Data Cleaning About Student Information Based on Massive Open Online Course System

Shengjun Yin, Yaling Yi, and Hongzhi Wang(✉)

Harbin Institute of Technology, Harbin, China
441607987@qq.com, limeyiyaling@163.com, wangzh@hit.edu.cn

Abstract. Recently, Massive Open Online Courses (MOOCs) is a major way of online learning for millions of people around the world, which generates a large amount of data in the meantime. However, due to errors produced from collecting, system, and so on, these data have various inconsistencies and missing values. In order to support accurate analysis, this paper studies the data cleaning technology for online open curriculum system, including missing value-time filling for time series, and rule-based input error correction. The data cleaning algorithm designed in this paper is divided into six parts: pre-processing, missing data processing, format and content error processing, logical error processing, irrelevant data processing and correlation analysis. This paper designs and implements missing-value-filling algorithm based on time series in the missing data processing part. According to the large number of descriptive variables existing in the format and content error processing module, it proposed one-based and separability-based criteria Hot+J3+PCA. The online course data cleaning algorithm was analyzed in detail on algorithm design, implementation and testing. After a lot of rigorous testing, the function of each module performs normally, and the cleaning performance of the algorithm is of expectation.

Keywords: MOOC · Data cleaning · Time series · Intermittent missing · Dimension reduction

1 Introduction

Data cleaning [1–4] or data scrubbing aims to detect the errors or mismatches of data and then remove them out to improve the quality of data [5,6]. Traditional data cleaning methods are generally rule-based methods [7]. The data cleaning method based on deep learning has been a research hot-spot in recent years.

The data cleaning algorithm in this paper is aimed at the problems of inaccurate data analysis results caused by various errors (such as inconsistencies and missing values) in the input data due to input errors, system errors, and other reasons. In order to optimize the data to support accurate analysis, this project

J. Zeng et al. (Eds.): ICPCSEE 2020, CCIS 1257, pp. 33–43, 2020.
https://doi.org/10.1007/978-981-15-7981-3_3

intends to study data cleaning techniques for online open course system students, including missing value filling for time series, and rule-based entry error correction [8]. The overall idea of the algorithm is similar to other data cleaning algorithms, but due to the particularity of online course data, this algorithm focuses on the problem of missing data filling on time series [9], encoding of category features, and feature extraction [10]. Time series-oriented missing value filling needs to establish a time series prediction model [11] based on the existing data, and predict and populate the results of missing values [9]. Category characteristics mainly refer to those discrete variables with special meanings. For example, the academic information of students includes primary school, junior high school, high school, and so on. Current machine learning algorithms struggle to handle these features. In the data cleaning phase, one-hot coding is needed to digitize these discrete features.

2 Technology

2.1 Data Cleaning Process

According to the modular design principle, we divide the data cleaning algorithm into 6 parts: preprocessing, missing data processing, format and content error processing, logical error processing, irrelevant data processing and correlation analysis as shown in Fig. 1.

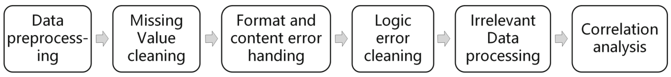

Fig. 1. The overall process of the data cleaning algorithm in this paper

2.2 Preprocessing

The data of student information in the online open course system is huge, and it will take a lot of time to read the data using traditional methods. In order to solve this problem, we used a data structure "DataFrame" that supports the parallel reading of large amounts of data, and designed a data reading algorithm for this data structure. The DataFrame includes three parts: the header, the content, and the index. The internal implementation of the DateFrame is the nesting of multiple lists. We design two parallel processing methods for DataFrame:

1. Overall parallel processing method: For the reading of two-dimensional matrix data of DataFrame, we have designed an overall parallel processing algorithm that can perform input parallelization by performing overall feature conversion on the matrix. The implementation is similar to the FeatureUnion class in the pipeline package, thich improve the overall efficiency by calling multiple transform methods to read data from multiple parts at the same time.

2. Improved parallel processing method: we design a parallel processing algorithm for a large number of simple repetitive tasks. We have improved the FeatureUnion class by the following process. First, we add the attribute idx_list to record each column of the feature matrix that needs to be processed in parallel; then Modify the fit and fit_transform method so that it can read a part of the feature matrix; finally, modify the transform method so that it can output part of the feature matrix.

