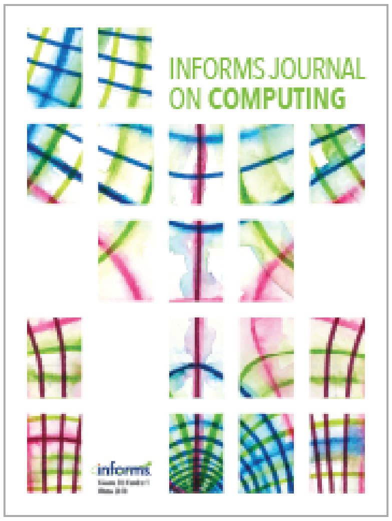


INFORMS Journal on Computing

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

Feature Selection and Grouping Effect Analysis for Credit Evaluation via Regularized Diagonal Distance Metric Learning

Tie Li; , Gang Kou; , Yi Peng; , Philip S. Yu

To cite this article:
Tie Li; , Gang Kou; , Yi Peng; , Philip S. Yu (2024) Feature Selection and Grouping Effect Analysis for Credit Evaluation via Regularized Diagonal Distance Metric Learning. INFORMS Journal on Computing

Published online in Articles in Advance 11 Dec 2024

. https://doi.org/10.1287/ijoc.2023.0322

INFORMS JOURNAL ON COMPUTING
Articles in Advance, pp. 1–22
ISSN 1091-9856 (print), ISSN 1526-5528 (online)
https://pubsonline.informs.org/journal/ijoc

Feature Selection and Grouping Effect Analysis for Credit Evaluation via Regularized Diagonal Distance Metric Learning

Tie Li,[a] Gang Kou,[b,c,d] Yi Peng,[a,*] Philip S. Yu[e]

[a] School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China; [b] Xiangjiang Laboratory, Changsha 410205, People's Republic of China; [c] School of Business Administration, Southwestern University of Finance and Economics, Chengdu 610074, People's Republic of China; [d] Big Data Laboratory on Financial Security and Behavior, Southwestern University of Finance and Economics, Chengdu 610074, People's Republic of China; [e] Department of Computer Science, University of Illinois at Chicago, Chicago, Illinois 60607

*Corresponding author

Contact: lteb2002@uestc.edu.cn, https://orcid.org/0000-0003-2795-9724 (TL); kougang@swufe.edu.cn, https://orcid.org/0000-0002-9220-8647 (GK); pengyi@uestc.edu.cn, https://orcid.org/0000-0003-0373-6665 (YP); psyu@uic.edu, https://orcid.org/0000-0002-3491-5968 (PSY)

Received: September 7, 2023
Revised: April 6, 2024; July 30, 2024; October 3, 2024
Accepted: October 4, 2024
Published Online in Articles in Advance: December 11, 2024

https://doi.org/10.1287/ijoc.2023.0322

Copyright: © 2024 The Author(s)

Abstract. In credit evaluation, feature selection and grouping effect analysis are used to identify the most relevant credit risk features. Most feature selection and grouping effect analysis are implemented via regularizing linear models. Nevertheless, substantial evidence shows that credit data are linearly inseparable due to heterogeneous credit customers and various risk sources. Although many nonlinear models have been proposed in the last two decades, the majority of them required recombination of the original features, which made it difficult to interpret the results of the models. To cope with this dilemma, we propose a diagonal distance metric learning model that improves distance metrics by rescaling the features. Meanwhile, feature selection and grouping effect analysis are realized by adding regularizations to the model. The main merit of the proposed model is that it avoids the limitation of the linear models by not pursuing linear separability, yet guaranteeing the interpretability. We also prove and explain why feature selection and grouping effect can be achieved and decompose the optimization problem into parallel linear programming problems, plus a small quadratic consensus-reaching problem, such that the optimization can be efficiently solved. Experiments using a real credit data set of 96,000 instances show that the proposed model improves the area under the receiver operating characteristic curve (AUC) of the distance-based classifier k-nearest neighbors by 14% in two-class credit evaluation and surpasses linear models in terms of accuracy, true positive rate, and AUC. The proposed regularized diagonal distance metric learning approach also has the potential to be applied to other fields where data are linearly inseparable.

History: Accepted by Ram Ramesh, Area Editor for Data Science and Machine Learning.

