Causal Feature Selection with Missing Data

KUI YU and YAJING YANG, Hefei University of Technology
WEI DING, University of Massachusetts Boston

Causal feature selection aims at learning the Markov blanket (MB) of a class variable for feature selection. The MB of a class variable implies the local causal structure among the class variable and its MB and all other features are probabilistically independent of the class variable conditioning on its MB, this enables causal feature selection to identify potential causal features for feature selection for building robust and physically meaningful prediction models. Missing data, ubiquitous in many real-world applications, remain an open research problem in causal feature selection due to its technical complexity. In this article, we discuss a novel multiple imputation MB (MimMB) framework for causal feature selection with missing data. MimMB integrates Data Imputation with MB Learning in a unified framework to enable the two key components to engage with each other. MB Learning enables Data Imputation in a potentially causal feature space for achieving accurate data imputation, while accurate Data Imputation helps MB Learning identify a reliable MB of the class variable in turn. Then, we further design an enhanced kNN estimator for imputing missing values and instantiate the MimMB. In our comprehensively experimental evaluation, our new approach can effectively learn the MB of a given variable in a Bayesian network and outperforms other rival algorithms using synthetic and real-world datasets.

CCS Concepts: • **Computing methodologies → Feature selection;**

Additional Key Words and Phrases: Causal feature selection, markov blanket, bayesian network, missing data

ACM Reference format:
Kui Yu, Yajing Yang, and Wei Ding. 2022. Causal Feature Selection with Missing Data. *ACM Trans. Knowl. Discov. Data.* 16, 4, Article 66 (January 2022), 24 pages.
https://doi.org/10.1145/3488055

1 INTRODUCTION

Causal feature selection as an emerging type of efficient feature selection for high-dimensional data analytics has gradually attracted attention [26, 37], which learns the **Markov blanket** (**MB**) of a class attribute for feature selection [1, 12]. The notion of MB was invented in the context of a **Bayesian network** (**BN**) [18]. In a BN, the MB of a node (feature or variable) consists of parents

This work is supported by the National Key Research and Development Program of China (under grant 2020AAA0106100), National Science Foundation of China (under grant 61876206), and Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province (under grant CICIP2020003).
Authors' addresses: K. Yu and Y. Yang, Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education (Hefei University of Technology), and School of Computer Science and Information Engineering, Hefei University of Technology, 485 Daxia Road, Hefei, 230601, China; emails: yukui@hfut.edu.cn, yyj13865934683@163.com; W. Ding, Department of Computer Science, University of Massachusetts Boston, 100 Morrissey Blvd, Boston, 02125-3393, USA; email: Wei.Ding@umb.edu.

1556-4681/2022/01-ART66 $15.00
https://doi.org/10.1145/3488055

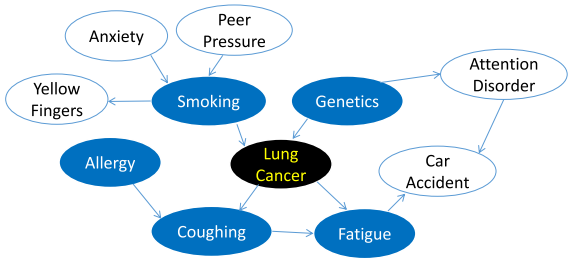


Fig. 1. A symbolic example of an MB in a lung-cancer BN.

(direct causes), children (direct effects), and spouses (other parents of the variable's children) of the variable. Figure 1 gives an example of an MB in the BN of lung cancer [12]. The MB of Lung cancer includes: Smoking and Genetics (parents), Coughing and Fatigue (children), and Allergy (spouse).

In a BN, all other nodes are probabilistically independent of a node conditioning on its MB. For example, given the MB of Lung cancer in Figure 1, i.e., Smoking, Genetics, Coughing, Fatigue, and Allergy, Lung cancer is independent of the remaining nodes. With this property, recent studies have theoretically proved that the MB of the class variable in a dataset is the optimal solution to the feature selection problem [38]. Furthermore, as can be seen in Figure 1, the MB of Lung cancer implies the local causal relationships between Lung cancer and the features (variables) in its MB, thus, the variables in the MB are potentially causal features, which can improve the robustness of predictive models and help to better understand many important problems in industrial and medical fields [19]. For example, in DNA microarray data analysis, it is crucial to select a small number of causal features (genes) from a high-dimensional gene dataset to help experts build robust prediction models for disease diagnosis or direct future experiments and studies [25]. Thus in recent years, many causal feature selection algorithms have been proposed for learning MB for feature selection [37].

Currently, almost all existing causal feature selection algorithms were designed only for complete datasets. However, missing data is a common phenomenon in many real-world applications. For instance, in medical diagnosis, some tests cannot be performed because the hospital lacks the necessary medical equipment, or certain medical examinations may not be suitable for certain patients, resulting in inevitable missing values in the dataset. It is not surprising that more than 40% of datasets in the UCI Machine Learning Repository have missing values [9], which is one of the most commonly used datasets collection for benchmarking machine learning procedures. The ubiquitous missing data problem becomes an obstacle of efficient causal feature selection. But it is still an open research problem for causal feature selection with missing data due to its technical complexity.

In this article, we propose a novel causal feature selection framework to learn MB from missing data, and our main contributions are as follows:

— We design the **multiple imputation MB** (**MimMB**) framework for causal feature selection with missing data. MimMB integrates Data Imputation with MB Learning in a unified framework to enable the two key components to engage with each other. MB Learning helps missing Data Imputation in a potentially causal feature space while Data Imputation provides an accurately imputed dataset for reliable MB Learning. Furthermore, this framework can easily be instantiated not only by existing causal feature selection algorithms, but also by classic feature selection methods.
— We propose a new **Enhanced kNN** (**EkNN**)-MB algorithm to instantiate the MimMB framework. In the EkNN-MB algorithm, an EkNN estimator is proposed for imputing

missing values by leveraging both complete and incomplete data samples. In addition, integrated with MB learning, EkNN only imputes missing values of the selected MB features instead of all features and thus it can tackle a large portion of missing values in a dataset.
— Using two synthetic datasets and five real-world datasets, we conduct extensive experiments to thoroughly validate the effectiveness of EkNN-MB and experimental results have shown that EkNN-MB achieves better performance than all rival algorithms under comparison.

The remaining of the article is organized as follows. Related work is presented in Section 2. Related notations, the MimMB framework and its instantiation are presented in Section 3. The experiments are reported in Section 4 and the article is concluded in Section 5.

2 RELATED WORK

In this section, we first review the methods that handle missing data and then discuss the methods of selecting features with incomplete datasets in literature.

2.1 Missing Data Treatment

Missing data has serious implications on the learning performance in machine learning [5, 13, 16, 17, 43]. Existing methods for dealing with missing data are roughly categorized into three strategies: case deletion, missing values imputation, and learning directly with missing data [8].

Case deletion directly discards the data samples with missing values [24]. When the ratio of missing data is high, this strategy easily results in serious bias and erroneous decisions. Learning directly with missing data does not require any imputation or deletion phase [34].

However, the majority of existing machine learning methods cannot be directly applied to a dataset with missing values. Hence, as the most commonly used method for dealing with incomplete datasets, missing values imputation has been used widely in various applications, such as wireless sensor networks [17] and software engineering [32]. For example, a simple imputation approach is the mean/mode imputation approach which imputes missing values with the average/mode of complete values for the corresponding numerical/categorical features [10]. But this approach may result in that all missing values in each feature are replaced with the same value.

More sophisticated imputation methods include the iterative imputation methods [6, 16, 43, 44], kNN [40], and decision trees [5]. As a popular method to deal with missing data, the kNN imputation approach has been gradually accepted by researchers because of its good performance and simplicity [28], and many kNN imputation methods have been proposed and successfully been applied to a broad range of real-world applications. Garcia et al. [11] proposed a kNN imputation method using a feature-weighted distance metric based on mutual information, which selects the k nearest instances considering the input attribute relevance to the target class. Pujianto et al. [22] demonstrated that the kNN imputation methods approach the accuracy of the complete data with different missing data. However, these methods only use complete instances for imputation, and the performances are seriously limited when the proportion of incomplete instances is high. A simple improvement consists in searching for the nearest neighbors of a missing data sample from both incomplete and complete data instances [13, 32, 41], which could cause the results have lower bias and higher variance and provide a relatively superior performance. However, the kNN imputation methods are effective for numerical variables, but do not perform well for categorical ones [35]. To address this problem, in this article, we will design a new strategy to calculate similarity for numerical attributes and categorical attributes.

2.2 Causal Feature Selection

As an emerging approach for feature selection, many causal feature selection methods have been proposed for learning the MB of the class attribute for feature selection [37]. The GS algorithm [15]

is the first MB learning algorithm. Tsamardinos et al. [31] proposed the IAMB based on the GS algorithm. Since then, many variants of IAMB have been proposed, such as inter-IAMB, IAMBnPC, inter-IAMBnPC, and Fast-IAMB [36]. GS and its variants greedily find PC (parents and children) and SP (spouses) of the class variable simultaneously without distinguishing PC of the class variable from its SP during MB learning. However, these algorithms require the number of samples to be exponential to the size of the MB, leading to inferior performance for feature selection when data samples are insufficient. To reduce the requirements of data samples of GS and its variants, the MMMB [1], HITON-MB [1], and PCMB algorithms [20] were proposed to learn PC and SP separately. Although these algorithms improve the data efficiency, but they are time inefficiency. To balance the learning efficiency and learning accuracy, the EEMB algorithm [33] was designed and it has a competitive efficiency with IAMB and its variants and achieves almost the same learning accuracy as MMMB, HITON-MB, and PCMB. However, all existing MB learning algorithms are designed for complete data and cannot deal with data with missing values. Neither was well-established classic/traditional feature selection methods designed for missing data [21, 30, 39].

