Combining Imputation Method and Feature Weighting Algorithms to Improve the Classification Accuracy of Incomplete Data

Yue Wang¹, Cong Guo¹, Chunjing Xiao¹, Wei Yang¹

¹School of Computer and Information Engineering, Henan University, Kai Feng, China
*email: yangwei@henu.edu.cn

Abstract. It is a common problem that datasets in pattern classification tasks contain missing values. In order to deal with missing values, the K-Nearest Neighbor Imputation method with Euclidean distance is often used to find K nearest neighbors, and then the mean of these samples is considered as the filling value. Although this method can deal with the problem of missing values well, it cannot effectively deal with the classification task with noise and redundant features because it ignores the importance of features. In order to do this, the paper proposes a novel imputation method. In our method, missing values are firstly filled by using the K-Nearest Neighbor Imputation method based on Euclidean distance, then a feature weighting vector can be obtained by using a feature weighting algorithm trained on the initially filled data, and finally the K-Nearest Neighbor Imputation with weighted Euclidean distance is used to fill missing values. The experimental results on six datasets show that the proposed method can improve the classification accuracy of incomplete data.

1. Introduction
Classification is a common task in machine learning, which widely exists in many fields, such as computer vision, image processing[1], natural language processing and bioinformatics. The current mainstream classification algorithms include support vector machines[2], neural networks and K-Nearest Neighbors. In particular, these algorithms are mainly designed for complete datasets and are not suitable for incomplete datasets with missing values. However, incomplete datasets are common in practical applications. For example, the attribute values of more than 40% of the datasets in the UCI machine learning database[3] are missing to vary degrees. In the social questionnaire survey data, the questions that respondents are unwilling to answer can lead to missing data[4]. Therefore, it is very meaningful to improve the classification accuracy of incomplete data.

For the classification problem with missing values, there are three mainstream solutions at present: (1) Sample deletion method[5], that is, delete all samples with missing values, and then provide the remaining complete data for the classifier. This method is applicable to classification tasks with few missing samples. (2) Decision tree algorithms that can directly classify incomplete data, such as C4.5, the classification performance of the decision tree classifier is lower. (3) Imputation method. It firstly adopts the set rules to fill in the missing attribute values, and then uses mainstream classifiers based on the filled complete dataset for classification. Because this method can usually achieve high classification accuracy, it has become the mainstream method to deal with incomplete data classification[6].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI. Published under licence by IOP Publishing Ltd
At present, there are three common imputation methods: Zero Imputation (ZI), Mean Imputation (MI)[7] and K-Nearest Neighbor Imputation (KNNI)[8]. ZI is very simple, which sets all missing values to 0. For a given feature, MI allows the mean of all its observations to be the predicted value of the missing value, thus maintaining the invariance of the feature mean. Since all missing values of a given feature are the same, both MI and ZI ignore the volatility of the data. Different from the first two methods, KNNI is based on the sample to fill in missing values. For samples with features of missing values, KNNI firstly finds its K nearest neighbor samples based on Euclidean distance, and then uses the mean of the observable values in these samples to fill in the missing values. KNNI can better deal with incomplete data classification since it considers the similarity between samples when filling missing values.

Although KNNI can handle missing value filling better, it ignores the importance of features when filling missing attribute values. For this reason, this paper proposes an algorithm that combines the imputation method and feature selection[9] to solve the classification problem of incomplete data. The algorithm firstly uses KNNI to fill in the missing values of the incomplete dataset, and then trains the feature weighting algorithm based on the filled data to obtain the importance of the feature, and finally fills in the missing value of the original incomplete dataset based on the feature importance.

The rest of this article is arranged as follows: Section 2 introduces the algorithm proposed in this article in detail; Section 3 introduces the comparative experiment and result analysis of the proposed algorithm and the traditional method. Finally, Section 4 summarizes the content of this article.

2. Method

In this section, we propose an algorithm for missing value filling based on feature importance. It is assumed that the missing value filling classification task consists of a training set and a testing set. The proposed algorithm firstly fills the training set with initial missing values based on the K-Nearest Neighbor Imputation of Euclidean distance, and then uses the feature weighting algorithm to obtain the feature weighting vector based on the filled training set. For the obtained feature weighting vector, the weighted Euclidean distance can be obtained by considering it as the weight of the corresponding feature. Finally, the proposed algorithm uses the K-Nearest Neighbor Imputation algorithm to fill in the missing values of the original unfilled training set based on the weighted Euclidean distance. We also perform similar processing for missing values in the testing set.

