Financial Innovation

RESEARCH Open Access

A high-dimensionality-trait-driven learning paradigm for high dimensional credit classification

Lean Yu[1,2*], Lihang Yu[1] and Kaitao Yu[3]

*Correspondence:
yulean@amss.ac.cn
[1] School of Economics and Management, Beijing University of Chemical Technology, 15 Beisanhuan East Road, Chaoyang District, Beijing 100029, China
Full list of author information is available at the end of the article

Abstract

To solve the high-dimensionality issue and improve its accuracy in credit risk assessment, a high-dimensionality-trait-driven learning paradigm is proposed for feature extraction and classifier selection. The proposed paradigm consists of three main stages: categorization of high dimensional data, high-dimensionality-trait-driven feature extraction, and high-dimensionality-trait-driven classifier selection. In the first stage, according to the definition of high-dimensionality and the relationship between sample size and feature dimensions, the high-dimensionality traits of credit dataset are further categorized into two types: 100 < feature dimensions < sample size, and feature dimensions ≥ sample size. In the second stage, some typical feature extraction methods are tested regarding the two categories of high dimensionality. In the final stage, four types of classifiers are performed to evaluate credit risk considering different high-dimensionality traits. For the purpose of illustration and verification, credit classification experiments are performed on two publicly available credit risk datasets, and the results show that the proposed high-dimensionality-trait-driven learning paradigm for feature extraction and classifier selection is effective in handling high-dimensional credit classification issues and improving credit classification accuracy relative to the benchmark models listed in this study.

Keywords: High dimensionality, Trait-driven learning paradigm, Feature extraction, Classifier selection, Credit risk classification

Introduction

Credit risk classification has always been a hot issue in scientific research, especially in the context of globalization (Ma and Wang 2020), where it has become increasingly important in the field of financial risk management. Credit risk assessment, the early stage of credit risk management, has attracted much attention from academics and practitioners (Niu et al. 2020). How to distinguish bad customers from good, minimize credit risk, and prevent credit fraud in advance are the most important issues for commercial banks and other related credit granting institutions (Yu et al. 2008; Wang et al. 2020).

With the development of information technology, the Internet and numerous intelligent devices produce more and more data, and accordingly, these data reflect an increasing amount of traits (Yu and Liu 2003), such as data noise (Yu et al. 2020a), data missing (Yu et al. 2020b), data imbalance (Yu et al. 2018), and small sample data (Yu and Zhang 2021). With the increase of feature dimensions, more redundant features appear. The problem of sparse data and complicated calculation caused by too many features is known as the curse of dimensionality. This kind of high-dimensionality problem becomes particularly important in credit risk classification. It increases not only the cost of credit classification but also the calculation time exponentially; therefore, the accuracy of classification will decline (Kou et al. 2020). Many traditional data mining algorithms will fail or become less effective when directly applied to the high-dimensional data. Therefore, credit risk classification of high-dimensional features has become a challenging task. Although many research achievements have been made in the past decades, there are still many problems and challenges to be solved in this field.

For the high dimensional credit dataset, reducing the data dimension is an essential operation in credit classification, and feature extraction is one of the main methods to do that. Latent semantic analysis (LSA) (Deerwester et al. 2010) is one of the earliest feature extraction methods, and its main idea is to transform the feature values with singular value decomposition (SVD), change the spatial relationship of the original features, and combine them to obtain new variables through the analysis of the relationship between features. Feature extraction mainly includes two methods: linear and nonlinear feature extraction. One of the prominent linear dimensionality reduction methods is principal component analysis (PCA) (Kambhatla and Leen 1997), followed by locality preserving projections (LPP) (He 2003), neighborhood preserving embedding (NPE) (He et al. 2005), and linear discriminant analysis (LDA) (Fisher 1936). However, LDA can only reduce the dimension to (n-1) at most after feature extraction for n-element classification problem (Fisher 1936), so it cannot be used as an effective feature extraction strategy for a binary credit classification problem in this paper. Furthermore, LPP focuses on the local structure information of data without considering the global structure sufficiently (He 2003). Therefore, PCA is selected as the linear feature extraction method in this study.

Nonlinear feature extraction methods mainly include the nonlinear feature extraction based on kernel methods and the manifold learning, such as kernel PCA (KPCA) (Zhang 2009), locally linear embedding (LLE) (Roweis and Saul 2000), and isometric mapping (ISOMAP) (Tenenbaum et al. 2000). It is important to note that nonlinear feature extraction has a certain advantage in reducing original data dimensions, but in practical applications, there are many problems regarding its performance. For example, nonlinear feature extraction based on kernel methods needs to perform nonlinear transformation for each sample, which will result in a huge calculation burden and a dimension disaster problem. Meanwhile, the nonlinear feature extraction method based on manifold learning requires dense sampling (Roweis and Saul 2000), which is a major hurdle for high-dimensional circumstances. Therefore, the performance of the nonlinear dimension reduction method in practice is often worse than expected.

After reducing the feature dimension through feature extraction, the next step is to use classification algorithms to classify the data samples. The most popular credit scoring methods include expert systems, statistical and econometric models, artificial intelligence (AI) techniques, and their hybrid forms (Yu et al. 2015). At present, the most commonly used statistical and econometric methods include linear discriminant analysis (LDA) (Blei et al. 2003), logistic regression (LogR) (Grablowsky and Talley 1981; Feder and Just 1977), wavelet method (Mabrouk 2020), mathematical programming model (MPM) (Mangasarian 1965), and *k*-nearest neighbor (KNN) algorithm (Henley and Hand 1996). With the rapid development of AI and machine learning (Kou et al. 2019; Pabuçcu et al. 2020), AI-based algorithms have proved to be more effective in credit risk classification compared with traditional methods; hence, more and more scholars apply those. The main AI algorithms include artificial neural networks (ANN) (Odom and Sharda 1990; Tam and Kiang 1992; Donskoy 2019), support vector machines (SVM) (Cortes and Vapnik 1995; Yu et al. 2020c), decision trees (DT) (Waheed et al. 2006; Rutkowski et al. 2014), and extreme learning machines (ELM) (Xin et al. 2014; Nayak and Misra 2020). These single classifiers can be divided into two types, linear and nonlinear. In addition to these single models, hybrid and ensemble classifiers, such as bagging and neural network ensemble classifiers, are other substantial types of credit risk classification and prediction methods (Yu et al. 2008; Yu et al. 2010; Wang and Ma 2010; Song and Wang 2019).

In summary, there are numerous dimensionality reduction methods and classification models applied to the high-dimensional credit risk classification, but the experimental results are also affected by the data itself, and the same model may perform differently with different data traits. However, little attention has been paid to the relationship between the data traits and model selection. Nelson and Plosser (1982) found that the traditional econometric models would show pseudo-regression when the data presented non-stationary traits. Tang et al. (2013, 2014) proposed a novel data-characteristic-driven modeling methodology for nuclear energy consumption forecasting and proved that the performance of this methodology is clearly superior. The data-trait-driven modeling methodology confirms that an effective model must match the data trait of the research sample. For this purpose, this paper tries to propose a high-dimensionality-trait-driven learning paradigm for feature extraction and classifier selection in credit classification with high dimensionality.

The main objective is to provide and select different feature extraction strategy and classifiers regarding the two categories of high dimensionality and establish the connection between high-dimensional data traits and feature extraction and classifier selection in credit risk classification. The rest of this paper is organized as follows: "Methodology formulation" section describes the proposed learning paradigm in detail. To verify and compare the validity of the proposed model, two real-world credit datasets are used, and the experimental design is presented in "Data descriptions and experimental design" section. "Results and discussion" section reports the results and further discussions. Finally, "Summary and discussion" section concludes the paper.