For classic feature selection, in general, there are three widely used approaches to dealing with missing values for feature selection [10]. The first yet the most common approach is to simply discard data samples containing missing values in a dataset, then apply existing feature selection methods only to the remaining complete data samples [2]. In this case, informative features may not be able to be selected and this will lead to the problem of false negatives. In addition, this approach is not applicable when the number of variables exceeds the number of data samples in a dataset. Finally, when a dataset contains a large number of data samples with missing values, it is not a good strategy to simply apply this approach only to complete data samples in the dataset. The second approach is to fill in missing values before feature selection starts and it may be beneficial to reduce false negatives [23]. However, when a dataset is high-dimensional and has a large number of missing values, this approach will select many irrelevant features, leading to the problem of false positives. The third approach is to first implement feature selection, then impute missing values on the selected feature set [8, 23]. When a large number of features have missing values in a dataset, this approach is not able to select correctly feature subset before filling in all missing values, leading to poor performance. Recently, Zheng et al. [42] proposed an unsupervised feature selection on incomplete data sets for clustering analysis. And, Seijo-Pardo et al. [27] analyzed the impact of missing value imputation on feature selection. So far, there are still no effective approaches on feature selection with missing data. In this article, we design a novel and effective framework that missing value imputation and MB learning are implemented alternatively and promote each other. Our framework is designed in a way that both existing MB learning methods and classic feature selection algorithms can easily instantiate it to deal with datasets with missing values.

3 PROPOSED ALGORITHMS

In this section, we will propose the MimMB framework to deal with feature selection with missing values. This section is organized as follows. We firstly introduce some notations and definitions related to BNs and MBs in Section 3.1, secondly, we discuss a new framework named as MimMB in Section 3.2, thirdly, we further instantiate the MimMB framework with the EkNN-MB algorithm in Section 3.3, and finally, we discuss a new EkNN estimator for imputing missing data in Section 3.4.

3.1 Notations and Definitions

In this section, we briefly introduce some notations and definitions related to BNs and MBs. In this article, an instance that at least one attribute value is missing, is referred to as an incomplete instance, otherwise, we call it a complete instance.

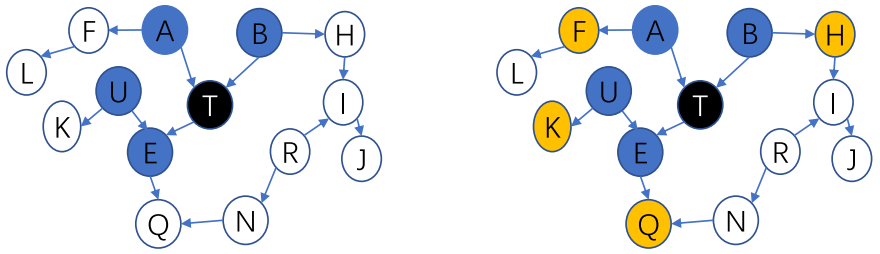


Fig. 2. (a) An example of the MB of T in a BN (the MB of T in blue) (left figure) and (b) an example of the extended MB of T in a BN. The extended MB of T consists of two parts of features: the MB of T in blue and the MBs of the variables within the MB of T in orange (excluding the MB of T in blue) (right figure).

Let $\boldsymbol{D} = \{x_i\}_{i=1}^{n}$ denote the set of n instances including incomplete and complete instances, where $x_i \in \mathbb{R}^{|v|}$ represents the ith instance and $|v|$ is the number of attributes of each instance. The set of instances $\boldsymbol{D} = \{x_i\}_{i=1}^{n}$ consists of two components: a set of incomplete instances $X_I = \{x_i\}_{i=1}^{|I|}$ and a set of complete instances $X_C = \{x_i\}_{i=1}^{|C|}$. Here, $|I|$ and $|C|$ are the numbers of incomplete instances and complete instances, respectively, and $|I| + |C| = n$. We denote the vth attribute values of the ith instance as $x[i, v]$ if it exists.

Assuming that V is the set of all variables, let P be a joint probability distribution and G denote a **directed acyclic graph** (**DAG**) defined on $\boldsymbol{V}$. The triplet $< \boldsymbol{V}, G, P >$ is called a BN if and only if every node of G is independent of any subset of its non-descendants given its parents.

Definition 3.1 (Faithfulness Ref. [29]). Given a BN $< \boldsymbol{V}, G, P >$, G is faithful to P if and only if every conditional independence present is entailed by G and the Markov condition. P is faithful if and only if there exists a DAG G such that G is faithful to P.

Definition 3.2 (Markov Blanket, MB Ref. [18]). In a faithful BN, the MB of a variable is unique and consists of its parents (direct causes), children (direct effects), and spouses (other parents of the variable's children).

Figure 2(a) gives a MB example. In a BN, the MB of a feature (node) renders the feature statistically independent of all the remaining features conditioning on the MB of this feature [18], as shown in Proposition 3.1 below.

Proposition 3.1 ([18]). *In a faithful BN, a variable is conditionally independent of all remaining variables in $V \setminus MB$ conditioning on this variable's MB.*

With Proposition 3.1, Corollary 3.1 as follows illustrates that a strongly relevant feature in traditional feature selection belongs to the MB of a class variable, and thus it states that learning the MB of a class variable is actually a procedure of feature selection.

Corollary 3.1 ([38]). *Under the faithfulness assumption, $\forall Y \in V$, Y belongs to $MB(T)$, if and only if Y is a strongly relevant feature.*

Furthermore, the work [38] has stated that under the faithfulness assumption, the MB of the class variable is the optimal feature subset for feature selection for classification.

Meanwhile, the MB of a class variable explicitly induces potential local causal relationships between the class attribute and the features, thus the MB naturally provides a causal and graphical interpretation about the optimal feature selection solution. Based on Definition 3.2, an extended MB of the class variable is defined as follows, which will be used in Section 4.

Definition 3.3 (Extended MB). In a faithful BN, the extended MB of a variable consists of two parts of features: the MB of the variable and the union of the MB of each variable within the MB of the variable (i.e., the MBs of the variables within the MB of T excluding the MB of T) (an example of the extended MB please see Figure 2(b)).

3.2 The MimMB Framework

As shown in Figure 3, the MimMB framework mainly consists of two subroutines: an Imputedata estimator for filling in missing values, and a FindMB procedure for learning the MB of the class variable. The two subroutines are implemented alternatively and promote each other, until the learned MB remains unchanged, that is, the remaining features being independent of the class variable is given the learnt MB.

Data imputation. To enable MimMB to handle a dataset containing a large portion of missing values, especially with the case of $|I| \gg |C|$, the Imputedata estimator aims at imputing missing values using both incomplete instances and complete instances in the current dataset (i.e., *Subdataset*).

Finding MB. After data imputation, the FindMB procedure will select an extended MB of the class variable instead of the MB of the class variable. As we discussed in Definition 3.3, an extended MB of the class variable consists of the MB of the class variable and the union of the MB of each variable within this MB of the class variable. The rationale behind this strategy is that since missing values are filled with plausible values using a data imputation method, according to the MB theory for feature selection, for a noise dataset, an extended MB of the class variable have a higher probability to containing the causally informative features than the MB of the class variable. In addition, selecting an extended MB may be the best strategy to select irrelevant features as few as possible. Together with multiple data imputations, selecting an extended MB can mitigate the problems of both false negatives and false positives in feature selection with missing data.

Updating *Subdataset*. The dataset *Subdataset* is updated as a new dataset only containing the extended MB. This makes the Imputedata estimator only fill in missing values for the causally informative features instead of all features in a dataset.

In summary, leveraging the unique property of MB for feature selection, the MimMB framework interleaves the Imputedata estimator and the FindMB procedure, and enable them to engage and help with each other, until the extended-MB achieves convergence. At each iteration, by selecting an extended MB of the class variable, the FindMB procedure not only reduces the impact of irrelevant features on missing value imputation, but also enables the Imputedata estimator to deal with a dataset with both high dimensionality and $|I| \gg |C|$. In turn, the Imputedata estimator only deals with missing values in the *Subdataset* dataset which is much smaller than D, this further improves the imputation accuracy of the Imputedata estimator. If missing values are imputed accurately, the FindMB procedure will have a high probability to return a stable and converged MB as soon as possible.

3.3 An Instantiation of MimMB: the EkNN-MB Algorithm

In this section, we propose an instantiation of the MimMB framework, called the EkNN-MB algorithm, as shown in Algorithm 1. EkNN-MB instantiates the *ImputeData(.)* function using a new missing data estimator, called EkNN, and the *FindMB(.)* function using the HITON-MB algorithm (a well-established MB learning algorithm) [1]. The EkNN estimator will be discussed in Section 3.4. At Lines 4–9, EkNN and HITON-MB are implemented alternatively and promote each other, until the extended MB of the class variable achieve stable (i.e., the remaining features in the dataset are independent of the class variable given the currently learnt MB). The details of EkNN-MB are discussed as follows.

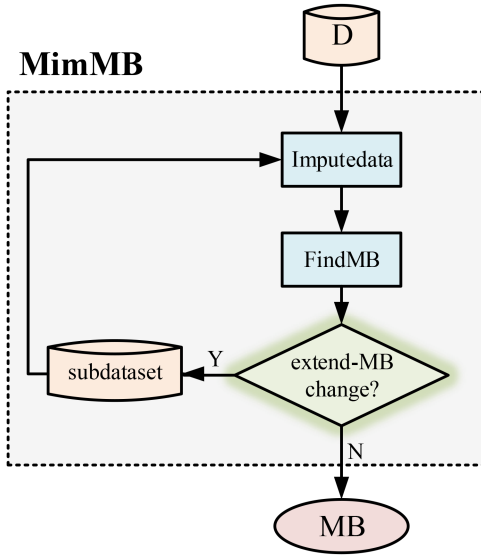


Fig. 3. The MimMB framework.