Funding: Financial support from the National Social Science Fund of China [Grant 23BTJ040] and the National Natural Science Foundation of China [Grants 72471047 and 71910107002] is gratefully acknowledged.

Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplemental Information (https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2023.0322) as well as from the IJOC GitHub software repository (https://github.com/INFORMSJoC/2023.0322). The complete IJOC Software and Data Repository is available at https://informsjoc.github.io/.

Keywords: diagonal distance metric learning • ElasticNet • feature selection • feature group • credit evaluation

1. Introduction

Credit evaluation is inherently complex due to the heterogeneous nature of customers and various forms of risks (Piramuthu 1999). Feature selection plays an important role in credit evaluation by identifying the most important features, yet it tends to omit correlated ones. Feature grouping effect analysis captures the underlying correlation structure of the features, and, thus, the combination of feature selection and grouping effect analysis is beneficial in capturing important risks in credit evaluation (Maldonado et al. 2017). Most existing feature selection and grouping effect analysis methods were built on the regularizations of linear models, such as Logistic Regression (LR) and Linear Support Vector Machine (L-SVM). A common assumption behind these linear models is that the data from different credit classes are linearly separable. Nevertheless, the assumption cannot be held (Zhang et al. 2021), which will be further explained in Section 3.

This work concerns the following aspects of credit evaluation:

1. Feature selection matters in credit evaluation. The significance of feature selection has been well recognized by existing studies in credit evaluation (Piramuthu 1999, Hong et al. 2014). Typical feature selection methods in credit evaluation include Least Absolute Shrinkage and Selection Operator (LASSO) (Piramuthu 1999) and L_0-regularized models (Gómez and Prokopyev 2021, Zheng et al. 2022). A limitation of existing feature selection methods is that many were built on linear models. When these linear models are unsuitable for the applied problem, the feature selection results will be unreliable.

2. Awareness of correlated feature groups is important for capturing credit risks. Most feature selection models tend to select only one feature from a group of highly correlated features (Zou and Hastie 2005). Nevertheless, selecting only one and omitting the other correlated ones may neglect important credit risk sources (features). For instance, "education" and "income" are all important features in credit scoring, but they are highly correlated. Most feature selection models will select either education or income because the two features are correlated. However, the omitted feature can contribute to credit risk analysis because it is related to credit risk. Many studies have noticed this issue and selected the grouped features or omitted them simultaneously (Won et al. 2020). Yet, these models still suffer from relying on linear models, which are vulnerable to complex distributions.

3. In credit evaluation, the interpretability of the model is the first consideration (Basu and Naughton 2020). Although many high-performance nonlinear models have been proposed in the last two decades, such as kernel methods and neural networks, the majority of them required recombination of the original features, which made the models hard to interpret (Xiao et al. 2023). As a result, most financial institutions still rely on linear models, such as LR, L-SVM, and Linear Discriminant Analysis (LDA), to conduct credit risk analysis because the results of these linear models are interpretable (Hilscher and Wilson 2017).

Credit data are linearly inseparable, due to the inherent heterogeneity of credit customers and various risk types. For instance, people from different areas or countries have diverse preferences and exhibit different behaviors (Ferman 2016). Companies from different industries have varying characteristics (Basu and Naughton 2020). There are also many varying types of risk sources, such as soft factor risk and loan risk (Bhat et al. 2019). All these heterogeneities resulted in many subpatterns in credit data and caused linear inseparability. This was a well-recognized problem and made kernel tricks of Support Vector Machine (SVM) popular before the renaissance of deep learning (Won et al. 2020).

To summarize, linearly inseparable credit data need nonlinear models, whereas the uninterpretable results produced by nonlinear models are unacceptable in financial applications. To cope with this dilemma, we turn to Distance Metric Learning (DML), which does not pursue linear separability. Most traditional DML approaches have unreasonable time complexities, which make them inappropriate for credit valuation. However, given the recent technical progress of machine learning and optimization, we see the possibility of solving this problem.

This study proposes a DML-based feature selection and grouping effect analysis approach, which is free from the assumption of linear separability. The contributions of the study are two-fold: (1) This study formulates the feature selection and grouping effect analysis as an L_1 and ElasticNet regularized optimization problem and proves why such tasks can be solved by the optimization problem. (2) To solve the proposed model effectively, this study proposes a new solver based on the Alternating Direction Method of Multipliers (ADMM), which decomposes the regularized DML optimization into many parallel linear programming problems and a small quadratic consensus-reaching problem. The ADMM-based solver is substantially faster than traditional methods.