Methodology formulation

In this section, a high-dimensionality-trait-driven learning paradigm for feature extraction and classifier selection is proposed for the high-dimensional credit risk classification problem. In particular, the main purpose is to select the most suitable feature extraction method and classifier considering the two categories of high dimensionality, and to establish the connection between the trait of high dimensionality and feature extraction and classifier selection in credit risk classification. The general framework of the proposed high-dimensionality-trait-driven learning paradigm is shown in Fig. 1.

As can be seen from Fig. 1, the proposed paradigm includes three main stages: categorization of high dimensionality, high-dimensionality-trait-driven feature extraction, and high-dimensionality-trait-driven classifier selection. When 100 < feature dimensions < sample size (Chandrashekar and Sahin 2014), PCA is selected as the feature extraction strategy, and single linear classifier is selected as the classification model. When feature dimensions ≥ sample size (Mwangi et al. 2014; Hua et al. 2009), non-feature extraction is selected as the feature extraction strategy, and linear ensemble classifier is selected as the classification model. The detailed descriptions and related methodology of the three stages are given in "Categorization of high dimensionality"–"High-dimensionality-trait-driven classifier selection" sections below.

Categorization of high dimensionality

In the existing literature, there are two different definitions of high dimensionality. On the one hand, high dimensionality means that the number of attribute features is larger than sample size (Mwangi et al. 2014; Hua et al. 2009). For example, Bai and Li (2012) claim that the number of attribute features equal to, or greater than sample size, can be called a high dimensionality in the sample. On the other hand, some studies have found that no matter how many samples there are, the number of attribute features will significantly affect the performance of the classifier. For example, Chandrashekar and Sahin (2014) reported that hundreds of variables could lead to high dimensionality and even to the curse of dimensionality problem.

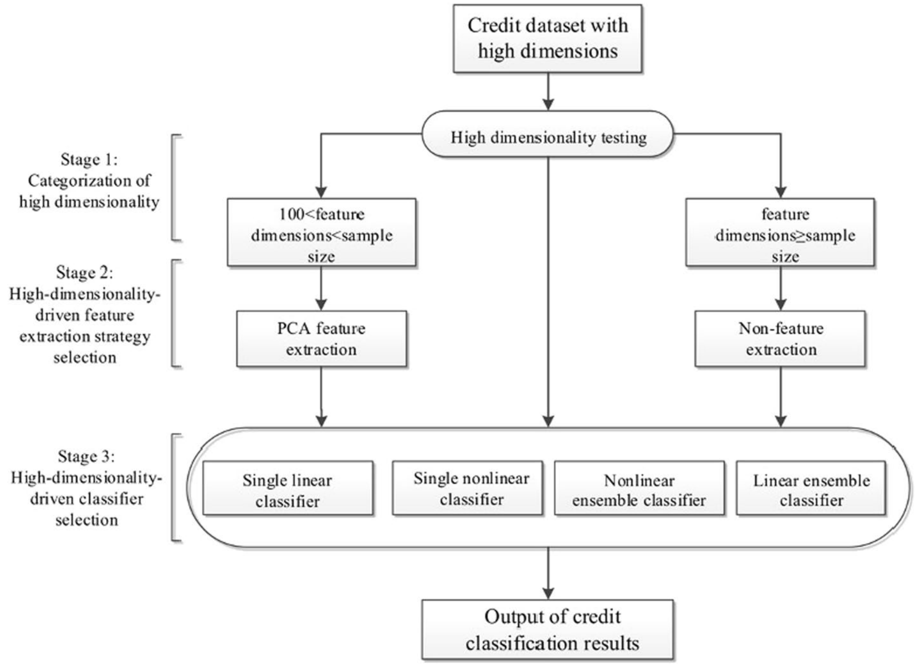
Credit dataset with high dimensions
High dimensionality testing
Stage 1: Categorization of high dimensionality
100<feature dimensions<sample size
feature dimensions≥sample size
Stage 2: High-dimensionality-driven feature extraction strategy selection
PCA feature extraction
Non-feature extraction
Stage 3: High-dimensionality-driven classifier selection
Single linear classifier
Single nonlinear classifier
Nonlinear ensemble classifier
Linear ensemble classifier
Output of credit classification results

Fig. 1 General framework of high-dimensionality-trait-driven learning paradigm

Based on the definitions of high-dimensionality and the quantitative relationship between the number of attribute features and the number of samples, the high-dimensionality traits of credit dataset are further categorized into two categories, 100 < feature dimensions < sample size (Chandrashekar and Sahin 2014) and feature dimensions ≥ sample size (Mwangi et al. 2014; Hua et al. 2009), for the convenience of analysis and computation. In terms of high-dimensionality-trait-driven idea, feature extraction strategy and classifier selections are carried out under two categories of high-dimensional conditions, which will be elaborated below.

High-dimensionality-trait-driven feature extraction strategy selection

The concept of the curse of dimensionality was proposed by Bellman in 1961, and was later used to refer to various high dimensionality problems in data analysis caused by an excessive number of features. To overcome the problems caused by the dimensionality disaster in credit risk classification, feature extraction is one of the effective dimensionality reduction methods. Based on the idea of high-dimensionality trait-driven modeling, this paper selects different feature extraction strategies according to different categories of high dimensionality. In particular, we focus mainly on the selection of feature extraction method when 100 < feature dimensions < sample size and feature dimensions ≥ sample size. In these circumstances, linear feature extraction, nonlinear feature extraction, and non-feature extraction are used as three typical extraction strategies, which will be illustrated below. In particular, non-feature extraction means that the classification is performed directly without dimensionality reduction.

Feature extraction strategy selection for 100 < feature dimensions < sample size

When 100 < feature dimensions < sample size (Chandrashekar and Sahin 2014), there are many attribute features, and the number of samples is relatively large; hence, a large amount of redundant information may be easily produced. In order to reduce the impact of redundant features on the classification performance and the calculation cost caused by data size, the dimension reduction is performed on the high-dimensionality dataset. This dimension reduction includes both nonlinear and linear feature extractions.

The nonlinear feature extraction mainly includes kernel based feature extraction, such as kernel principal component analysis (KPCA), and manifold learning based feature extraction, such as isometric mapping (ISOMAP) and locally linear embedding (LLE). In this case, the data sample size is relatively large compared to the high-dimensional trait. Using the nonlinear feature extraction method to reduce the dimension has certain advantages in theory, but in practical applications, especially if the sample size is large, it would produce huge calculation burden (Li and Lu 1999).The nonlinear dimension reduction performance is greatly influenced by the noise in the data, which is often worse than expected (Geng et al. 2005).

Therefore, the linear feature extraction method is chosen for dimensionality reduction under the condition of "100 < feature dimensions < sample size" in order to reduce the computational time and complexity, which exist in the nonlinear feature extraction methods. Meanwhile, the linear relationships of attribute features are strong when the datasets are directly classified, which proves the necessity of using the linear feature extraction strategy (Rosenblatt 1988). Usually, linear feature extraction refers to a method of constructing the linear dimension reduction mapping and obtaining the low-dimensional representation of the high-dimensional data. This type of method is not only suitable for dealing with linear structure but also for dealing with high dimensional traits with more samples, such as 100 < feature dimensions < sample size. Some conventional linear feature extraction methods include principal component analysis (PCA), linear discriminant analysis (LDA), and locality preserving projections (LPP). It is worth noting that LDA can only reduce the dimension to (n-1) at most after feature extraction for n-element classification problem (Fisher 1936), so it cannot be used as an effective feature extraction strategy for the binary credit classification problem in this study. However, LPP focuses on the local structure information of data, without considering the global structure sufficiently (He 2003). Therefore, when 100 < feature dimensions < sample size, PCA is selected as the high-dimensionality-trait-driven feature extraction strategy, according to the trait of high dimensionality. The details of PCA can be found in other papers, such as Kambhatla and Leen's (1997).