ALGORITHM 1: The EkNN-MB Algorithm

Input:

1: ***D***: an incomplete dataset, k: the number of neighbors, class: the class variable

Output: *MB*: the MB of the class variable

2: Initialize $Subdataset \leftarrow \boldsymbol{D}$
3: **repeat**
4: $\hat{Subdataset} \leftarrow EkNN(Subdataset, \text{k})$
5: $MB \leftarrow HITON\text{-}MB(\hat{Subdataset}, \text{class})$
6: $neigh\text{-}MB \leftarrow \cup\{HITON\text{-}MB(\hat{Subdataset}, x) | x \in MB\}$
7: $extended\text{-}MB \leftarrow MB \cup neigh\text{-}MB$
8: //Generate a new *Subdataset*
9: *Subdataset* ← generate a new *Subdataset* only containing the class variable and the variables in *extended-MB*
10: **until** *extended-MB* does not change or achieve stable
11: **return** *MB*

Line 2: Initialization. Initially, at Line 2, the imputation dataset *Subdataset* is set to D. Then at the first iteration of EkNN-MB (i.e., Lines 4–9), the EkNN estimator imputes all missing values in D, then HITON-MB learns an initial extended MB of the class variable on the imputed D (i.e., $\hat{Subdataset}$).

Line 4: Data imputation. At Line 4, EkNN imputes missing values using observed information of both incomplete instances and complete instances in the dataset *Subdataset*, and thus it enables EkNN-MB to tackle a large portion of missing values in a dataset, especially when $|I| \gg |C|$ holds.

Lines 5–7: Finding extended MB. At Lines 5–7, HITON-MB selects an extended MB of the class variable instead of the MB of the class variable. At Line 5, HITON-MB first learns the MB of the class variable, and at Line 6, it identifies the MB of each variable within this learnt MB of the class variable, called neigh-MB. At Line 7, the extended MB of the class variable is the union of

the learnt MB of the class variable and the set of neigh-MB. At Line 5, the *FindMB(.)* function is instantiated by a well-established MB learning algorithm, HITON-MB. It is a divide-and-conquer MB learning algorithm and breaks the problem of learning the MB of a variable into two subproblems: (1) learning parents and children of the variable; and (2) identifying the spouses of the variable based on the learnt parents and children of the variable. In fact, the *FindMB(.)* function in the MimMB framework can be instantiated in a straightforward way in any state-of-the-art MB learning algorithms, such as MMMB [1] and EEMB [33].

Line 9: Updating *Subdataset*. At Line 9, the dataset *Subdataset* is updated as a new dataset only containing the extended MB found at Lines 5–7.

3.4 The EkNN Estimator

Given an incomplete instance $x_i \in X_I$, to impute its missing value of the vth attribute, a general approach includes two steps: (1) finding the k nearest neighbors for x_i in the set of complete instances X_C, and (2) imputing $x[i, v]$ with the mode (mean) of vth attribute values of N_i if $x[i, v]$ is categorical (numerical). This imputation process can be formalized as follows.

$$x[i,v] = \begin{cases} \arg\max_v \sum_{j \in N_i} \mathbb{1}(x[j,v] = a), & \text{if } x[j,v] \text{ is categorical} \\ \frac{1}{k} \sum_{j \in N_i} x[j,v], & \text{if } x[j,v] \text{ is numerical} \end{cases}, \tag{1}$$

In Equation (1), N_i denote the k nearest neighbors of x_i, a is a value in the domain of the target variable v, and $\mathbb{1}$ (x[j,v]=a) is an indicator function that returns the value 1 if its argument is true and 0 otherwise. Existing kNN estimators perform the two steps described above for imputing missing values of each incomplete data instances.

However, almost all existing kNN estimators only use complete data instances for missing value imputation. In practice, many real-world datasets have a high ratio of missing values and the complete instances are insufficient for robust data imputation. Hence, this strategy is extremely vulnerable for identifying reliable k nearest neighbors, leading to that good nearest neighbors in the incomplete instances may be mistakenly discarded, resulting in biases in estimating missing values. Moreover, this approach may fail to handle a dataset with a large number of incomplete instances, especially when $|I| \gg |C|$ holds. All those problems motivate us to select k nearest neighbors of an incomplete instance by leveraging both incomplete and complete data instances. Furthermore, in order to further improve the performance of the imputation method, we use Minkowski distance metric for calculating similarity between numerical attributes and the weighted Hamming distance metric for categorical attributes.

We design an **E**nhanced k **N**earest **N**eighbor (EkNN) estimator. EkNN can impute missing values using observed information of both incomplete and complete data samples and uses different calculation criteria for different data types. It consists of three steps: (1) initializing missing values, (2) computing k nearest neighbors, and (3) imputing missing values, as shown in Algorithm 2.

Line 1: Initializing missing values. At Line 1, EkNN initializes missing values of a variable only using the mean/mode of the variable in the complete instances.

Lines 4 to 8: Computing k nearest neighbors. At Lines 4–11, EkNN aims at selecting the k nearest neighbors for each incomplete data instance. At Line 6, the distance(x,y) function calculates the distance between an incomplete instance x and an instance y in $\boldsymbol{D}$ and stores the computed distance in the set ***Dis***. At Line 8, all distances in the set ***Dis*** are sorted in descending. Using the sorted ***Dis***, EkNN gets the k nearest neighbors for x and stores them in the set $\boldsymbol{L}$. To instantiate the distance(x,y) function, according to the type of variable T, we select the Minkowski distance for

ALGORITHM 2: The EkNN Estimator

Input: $\boldsymbol{D}$, k
Output: $\hat{\mathbf{D}}$: The imputed dataset
1: Initialize missing values of a variable using the mean/mode of the variable in complete instances.
2: **repeat**
3: T ← ***CanA***.*pop()* // ***CanA*** is the set of attributes with missing values.
4: **for** each instance x in $\boldsymbol{X}_I$ **do**
5: **for** each instance y in $\boldsymbol{D}$ **do**
6: ***Dis*** ← distance(x, y) based on Equation (2)/(3) //***Dis*** sorts the distances between instances.
7: **end for**
8: $\boldsymbol{L}$← the first k $arg\min_{distance \in \boldsymbol{Dis}}$**sort**(***Dis***)
9: ***CanV***← get k values of attribute T according to $\boldsymbol{L}$
10: Calculate $\hat{x}$[i, T] based on ***CanV*** using formula (1)
11: **end for**
12: **until** ***CanA*** is empty
13: **return** $\hat{D}$

numerical attributes that has shown the advantage on calculating distance between data instances with numerical attributes and the weighted Hamming distance that is the best for categorical attributes, as defined as follows.

$$d(x_i, x_j) = (|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{iq} - x_{jq}|^p)^{\frac{1}{p}} \tag{2}$$

p is the Minkowski coefficient. When p equals 1, Equation (2) is treated as the Manhattan distance; when p equals 2, it is treated as the Euclidean distance.

Suppose instances x and y contain m variables, $x = (x_1, x_2, \ldots, x_m)$, $y = (y_1, y_2, \ldots, y_m) \in \Omega_1 \times \Omega_2 \times \cdots \times \Omega_m$. We use w_k to represent the weighted hamming distance between x_k and y_k, then the weighted hamming distance between instances x and y is defined as follows.

$$w(x, y) = \frac{1}{\frac{1}{w_1} + \frac{1}{w_2} + \cdots + \frac{1}{w_m}} \tag{3}$$

Lines 9 to 10: Imputing missing values. Using the set $\boldsymbol{L}$, at Line 9, we can acquire the candidate set ***CanV*** with k attribute values that are most relevant to the missing value $x[i, T]$. Then, according to Equation(1), the mean/mode of the set ***CanV*** is calculated as an estimate of $x[i, T]$ to replace the corresponding missing value initialized at Line 1.

4 EXPERIMENTS

In this section, we have carefully designed our experimental evaluations as follows.

(1) Datasets.

— Synthetic datasets. We carefully use two benchmark BNs to generate synthetic datasets with different missing rates to evaluate EkNN-MB in Section 4.1. For synthetic datasets, we evaluate EkNN-MB in two aspects. Firstly, since we can read off the MB of any variable from a benchmark BN, we evaluate EkNN-MB for the MB learning task by comparing the true MB of a variable in a benchmark BN with the MB learnt by EkNN-MB using a missing dataset

generated from this benchmark BN. Secondly, we consider a variable in a benchmark BN as the class variable and use the learnt MB of the variable for feature selection for classification.
— Real-world datasets. We conduct experiments to validate EkNN-MB using five real-world datasets for feature selection for classification in Section 4.2 and give a further analysis of EkNN in Section 4.3.

(2) Compared algorithms. We have adopted two strategies to validate the effectiveness of our proposed method. One strategy is that we first impute missing values in a dataset with the following data imputation methods, then we conduct feature selection on this imputed dataset using three classic feature selection methods, i.e., FCBF [39], LASSO [30], and mRMR [21], and a causal feature selection algorithm HITON-MB [1].

— **kNN.** The kNN imputation refers to the KNNImputer class from the Python Scikit-Learn library.[1]
— **ICkNN [32].** ICkNN, incomplete case k nearest neighbor imputation, has been proved to increase the effectiveness of nearest neighbor imputation compared to complete case kNN imputation.
— **MI [14].** MI, standing for mean imputation, is a method in which the missing value on a certain variable is replaced by the mean of the corresponding variable.

The second strategy is that we compare EkNN-MB with FCBF, LASSO, and mRMR by instantiating the MimMB framework using FCBF, LASSO, and mRMR. Specifically, in the MimMB framework, by employing the EkNN estimator for imputing missing values, we instantiate the function *FindMB(.)* using FCBF, LASSO, and mRMR, respectively.

(3) Implementation details.

— FCBF, LASSO, and mRMR are implemented in Matlab, while the other approaches are implemented in Python. The information threshold of FCBF is set to 0.01. The number of selected features of LASSO and mRMR is set to 15 for Alarm, splice-junction, breast-cancer, krvskp and 30 for Child5, bankrupty, madelon, respectively.
— We apply the 10-fold cross-validation for all datasets and adopt the **Support Vector Machine** (**SVM**) to compute the classification accuracy for all compared algorithms. All experiments are conducted on a computer with Intel(R) i7-8700 3.2GHz CPU and 16GB memory.
— The number of neighbors k for kNN is set to 9. G^2 tests are employed for the conditional independence tests with the statistical significance level of 0.01. Other parameters in the compared algorithms are set as suggested in the corresponding literature. The source codes of HITON-MB in the causal feature selection and structure learning package are available at our project GitHub website https://github.com/kuiy.