In order to briefly describe the proposed algorithm, the paper defines some symbols. Let \( D = (X, y) \) represents an incomplete dataset, where \( X = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{n \times d} \) denotes the feature matrix containing missing values, \( y = [y_1, y_2, \ldots, y_n]^T \in \mathbb{R}^n \) indicates the corresponding label vector, \( n \) represents sample’s total number, and \( d \) is the feature dimension. The filled dataset is represented by \( \hat{D} = (\hat{X}, \hat{y}) \). In addition, this paper uses feature weighting vector \( w = [w_1, w_2, \ldots, w_p] \) to express the importance of features. In the following, we will introduce the K-Nearest Neighbor Imputation based on Euclidean distance, the feature weighting algorithm for obtaining a feature importance vector, and the K-Nearest Neighbor Imputation based on weighted Euclidean distance.

2.1. K-Nearest Neighbor Imputation algorithm based on Euclidean distance

In this section, the K-Nearest Neighbor Imputation method based on Euclidean distance is used to fill the missing values in incomplete dataset \( D \) and transform it into a complete dataset \( \hat{D} = (\hat{X}, \hat{y}) \). In order to fill a missing value \( x_i \) in a given sample \( x_j \), the K-Nearest Neighbor Imputation method uses the following steps:

1) Use all non-missing samples and all missing samples in \( D \) to construct sample sets \( C \) and \( M \) respectively.

2) In order to fill in any missing item \( x_i \) in \( M \), We need to use the observable value in \( x_j \) to calculate the distance from all samples in \( C \). According to them, find the \( K \) nearest samples of \( x_j \).
in $C$. The Euclidean distance between sample $x_i$ and sample $x_j$ in $C$ is calculated as follows:

$$D(x_i, x_j) = \sum_{m} (x_{im} - x_{jm})^2 .$$

(1)

3) Finally, the missing value is filled in with the mean value of the $t$th feature of the $K$ Nearest Neighbor samples:

$$\hat{x}_{it} = \frac{1}{K} \sum_{x_t \in \text{Nearest}(x_i)} x_{it} ,$$

(2)

where $\text{Nearest}(x_i)$ represents the set of $K$ nearest neighbor samples of $x_i$.

2.2. Feature weighting algorithm

In order to obtain the feature importance vector, the feature weighting algorithm can be trained based on the initially filled training dataset. In this study, we choose three feature weighting algorithms based on nearest neighbor model: ReliefF[10], NCFS[11] and WKNN[12] to perform experiments.

2.2.1. ReliefF

ReliefF is a filtered feature selection algorithm. Its specific idea is to randomly select samples $\hat{x}_i$ from the training set. According to $k$ neighboring samples of $\hat{x}_i$ the same class, the attribute values of $k$ nearest neighbor samples of different classes are used to assign scores to the features.

2.2.2. NCFS

NCFS solves the feature weighting vector by maximizing the classification accuracy of the leave-one-out method. In NCFS, the probability that the sample $\hat{x}_i$ refers to $\hat{x}_j$ is

$$P_{ij} = \frac{\exp(-\sum_{q=1}^{d} w_q | \hat{x}_{iq} - \hat{x}_{jq}|)}{\sum_{j' \neq j} \exp(-\sum_{q=1}^{d} w_q | \hat{x}_{iq} - \hat{x}_{iq'}|)} .$$

(3)

The probability of sample $\hat{x}_i$ being correctly predicted can be expressed as:

$$P_{i} = \sum_{j' \neq i \text{ and } y_j \neq y_i} P_{ij} .$$

(4)

Based on the above definition, we can obtain the approximate classification accuracy of the leave-one-out method. In addition, in order to obtain a sparse weighting vector solution, NCFS further introduces $L_1$ norm in the objective function. Therefore, the final optimization problem can be formalized as:

$$\hat{\xi}(w) = \sum_{i=1}^{n} P_i - 2 \sum_{q=1}^{d} |w_q| ,$$

$$s.t. \ w_q \geq 0 \ (q = 1, ..., d) .$$

(5)

The balance parameter $\lambda$ is a positive constant number. In particular, by replacing $w_q$ with $\tilde{w}_q^2$, NCFS removes inequality constraints during optimization.