Feature extraction strategy selection for feature dimensions ≥ sample size

When feature dimensions ≥ sample size (Mwangi et al. 2014; Hua et al. 2009), the number of samples is relatively small compared to the usual sample. Then, dimensionality reduction could further compress the amount of data, which may lead to insufficient information for classification and affect the subsequent classification performance (Li et al. 2011). In this case, using feature extraction has no obvious advantage in reducing computational complexity and saving computational time. Therefore, when feature dimensions ≥ sample size, non-feature extraction is selected as the high-dimensionality-trait-driven feature extraction strategy, according to the trait of high dimensionality. With this strategy, the classification is carried out directly, without dimensionality reduction. It is often used in small-scale datasets with high dimensionality traits, which is a typical case of feature dimensions larger than sample size. This study will select different feature extraction strategies for different high-dimensional traits. In particular, PCA is used for feature extraction when 100 < feature dimensions < sample size, and non-feature extraction is performed when feature dimensions ≥ sample size in the experimental analysis, as illustrated in Fig. 1.

High-dimensionality-trait-driven classifier selection

In order to obtain the good classification performance, different classifiers will be used. In this paper, single linear classifier, single nonlinear classifier, and their corresponding linear or nonlinear ensemble classifiers will be used in terms of high-dimensionality traits. In particular, single classifier refers to the single classification model rather than the integration of multiple classifiers.

Classifier selection when 100 < feature dimensions < sample size

As mentioned in the previous section, when 100 < feature dimensions < sample size (Chandrashekar and Sahin 2014), a typical linear feature extraction strategy, PCA, will be used for dimension reduction. In this case, 12 classifiers are utilized, and the experimental results show that the linear classifier performs the best. The main reasons are two-fold: On the one hand, there are strong linear relationships among the attribute features in the testing dataset, thus the linear classifier can fit those linear features well. On the other hand, under the condition of "100 < feature dimensions < sample size", if the number of samples is relatively large, resulting in a large data scale, the linear classifier can reduce the computational complexity greatly. Therefore, when 100 < feature dimensions < sample size, PCA is selected as the feature extraction strategy, and single linear classifier is selected as the classification model, according to the high dimensionality trait.

Usually, the linear classifier is a typical classification model that can separate positive and negative samples with a hyperplane. The single linear classifier used in this study includes LDA (Blei et al. 2003) and LogR (Grablowsky and Talley 1981).

Classifier selection when feature dimensions ≥ sample size

As mentioned earlier, when feature dimensions ≥ sample size (Mwangi et al. 2014; Hua et al. 2009), non-feature extraction strategy is selected. In this case, the number of samples is relatively small, and the classification performance of a single classifier is unstable and prone to errors. Therefore, the ensemble classifier is chosen to reduce the fluctuation error of the single linear classifier. Meanwhile, because of the strong linear relationship between the data features, the linear ensemble classifier is selected as the generic classification model under the high-dimensional features.

The linear ensemble classifier used in this paper is obtained by integrating linear single classifier with bootstrap aggregating (bagging) and majority voting method. The bagging algorithm (Breiman 1996) selects m subsets from the training set uniformly and uses the algorithms of classification and regression on the m training sets to obtain m single classification models, before obtaining the results of bagging through the methods of majority voting. The main reason for selecting bagging as an ensemble method is that it has lower computational complexity compared to the other ensemble method, such as boosting.

Data descriptions and experimental design

Data descriptions and evaluation criteria

In this section, two real-world credit datasets, Kaggle loan default prediction dataset (Kreienkamp and Kateshov 2014) and China Unionpay credit dataset (Liu et al. 2019), are used to test the effectiveness of the proposed high-dimensionality-trait-driven learning paradigm. The specific descriptions of the two datasets are shown below.

Kaggle loan default prediction dataset

This publicly available credit dataset is obtained from Kaggle's website (https://www.kaggle.com/c/loan-default-prediction) and consists of 105,471 samples with 769 attributes. For the original dataset, some preprocessing steps are performed. First, data cleaning was performed, where some samples with missing values were deleted directly. Second, classification label transformation was conducted. After deleting the samples with missing values, the default loss value of each sample was transformed into a binary classification problem illustrating whether or not to default, represented by 0 or 1. Third, imbalance data processing was performed. After classification label transformation, the ratio of the non-default samples (good samples) to the default samples (bad samples) was about 10:1, thus the dataset is highly imbalanced. In order to reduce the influence of imbalance on the experimental results, the dataset was undersampled based on clustering results (Chao et al. 2020). The main idea of clustering is to generate 10 clusters by clustering the good samples of the dataset, undersampling them at the specified sampling rate in each cluster, and integrating the 5275 good samples from the undersampled samples with the same number of bad samples to formulate a new analytical dataset with an imbalance rate of 1:1.

After that, the new dataset is composed of 10,550 samples with 769 features, and it meets the high dimensional trait condition because 100 < feature dimensions (769) < sample size (10,550). In order to meet the standards of two high dimensional traits simultaneously, 700 samples are randomly selected from the dataset after preprocessing to construct a dataset with other high dimensional traits, so the condition of "feature dimensions (769) ≥ sample size (700)" can be satisfied.

China Unionpay credit dataset

The China Unionpay credit dataset is obtained from the data competition created by China Unionpay (https://open.chinaums.com/#/intro). This dataset is a binary classification problem, which divides 11,017 observations into two classes: good credits (8873 observations) and bad credits (2144 observations). The dataset describes the observations on 199 feature attributes, including six major dimensions: identity information and property status, cardholder information, trading information, loan information, repayment information, and loan application information.

Similar to the Kaggle dataset, the pre-processing steps are conducted. For the missing values of samples, the average interpolation method is initially used for imputation. Then, using the random undersampling, 2144 samples from the good credit ones are randomly selected. Finally, all bad credit samples are combined with randomly selected 2144 good samples into analytical samples with an imbalance rate of 1:1.

After processing, the new analytical dataset is composed of 4288 samples with 199 features, and it meets the high dimensional trait condition since 100 < feature dimensions (199) < sample size (4288). In order to meet another high dimensional trait condition, 199 samples are selected from the dataset by random undersampling after pre-processing, so that the condition of "feature dimensions (199) ≥ sample size (199)" is satisfied.

To evaluate the performance of the analytical model, Total accuracy (Total for short), true positive accuracy (TP for short), true negative accuracy (TN for short), and area under curve (AUC for short) (Bradley 1997; Shen et al. 2020) are selected as evaluation criteria.

Experimental design

In the experimental design, two main operations, including feature extraction and classifier selection, are conducted. In the selection of the feature extraction strategy, PCA-based linear feature extraction was performed when 100 < feature dimensions < sample size, according the high dimensionality traits, and non-feature extraction was selected as a comparing benchmark model. Similarly, non-feature extraction was carried out when feature dimensions ≥ sample size, and the PCA feature extraction was selected as a benchmark model. The performance of 12 classifiers are compared to evaluate the effectiveness of high-dimensionality-trait-driven feature extraction strategy selection. When PCA was used for dimension reduction, principal components with cumulative variance contribution rate up to 98% were selected as the new attribute features after dimensionality reduction.

In the classifier selection, this study chose 12 classifiers: LDA, LogR, KNN, SVM, back propagation neural network (BPNN), classification and regression tree (CART), and the ensemble classifiers with bagging corresponding to these 6 single classifiers. Among them, LDA and LogR belong to the single linear classifiers, and their respective ensembles belong to the linear ensemble classifiers. KNN, SVM, BPNN, and CART belong to the single nonlinear classifiers, and their respective ensembles belong to the nonlinear ensemble classifiers. When 100 < feature dimensions < sample size, single linear classifier is used after the dimension reduction of PCA. When feature dimensions ≥ sample size, the linear ensemble classifier is used directly, without feature extraction considering the description of "Methodology formulation" section.

Regarding model specification, LDA selects the "diaglineard" discriminant function, and the tolerance of LogR iteration termination condition is set to 0.7. The k value of KNN is set to 12. SVM uses RBF kernel function with regularization parameter $C=12$ and $\sigma^2=2$. The number of neurons in each layer of BPNN is set to 5, the transfer functions for hidden lalyer and output layer are "logsig" and "purelin", respectively, and the training function is "traincgf". For the decision tree method, the CART uses default parameters. When bagging is utilized for the ensemble of the base classifier, the number of base classifiers is set to 3.