The above process can be summarized as follows,

– Step 2–1 = (‘OneHoutEncoder’, OneHotEncoder(sparse = False))
– Step 2–2 = (‘Tolog’, FunctionTransfer(log1p))
– Step 2–3 = (‘ToBinary’, Binarizer())
– Step 2 = (‘FeatureUnionExt’, FeatureUnionExt(tansformer_list = [Step2–1, Step2–2, Step2–3], idx_list=[[0], [1, 2, 3], [4]]))

2.3 Missing Data Processing

There are several ways to deal with missing value data:

– Feature discarding method: For the features with a higher degree of missing (the degree of missing is greater than 10
– Null value feature method: In some data sets, a null value can be used as a feature, but in order to facilitate processing, the null value needs to be specified as a special value.
– Mean filling method: If there are missing values in the training set but no missing values in the test set, the conditional mean or conditional median is usually taken for the missing values, because the training set data does not need to be particularly high quality.
– Upper and lower value or difference filling method: These two methods are filled using the feature information in the data table. The interpolation method needs to calculate the missing interval.
– Algorithm-Filled Filling: This is the most effective missing-value processing method currently in use. It learns from existing data to build an analysis model to predict missing values, including time-series prediction methods and logical prediction methods.

In this paper, we mainly use the time series prediction method in algorithm fitting and filling method to process missing values. The advantage is that it can analyze the time series, predict and fill the missing data at different time nodes, and the specific implementation of the algorithm.

2.4 Format and Content Error Handling

In this paper, due to format errors and content errors in the data, regular expressions are mainly used to reduce the format errors caused by user input, but this method is only for error prevention. If wrong data has been generated already, you need to use data binning, clustering, regression and other methods for processing. In this algorithm, data binning is mainly used to discretize continuous data. For example, the user's age data is distributed for small dispersion data in the range from 0 to 120, we use data binning to convert the data into large dispersion data for data analysis. For different data sets, we design supervised and unsupervised discretization algorithms. Supervised discretization algorithm: As for data with fixed category information, supervised discretization algorithm is required to discretize the data. Optimal criteria need to be formulated based on the category information. The determination of the optimal criteria needs to be judged based on the characteristics of the data set size Including but not limited to chi-square test, entropy discrimination, information gain, Gini coefficient and other methods. Unsupervised discretization algorithm: Unsupervised discretization algorithms need to be used to process data with unfixed class information. The main classification methods include dividing the value range of the feature's numerical attributes into K-divisions. When the data is approximately subject to uniform distribution, this kind of discretization algorithm has the best effect; another classification method is to classify the characteristic data by isoquantity by calculating the quantiles to ensure that the number of data samples in each interval is almost the same. After data binning, you need to perform data smoothing and other operations. This part of the algorithm involves processing category features.

2.5 Logical Error Handling

This part aims to judge the legality and validity of the data through simple logical reasoning. It mainly includes the three steps of removing duplicate values, removing irrational values, and correcting contradictions. Due to the intersection of the last two steps and formats with the content error processing algorithm, the data deduplication algorithm will be designed in this section. The deduplication algorithm is mainly divided into two parts: determining duplicate data and removing duplicate data. Determining duplicate data: The determination of duplicate data requires similarity analysis of the data. The algorithm processing flow is shown in Fig. 2. First, we perform pairwise similarity analysis on all the data in the data set; Calculate the threshold; finally use the threshold to eliminate duplicate data.

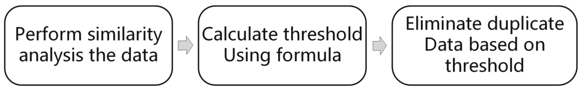
Perform similarity analysis the data
Calculate threshold Using formula
Eliminate duplicate Data based on threshold

Fig. 2. Process of eliminating duplicate data

Using the operation pool to remove duplicate data: Due to the huge amount of data in the system, deleting one by one when deleting duplicate data will cause the DataFrame index to be re-write after each delete operation, which will cause serious performance problems. The solution to this problem is to use the operation pool to save the operation data to prevent performance loss caused by repeatedly calling and releasing the upper-level API of the DataFrame each time a delete operation is performed.

2.6 Irrelevant Data Processing

Irrelevant data refers to irrelevant features found during the analysis, these features or data will not affect the final results of the analysis, but deleting these data will adversely affect the readability and integrity of the analysis results. So you need to consider how to deal with this irrelevant data. As for redundant fields, direct deletion can be used for irrelevant data processing; for multi-index data features, it is necessary to determine whether to delete this feature based on the needs of the results.

