test instances to be modified to suit the models. Instead, a suitable model is mapped to each test instance. Our method performs seamlessly and efficiently for test instances with blockwise missing patterns. The global feature importance score is useful for ranking feature contribution.

We simulated blockwise missing patterns using three complete data sets and evaluated the degradation of prediction performance as the missing proportion in features increased. Thereafter, using three incomplete data sets across different problem domains, we demonstrated the utility and superior predictive performance of BRM when applied to data with diverse sets of features having actual blockwise missing patterns. Our method involves minimal synthetic modification to the original data through imputation; it also avoids discarding important information via pruning. Our method is therefore a valuable tool for predictive modeling with blockwise missing patterns.

Our current work has some limitations and considerations. We have not considered temporal autocorrelations and other heterogeneity in model errors for simplicity. We leave it to future research to explore the extension of BRM to longitudinal analysis. We have primarily focused on cross-sectional data sets but have not discussed text, images, videos, and other data types. For such data, incomplete data may have atypical structural patterns, and therefore, the missing value handling process will be different. We have presented a simple weighted mechanism to compute feature importance score as it was not the focus of our study.

Figure 7. (Color online) Missing Value Distribution Across Input Features for ADNI, WB2, and COVID Data Sets

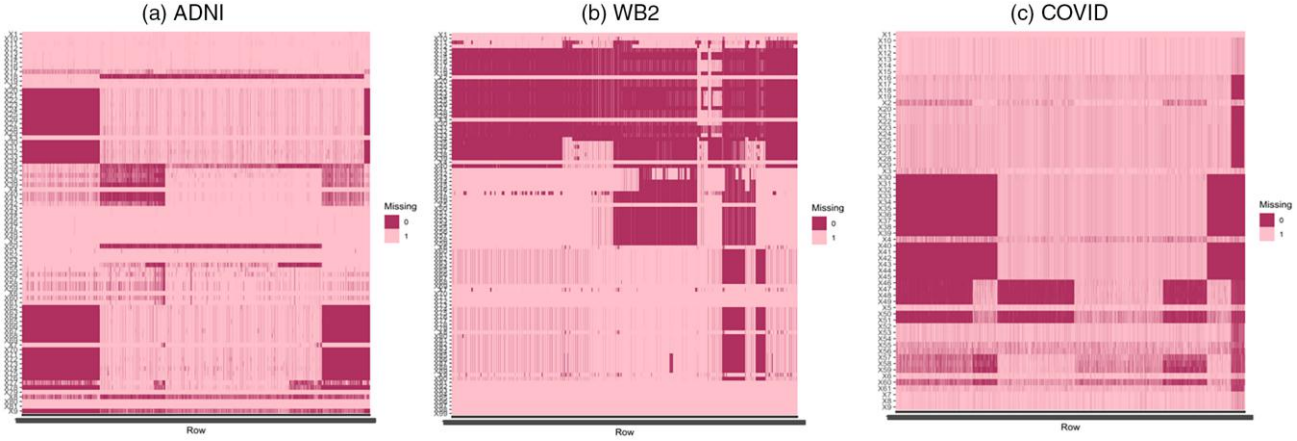


Table 5. Prediction Performance Evaluation on ADNI Data Set

Model	Benchmark	AUC	F1
GLM	IMSF	0.5178	0.7057
	SI	0.8431	0.8610
	GAIN	0.8563	0.8755
	COLUMNWISE	0.8633	0.8810
	REFE	0.8687	0.8846
	BRM_NOV	0.8815	0.8984
	BRM	**0.8838**	**0.9005**
GBM	REFE	0.8771	0.8963
	SI	0.8709	0.8967
	COLUMNWISE	0.8726	0.8967
	GAIN	0.8788	0.8989
	BRM_NOV	0.9083	0.9228
	BRM	**0.9114**	**0.9269**
CART	CART	0.8658	0.8943
	CART_I	0.8798	0.9022
	LAZY (TREE)	0.8773	0.8982

However, future studies can extend our work in terms of developing a model averaging procedure for BRM that results in coefficients that approximate the true model coefficients with complete data. We have considered the K-means algorithm with Euclidean distance measure for identifying blockwise patterns because it is the most widely used clustering method. However, other methods such as agglomerative or spectral or distribution-based clustering with other distance measures can be used instead without loss of generality. Soft clustering methods such as fuzzy C-means clustering may be explored instead of K-means clustering and overlapping subsets in future research exploiting incomplete data with blockwise missing patterns. It is important to note that, although the operationalization of BRM appears similar to common ensemble learning methods such as randomforest or bagging, it is different from them and not an alternative ensemble learner for complete data sets. That is, the candidate models are trained on subsets specific to the blockwise missing patterns rather than random subsamples of the training data set. We reported the results of experiments evaluating the prediction performance of BRM and benchmarks using default parameters for the nonlinear models to ensure uniformity. We conducted experiments that included a model tuning step; however, the relative performance of BRM and benchmarks did not vary significantly due to either model tuning or choice of the underlying model.