In addition, the datasets are divided into training sets and test sets, with a 7:3 ratio. Due to the random initial conditions and the randomness generated by the training set, each model would be run 10 times (Yu et al. 2008). The average values of the Total, TP, TN, and AUC, and the corresponding standard deviation are used as the results of these models. These results were then used to select suitable feature extraction methods and classifiers under different high-dimensionality trait conditions, as well as further to verify the effectiveness of the proposed high-dimensionality-trait-driven learning paradigm.

Results and discussion

Experimental results when 100 < feature dimensions < sample size

According to the experimental design, the empirical results of two datasets when 100 < feature dimensions < sample size are shown in Tables 1, 2, 3 and 4.

Tables 1 and 2 present the performance of the two datasets in the 12 classification algorithms after PCA feature extraction, for 100 < feature dimensions < sample size. To make the comparison more intuitive and easy to comprehend, the classification results without the feature extraction method are marked in the brackets as the benchmark model for comparison. In addition, the results with bold font are the best in the tables.

As seen from Tables 1 and 2, under the condition of 100 < feature dimensions < sample size, the high-dimensionality-trait-driven learning paradigm with PCA feature extraction method has a better classification performance compared to the non-feature extraction strategy in most of the 12 classifiers. This indicates the effectiveness of the high-dimensionality-trait-driven learning paradigm for feature extraction selection. This can be interpreted with the following two reasons: On the one hand, in the

Table 1 Results of different classifiers with/without PCA in Kaggle dataset when 100 < feature dimensions < sample size

Feature extraction	Classifer		Total	TP	TN	AUC
PCA-based feature extraction (Non-feature extraction)	Single linear classifier	LogR	**0.6884** (0.6803)	**0.7030** (0.6865)	0.6737 **(0.6743)**	**0.7574** (0.7528)
		LDA	**0.6897** (0.6011)	**0.7009** (0.6360)	**0.6785** (0.5658)	**0.7592** (0.6351)
	Single nonlinear classifier	KNN	**0.6360** (0.6144)	**0.8205** (0.7434)	0.4485 **(0.4835)**	**0.7015** (0.6653)
		SVM	**0.7112** (0.7012)	**0.7411** (0.7048)	0.6809 **(0.6976)**	**0.7822** (0.7707)
		BPNN	**0.6771** (0.6537)	**0.6731** (0.6515)	**0.6812** (0.6568)	**0.7468** (0.7131)
		CART	0.5676 **(0.6382)**	0.5686 **(0.6402)**	0.5667 **(0.6364)**	0.5636 **(0.6359)**
	Linear ensemble classifier	LogR Bagging	0.6791 **(0.6861)**	0.6948 **(0.7079)**	0.6631 **(0.6639)**	0.7447 **(0.7579)**
		LDA Bagging	**0.6793** (0.6003)	**0.6917** (0.6387)	**0.6670** (0.5614)	**0.7441** (0.6363)
	Nonlinear ensemble classifier	KNN Bagging	0.6120 **(0.6331)**	**0.8307** (0.7305)	0.3892 **(0.5345)**	0.6702 **(0.6876)**
		SVM Bagging	**0.6972** (0.6813)	**0.7290** (0.6243)	0.6650 **(0.7396)**	**0.7652** (0.7459)
		BPNN Bagging	**0.6747** (0.6686)	**0.6825** (0.6618)	0.6668 **(0.6752)**	**0.7340** (0.7327)
		CART Bagging	0.5991 **(0.6372)**	**0.6313** (0.6138)	0.5662 **(0.6610)**	0.5908 **(0.6330)**

dataset with high-dimensional traits, the data has a strong linear relationship and the information redundancy is high. PCA-based feature extraction can reduce noise and improve the accuracy of classification. On the other hand, in this case, PCA-based feature extraction can also reduce computational complexity and save computational time for a large sample size.

After demonstrating the importance of PCA-based feature extraction, the subsequent task is to illustrate the effectiveness of the classifier selection in terms of the proposed learning paradigm. According to the experimental results in Tables 3 and 4, when 100 < feature dimensions < sample size, single linear classifier should be selected as a classification model because single linear classifiers perform better than other classifiers listed in this paper, considering the experimental results of two credit datasets.

To verify its effectiveness further, single nonlinear classification, linear ensemble classifier, and nonlinear ensemble classifier are compared with the benchmark models. In 10 experiments, the corresponding results of each type of classifier are composed of the average results of the multiple base classifiers. For example, the average of the classification results of LDA and logR in each experiment is expressed as the result of a linear single classifier. In each experiment, the performance of four categories of classifiers under four evaluation criteria can be compared, as reported in Tables 3 and 4. It should be noted that the results with bold font illustrate the best performance under the same evaluation indicators in each experiment.

Table 2 Results of different classifiers with/without PCA in China Unionpay dataset when 100 < feature dimensions < sample size

Feature extraction	Classifer		Total	TP	TN	AUC
PCA-based feature extraction (Non-feature extraction)	Single linear classifier	LogR	**0.6965** (0.6959)	**0.7473** (0.7348)	0.6459 **(0.6571)**	**0.7481** (0.7463)
		LDA	**0.6982** (0.6696)	**0.7449** (0.7447)	**0.6517** (0.5944)	**0.7477** (0.7137)
	Single nonlinear classifier	KNN	0.6468 **(0.6699)**	**0.7475** (0.7237)	0.5462 **(0.6163)**	0.7020 **(0.7208)**
		SVM	0.7024 **(0.7029)**	**0.7479** (0.6926)	0.6571 **(0.7134)**	**0.7574** (0.7561)
		BPNN	**0.6890** (0.6860)	**0.7111** (0.7091)	**0.6672** (0.6626)	**0.7355** (0.7321)
		CART	0.6103 **(0.6277)**	0.6084 **(0.6304)**	0.6123 **(0.6250)**	0.6039 **(0.6349)**
	Linear ensemble classifier	LogR Bagging	**0.6970** (0.6912)	**0.7500** (0.7345)	0.6440 **(0.6480)**	**0.7472** (0.7430)
		LDA Bagging	**0.6954** (0.6697)	0.7415 **(0.7475)**	**0.6496** (0.5918)	**0.7455** (0.7154)
	Nonlinear ensemble classifier	KNN Bagging	0.6100 **(0.6581)**	**0.7612** (0.7258)	0.4594 **(0.5903)**	0.6638 **(0.7103)**
		SVM Bagging	**0.6618** (0.6489)	**0.6836** (0.6219)	0.6404 **(0.6757)**	0.7005 **(0.7062)**
		BPNN Bagging	**0.7005** (0.6900)	**0.7075** (0.7062)	**0.6940** (0.6739)	**0.7434** (0.7364)
		CART Bagging	**0.6484** (0.5446)	**0.6357** (0.5540)	**0.6614** (0.5370)	**0.6600** (0.5583)

As can be seen from Tables 3 and 4, no matter what the Kaggle or the Unionpay dataset is utilized in 10 experiments, single linear classifier performs the best in terms of Total, TN, and AUC, and nonlinear ensemble classifier performs the best in terms of TP. Furthermore, regarding average and standard deviation, the PCA-based single linear classifier obtains the best results considering the four evaluation criteria, indicating that the proposed high-dimensionality-trait-driven learning paradigm has strong robustness. The main reasons involve the following two aspects: First, when 100 < feature dimensions < sample size, PCA-based feature extraction can reduce the impact of redundant features on the classification performance as well as the calculation cost caused by the data sample size. Second, the experimental results show that the datasets have a strong linear relationship without feature extraction, so the linear single classifier can be effective.