4.1 Benchmark BN Datasets

The benchmark BN datasets are sampled from two known benchmark BN networks, Alarm [3] and Child [4], as shown in Table 1. In order to test the performance of the proposed method, six different levels of missing values: 5%, 10%, 20%, 30%, 40%, and 50% were generated in every attribute except the class attribute in a dataset, respectively. We choose 10 variables in a BN as the class variables for each dataset. For each class variable and each level of missing values in a dataset, we have run experiments 10 times. Therefore, for each level of missing values, $100(= 10 \times 10)$ datasets containing missing values were generated. Hence, from one complete dataset, $600(= 100 \times 6)$

[1] https://scikit-learn.org/stable/modules/impute.html.

Table 1. Summary of Benchmark BNs

Network	Num. Vars	Num. Edges	Max In-/out-Degree	Min/Max MBset	Data Size
Alarm	37	46	4/5	1/8	5,000
Child5	100	126	2/7	1/8	5,000

datasets with missing values were generated and a total of 1200($= 600 \times 2$) incomplete datasets were used in the experiments.

For a benchmark BN dataset, since the MB of each feature can be read from the corresponding BN network, we firstly evaluate the performance of EkNN-MB for MB learning with missing data using the following metrics.

- **Precision.** Precision is the number of true positives in the output (the features in the output belonging to the true MB of a variable) divided by the number of variables in the output of an algorithm.
- **Recall.** Recall is the number of true positives in the output divided by the number of true positives (true MB of a variable) in the BN.
- **Distance.** Distance is calculated as $\sqrt{(1-precision)^2 + (1-recall)^2}$ for balancing precision and recall. Smaller value of the distance is better.

Secondly, we consider a variable in a benchmark BN as the class variable and use the learnt MB of the variable for classification using the metric **Classification Accuracy** that is the percentage of the correctly classified test instances that were previously unseen. In the following tables, the distance, precision, recall, and classification accuracy for each missing rate of each dataset are calculated on averaging the 100 different results. The best results are highlighted in bold face in the tables.

4.1.1 Results of EkNN-MB, kNN-MB, ICkNN-MB, and MI-MB using the First Strategy. Using the first strategy, we first impute missing values in a dataset with the data imputation methods as we previously discussed (i.e., kNN, ICkNN, MI), then learn the MB from the imputed dataset using HITON-MB. With this strategy, we produce three MB learning algorithms to handle missing data, kNN-MB, ICkNN-MB, and MI-MB, respectively.

Table 2 summarizes the average Distance, Precision, Recall, and Classification Accuracy for six different missing rates on the Benchmark BN datasets. From the table, we observe that EkNN-MB is significantly better or at least has the similar results as kNN-MB, ICkNN-MB, and MI-MB on the precision, recall and distance metrics. Meanwhile, EkNN-MB is significantly better than kNN-MB, ICkNN-MB, and MI-MB on the classification accuracy. The experimental results indicate that it is crucially important to learn an accurate MB from a missing dataset for feature selection for classification.

Figure 4 shows the classification accuracies of different methods with different levels of missing rates. We can see that EkNN-MB is significantly better than the other three methods at all levels of missing rates using the Alarm datasets. For the Child5 datasets, EkNN-MB is significantly better than its rivals when the missing rate is greater than 10% or above, especially when the missing rate is bigger than 20%.

4.1.2 Results of EkNN-MB, FCBF, LASSO, and mRMR Using the Second Strategy. In this section, with the second strategy for evaluating EkNN-MB, in the MimMB framework, we instantiate the function *FindMB(.)* using FCBF, LASSO, and mRMR, respectively. Then, we compare the feature

Table 2. Average Distance, Precision, Recall, and Accuracy of EkNN-MB, kNN-MB, ICkNN-MB, and MI-MB

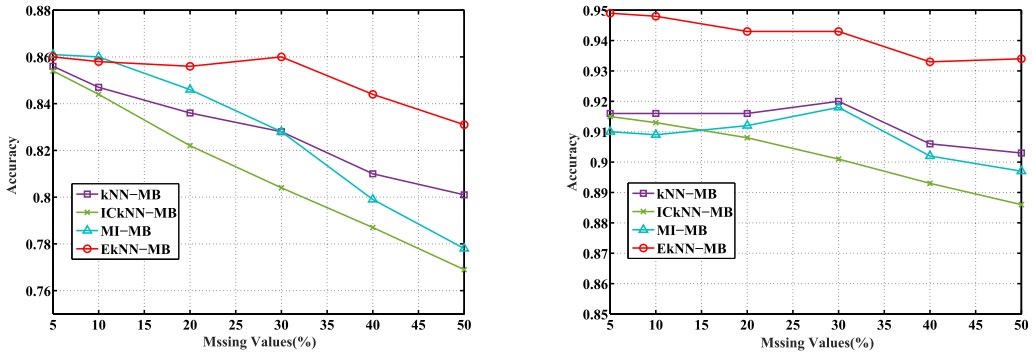
Dataset	Metric	Algorithm	5%	10%	20%	30%	40%	50%
Alarm	Distance	kNN-MB	0.033	0.112	0.233	0.512	0.614	0.662
		ICkNN-MB	**0.000**	0.062	0.205	0.412	**0.485**	0.629
		MI-MB	0.440	0.531	0.649	0.709	0.727	0.742
		EkNN-MB	**0.000**	**0.009**	**0.098**	**0.235**	0.512	**0.547**
	Precision	kNN-MB	0.967	0.888	0.767	0.492	0.385	0.340
		ICkNN-MB	**1.000**	0.951	0.795	0.593	**0.515**	0.376
		MI-MB	0.558	0.469	0.351	0.294	0.279	0.276
		EkNN-MB	**1.000**	**0.991**	**0.902**	**0.766**	0.490	**0.469**
	Recall	kNN-MB	**1.000**	**1.000**	**1.000**	0.972	**0.988**	**0.976**
		ICkNN-MB	**1.000**	**1.000**	**1.000**	0.974	0.973	0.964
		MI-MB	**1.000**	**1.000**	**1.000**	0.969	0.941	0.875
		EkNN-MB	**1.000**	**1.000**	**1.000**	**1.000**	**0.988**	**0.976**
	Accuracy	kNN-MB	0.916	0.916	0.916	0.920	0.906	0.903
		ICkNN-MB	0.915	0.913	0.908	0.901	0.893	0.886
		MI-MB	0.910	0.909	0.912	0.918	0.902	0.897
		EkNN-MB	**0.949**	**0.948**	**0.943**	**0.943**	**0.933**	**0.934**
Child5	Distance	kNN-MB	0.040	0.122	0.301	0.378	0.426	**0.443**
		ICkNN-MB	0.039	0.055	0.202	**0.328**	0.468	0.641
		MI-MB	0.150	0.290	0.593	0.667	0.730	0.779
		EkNN-MB	**0.000**	**0.029**	**0.131**	0.373	**0.395**	0.494
	Precision	kNN-MB	0.960	0.881	0.709	0.631	0.589	**0.574**
		ICkNN-MB	0.961	0.945	0.823	**0.685**	0.545	0.372
		MI-MB	0.850	0.712	0.412	0.340	0.278	0.255
		EkNN-MB	**1.000**	**0.972**	**0.840**	0.641	**0.619**	0.521
	Recall	kNN-MB	**1.000**	0.992	0.958	**0.950**	0.908	0.892
		ICkNN-MB	**1.000**	**1.000**	0.950	0.933	0.900	0.875
		MI-MB	**1.000**	0.992	0.950	0.925	0.908	0.783
		EkNN-MB	**1.000**	**1.000**	**0.959**	0.934	**0.923**	**0.900**
	Accuracy	kNN-MB	0.856	0.847	0.836	0.828	0.810	0.801
		ICkNN-MB	0.854	0.844	0.822	0.804	0.787	0.769
		MI-MB	**0.861**	**0.860**	0.846	0.828	0.799	0.778
		EkNN-MB	0.860	0.858	**0.856**	**0.860**	**0.844**	**0.831**

sets selected by FCBF, LASSO, and mRMR with the MB set selected by EkNN-MB at six levels of missing rates for classification using SVM.

Table 3 presents the average classification accuracies of EkNN-MB, FCBF, LASSO, and mRMR using SVM under six levels of missing rates. It is clear that EkNN-MB is significantly better than FCBF, LASSO, and mRMR. For example, using the Child5 dataset with missing rate 30%, EkNN-MB achieves 9.6%, 13.0%, and 15.3% higher classification accuracy than FCBF, LASSO, and mRMR, respectively. Furthermore, Figure 5 gives the classification accuracies of EkNN-MB, FCBF, LASSO, and mRMR with different levels of missing rates. We can observe that EkNN-MB is much more accurate than FCBF, LASSO, and mRMR for feature selection for classification with missing data. The possible explanation is that EkNN-MB can capture potential causal features and does not need to specify the number of selected features before it starts, while FCBF, LASSO, and mRMR cannot.