2.7 Correlation Analysis

Association analysis uses PCA or LDA to integrate features to eliminate useless information. In this algorithm, PCA is mainly used to extract features. PCA (Principal Component Analysis), principal component analysis, is a data analysis technique. The main idea is to project high-dimensional data to a lower-dimensional space, extract the main factors of multiple things, and reveal its essential characteristics. Advantages of association analysis:

1. It can eliminate the relevant impact between the evaluation indicators, because the principal component analysis method transforms the original data indicator variables to form independent principal components. The practice proves that the higher the degree of correlation between indicators, the better the effect of principal component analysis.
2. It can reduce the workload of index selection. For other evaluation methods, it takes a lot of effort to select the index because it is difficult to eliminate the relevant impact between the evaluation indices. But the principal component analysis method can eliminate this related impact. As a result, index selection is relatively easy.
3. In the principal component analysis, each principal component is arranged in order according to the magnitude of the variance. When analyzing a problem, you can discard a part of the principal components and only take a few principal components with larger variances to represent the original variables, thereby reducing the calculation work. When using the principal component analysis method for comprehensive evaluation, the principle of selection is that the cumulative contribution rate is $geq85$.

3 Design and Implementation of Time Series Filling Algorithm

3.1 Data Smoothing

Time series data generally have periodic data fluctuations, which are the key characteristics of the entire time series. But these characteristics are too obvious. As a result, they will obscure other key characteristics to a certain extent, so it is necessary to retain the characteristics of the periodic data. Data difference algorithm: The original time series data is segmented by generating adaptive time series, and part of the data is intercepted every fixed length to divide the data into subsets. In the case where the intercept length is an integer multiple of the period T, each subset is data with the period characteristics removed, and features that are obscured by the period characteristics can be extracted. The implementation of the data difference algorithm is shown in Fig. 3.

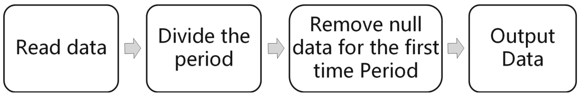

Fig. 3. Main flow of data difference algorithm

And the comparison of the data distribution before and after the data difference processing is shown in Fig. 4.

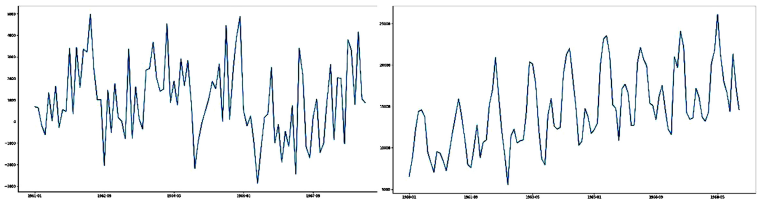

Fig. 4. Comparison before and after processing of the difference algorithm

3.2 Autocorrelation Graph Generation

By analyzing the correlation coefficients of the output variables of the time series to determine the characteristics of the time series, this method is called an autocorrelation method, also known as an autocorrelation map. The autocorrelation graph will show the correlation of each lagging feature and label these features by statistical feature analysis. The X-axis in the figure represents the value of the lagging feature, and the interval distribution of the Y-axis is (−1, 1), showing the positive and negative correlations of these lag values. The points in the blue area are statistically significant. If the lag value of a feature is 0 but the correlation n is 1, it means that the observed value of the feature and itself 100% positive correlation. The implementation of the autocorrelation feature is shown in Fig. 5, which calls the DateFrame data structure we mentioned earlier.

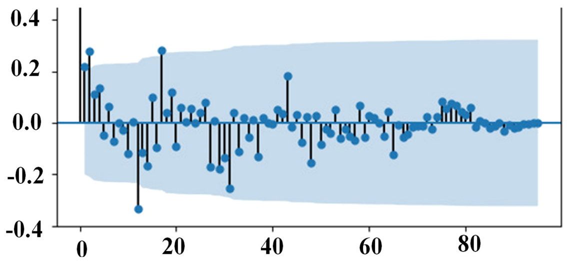

Fig. 5. Sample autocorrelation diagram

3.3 Optimization for Supervised Learning

If we take the lag variable as the input variable and the current observation variable as the output variable, we can transform the time series prediction problem into a supervised learning problem. The specific conversion steps are shown in Fig. 6. First, we use the shift function in the Pandas library to generate a new queue after the observations are transformed. Then we traverse the DataFrame to constrain the data specifications. Finally, we generate a prediction label by supervising the learning classifier.