2.2.3. WKNN

WKNN is also an embedded feature selection algorithm for regression tasks. When used for classification, it is similar to the algorithm principle of NCFS. The most obvious difference is that it uses Euclidean distance to calculate the distance between samples.
2.3. Weighted K-Nearest Neighbor Imputation algorithm

In order to fill in missing values based on the feature importance vector, the weighted Nearest Neighbor Imputation algorithm needs to use the unfilled training dataset $D$ and the filled training set $\hat{D}$ at the same time. For a certain missing value $x_i$ in a given sample $x_i$, the weighted K-Nearest Neighbor Imputation algorithm uses the following steps to fill in missing values:

1) Let the corresponding sample of $x_i$ in $\hat{D}$ be $\hat{x}_i$, the set of all samples in the set $C$ and $M$ corresponding to the samples are $\hat{C}, \hat{M}$ in $\hat{D}$ respectively.

2) Calculate the weighted Euclidean distance between $\hat{x}_i$ and all samples in $\hat{C}$. The formula for calculating the weighted distance between samples is:

$$D(\hat{x}_i, \hat{x}_j) = \sum_{q=1}^{d} w_q \times (\hat{x}_{iq} - \hat{x}_{jq})^2,$$

Get the index set $I$ of the $K$ Nearest Neighbor samples of $\hat{x}_i$ through Nearest Neighbor search.

3) Based on the index set $I$ and the original training data, the final filling value is:

$$\frac{1}{K} \sum_{i \in I} x_{i}.$$

3. Experimental result

In order to verify the effectiveness of the proposed algorithm, we compared the proposed algorithm with other algorithms on six datasets. Next, we firstly introduce the dataset and experimental settings, and then give the experimental results and related analysis.

3.1. Datasets

The six datasets used in the experiment are glass, hayes-roth, iris, lymphography, monk-2 and tae. Table 1 shows their specific information. In order to calculate the classification accuracy on a given data set, we randomly select 70% of the samples as the training set and 30% of the samples as the testing set. Repeat the above process 10 times, the experimental result is the average classification accuracy of 10 times. In order to avoid the influence of the feature value range, we normalized the features.

| Tab. 1 Details of the datasets |
|--------------------------------|
| Dataset          | Instances | Features | Classes |
| glass            | 214       | 9        | 7       |
| hayes-roth       | 160       | 40       | 3       |
| iris             | 150       | 4        | 3       |
| lymphography     | 148       | 18       | 4       |
| monk-2           | 432       | 6        | 2       |
| tae              | 151       | 4        | 3       |

3.2. Experimental settings

The parameter $K$ of the KNNI and KNN used in this paper is 3. The relevant parameter settings of the feature selection method NCFS are the same as those in [11]. The codes of WKNN and reliefF can be downloaded from github2,3 respectively. The parameter $k_i$ in ReliefF in section 2.2.1 takes 5. After the feature weighting vector is obtained by the feature selection method, we normalize it. In addition, we used MCAR [13] mechanism to generate corresponding datasets with missing rates of 5%, 10%, and 15% for each dataset. Besides, in order to further verify the effectiveness of the proposed
algorithm, we add noise to the six datasets. The number of noise features added to each dataset is 10, and the noise data are extracted from the normal distribution with mean value of 0 and standard deviation of 1. We only use MCAR mechanism to generate datasets with a missing rate of 5% on noisy datasets.

3.3. Experimental results and analysis
This section shows the comparison results of KNNI, KNNI+ReliefF, KNNI+NCFS and KNNI+WKNN on 6 datasets. Figure 1 shows the average classification accuracy of each algorithm on a dataset without adding noise. In Figure 1, the abscissa represents the missing rate of data, and the ordinate represents the average classification accuracy of the algorithm. It can be seen from Fig. 1 that the classification accuracy of KNNI+NCFS and KNNI+WKNN is higher than that of KNNI algorithm on the whole, while the comparison results of KNNI+ReliefF and KNNI are unstable. Therefore, we can say that considering the importance of features in KNNI can indeed improve the classification accuracy of incomplete data, but we need to choose a feature weighting algorithm with good effect. In Figure 1, we can also see that the classification accuracy of KNNI+WKNN is higher than that of KNNI+NCFS. Therefore, in the experiment of this paper, the effect of selecting WKNN algorithm is the best. In order to further verify that the proposed algorithm can improve the classification accuracy of incomplete data, we also compare the classification capabilities of KNNI, KNNI+ReliefF, KNNI+NCFS and KNNI+WKNN on the dataset with noise. The specific results are shown in Table 2. Table 2 shows the average classification accuracy and standard deviation of each algorithm. Bold characters in Table 2 indicate the optimal value of each row. It can be seen from Table 2 that among the four algorithms, KNNI+WKNN has the highest classification accuracy, followed by KNNI+NCFS, but the classification accuracy of KNNI+ReliefF is indeed not as good as KNNI. In summary, we finally choose KNNI+WKNN as the algorithm proposed in this article.