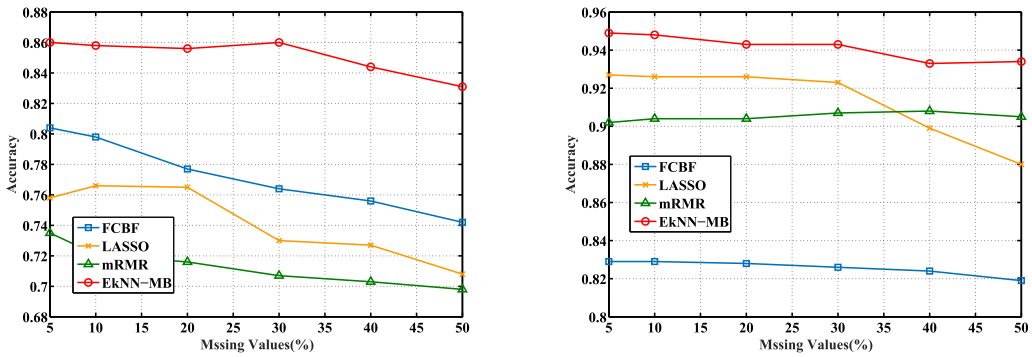
Table 3. Average Classification Accuracies of EkNN-MB, FCBF, LASSO and mRMR Using SVM

Dataset	Algorithm	5%	10%	20%	30%	40%	50%
Alarm	FCBF	0.829	0.829	0.828	0.826	0.824	0.819
	LASSO	0.927	0.926	0.926	0.923	0.899	0.880
	mRMR	0.902	0.904	0.904	0.907	0.908	0.905
	EkNN-MB	**0.949**	**0.948**	**0.943**	**0.943**	**0.933**	**0.934**
Child5	FCBF	0.804	0.798	0.777	0.764	0.756	0.742
	LASSO	0.758	0.766	0.765	0.730	0.727	0.708
	mRMR	0.735	0.721	0.716	0.707	0.703	0.698
	EkNN-MB	**0.860**	**0.858**	**0.856**	**0.860**	**0.844**	**0.831**



(a) Child5 dataset (b) Alarm dataset

Fig. 4. Classification accuracies of EkNN-MB, kNN-MB, ICkNN-MB and MI-MB by varying missing rates.



(a) Child5 dataset (b) Alarm dataset

Fig. 5. Classification accuracies of EkNN-MB, FCBF, LASSO, and mRMR by varying missing rates.

4.2 Real-World Datasets

Five real-world complete datasets are selected from the UCI Machine Learning Repository and NIPS2003 feature selection challenge datasets. Table 4 summarizes the main characteristics of each dataset including the number of instances and the number of features. To test the performance of EkNN-MB, missing values for each dataset are randomly introduced into all attributes except the

Table 4. Summary of Real-World Datasets

Dataset	Number of features	Number of instances
breast-cancer	32	569
krvskp	36	3,196
splice-junction	61	3,190
bankruptcy	147	7,063
madelon	500	2,000

class variable using missing rates of 5%, 10%, 20%, 30%, 40%, 50%. At the same time, in order to reduce the likelihood of obtaining biased results by randomly introducing missing values, each missing level is performed 10 times over each dataset. Therefore, from one complete dataset 60(= 10×6) datasets containing missing values were generated and a total of 300(= 60×5) incomplete datasets were used in the experiments.

Since the MB of each variable in a real-world dataset is not known, we use classification accuracy to evaluate the performance of EkNN-MB and its rivals for feature selection for classification with missing data. In addition to EkNN-MB, in this section, we also conduct the experiments by using the extended MB found by EkNN-MB for classfiction and the corresponding algorithm is called EkNN-EMB. In the following tables, the classification accuracy for each missing rate of each dataset is based on averaging 10 different classification results. The best results are highlighted in bold face in the tables.

4.2.1 Results of EkNN-MB, EkNN-EMB, and the 12 Rivals Using the First Strategy. With the first strategy, for each data imputation method, we conduct feature selection on the imputed dataset using FCBF, mRMR, LASSO, and HITON-MB, and thus produce the following 12 feature selection algorithms for handling missing data. Then, we compare EkNN-MB with those 12 feature selection algorithms using SVM.

— kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB;
— ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB;
— MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB.

Tables 5 to 9 present the average classification accuracies of EkNN-MB, EkNN-EMB, and the 12 feature selection algorithms using five real-world datasets with six missing data rates, respectively. We observe that EkNN-MB is better than kNN-FCBF, kNN-mRMR, and kNN-LASSO. ICkNN-MB is better than ICkNN-FCBF, ICkNN-mRMR, and ICkNN-LASSO. MI-MB is better than MI-FCBF, MI-mRMR, and MI-LASSO. We can also see that EkNN-MB is significantly better than kNN-MB, ICkNN-MB, and MI-MB when the missing rate is above 10%. In addition, these results strongly suggest the effectiveness of our method on datasets with high missing rates.

For EkNN-EMB and EkNN-MB, we find that EkNN-MB (using the MB for classification) is better than EkNN-EMB (using the extended MB for classification) on a dataset when the dataset has a large number of data instances, such as the krvskp and bankruptcy datasets. When the number of data instances becomes small, although EkNN-EMB outperforms EkNN-MB, the size of the extended MB is much larger than that of the MB, as shown in Table 12. The possible explanation is

Table 5. Average Accuracies of EkNN-EMB, EkNN-MB, kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB, ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB, MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB Using SVM on Splice-Junction

Algorithm	5%	10%	20%	30%	40%	50%
kNN-FCBF	0.910	0.900	0.865	0.849	0.839	0.829
kNN-mRMR	0.641	0.633	0.672	0.663	0.636	0.641
kNN-LASSO	0.848	0.841	0.800	0.795	0.796	0.778
kNN-MB	0.896	0.884	0.858	0.825	0.785	0.747
ICkNN-FCBF	0.907	0.897	0.847	0.840	0.808	0.777
ICkNN-mRMR	0.635	0.626	0.616	0.598	0.583	0.573
ICkNN-LASSO	0.848	0.845	0.807	0.790	0.756	0.718
ICkNN-MB	0.895	0.885	0.853	0.824	0.775	0.744
MI-FCBF	0.902	0.904	0.852	0.844	0.834	0.826
MI-mRMR	0.692	0.702	0.705	0.702	0.679	0.673
MI-LASSO	0.860	0.866	0.839	0.792	0.806	0.723
MI-MB	0.911	0.893	0.858	0.843	0.824	0.764
EkNN-MB	0.902	0.893	0.877	0.868	0.857	0.838
EkNN-EMB	**0.913**	**0.906**	**0.891**	**0.883**	**0.876**	**0.849**

Table 6. Average Accuracies of EkNN-EMB, EkNN-MB, kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB, ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB, MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB Using SVM on Breast-Cancer

Algorithm	5%	10%	20%	30%	40%	50%
kNN-FCBF	0.911	0.920	0.923	0.927	0.920	0.901
kNN-mRMR	0.948	0.944	0.956	0.948	0.947	0.940
kNN-LASSO	0.940	0.939	0.943	0.950	0.953	0.940
kNN-MB	0.960	0.956	0.960	0.951	0.953	0.962
ICkNN-FCBF	0.917	0.917	0.921	0.912	0.903	0.910
ICkNN-mRMR	0.946	0.943	0.937	0.932	0.927	0.911
ICkNN-LASSO	0.927	0.944	0.931	0.945	0.925	0.929
ICkNN-MB	0.960	0.958	0.957	0.947	0.946	0.941
MI-FCBF	0.927	0.941	0.941	0.935	0.938	0.932
MI-mRMR	0.957	0.960	0.950	0.939	0.938	0.933
MI-LASSO	0.953	0.955	0.941	0.935	0.932	0.931
MI-MB	**0.967**	**0.968**	0.951	0.959	0.964	0.966
EkNN-MB	0.961	0.961	0.967	0.961	0.968	**0.978**
EkNN-EMB	0.966	0.963	**0.971**	**0.968**	**0.973**	**0.978**

that a small number of data instances has an important impact on learning MB for feature selection. The problem of missing data further deteriorates the performance of MB learning on small-sized data. In this case, the extended MB may provide an alternative to the MB for feature selection with missing data under the MimMB framework.

Furthermore, we give the classification accuracies of EkNN-MB, EkNN-EMB, kNN-MB, ICkNN-MB, and MI-MB by varying missing rates, as shown in Figure 6.

Table 7. Average Accuracies of EkNN-EMB, EkNN-MB, kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB, ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB, MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB Using SVM on Krvskp

Algorithm	5%	10%	20%	30%	40%	50%
kNN-FCBF	0.688	0.700	0.713	0.708	0.701	0.711
kNN-mRMR	0.704	0.708	0.718	0.710	0.700	0.703
kNN-LASSO	0.717	0.721	0.722	0.715	0.710	0.718
kNN-MB	**0.972**	0.969	0.956	0.946	0.933	0.913
ICkNN-FCBF	0.690	0.688	0.691	0.693	0.677	0.650
ICkNN-mRMR	0.704	0.706	0.702	0.707	0.711	0.710
ICkNN-LASSO	0.713	0.715	0.709	0.707	0.693	0.669
ICkNN-MB	0.971	0.968	0.953	0.936	0.915	0.892
MI-FCBF	0.704	0.726	0.716	0.720	0.723	0.713
MI-mRMR	0.713	0.728	0.713	0.718	0.728	0.723
MI-LASSO	0.724	0.737	0.723	0.716	0.718	0.707
MI-MB	0.963	0.957	0.961	0.942	0.947	0.936
EkNN-MB	0.967	**0.972**	**0.969**	**0.970**	**0.972**	**0.971**
EkNN-EMB	**0.973**	0.966	0.964	0.959	0.964	0.953

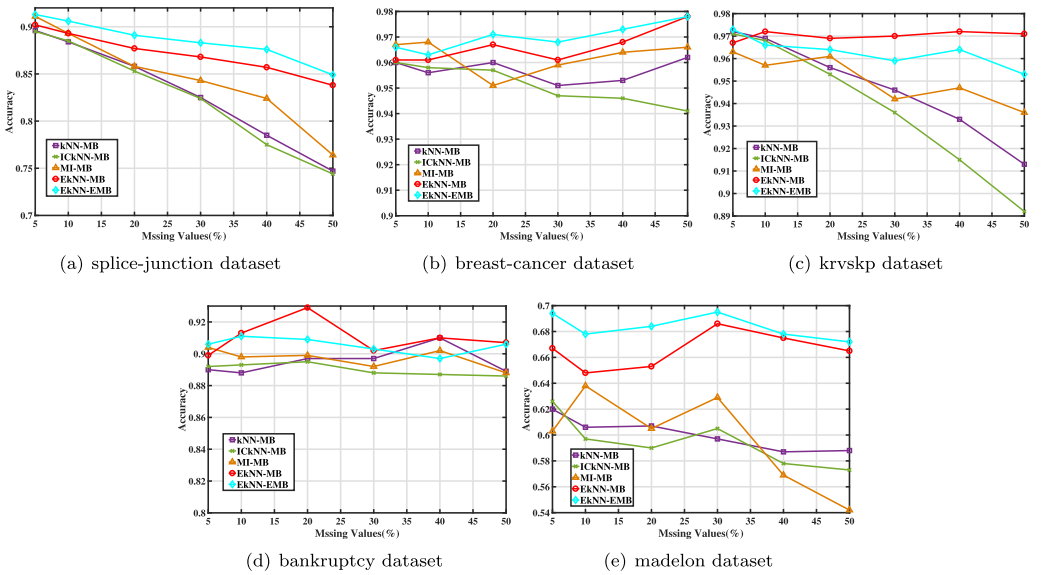
Table 8. Average Accuracies of EkNN-EMB, EkNN-MB, kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB, ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB, MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB Using SVM on Bankruptcy