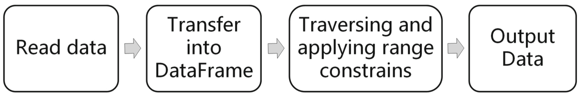

Fig. 6. Supervised learning conversion process

3.4 Feature Selection for Lag Variables

Usually, a large number of features are weighted using decision tree algorithms (bagged trees, random forests, etc.). By calculating the weights of these features, the relative effectiveness of the features in the prediction model can be predicted. By using lag features, you can analyze other types of features based on time stamps, rolling statistics, and so on. Feature selection according to lag features needs to be considered from the following two aspects. Generation of feature weights for lag variables: In this algorithm, we use a random forest model to generate feature weights, as shown in Fig. 7, where RandomForestRegressor is a random forest model. To prevent accidental errors from affecting the experimental results, we use random seed initialization to randomly generate data.

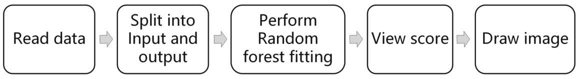

Fig. 7. The process of fitting using random forest

The effect of the weight generation algorithm is shown in Fig. 8. The figure shows that the relative importance of t − 12, t − 2, and t − 4 is high. In fact, similar results can be obtained by other design tree algorithms.

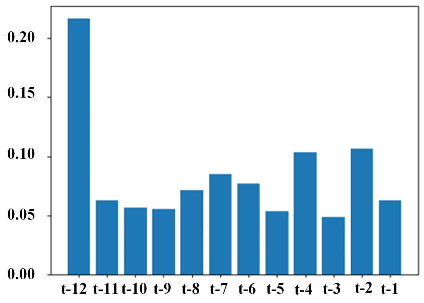

Fig. 8. Running effect of weight generation algorithm

Feature selection for lag variables: By using a recursive feature selection method (RFE) to create a predictive model that assigns different weights to each feature, and deletes those features with the smallest weight. Each time a feature is deleted, RFE is used to generate Model and repeat the above process until the number of features reaches the expected requirements. The implementation of the RFE algorithm is similar to Fig. 8. Running the algorithm found that the output results were t − 12, t − 2, and t − 4. This result is the same as the highest weight result generated during the weight generation phase. The bar graph of the feature weight ranking is shown in Fig. 9.

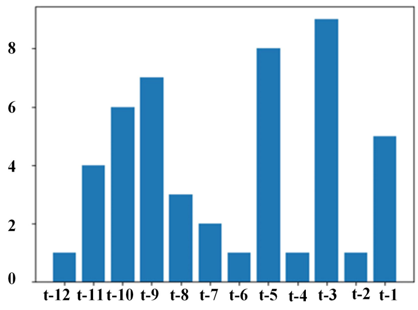

Fig. 9. Bar chart of feature weight ordering (smaller is better)

3.5 Selection of Core Algorithms

In order to solve the problems caused by vanishing gradients and local optimization, people find that local optimization can be solved by pre-training after years of exploration. At the same time, using maxout or RelU activation function instead of sigmoid can alleviate the problem of vanishing gradients. After this replacement, the structure is the basic structure of DNN. However, DNN is still a fully connected multi-layer perceptron in nature. As the number of layers increases, the exponential expansion of the number of parameters increase as well which mainly occurs in image recognition. CNN [12,13] uses convolution kernels and convolution layers to achieve the reuse of regional features, which perfectly solves many problems caused by the increase in the amount of calculations (such as the problem of overfitting). CNN uses the common features of the local structure to propagate forward. The execution efficiency of the algorithm depends on the size of the convolution kernel window. The time series is a very important feature in the user information data analyzed in this paper, so we select RNN-type algorithms as the core algorithm of the analysis model. However, to avoid the phenomenon of gradient disappearance, LSTM [14,15] needs to be used as the core algorithm for the amount of data is large.

4 Design and Implementation of Category Feature Processing Algorithm

There are so many data composed of category features in the data, such as: age, time, UID and so on, that the data corresponding to these features are often discrete and have large distribution intervals. In most cases, these data have no practical significance, so directly applying these features to model calculations affects the overall accuracy of the prediction model. We choose to use data binning and one-hot encoding to deal with these category features. Since one-hot encoding will cause the number of features to increase, we need to use feature selection and extraction algorithms to reduce the feature dimensions.