Therefore, based on the four evaluation criteria, it can be conclude that when 100 < feature dimensions < sample size, the single linear classifier performs the best after PCA feature extraction, demonstrating the effectiveness of the proposed high-dimensionality-trait-driven learning paradigm. This indicates that different feature extraction strategies and classifiers selection should be carefully determined by the different traits of high dimensionality.

Table 3 Robustness analysis of classification with PCA in Kaggle dataset when 100 < feature dimensions < sample size

	Single linear classifier					**Single nonlinear classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	**0.6972**	0.7090	**0.6845**	**0.7629**	Results of 10 experiments	0.6537	0.7096	0.5941	0.7010
	0.6888	0.7066	**0.6714**	**0.7523**		0.6514	0.7150	0.5891	0.6997
	0.6986	0.7040	**0.6929**	**0.7612**		0.6459	0.7008	0.5887	0.6921
	0.6848	0.6850	**0.6846**	**0.7549**		0.6498	0.6987	0.5992	0.7036
	0.6845	0.6938	**0.6751**	**0.7539**		0.6437	**0.7059**	0.5800	0.6975
	0.6867	0.7142	**0.6588**	**0.7545**		0.6423	0.6793	0.6049	0.6901
	0.6913	0.6954	**0.6870**	**0.7604**		0.6527	0.6951	0.6081	0.7054
	0.6836	0.7094	0.6588	**0.7599**		0.6414	0.7208	0.5652	0.6937
	0.6754	0.6864	**0.6645**	**0.7517**		0.6419	0.6656	0.6186	0.6931
	0.6995	0.7153	**0.6835**	**0.7714**		0.6568	**0.7174**	0.5954	0.7090
Average	0.6890	0.7019	0.6761	0.7583	Average	0.6480	0.7008	0.5943	0.6985
Standard deviation	0.0077	0.0110	0.0123	0.0061	Standard deviation	0.0056	0.0174	0.0150	0.0063
	Linear ensemble classifier					**Nonlinear ensemble classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	0.6921	0.7017	0.6819	0.7624	Results of 10 experiments	0.6595	**0.7148**	0.6004	0.7054
	0.6883	0.7082	0.6689	0.7516		0.6593	**0.7155**	0.6043	0.7089
	0.6919	0.6947	0.6890	0.7551		0.6570	**0.7159**	0.5956	0.7004
	0.6479	0.6704	0.6246	0.6969		0.6077	**0.7746**	0.4355	0.6367
	0.6852	0.7006	0.6693	0.7528		0.6456	0.6825	0.6078	0.6931
	0.6367	0.6493	0.6238	0.6884		0.6000	**0.7323**	0.4658	0.6283
	0.6889	0.6920	0.6857	0.7593		0.6570	**0.7170**	0.5938	0.7062
	0.6847	0.7013	**0.6687**	0.7567		0.6513	**0.7219**	0.5836	0.7054
	0.6796	0.6985	0.6610	0.7505		0.6518	**0.6990**	0.6054	0.7008
	0.6968	0.7159	0.6775	0.7702		0.6681	0.7100	0.6256	0.7154
Average	0.6792	0.6933	0.6650	0.7444	Average	0.6457	0.7184	0.5718	0.6901
Standard deviation	0.0202	0.0195	0.0232	0.0280	Standard deviation	0.0229	0.0239	0.0652	0.0310

Experimental results when feature dimensions ≥ sample size

Similar to "Experimental results when 100 < feature dimensions < sample size" section, this section will report the experimental results of two datasets under the condition of "feature dimensions ≥ sample size" in terms of the framework of Fig. 1. Accordingly, the computational results are presented in Tables 5, 6, 7 and 8.

In detail, Tables 5 and 6 show the performance of the two datasets in the 12 classification algorithms without feature extraction, when feature dimensions ≥ sample size. Similarly, the classification results with the PCA feature extraction method are marked in brackets as the benchmark model, for comparison purpose. In addition, the results with bold font are the best in the tables.

As seen from Tables 5 and 6, when feature dimensions ≥ sample size, the high-dimensionality-trait-driven learning paradigm without feature extraction performs better compared to the PCA-based feature extraction strategy in most of the 12 classifiers, proving the effectiveness of the high-dimensionality-trait-driven learning paradigm for feature extraction selection. There are two possible reasons: On the one hand, under the condition of "feature dimensions ≥ sample size", there is only a small

Table 4 Robustness analysis of classification with PCA in China Unionpay dataset when 100 < feature dimensions < sample size

	Single linear classifier					Single nonlinear classifier			
	Total	TP	TN	AUC		Total	TP	TN	AUC
Results of 10 experiments	**0.7077**	0.7512	**0.6646**	**0.7593**	Results of 10 experiments	0.6638	0.7274	0.6009	0.7019
	0.6883	**0.7433**	0.6344	**0.7407**		0.6564	0.6864	0.6271	0.6911
	0.7030	0.7286	**0.6746**	**0.7420**		0.6693	0.6895	0.6467	0.7038
	0.6879	0.7516	**0.6250**	**0.7458**		0.6658	0.7219	0.6103	0.7093
	0.6933	**0.7431**	0.6420	0.7502		0.6665	0.7030	0.6289	0.7047
	0.6960	**0.7348**	**0.6574**	**0.7548**		0.6660	0.6983	0.6337	0.7108
	0.6972	**0.7461**	0.6492	0.7397		0.6508	0.6889	0.6135	0.6830
	0.7108	**0.7857**	0.6375	**0.7653**		0.6636	0.7170	0.6114	0.7024
	0.6960	**0.7329**	0.6592	0.7427		0.6617	0.7026	0.6207	0.6976
	0.6933	**0.7433**	0.6444	0.7385		0.6574	0.7021	0.6137	0.6923
Average	0.6974	0.7461	0.6488	0.7479	Average	0.6621	0.7037	0.6207	0.6997
Standard deviation	0.0077	0.0158	0.0151	0.0092	Standard deviation	0.0056	0.0142	0.0135	0.0087
	Linear ensemble classifier					**Nonlinear ensemble classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	0.6921	0.7017	0.6819	0.7624	Results of 10 experiments	0.6595	**0.7148**	0.6004	0.7054
	0.6883	0.7082	**0.6689**	0.7516		0.6593	0.7155	0.6043	0.7089
	0.6919	0.6947	0.6890	0.7551		0.6570	**0.7159**	0.5956	0.7004
	0.6479	0.6704	0.6246	0.6969		0.6077	**0.7746**	0.4355	0.6367
	0.6852	0.7006	**0.6693**	**0.7528**		0.6456	0.6825	0.6078	0.6931
	0.6367	0.6493	0.6238	0.6884		0.6000	0.7323	0.4658	0.6283
	0.6889	0.6920	**0.6857**	**0.7593**		0.6570	0.7170	0.5938	0.7062
	0.6847	0.7013	**0.6687**	0.7567		0.6513	0.7219	0.5836	0.7054
	0.6796	0.6985	**0.6610**	**0.7505**		0.6518	0.6990	0.6054	0.7008
	0.6968	0.7159	**0.6775**	**0.7702**		0.6681	0.7100	0.6256	0.7154
Average	0.6792	0.6933	0.6650	0.7444	Average	0.6457	0.7184	0.5718	0.6901
Standard deviation	0.0202	0.0195	0.0232	0.0280	Standard deviation	0.0229	0.0239	0.0652	0.0310

number of samples. If feature extraction is conducted, the samples cannot provide sufficient information for classification task, which can affect the classification performance (Li et al. 2011). One the other hand, when samples are small, using the feature extraction method to reduce cost and improve the calculation efficiency has no clear advantage.

After proving the importance of non-feature extraction when feature dimensions ≥ sample size, the subsequent task is to demonstrate the effectiveness of the classifier selection. According to the experimental results in Tables 5 and 6, when feature dimensions ≥ sample size, linear ensemble classifier should be selected because the performance of this type of classifiers is superior considering the experimental results of two credit datasets.