Algorithm	5%	10%	20%	30%	40%	50%
kNN-FCBF	0.886	0.885	0.886	0.886	0.886	0.886
kNN-mRMR	0.886	0.886	0.886	0.886	0.886	0.886
kNN-LASSO	0.886	0.886	0.886	0.886	0.886	0.886
kNN-MB	0.890	0.888	0.897	0.897	**0.910**	0.889
ICkNN-FCBF	0.886	0.886	0.886	0.885	0.886	0.886
ICkNN-mRMR	0.886	0.886	0.886	0.886	0.886	0.886
ICkNN-LASSO	0.886	0.886	0.886	0.886	0.886	0.886
ICkNN-MB	0.892	0.893	0.895	0.888	0.887	0.886
MI-FCBF	0.886	0.886	0.885	0.885	0.886	**0.908**
MI-mRMR	0.886	0.886	0.886	0.889	0.886	0.886
MI-LASSO	0.886	0.886	0.886	0.889	0.906	0.905
MI-MB	0.904	0.898	0.899	0.892	0.902	0.888
EkNN-MB	0.899	**0.913**	**0.929**	0.902	**0.910**	0.907
EkNN-EMB	**0.906**	0.911	0.909	**0.903**	0.897	0.906

4.2.2 Results of EkNN-MB, EkNN-EMB, FCBF, LASSO, and mRMR Using the Second Strategy. With the second strategy, we compare EkNN-MB and EkNN-EMB with FCBF, LASSO, and mRMR using five real-world datasets. Table 10 presents the average classification accuracies of EkNN-MB, EkNN-EMB, FCBF, LASSO, and mRMR using five real-world datasets with six missing rates using SVM. In Table 10, EkNN-MB achieves better classification accuracy than FCBF, LASSO, and mRMR on all datasets excluding the splice-junction dataset. For example, for six different missing rates of the krvskp dataset, EkNN-MB outperforms FCBF, LASSO, and mRMR by at least 27.9%, 26.4%,

Table 9. Average Accuracies of EkNN-EMB, EkNN-MB, kNN-FCBF, kNN-mRMR, kNN-LASSO, kNN-MB, ICkNN-FCBF, ICkNN-mRMR, ICkNN-LASSO, ICkNN-MB, MI-FCBF, MI-mRMR, MI-LASSO, and MI-MB Using SVM on Madelon

Algorithm	5%	10%	20%	30%	40%	50%
kNN-FCBF	0.505	0.503	0.498	0.501	0.494	0.513
kNN-mRMR	0.506	0.515	0.509	0.507	0.514	0.526
kNN-LASSO	0.506	0.519	0.509	0.511	0.512	0.514
kNN-MB	0.620	0.606	0.607	0.597	0.587	0.588
ICkNN-FCBF	0.494	0.512	0.483	0.517	0.501	0.497
ICkNN-mRMR	0.504	0.507	0.501	0.509	0.509	0.510
ICkNN-LASSO	0.509	0.509	0.521	0.506	0.504	0.506
ICkNN-MB	0.626	0.597	0.590	0.605	0.578	0.573
MI-FCBF	0.506	0.497	0.507	0.520	0.484	0.510
MI-mRMR	0.509	0.521	0.518	0.540	0.540	0.535
MI-LASSO	0.507	0.522	0.531	0.532	0.539	0.537
MI-MB	0.603	0.638	0.605	0.629	0.569	0.542
EkNN-MB	0.667	0.648	0.653	0.686	0.675	0.665
EkNN-EMB	**0.694**	**0.678**	**0.684**	**0.695**	**0.678**	**0.672**



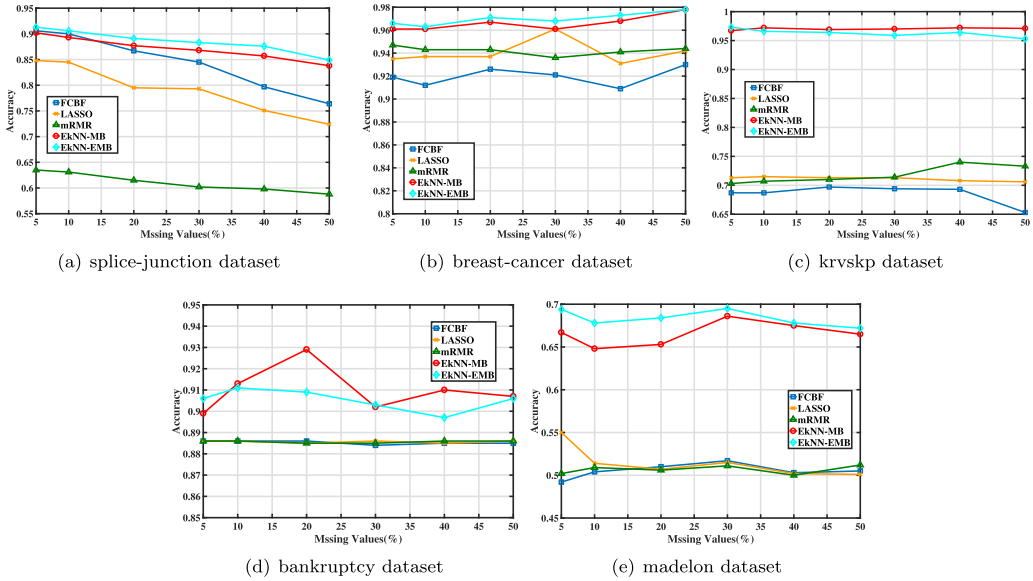
(a) splice-junction dataset (b) breast-cancer dataset (c) krvskp dataset

(d) bankruptcy dataset (e) madelon dataset

Fig. 6. Classification accuracies of EkNN-EMB, EkNN-MB, kNN-MB, ICkNN-MB, and MI-MB by varying missing rates.

23.2% in classification accuracy. Furthermore, we can see that the classification accuracy of FCBF, LASSO, and mRMR are almost the same on the bankruptcy dataset. The possible explanation is that the bankruptcy dataset has a highly imbalanced class distribution. Figure 7 shows that EkNN-MB and EkNN-EMB are much better than the other three feature selection methods.

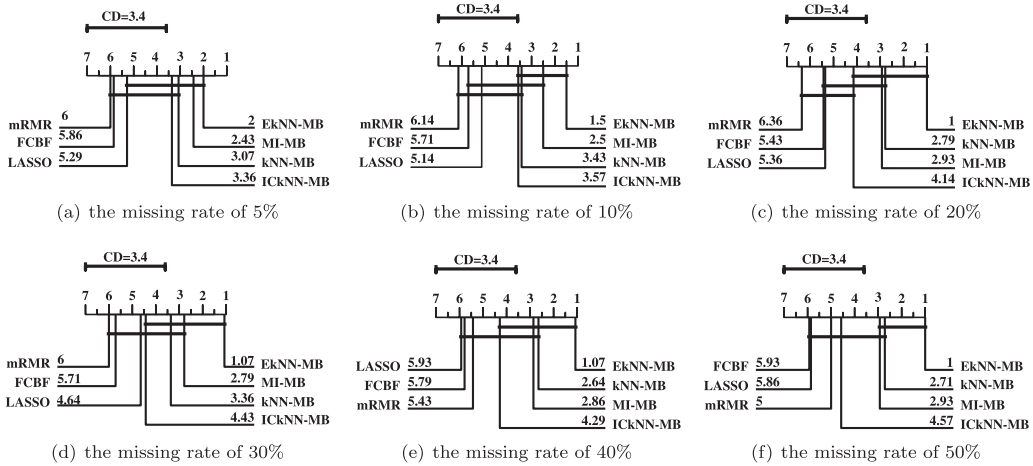
To further analyze the significant difference between EkNN-MB and its rivals on both synthetic and real-world datasets, we perform the Nemenyi test [7], which states that the performance of



(a) splice-junction dataset (b) breast-cancer dataset (c) krvskp dataset

(d) bankruptcy dataset (e) madelon dataset

Fig. 7. Classification accuracies of EkNN-EMB, EkNN-MB, FCBF, LASSO, and mRMR by varying missing rates.



(a) the missing rate of 5% (b) the missing rate of 10% (c) the missing rate of 20%

(d) the missing rate of 30% (e) the missing rate of 40% (f) the missing rate of 50%

Fig. 8. Crucial difference diagram of the Nemenyi test for classification accuracy with different missing data rates.

two algorithms is significantly different if the corresponding average ranks differ by at least one **critical difference** (**CD**). Figure 8 provides the CD diagrams, where the average rank of each algorithm is marked along the axis (lower ranks to the right). We observe that EkNN-MB is the only method that achieves the lowest rank at any level of missing rate. In particular, when the missing rate of a dataset reaches 10% or above, the rank value of EkNN-MB is below 1.5.