Table 6. Prediction Performance Evaluation on WB2 Data Set

Model	Benchmark	RMSE	SMAPE
GLM	BRM	**16.06**	**26.11**
	BRM_NOV	16.60	26.89
	REFE	17.42	28.51
	COLUMNWISE	17.46	28.57
	SI	17.84	29.58
	MI	19.12	30.46
	GAIN	19.72	30.99
	LAZY	17.10	26.39
GBM	BRM_NOV	**13.23**	**21.48**
	BRM	13.92	22.85
	GAIN	14.78	25.03
	COLUMNWISE	15.85	26.09
	REFE	15.89	26.09
	SI	16.22	26.53
	MI	17.50	28.43
CART	LAZY (TREE)	15.48	25.05
	CART_I	17.78	29.18

Table 7. Prediction Performance Evaluation on COVID Data Set

Model	Benchmark	AUC	F1
GLM	IMSF	0.5359	0.4384
	LISTWISE	0.6178	0.4969
	DISCOM	0.6311	0.5689
	GAIN	0.6352	0.5944
	SI	0.6480	0.6346
	COLUMNWISE	0.6532	0.6359
	MI	0.6366	0.6375
	REFE	0.6552	0.6401
	BRM	0.6570	0.6473
	BRM_NOV	**0.6582**	**0.6590**
GBM	LISTWISE	0.6034	0.4390
	GAIN	0.6261	0.4855
	MI	0.6360	0.5623
	REFE	0.6589	0.5976
	COLUMNWISE	0.6579	0.6203
	BRM	0.6651	0.6223
	BRM_NOV	0.6708	0.6522
	SI	**0.7099**	0.6886
CART	CART	0.6660	0.6762
	CART_I	0.6854	**0.6966**

6. Conclusion

In this study, we introduced a new method to analyze data with blockwise missing patterns that is based on the approach proposed by Friedman et al. (1996) geared toward circumventing the need for imputations. In contrast to the classical reduced modeling approach that suffers from the computational burden of training models at run-time based on the missing pattern of test instances, our method (BRM) limits the number of trained models by exploiting blockwise missing patterns identified in the training data. We benchmark BRM against existing methods using three complete data sets to show its advantages in prediction performance and running time. We further demonstrate its application to three incomplete data sets with actual blockwise missing patterns. BRM is applicable for predictive modeling using any tabular supervised learning method. It is particularly beneficial for applications that require the use of incomplete data with complex blockwise missing patterns that cannot be

identified through manual inspection. We anticipate that our method will facilitate consideration of incomplete data such as those with blockwise missing patterns that often have a story to tell but have traditionally been discounted in the past due to the inherent difficulty in analyzing them.

References

Apley DW, Zhu J (2020) Visualizing the effects of predictor variables in black box supervised learning models. *J. Roy. Statist. Soc. Ser B Statist. Methodology* 82(4).

Banner KM, Higgs MD (2017) Considerations for assessing model averaging of regression coefficients. *Ecological Appl.* 27(1).

Batista GEAPA, Monard MC (2003) An analysis of four missing data treatment methods for supervised learning. *Appl. Artificial Intelligence* 17(5–6):519–533.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and Regression Trees*, vol. 19 (Routledge).

Camino RD, Hammerschmidt CA, State R (2020) Working with deep generative models and tabular data imputation. *Proc. Internat. Conf. Machine Learn. (Vienna)*.

Chen X, Xie MG (2014) A split-and-conquer approach for analysis of extraordinarily large data. *Statist. Sinica* 24(4).

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B Methodological* 39(1):1–22.

Fanaee-T H, Gama J (2013) Event labeling combining ensemble detectors and background knowledge. *Progress Artificial Intelligence* 2:113–127.

Friedman N (1997) Learning belief networks in the presence of missing values and hidden variables. *Proc. 14th Internat. Conf. Machine Learn.* (Morgan Kaufmann Publishers Inc., San Francisco), 125–133.

Friedman JH, Kohavi R, Yun Y (1996) Lazy decision tree. *Proc. Thirteenth Natl. Conf. Artificial Intelligence*, vol. 1 (AAAI Press, Palo Alto, CA), 717–724.

GSA (2021) Wellbuilt for wellbeing. Accessed November 18, 2024, https://www.gsa.gov/governmentwide-initiatives/federal-highperformance-green-buildings/resource-library/health/wellbuilt-for-wellbeing.

Han J, Kamber M, Pei J (2022) *Data Mining: Concepts and Techniques*, 4th ed. (Morgan Kaufmann).

Hansen BE, Racine JS (2012) Jackknife model averaging. *J. Econometrics* 167(1):38–46.