To verify its effectiveness further, single linear classifier, single nonlinear classifier, and nonlinear ensemble classifier are selected as the benchmark models for comparison purposes. In 10 experiments, the results of each type of classifier are composed of the average results of the multiple base classifiers it contains. In each experiment, the performance of four categories of classifiers under four evaluation criteria was compared

Table 5 Results of different classifiers without/with PCA in Kaggle dataset when feature dimensions ≥ sample size

Feature extraction	Classifier		Total	TP	TN	AUC
No Feature extraction (PCA feature extraction)	Single linear classifier	LogR	**0.6329** (0.6219**)**	**0.6370** (0.6229)	**0.6321** (0.6232)	**0.6792** (0.6761)
		LDA	0.6081 **(0.6200)**	**0.6500** (0.6194)	0.5673 **(0.6233)**	0.6466 **(0.6791)**
	Single nonlinear classifier	KNN	**0.5714** (0.5500)	**0.7080** (0.6887)	**0.4390** (0.4108)	**0.6156** (0.5901)
		SVM	**0.6381** (0.6271)	**0.6380** (0.6279)	**0.6420** (0.6311)	**0.6935** (0.6873)
		BPNN	**0.6081** (0.5586)	**0.5584** (0.5150)	**0.6604** (0.6018)	**0.6537** (0.5859)
		CART	**0.5786** (0.5376)	**0.5784** (0.5401)	**0.5808** (0.5361)	**0.5896** (0.5436)
	Linear ensemble classifier	LogR Bagging	**0.6257** (0.6224)	0.6376 **(0.6401)**	**0.6162** (0.6064)	**0.6727** (0.6687)
		LDA Bagging	0.6171 **(0.6195)**	0.6348 **(0.6516)**	**0.5988** (0.5893)	0.6457 **(0.6731)**
	Nonlinear ensemble classifier	KNN Bagging	**0.5700** (0.5586)	0.6852 **(0.6969)**	**0.4584** (0.4197)	**0.6170** (0.5909)
		SVM Bagging	0.5819 **(0.6281)**	0.5314 **(0.6616)**	**0.6433** (0.5960)	0.6401 **(0.6771)**
		BPNN Bagging	**0.6343** (0.5986)	**0.6228** (0.6140)	**0.6505** (0.5870)	**0.6709** (0.6497)
		CART Bagging	**0.5890** (0.5495)	**0.5951** (0.5552)	**0.5847** (0.5446)	**0.5993** (0.5734)

Table 6 Results of different classifiers without/with PCA in China Unionpay dataset when feature dimensions ≥ sample size

Feature extraction	Classifier		Total	TP	TN	AUC
		LDA	0.5220 **(0.6220)**	0.4730 **(0.6597)**	**0.5907** (0.5836)	**0.6577** (0.6534)
	Single nonlinear classifier	KNN	**0.6559** (0.5932)	0.7535 **(0.8672)**	**0.5521** (0.3004)	**0.7145** (0.6635)
		SVM	**0.6627** (0.6085)	0.7111 **(0.7441)**	**0.6183** (0.4711)	**0.7014** (0.6726)
		BPNN	**0.6017** (0.5831)	**0.7474** (0.7183)	**0.4408** (0.4239)	**0.6096** (0.6022)
		CART	**0.5780** (0.5644)	0.5601 **(0.5962)**	**0.5957** (0.5314)	**0.5958** (0.5686)
	Linear ensemble classifier	LogR Bagging	**0.6407** (0.6237)	0.6562 **(0.6787)**	**0.6287** (0.5704)	0.6532 **(0.6576)**
		LDA Bagging	**0.6729** (0.6271)	**0.7594** (0.6847)	**0.5828** (0.5738)	**0.6981** (0.6733)
	Nonlinear ensemble classifier	KNN Bagging	**0.6356** (0.5627)	0.7996 **(0.8975)**	**0.4634** (0.2069)	**0.6868** (0.6630)
		SVM Bagging	**0.6695** (0.6305)	0.6824 **(0.6977)**	**0.6570** (0.5549)	**0.7158** (0.6798)
		BPNN Bagging	**0.6576** (0.5949)	**0.7621** (0.6747)	**0.5379** (0.5050)	**0.6892** (0.6522)
		CART Bagging	0.5542 **(0.5831)**	0.4748 **(0.6842)**	**0.6371** (0.4779)	**0.5665** (0.5547)

Table 7 Robustness analysis of classification without feature extraction in Kaggle dataset when feature dimensions ≥ sample size

	Single linear classifier					**Single nonlinear classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	0.6190	0.6538	0.5849	0.6636	Results of 10 experiments	0.6000	**0.6827**	0.5189	0.6463
	0.6381	0.6887	0.5865	0.6910		**0.6488**	0.6533	0.6442	**0.6919**
	0.5667	0.5802	0.5529	0.6020		0.5750	0.4976	**0.6538**	0.6044
	0.6143	0.5872	0.6436	**0.6590**		0.5940	0.5734	0.6163	0.6268
	0.6429	0.6228	**0.6667**	**0.6894**		0.6119	0.6053	0.6198	0.6483
	0.6357	0.6250	**0.6462**	0.6828		0.6107	0.6250	0.5967	0.6454
	0.6548	**0.6905**	0.6190	**0.6806**		0.5917	0.5738	0.6095	0.6378
	0.5762	0.6237	0.5354	0.6201		0.5845	0.6134	0.5597	**0.6320**
	0.6381	0.7050	0.5773	0.6842		0.6179	**0.7600**	0.4886	0.6593
	0.6190	**0.6582**	0.5848	**0.6562**		0.5560	0.6224	0.4978	0.5887
Average	0.6205	0.6435	0.5997	0.6629	Average	0.5990	0.6207	0.5805	0.6381
Standard deviation	0.0287	0.0431	0.0427	0.0303	Standard deviation	0.0255	0.0701	0.0604	0.0285
	Linear ensemble classifier					**Nonlinear ensemble classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	**0.6357**	0.6394	**0.6321**	**0.6644**	Results of 10 experiments	0.6190	0.6563	0.5825	0.6595
	0.6452	**0.6887**	0.6010	0.6830		0.6417	0.6132	**0.6707**	0.6748
	0.5952	**0.5849**	0.6058	**0.6142**		0.5690	0.4906	0.6490	0.5846
	0.6095	**0.6101**	0.6089	0.6497		0.5881	0.5321	**0.6485**	0.6297
	0.6333	**0.6360**	0.6302	0.6810		0.5702	0.5044	0.6484	0.6146
	0.6429	0.6538	0.6321	**0.6887**		0.6167	**0.6995**	0.5354	0.6379
	0.6214	0.6333	0.6095	0.6593		0.6143	0.5786	**0.6500**	0.6590
	0.5952	0.6237	**0.5708**	0.6203		0.5750	**0.6289**	0.5288	0.6133
	0.6357	0.6900	**0.5864**	**0.6959**		0.5595	0.7500	0.3864	0.6296
	0.6000	0.6020	**0.5982**	0.6355		0.5845	0.6327	0.5424	0.6150
Average	0.6214	0.6362	0.6075	0.6592	Average	0.5938	0.6086	0.5842	0.6318
Standard deviation	0.0199	0.0343	0.0202	0.0288	Standard deviation	0.0273	0.0837	0.0886	0.0270

and presented in Tables 7 and 8. It should be noted that the results with bold font are the best performance under the same evaluation criteria in every experiment.

As seen from Tables 7 and 8, in the two datasets, the linear ensemble classifier performs better in the Total index, with excellent classification performance for the other three evaluation criteria, as well. Overall, the linear ensemble classifier performs the best compared to other benchmark classifiers. Moreover, it has the lowest standard deviation for most evaluation criteria in the 10 experiments, indicating strong robustness. The evidence explaining this phenomenon is that bagging ensemble helps to reduce the errors caused by the fluctuation of training data. Due to the small sample size and unstable performance of the single classifier in this case, bagging can reduce the variance of the base classifier and improve the generalization performance.