4.3 Analysis of EkNN

In this section, we analyze the effective of EkNN for missing value imputation against the two kNN imputation methods, kNN and ICkNN. In the EkNN-MB algorithm, we simply replace EkNN

Table 10. Average Accuracies of EkNN-EMB, EkNN-MB, FCBF, LASSO, and mRMR Using SVM

Dataset	Algorithm	5%	10%	20%	30%	40%	50%
splice-junction	FCBF	0.906	0.900	0.867	0.845	0.797	0.764
	LASSO	0.848	0.845	0.795	0.793	0.751	0.724
	mRMR	0.635	0.631	0.615	0.602	0.598	0.588
	EkNN-MB	0.902	0.893	0.877	0.868	0.857	0.838
	EkNN-EMB	**0.913**	**0.906**	**0.891**	**0.883**	**0.876**	**0.849**
breast-cancer	FCBF	0.919	0.912	0.926	0.921	0.909	0.930
	LASSO	0.935	0.937	0.937	0.961	0.931	0.942
	mRMR	0.947	0.943	0.943	0.936	0.941	0.944
	EkNN-MB	0.961	0.961	0.967	0.961	0.968	**0.978**
	EkNN-EMB	**0.966**	**0.963**	**0.971**	**0.968**	**0.973**	**0.978**
krvskp	FCBF	0.687	0.687	0.697	0.694	0.693	0.653
	LASSO	0.713	0.715	0.713	0.713	0.708	0.706
	mRMR	0.703	0.707	0.710	0.714	0.740	0.733
	EkNN-MB	0.967	**0.972**	**0.969**	**0.970**	**0.972**	**0.971**
	EkNN-EMB	**0.973**	0.966	0.964	0.959	0.964	0.953
bankruptcy	FCBF	0.886	0.886	0.886	0.884	0.885	0.885
	LASSO	0.886	0.886	0.885	0.886	0.885	0.886
	mRMR	0.886	0.886	0.885	0.885	0.886	0.886
	EkNN-MB	0.899	**0.913**	**0.929**	0.902	**0.910**	0.907
	EkNN-EMB	**0.906**	0.911	0.909	**0.903**	0.897	0.906
madelon	FCBF	0.492	0.504	0.510	0.517	0.503	0.505
	LASSO	0.500	0.514	0.507	0.515	0.502	0.501
	mRMR	0.502	0.509	0.506	0.511	0.500	0.512
	EkNN-MB	0.667	0.648	0.653	0.686	0.675	0.665
	EkNN-EMB	**0.694**	**0.678**	**0.684**	**0.695**	**0.678**	**0.672**

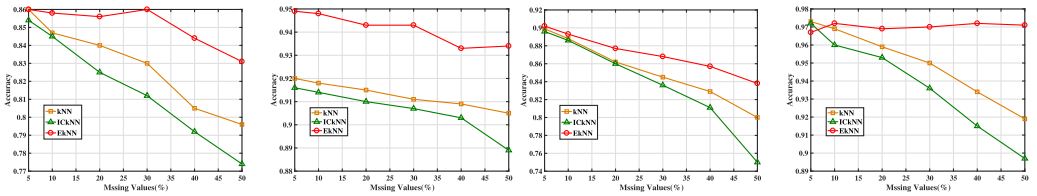
with the kNN and ICkNN methods, respectively, for data imputation. Table 11 summarizes the classification accuracies of different missing data rates on both synthetic and real-world datasets using SVM. We can see that the EkNN estimator outperforms the kNN and ICkNN methods for data imputation at different missing rates, especially when the missing rate is larger than 10%. Figure 9 shows the curve of classification accuracy of these three kNN imputation methods by varying missing rates. When the missing rate is larger than 10%, the performance of kNN and ICkNN will deteriorate caused by the inaccurate nearest neighbors identified, since ICkNN and kNN compute nearest neighbors only based on complete data samples and neglect informative data samples with missing values. On the contrary, EkNN uses not only complete instances but also incomplete instances for selecting the nearest neighbors of an incomplete data instance.

4.4 Convergence of EkNN-MB

In this section, we analyze the convergence of EkNN-MB from two perspectives: the variations of the extended MB and the classification accuracy of EkNN-MB using five real-world datasets, as shown in Figures 10 and 11. From the view of variations of the extended MB, we use the F1 metric to measure the difference between the extended MB learnt at each iteration and the extended MB achieved in the final iteration. $F1 = Precision * Recall/(Precision + Recall)$. The F1 score is the harmonic average of the precision and recall, where F1 = 1 is the best case that both precision

Table 11. Average Accuracies of EkNN, kNN, and ICkNN Using SVM

Dataset	Algorithm	5%	10%	20%	30%	40%	50%
Child5	IkNN	**0.860**	0.847	0.840	0.830	0.805	0.796
	ICkNN	0.854	0.845	0.825	0.812	0.792	0.774
	EkNN	**0.860**	**0.858**	**0.856**	**0.860**	**0.844**	**0.831**
Alarm	kNN	0.920	0.918	0.915	0.911	0.909	0.905
	ICkNN	0.916	0.914	0.910	0.907	0.903	0.889
	EkNN	**0.949**	**0.948**	**0.943**	**0.943**	**0.933**	**0.934**
splice-junction	kNN	0.899	0.888	0.862	0.845	0.829	0.800
	ICkNN	0.896	0.886	0.860	0.836	0.811	0.750
	EkNN	**0.902**	**0.893**	**0.877**	**0.868**	**0.857**	**0.838**
krvskp	kNN	**0.973**	0.969	0.959	0.950	0.934	0.919
	ICkNN	0.972	0.960	0.953	0.936	0.915	0.897
	EkNN	0.967	**0.972**	**0.969**	**0.970**	**0.972**	**0.971**



(a) Child5 (b) Alarm (c) splice-junction (d) krvskp

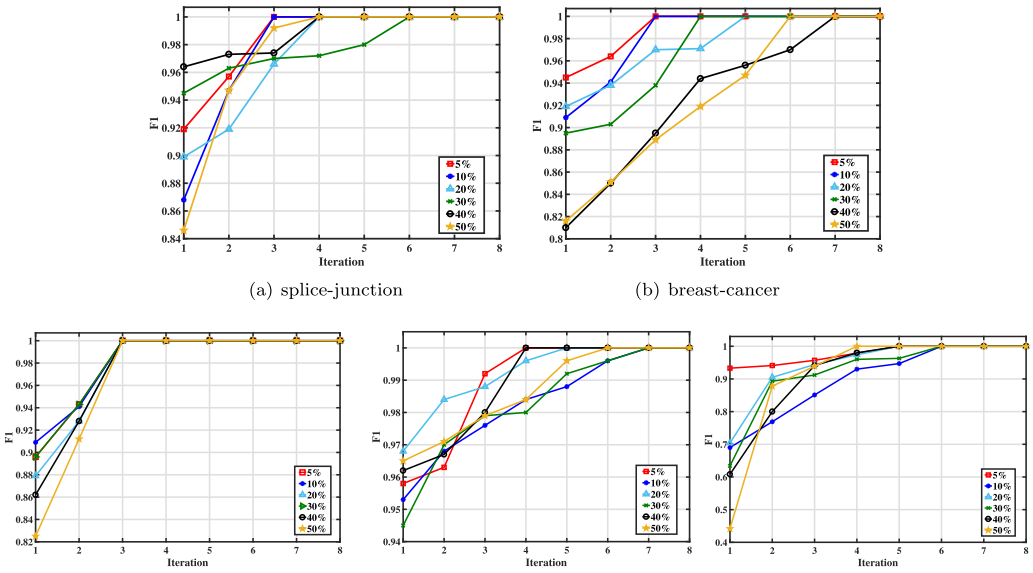
Fig. 9. Classification accuracies of EkNN, kNN, and ICkNN by varying missing rates.

and recall values equal to 1. This is to say, the extended MB learnt at the current iteration is the same as the extended MB found in the final iteration. Figure 10 shows the F1 values of EkNN-MB at each iteration on five real-world datasets with missing ratio 5%, 10%, 20%, 30%, 40%, and 50%, respectively. We can see that after several iterations, the extended MB learnt by EkNN-MB does not change any more and converges.

From the view of the classification accuracy of EkNN-MB, Figure 11 shows the classification accuracy using the MB learnt by EkNN-MB at each iteration on five real-world datasets with missing ratio 5%, 10%, 20%, 30%, 40%, and 50%, respectively. Referring to Figure 10, we can see that when the learnt extended MB does not change, the final classification accuracy of EkNN-MB will also become stable. In Figure 11, these results show that the performance of EkNN-MB becomes better as the number of iterations increases. After several iterations, the classification accuracy of EkNN-MB becomes stable and unchanges.

5 CONCLUSION AND FURTURE WORK

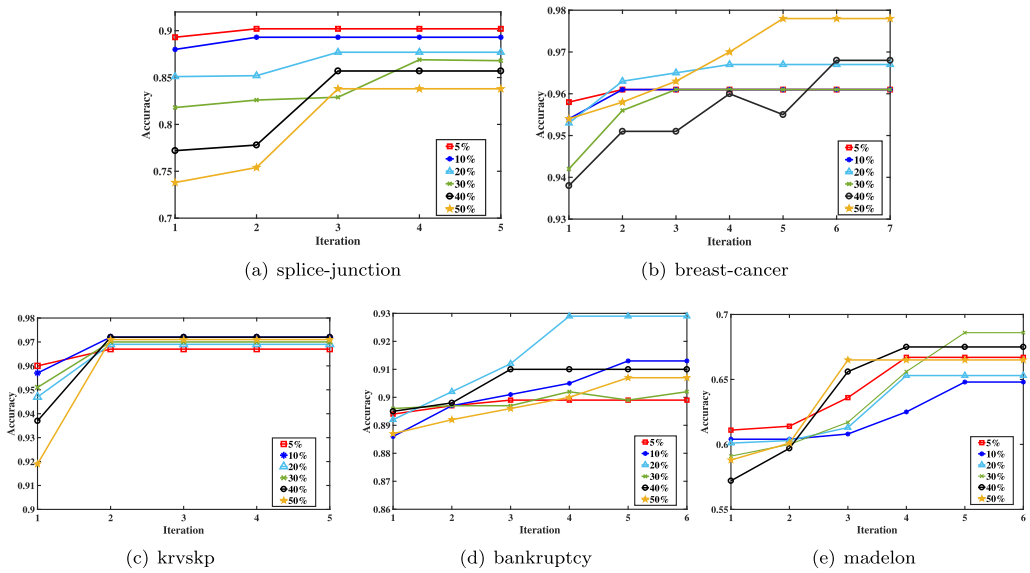
In this article, we have integrated data imputing and MB learning as a unified framework and proposed a new MimMB framework for feature selection with missing data. This framework can be instantiated not only by existing causal feature selection algorithms, but also by existing traditional feature selection methods in a straightforward way. The property of MimMB enables existing MB/traditional feature selection methods to deal with a dataset with missing values without any modifications. To address the problem of data imputation, we design a new EkNN algorithm and propose the EkNN-MB algorithm to instantiate the MimMB framework. We conduct



(a) splice-junction (b) breast-cancer

(c) krvskp (d) bankruptcy (e) madelon

Fig. 10. Convergence of EkNN-MB using F1 on different datasets with different missing rates.