Hastie T, Mazumder R, Lee JD, Zadeh R (2015) Matrix completion and low-rank SVD via fast alternating least squares. *J. Machine Learn. Res.* 16:3367–3402.

Ipsen NB, Mattei PA, Frellsen J (2020) How to deal with missing data in supervised deep learning? *Proc. Internat. Conf. Machine Learn.*

Kowarik A, Templ M (2016) Imputation with the R package VIM. *J. Statist. Software* 74:1–16.

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

Little RJA, Rubin DB (2002) *Statistical Analysis with Missing Data* (John Wiley & Sons, Hoboken, NJ).

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems* (Curran Associates Inc., Red Hook, NY), 4766–4775.

Mattei PA, Freiisen J (2019) MIWAE: Deep generative modelling and imputation of incomplete data sets. *Proc. 36th Internat. Conf. Machine Learn.* (PMLR, New York).

Mazumder R, Hastie T, Tibshirani R (2010) Spectral regularization algorithms for learning large incomplete matrices. *J. Machine Learn. Res.* 11:2287–2322.

Rubin DB (1976) Inference and missing data. *Biometrika* 63(3): 581–592.

Rubin DB (2004) *Multiple Imputation for Nonresponse in Surveys* (John Wiley & Sons, Hoboken, NJ).

Saar-Tsechansky M, Provost F (2007) Handling missing values when applying classification models. *J. Machine Learn. Res.* 8: 1625–1657.

Salomon JA, Reinhart A, Bilinski A, Chua EJ, la Motte-Kerr W, Rönn MM, Reitsma MB, et al. (2021) The US COVID-19 Trends and Impact Survey: Continuous real-time measurement of COVID-19 symptoms, risks, protective behaviors, testing, and vaccination. *Proc. Natl. Acad. Sci. USA* 118(51):e2111454118.

Schafer JL, Graham JW (2002) Missing data: Our view of the state of the art. *Psych. Methods* 7(2):142–177.

Schuurmans D, Greiner R (1997) Learning to classify incomplete examples. *Comput. Learn. Theory Natural Learn. Systems Making Learn. Systems Practice*, vol. 4 (MIT Press, Cambridge, MA), 87–105.

Smieja M, Struski L, Tabor J, Zielinski B, Spurek P (2018) Processing of missing data by neural networks. Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc., Red Hook, NY).

Srinivasan K, Currim F, Lindberg CM, Razjouyan J, Gilligan B, Lee H, Canada KJ, et al. (2023) Discovery of associative patterns between workplace sound level and physiological wellbeing using wearable devices and empirical Bayes modeling. *NPJ Digital Medicine* 6(1):1–10.

Srinivasan K, Currim F, Ram S, Lindberg C, Sternberg E, Skeath P, Najafi B, et al. (2016) Feature importance and prediction modeling for multi- source healthcare data with missing values. *Proc. 6th Internat. Conf. Digital Health* (ACM, New York), 1–8.

Stekhoven DJ, Buhlmann P (2012) Missforest: Non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28(1):112–118.

Ushey K, Allaire J, Tang Y (2018) reticulate: Interface to "Python." *CRAN: Contributed Packages*. https://doi.org/10.32614/CRAN.package.reticulate.

van Buuren S (2018) *Flexible Imputation of Missing Data*, 2nd ed. (CRC Press, Boca Raton, FL).

Van Buuren S, Groothuis-Oudshoorn K (2011) Multivariate imputation by chained equations. *J. Statist. Software* 45(3):1–67.

Woźnica K, Biecek P (2020) Does imputation matter? Benchmark for predictive models. *Proc. Internat. Conf. Machine Learn.*

Xiang S, Yuan L, Fan W, Wang Y, Thompson PM, Ye J (2013) Multi-source learning with block-wise missing data for Alzheimer's disease prediction. *Proc. 19th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York).

Xue F, Qu A (2021) Integrating multisource block-wise missing data in model selection. *J. Amer. Statist. Assoc.* 116(536): 1914–1927.

Yoon J, Jordon J, Van Der Schaar M (2018) GAIN: Missing data imputation using generative adversarial nets. *Proc. 35th Internat. Conf. Machine Learn.*, vol. 13 (PMLR, New York).

Yu G, Li Q, Shen D, Liu Y (2020) Optimal sparse linear prediction for block-missing multi-modality data without imputation. *J. Amer. Statist. Assoc.* 115(531):1406–1419.

Yuan L, Wang Y, Thompson PM, Narayan VA, Ye J (2012) Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *Neuroimage* 61(3):622–632.

Zhou D, Cai T, Lu J (2023) Multi-source learning via completion of block-wise overlapping noisy matrices. *J. Machine Learn. Res.* 221:1–43.

Zhu H, Li G, Lock EF (2020) Generalized integrative principal component analysis for multi-type data with block-wise missing structure. *Biostatistics (Oxford, England)* 21(2):302–318.