![Fig. 1 Comparison results of the four algorithms on the dataset without adding noise](image)

**Tab. 2 The average classification accuracy of the four algorithms on the noise-added dataset**

| Datasets       | KNNI  | KNNI+ReliefF | KNNI+NCFS | KNNI+WKNN |
|----------------|-------|--------------|-----------|-----------|
| Glass          | 0.6308±0.07 | 0.6815±0.07 | 0.8538±0.04 | 0.8938±0.04 |
| hayes-roth     | 0.48±0.08  | 0.7075±0.05 | 0.7425±0.06 | 0.7525±0.06 |
| iris           | 0.8889±0.04 | 0.9378±0.02 | 0.9467±0.01 | 0.9556±0.02 |
| lymphography   | 0.7733±0.04 | 0.7578±0.03 | 0.7844±0.03 | 0.7867±0.05 |
| monk-2         | 0.7915±0.05 | 0.94±0.02  | 0.9646±0.01 | 0.9669±0.01 |
| tae            | 0.3783±0.06 | 0.3543±0.03 | 0.4196±0.07 | 0.3783±0.06 |
4. Conclusion

This paper combines the importance of features with the $K$-Nearest Neighbor Imputation method, and proposes a filling method which can effectively improve the classification accuracy of incomplete data. The proposed method firstly uses $K$-Nearest Neighbor Imputation method to fill the original incomplete dataset, thereby transforming it into a complete dataset. Second, the feature weighting algorithm is used to obtain a feature weighting vector. Finally, the $K$-Nearest Neighbor Imputation method with the weighted Euclidean distance is adopted to fill the original incomplete dataset again. This paper compares KNNI, KNNI+ReliefF, KNNI+NCFS and KNNI+WKNK on the six datasets. The experimental results show that KNNI+NCFS and KNNI+WKNK are better than KNNI. Therefore, the proposed algorithm can improve the classification accuracy of incomplete data, but it is necessary to select a feature weighting algorithm with good effect to improve the classification accuracy of incomplete data. In the future, we will consider how to fill regression datasets with missing values.

Acknowledgement

This work was supported by the National Science Foundation of China (no.61806074) and the Science and Technology Foundation of Henan Province of China (No.212102210387).

References

[1] Zhang J, Clayton M K and Townsend P A 2015 Missing Data and Regression Models for Spatial Images IEEE T GEOSCI REMOTE 53 pp 82-1574
[2] Suthaharan S 2016 Machine learning models and algorithms for big data classification: thinking with examples for effective learning. Computing reviews. 57 pp 4-283
[3] Bache K and Lichman M 2013 UCI Machine Learning Repository: http://archive.ics.uci.edu.
[4] Pedersen A B, Mikkelsen E M and Cronin-Fenton D, et al. 2017 Missing data and multiple imputation in clinical epidemiological research Clinical Epidemiology 9 pp 66-157
[5] Hooke M, Mrozinski J and DiNicola M 2021 IEEE Aerospace Conference on Big Sky MT USA IEEE pp 1-14
[6] Farhangfar A, Kurgan Land Dy J 2008 Impact of imputation of missing values on classification error for discrete data. Pattern Recognition 41 pp 705-3692
[7] Donders A R T, Heijden G J M G V D and Stijnen T, et al. 2006 Review: A gentle introduction to imputation of missing values Journal of Clinical Epidemiology 59 pp 91-1087
[8] Batista G E and Monard M C. 2003 A Study of K-Nearest Neighbour as an Imputation Method 87 pp 251–260
[9] Gnana D A A, Balamurugan S A A and Leavline E J. 2016 Literature Review on Feature Selection Methods for High-Dimensional Data. International Journal of Computer Applications 136 pp 9-17
[10] Robnik-Sikonja M and Kononenko I 2003 Theoretical and Empirical Analysis of ReliefF and RReliefF. Machine Learning 53 pp 23-69
[11] Yang W, Wang Q Q and Zuo W M. 2012 Neighborhood Component Feature Selection for High-Dimensional Data Journal of Computers 7 pp 8-161
[12] Bugata P and Droter P. 2019 Weighted nearest neighbors feature selection Knowledge-Based Systems 163 pp 61-749
[13] Little R J and Rubin D B 1986 Statistical Analysis With Missing Data Wiley 26 pp 374-375