(a) splice-junction (b) breast-cancer

(c) krvskp (d) bankruptcy (e) madelon

Fig. 11. Convergence of EkNN-MB using classification accuracy on different datasets with different missing rates.

comprehensive experiments using both synthetic and real-world datasets and the experimental results have validated the effectiveness of EkNN-MB.

In future, first, although the EkNN algorithm achieves better performance than existing data imputation methods, it needs to search through the entire dataset in the first iteration of the

Table 12. Number of Features in the MB and Extended-MB(EMB) at the Final Iteration of EkNN-MB

	5%		10%		20%		30%		40%		50%	
	MB	EMB	MB	EMB	MB	EMB	MB	EMB	MB	EMB	MB	EMB
splice-junction	14	56	16	56	17	57	16	52	18	55	22	59
breast-cancer	6	28	4	18	4	17	4	17	8	17	8	20
krvskp	16	35	16	36	19	36	13	36	15	36	18	36
bankruptcy	42	99	47	102	45	107	46	101	43	103	52	116
madelon	8	27	7	20	7	19	9	26	12	24	9	23

EkNN-MB algorithm. This limits the application of EkNN to a large dataset, we plan to design new data imputation methods. In addition, for finding advantages and disadvantages of each data imputation method for feature selection, we will extensively examine the impact of all representative and state-of-the-art data imputation methods on feature selection using the MimMB framework. Furthermore, in the MimMB framework, we simply employ an existing MB learning algorithm for feature selection. We plan to design new MB learning algorithms that are more suitable for the MimMB framework to deal with missing data. Finally, in general, the MB of a class variable is the optimal feature subset for feature selection under the independent and identically distributed data setting. That is to say, parents, children, and spouses of the class variable are all strongly relevant features for feature selection. However, under the out-of-distribution data setting, only the parents in the MB of the class variable are potential causes and would be invariant features. Clearly, it is not easy to distinguish parents (causes) from children (effects) from observational data. Then, we will explore how to leverage the MimMB framework for feature selection with missing data when training data and testing data are sampled from different distributions.

REFERENCES

[1] Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos. 2010. Local causal and markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research* 11, 1 (2010), 171–234.

[2] Alex Aussem and Sergio Rodrigues de Morais. 2010. A conservative feature subset selection algorithm with missing data. *Neurocomputing* 73, 4–6 (2010), 585–590.

[3] Ingo A. Beinlich, Henri Jacques Suermondt, R. Martin Chavez, and Gregory F. Cooper. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*. Springer, 247–256.

[4] Robert G. Cowell, Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. 2006. *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*. Springer.

[5] Rupam Deb and Alan Wee-Chung Liew. 2016. Missing value imputation for the analysis of incomplete traffic accident data. *Information Sciences* 339 (2016), 274–289.

[6] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[7] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, Jan (2006), 1–30.

[8] Gauthier Doquire and Michel Verleysen. 2012. Feature selection with missing data using mutual information estimators. *Neurocomputing* 90 (2012), 3–11.

[9] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. Retrieved October 22, 2020 from http://archive.ics.uci.edu/ml.

[10] Pedro J. García-Laencina, José-Luis Sancho-Gómez, and Aníbal R. Figueiras-Vidal. 2010. Pattern classification with missing data: A review. *Neural Computing and Applications* 19, 2 (2010), 263–282.

[11] Pedro J. García-Laencina, José-Luis Sancho-Gómez, Aníbal R. Figueiras-Vidal, and Michel Verleysen. 2009. K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing* 72, 7–9 (2009), 1483–1493.

[12] Isabelle Guyon, Constantin Aliferis, and André Elisseeff. 2007. Causal feature selection. In *Computational Methods of Feature Selection* (2007). Chapman and Hall/CRC, 63–82.
[13] Jianglin Huang, Jacky Wai Keung, Federica Sarro, Yan-Fu Li, Yuen-Tak Yu, WK Chan, and Hongyi Sun. 2017. Cross-validation based K nearest neighbor imputation for software quality datasets: An empirical study. *Journal of Systems and Software* 132 (2017), 226–252.
[14] Lawrence R. Landerman, Kenneth C. Land, and Carl F. Pieper. 1997. An empirical evaluation of the predictive mean matching method for imputing missing values. *Sociological Methods & Research* 26, 1 (1997), 3–33.
[15] Dimitris Margaritis and Sebastian Thrun. 1999. Bayesian network induction via local neighborhoods. In *Proceedings of the Advances in Neural Information Processing Systems*. 505–511.
[16] Vahid Nassiri, Geert Molenberghs, Geert Verbeke, and João Barbosa-Breda. 2020. Iterative multiple imputation: A framework to determine the number of imputed datasets. *The American Statistician* 74, 2 (2020), 125–136.
[17] Liqiang Pan and Jianzhong Li. 2010. K-nearest neighbor based missing data estimation algorithm in wireless sensor networks. *Wireless Sensor Network* 2, 02 (2010), 115.
[18] Judea Pearl. 2014. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier.
[19] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. Basic Books.
[20] Jose M. Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. 2007. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning* 45, 2 (2007), 211–232.
[21] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 8 (2005), 1226–1238.
[22] Utomo Pujianto, Aji Prasetya Wibawa, and Muhammad Iqbal Akbar. 2019. K-nearest neighbor (K-NN) based missing data imputation. In *Proceedings of the 5th International Conference on Science in Information Technology*. IEEE, 83–88.
[23] Wenbin Qian and Wenhao Shu. 2015. Mutual information criterion for feature selection from incomplete data. *Neurocomputing* 168 (2015), 210–220.
[24] J. Ross Quinlan. 1989. Unknown attribute values in induction. In *Proceedings of the 6th International Workshop on Machine Learning*. Elsevier, 164–168.
[25] Beatriz Remeseiro and Veronica Bolon-Canedo. 2019. A review of feature selection methods in medical applications. *Computers in Biology and Medicine* 112 (2019), 103375.
[26] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proceedings of the IEEE* 109, 5 (2021), 612–634.
[27] Borja Seijo-Pardo, Amparo Alonso-Betanzos, Kristin P. Bennett, Veronica Bolon-Canedo, Julie Josse, Mehreen Saeed, and Isabelle Guyon. 2019. Biases in feature selection with missing data. *Neurocomputing* 342 (2019), 97–112.
[28] J. G. Skellam. 1952. Studies in statistical ecology: I. Spatial pattern. *Biometrika* 39, 3/4 (1952), 346–362.
[29] Peter Spirtes, Clark N. Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, Prediction, and Search*. MIT press.
[30] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.
[31] Ioannis Tsamardinos, Constantin F. Aliferis, Alexander R. Statnikov, and Er Statnikov. 2003. Algorithms for large scale markov blanket discovery. In *Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference*. 376–380.
[32] Jason Van Hulse and Taghi M. Khoshgoftaar. 2014. Incomplete-case nearest neighbor imputation in software measurement data. *Information Sciences* 259 (2014), 596–610.
[33] Hao Wang, Zhaolong Ling, Kui Yu, and Xindong Wu. 2020. Towards efficient and effective discovery of markov blankets for feature selection. *Information Sciences* 509 (2020), 227–242.
[34] David Williams, Xuejun Liao, Ya Xue, and Lawrence Carin. 2005. Incomplete-data classification using logistic regression. In *Proceedings of the 22nd International Conference on Machine Learning*. 972–979.
[35] Zeshui Xu and J. Chen. 2008. An overview of distance and similarity measures of intuitionistic fuzzy sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 16, 04 (2008), 529–555.
[36] Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative markov blanket discovery for optimal feature selection. In *Proceedings of the 5th IEEE International Conference on Data Mining*. IEEE, 809–812.
[37] Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys* 53, 5 (2020), 1–36.
[38] Kui Yu, Lin Liu, and Jiuyong Li. 2021. A unified view of causal and non-causal feature selection. *ACM Transactions on Knowledge Discovery from Data* 15, 4 (2021), 1–46.
[39] Lei Yu and Huan Liu. 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5, Oct (2004), 1205–1224.

[40] Shichao Zhang. 2011. Shell-neighbor method and its application in missing data imputation. *Applied Intelligence* 35, 1 (2011), 123–133.
[41] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. 2017. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology* 8, 3 (2017), 1–19.
[42] Wei Zheng, Xiaofeng Zhu, Yonghua Zhu, and Shichao Zhang. 2018. Robust feature selection on incomplete data. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3191–3197.
[43] Xiaofeng Zhu, Jianye Yang, Chengyuan Zhang, and Shichao Zhang. 2021. Efficient utilization of missing data in cost-sensitive learning. *IEEE Transactions on Knowledge and Data Engineering* 33, 6 (2021), 2425–2436.
[44] Xiaofeng Zhu, Shichao Zhang, Zhi Jin, Zili Zhang, and Zhuoming Xu. 2010. Missing value estimation for mixed-attribute datasets. *IEEE Transactions on Knowledge and Data Engineering* 23, 1 (2010), 110–121.

Received March 2021; revised July 2021; accepted September 2021