5 Evaluation and Conclusion

5.1 Parameter Tuning

Improved grid search algorithm: Optimize the repeated calculation steps of grid search. In the improved algorithm, we have modified the distance traveled by each search to reduce the number of searches, thereby reducing the number of cross-validations to improve efficiency. We use some data sets for grid search comparison tests, and use GBDT, Xgboost, RF, and SVM for testing. The model test results are shown in Fig. 10.

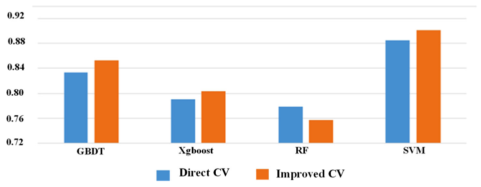

Fig. 10. Grid search comparison test results

5.2 Algorithm Test Results

The time series comparison of user behavior before and after the algorithm execution is shown in Fig. 11, and the data is successfully stabilized.

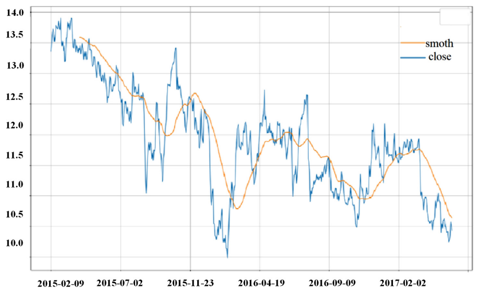

Fig. 11. Comparison of user behavior time series before and after algorithm execution

References

1. Li, W., Li, L., Li, Z., Cui, M.: Statistical relational learning based automatic data cleaning. Front. Comput. Sci. **13**(1), 215–217 (2019). https://doi.org/10.1007/s11704-018-7066-4
2. Zheng, Z., Quach, T.M., Jin, Z., Chiang, F., Milani, M.: Currentclean: interactive change exploration and cleaning of stale data. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, pp. 2917–2920. ACM (2019)
3. Wang, T., Ke, H., Zheng, X., Wang, K., Sangaiah, A.K., Liu, A.: Big data cleaning based on mobile edge computing in industrial sensor-cloud. IEEE Trans. Industr. Inform. **16**, 1321–1329 (2019)
4. Lin, X., Peng, Y., Xu, J., Choi, B.: Human-powered data cleaning for probabilistic reachability queries on uncertain graphs. In: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, 16–19 April 2018, pp. 1755–1756. IEEE Computer Society (2018)
5. Ilyas, I.F., Chu, X.: Data Cleaning. ACM, New York (2019)
6. Cichy, C., Rass, S.: An overview of data quality frameworks. IEEE Access **7**, 24634–24648 (2019)
7. Ganibardi, A., Ali, C.A.: Web usage data cleaning. In: Ordonez, C., Bellatreche, L. (eds.) DaWaK 2018. LNCS, vol. 11031, pp. 193–203. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98539-8_15
8. Polleres, A., Sobernig, S., Umbrich, J.: Data cleaning and preparation (basics) (2019)
9. Yi, X., Zheng, Y., Zhang, J., Li, T.: ST-MVL: filling missing values in geo-sensory time series data. In: Kambhampati, S. (ed.) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 2704–2710. IJCAI/AAAI Press (2016)
10. Joo, Y., Jeong, J.: Under sampling adaboosting shapelet transformation for time series feature extraction. In: Misra, S., et al. (eds.) ICCSA 2019. LNCS, vol. 11624, pp. 69–80. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24311-1_5
11. Mahalakshmi, G., Sridevi, S., Rajaram, S.: A survey on forecasting of time series data. In: International Conference on Computing Technologies and Intelligent Data Engineering (2016)
12. Barrow, E., Eastwood, M., Jayne, C.: Deep CNNs with rotational filters for rotation invariant character recognition. In: 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, 8–13 July 2018, pp. 1–8. IEEE (2018)
13. Cui, Y., Zhang, B., Yang, W., Yi, X., Tang, Y.: Deep CNN-based visual target tracking system relying on monocular image sensing. In: 2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, 8–13 July 2018, pp. 1–8. IEEE (2018)
14. Wang, D., Fan, J., Fu, H., Zhang, B.: Research on optimization of big data construction engineering quality management based on RNN-LSTM. Complexity **2018**, 9691868:1–9691868:16 (2018)
15. Ma, Y., Peng, H., Cambria, E.: Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI 2018) (2018)