Therefore, based on the four evaluation criteria, it is not hard to find that when feature dimensions ≥ sample size, the linear ensemble classifier has the best performance without feature extraction, demonstrating the effectiveness of the proposed

Table 8 Robustness analysis of classification without feature extraction in China Unionpay dataset when feature dimensions ≥ sample size

	Single linear classifier					**Single nonlinear classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	0.6525	0.7308	0.5909	0.6498	Results of 10 experiments	0.6864	0.7212	0.6591	0.7002
	0.5847	0.7241	0.4500	0.6040		0.6017	0.7155	0.4917	**0.6303**
	0.5085	**0.7069**	0.3167	0.5112		0.5636	0.6207	0.5083	**0.6013**
	0.6271	**0.8382**	0.3400	0.7203		**0.6949**	0.6765	**0.7200**	0.7413
	0.5508	0.4032	**0.7143**	**0.7175**		0.5975	**0.7742**	0.4018	0.6796
	0.5339	0.2727	**0.8654**	**0.7663**		0.6229	**0.7348**	0.4808	0.6435
	0.5932	0.3594	**0.8704**	0.7378		0.6017	0.5859	0.6204	0.6616
	0.5593	0.3276	**0.7833**	0.6046		**0.6229**	0.6983	0.5500	0.6068
	0.5593	**0.8226**	0.2679	0.6057		0.6525	0.7742	0.5179	0.6591
	0.5424	0.3485	**0.7885**	0.6544		0.6017	0.6288	0.5673	0.6295
Average	0.5712	0.5534	0.5987	0.6572	Average	0.6246	0.6930	0.5517	0.6553
Standard deviation	0.0438	0.2285	0.2372	0.0786	Standard deviation	0.0416	0.0644	0.0937	0.04315
	Linear ensemble classifier					**Nonlinear ensemble classifier**			
	Total	**TP**	**TN**	**AUC**		**Total**	**TP**	**TN**	**AUC**
Results of 10 experiments	**0.6949**	**0.7885**	0.6212	0.7086	Results of 10 experiments	0.6864	0.6923	**0.6818**	**0.7213**
	0.6356	0.7069	0.5667	0.6253		0.6059	0.6293	**0.5833**	0.6272
	0.5678	0.6207	**0.5167**	0.5534		0.5593	0.6034	**0.5167**	0.5902
	0.6864	0.7059	0.6600	0.6918		0.6864	0.7279	0.6300	**0.7431**
	0.6525	0.6774	0.6250	0.6964		0.6441	0.7097	0.5714	0.6707
	0.7288	0.7121	0.7500	0.7430		0.6059	0.5909	0.6250	0.6758
	0.6610	0.6875	0.6296	**0.7419**		0.6525	**0.7500**	0.5370	0.6671
	0.6017	0.7241	0.4833	0.6138		0.6017	**0.7759**	0.4333	**0.6353**
	0.6780	0.7581	0.5893	**0.6855**		0.6441	0.6210	**0.6696**	0.6580
	0.6610	**0.6970**	0.6154	**0.6970**		0.6059	**0.6970**	0.4904	0.6571
Average	0.6568	0.7078	0.6057	0.6757	Average	0.6292	0.6797	0.5739	0.6646
Standard deviation	0.0465	0.0452	0.0744	0.0601	Standard deviation	0.0406	0.0646	0.0803	0.0440

high-dimensionality-trait-driven learning paradigm, as shown in Fig. 1. This also indicates that different feature extraction strategies and classifiers should be carefully considered in terms of the different traits of high dimensionality hidden in the credit dataset.

Summary and discussion

From the experimental results and analysis in "Experimental results when 100 < feature dimensions < sample size" and "Experimental results when feature dimensions ≥ sample size" sections, several important findings and implications can be summarized.

First, when 100 < feature dimensions < sample size, the high-dimensionality-trait-driven learning paradigm with the PCA feature extraction method has a better classification performance compared to the non-feature extraction strategy. The single linear classifier performs the best after the PCA processing, according to the traits of high-dimensional data, as shown in "Experimental results when 100 < feature dimensions < sample size" section.

Second, when feature dimensions ≥ sample size, direct use of the linear ensemble classifier, without feature extraction, can achieve better classification performance, as shown in "Experimental results when feature dimensions ≥ sample size" section.

Third, for the selection of a feature extraction strategy, PCA-based linear feature extraction is carried out when 100 < feature dimensions < sample size and non-feature extraction is conducted when feature dimensions ≥ sample size. This demonstrates the effectiveness of the proposed high-dimensionality-trait-driven learning paradigm, as shown in Fig. 1.

Fourth, for the classifier selection inspired by the proposed high-dimensionality-trait-driven learning paradigm, when 100 < feature dimensions < sample size, the single linear classifier is selected as the generic classification model. In addition, when feature dimensions ≥ sample size, the linear ensemble classifier is chosen.

Finally, the above analysis proves the effectiveness of the high-dimensionality-trait-driven learning paradigm for feature extraction and classifier selection. When 100 < feature dimensions < sample size, PCA is selected as the feature extraction strategy, and single linear classifier is selected as the classification model. When feature dimensions ≥ sample size, non-feature extraction is selected as the feature extraction strategy, and linear ensemble classifier is selected as the classification model.

Although the proposed high-dimensionality-trait-driven learning paradigm provides a reliable guideline for feature extraction and classifier selection in high-dimensional credit classification, feature dimension categories are dependent on the sample data, and lack the strict mathematical reasoning and proof. This issue may limit the use of the proposed learning paradigm. In the future research, more datasets should be used to verify its effectiveness.

Conclusions

To solve the high-dimensionality issue and improve its accuracy in credit risk assessment, a high-dimensionality-trait-driven learning paradigm was proposed for feature extraction and classifier selection. For verification purposes, two credit datasets have been presented to test the classification capability and effectiveness of the learning paradigm proposed in this paper. The experimental results show that it can be better utilized to solve high-dimensionality issues in credit risk classification.

Moreover, the study can provide some important references for the selection of feature extraction and classifier for different high-dimensionality datasets, implying that the proposed high-dimensionality-trait-driven learning paradigm can be used as a promising credit risk assessment tool with high dimensionality traits. In practical applications, the proposed paradigm can help financial institutions to make suitable decisions and choose different strategies when faced with different situations of high dimensionality traits. This can not only improve the classification accuracy but also reduce the possible economic loss for financial institutions. Accordingly, it brings sufficient practical significance.

In addition, directions to improve the proposed learning paradigm further are suggested. Regarding the selection of feature extraction methods, the combination of different methods can be performed. In terms of classifier training, popular optimization algorithms, such as PSO, and powerful ensemble methods can be used to improve and optimize the classification performance of the classifier further. We plan to examine these issues in the future.

Abbreviations

LSA: Latent semantic analysis; SVD: Singular value decomposition; PCA: Principal component analysis; LPP: Locality preserving projections; NPE: Neighborhood preserving embedding; LDA: Linear discriminant analysis; KPCA: Kernel principal component analysis; LLE: Locally linear embedding; ISOMAP: Isometric feature mapping; AI: Artificial intelligence; LogR: Logistic regression; MPM: Mathematical programming model; KNN: K-nearest neighbor; ANN: Artificial neural networks; SVM: Support vector machines; DT: Decision trees; CART: Classification and regression tree; ELM: Extreme learning machine; BPNN: Back propagation neural network; Bagging: Bootstrap aggregating; TP: True positive; TN: True negative; AUC: Area under curve.

Acknowledgements

We would like to express our sincere appreciation to editors and four independent reviewers in making valuable comments and suggestions to the paper. Their comments have improved the quality of the paper immensely.

Authors' contributions

LY proposed the data-trait-driven methodology, participated in the framework design, drafted and edited the manuscript and supervised LY and KY. LY participated in the framework design, data preprocessing, software coding, and drafted the manuscript. KY carried out the data collection, literature investigation and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work is partially supported by grants from the Key Program of National Natural Science Foundation of China (NSFC Nos. 71631005 and 71731009), and the Major Program of the National Social Science Foundation of China (No. 19ZDA103).

Availability of data and materials

Data used in this paper are downloaded from the following websites: https://www.kaggle.com/c/loan-default-prediction and https://open.chinaums.com/#/intro.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Economics and Management, Beijing University of Chemical Technology, 15 Beisanhuan East Road, Chaoyang District, Beijing 100029, China. [2]School of Economics and Management, University of Chinese Academy of Sciences, 80 Zhongguancun East Road, Haidian District, Beijing 100190, China. [3]Canada International School of Beijing, Liangmaqiao Road, Chaoyang District, Beijing 100125, China.

Received: 8 June 2020 Accepted: 22 April 2021
Published online: 30 April 2021

References

Bai J, Li K (2012) Statistical analysis of factor models of high dimension. Ann Stat 40(1):436–465

Blei DM, Ng AY, Jordan MI, Lafferty J (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Bradley P (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28

Chao X, Kou G, Peng Y, Viedma EH (2020) Large-scale group decision-making with non-cooperative behaviors and heterogeneous preferences: an application in financial inclusion. Eur J Oper Res 288(1):271–293

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (2010) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407

Donskoy S (2019) BOMD: Building optimization models from data (neural networks based approach). Quant Finance Econ 3(4):608–623

Feder G, Just RE (1977) A study of debt servicing capacity applying logit analysis. J Dev Econ 4(1):25–38

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(7):179–188

Geng X, Zhan DC, Zhou ZH (2005) Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Trans Syst Man Cybern Part B Cybern 35(6):1098–1107

Grablowsky BJ, Talley WK (1981) Probit and discriminant functions for classifying credit applicants-a comparison. J Econ Bus 33(3):254–261

He X (2003) Locality Preserving Projections. Adv Neural Inf Process Syst 16(1):153–160

He X, Cai D, Yan S, Zhang HJ (2005) Neighborhood preserving embedding. In: IEEE international conference on computer vision, Beijing, 17–21 October 2005
Henley WE, Hand DJ (1996) A k-nearest-neighbour classifier for assessing consumer credit risk. J R Stat Soc Ser D (Stat) 45(1):77–95
Hua J, Tembe WD, Dougherty ER (2009) Performance of feature-selection methods in the classification of high-dimension data. Pattern Recogn 42(3):409–424
Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis. Neural Comput 9(7):1493–1516
Kou G, Chao X, Peng Y et al (2019) Machine learning methods for systemic risk analysis in financial sectors. Technol Econ Dev Econ 25(5):716–742
Kou G, Xu Y, Peng Y et al (2020) Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. Decis Support Syst. https://doi.org/10.1016/j.dss.2020.113429
Kreienkamp T, Kateshov A (2014) Credit risk modeling: combining classification and regression algorithms to predict expected loss. J Corporate Finance Res 4(32):4–10
Li DC, Liu CW, Hu SC (2011) A fuzzy-based data transformation for feature extraction to increase classification performance with small medical data sets. Artif Intell Med 52(1):45–52
Li S, Lu J (1999) Face recognition using the nearest feature line method. IEEE Trans Neural Networks 10(2):439–443
Liu Y, Ghandar A, Theodoropoulos G (2019) Island model genetic algorithm for feature selection in non-traditional credit risk evaluation. In: 2019 IEEE congress on evolutionary computation (CEC).
Ma GN, Wang Y (2020) Can the Chinese domestic bond and stock markets facilitate a globalising renminbi? Econ Polit Stud 8(3):291–311
Mabrouk AB (2020) Wavelet-based systematic risk estimation: application on GCC stock markets: the Saudi Arabia case. Quant Finance Econ 4(4):542–595
Mangasarian OL (1965) Linear and nonlinear separation of patterns by linear programming. Oper Res 13(3):444–452
Mwangi B, Tian TS, Soares JC (2014) A review of feature reduction techniques in neuroimaging. Neuroinformatics 12(2):229–244
Nayak SC, Misra BB (2020) Extreme learning with chemical reaction optimization for stock volatility prediction. Financ Innov. https://doi.org/10.1186/s40854-020-00177-2
Nelson CR, Plosser CR (1982) Trends and random walks in macroeconmic time series : Some evidence and implications. J Monet Econ 10(2):139–162
Niu K, Zhang Z, Liu Y et al (2020) Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. Inf Sci 536:120–134
Odom MD, Sharda R (1990) A neural network model for bankruptcy prediction. In: The 1990 international joint conference on neural networks (IJCNN), San Diego, CA, 17–21 June 1990
Pabuçcu H, Ongan S, Ongan A (2020) Forecasting the movements of Bitcoin prices: an application of machine learning algorithms. Quant Finance Econ 4(4):679–692
Rosenblatt F (1988) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65(6):386–408
Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
Rutkowski L, Jaworski M, Pietruczuk L, Duda P (2014) The CART decision tree for mining data streams. Inf Sci 266:1–15
Shen F, Zhao X, Kou G (2020) Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. Decis Support Syst. https://doi.org/10.1016/j.dss.2020.113366
Song JB, Wang X (2019) Customer concentration and management earnings forecast. Econ Polit Stud 7(4):454–479
Tam KY, Kiang MY (1992) Managerial applications of neural networks: the case of bank failure predictions. Manag Sci 38(7):926–947
Tang L, Yu L, He K (2014) A novel data-characteristic-driven modeling methodology for nuclear energy consumption forecasting. Appl Energy 128(3):1–14
Tang L, Yu L, Liu F, Xu W (2013) An integrated data characteristic testing scheme for complex time series data exploration. Int J Inf Technol Decis Mak 12(3):491–521
Tenenbaum J, De-Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
Waheed T, Bonnell RB, Prasher SO, Paulet E (2006) Measuring performance in precision agriculture: CART—A decision tree approach. Agric Water Manag 84(1–2):173–185
Wang G, Ma J (2010) A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. Expert Syst Appl 39(5):5325–5331
Wang H, Kou G, Peng Y (2020) Multi-class misclassification cost matrix for credit ratings in peer-to-peer lending. J Oper Res Soc 2:1–12
Xin J, Wang Z, Chen C, Ding L, Wang G, Zhao Y (2014) ELM: distributed extreme learning machine with mapreduce. World Wide Web 17(5):1189–1204
Yu L, Li X, Tang L et al (2015) Social credit: a comprehensive literature review. Financ Innov. https://doi.org/10.1186/s40854-015-0005-6
Yu L, Liu H (2003) Feature selection for high-dimensional data: A fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning, Washington, DC, 21–24 August 2003
Yu L, Wang S, Lai KK (2008) Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst Appl 34(2):1434–1444
Yu L, Yue W, Wang S, Lai KK (2010) Support vector machine based multiagent ensemble learning for credit risk evaluation. Expert Syst Appl 37(2):1351–1360
Zhang Y (2009) Enhanced statistical analysis of nonlinear processes using KPCA. KICA SVM Chem Eng Sci 64(5):801–811
Yu L, Zhang X (2021) Can small sample dataset be used for efficient internet loan credit risk assessment? Evidence from online peer to peer lending. Financ Res Lett 38:101521
Yu L, Zhou R, Tang L et al (2018) A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. Appl Soft Comput 69:192–202

Yu L, Huang X, Yin H (2020a) Can machine learning paradigm improve attribute noise problem in credit risk classification? Int Rev Econ Financ 70:440–455

Yu L, Zhou R, Chen R et al (2020b) Missing data preprocessing in credit classification: One-hot encoding or imputation? Emerg Mark Financ Trade. https://doi.org/10.1080/1540496X.2020.1825935

Yu L, Yao X, Zhang X et al (2020c) A novel dual-weighted fuzzy proximal support vector machine with application to credit risk analysis. Int Rev Financ Anal. https://doi.org/10.1016/j.irfa.2020.101577

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.