The rest of the study is organized as follows: Section 2 reviews related works. Section 3 presents the characteristics of the credit data. Section 4 introduces the proposed models. Section 5 conducts experiments and evaluates the results. Section 6 concludes the study. Online Appendix A contains the proof of Theorem 1, Online Appendix B contains the proof of Lemma 1, and Online Appendix C provides the performance evaluation of the ADMM solver.

2. Related Works

This work is closely related to three lines of research: feature selection and grouping effect analysis, traditional DML approaches, and their applications in credit evaluation.

2.1. Feature Selection and Grouping Effect Analysis with Linear Models

The mainstream research on feature selection and grouping effect analysis was accomplished by exerting regularizations on linear models. The basic theory behind the regularization methods is that the least-squares estimate often has a low bias, but a large variance. By adding regularization terms, it introduces bias, but shrinks the values of the coefficients, and the overall prediction accuracy can be improved. The following paragraphs go through the major feature selection and grouping effect analysis models.

2.1.1. The LASSO. The LASSO is a linear regression model that combines the least-squares loss with an L_1 regularization. It can be solved with the following optimization problem:

$$\min \frac{1}{2}\sum_{1}^{N}(y_i-\beta_0-\boldsymbol{x}_i^T\boldsymbol{\beta})^2+\lambda||\boldsymbol{\beta}||_1,$$

where each $\boldsymbol{x}_i \in \mathbb{R}^p$ represents a p-dimensional vector of features, and $y_i \in \mathbb{R}$ is the associated response variable (Tibshirani 1996). The first term in the optimization problem is the least-squares estimator, whereas the second term is a L_1 norm regularization. The key property of the LASSO is that it can result in a sparse solution, in which few coefficients are nonzero (Jiang et al. 2021). Zero coefficients indicate that the corresponding features take no effect in predicting the target/response variable (Shi et al. 2018).

2.1.2. The ElasticNet. The ElasticNet makes a compromise between the Ridge (L_2 regularization) and the LASSO. It solves the following optimization problem (Zou and Hastie 2005):

$$\min \frac{1}{2}\sum_{1}^{N}(y_i-\beta_0-\boldsymbol{x}_i^T\boldsymbol{\beta})^2+\lambda\left[\frac{1}{2}(1-\alpha)||\boldsymbol{\beta}||_2^2+\alpha||\boldsymbol{\beta}||_1\right],$$

where $\alpha \in [0,1]$ is a parameter that controls the compromise between the L_1 and L_2 terms. A prominent characteristic of the ElasticNet is that it selects within-group features together if the features are strongly correlated and poses similar coefficients to these features (Jiang et al. 2021). Thus, the ElasticNet is an ideal approach if we need to keep or eliminate the correlated features simultaneously.

2.1.3. The Group LASSO. The idea of this approach is that all coefficients within a group become nonzero (or zero). Suppose we have J groups of features, and $\boldsymbol{\theta}_j$ is the coefficient vector in group j, $\boldsymbol{\theta}_j \in \mathbb{R}^{pj}$. The group LASSO solves the following convex problem (Yuan and Lin 2006):

$$\min \frac{1}{2}\sum_{1}^{N}\left(y_i-\theta_0-\sum_{j=1}^{J}\boldsymbol{z}_{ij}^T\boldsymbol{\theta}_j\right)^2+\lambda\sum_{j=1}^{J}||\boldsymbol{\theta}_j||_2,$$

where $||\boldsymbol{\theta}_j||_2$ is the Euclidean norm of the parameter vector. The Euclidean norm term encourages either that the entire parameter vector θ_j be zero or all its elements be nonzero (Meier et al. 2008). This approach is also referred as $L_{2,1}$ regularization (Shi et al. 2018).

2.1.4. The Fused LASSO. The fused LASSO aims to learn sparse coefficients and make the consequent coefficients to be similar to each other. It solves the following problem (Tibshirani et al. 2005):

$$\min \frac{1}{2}\sum_{1}^{N}\left(y_i-\beta_0-\sum_{j=1}^{P}x_{ij}^T\beta_j\right)^2+\lambda_1\sum_{j=1}^{P}|\beta_j|+\lambda_2\sum_{j=2}^{P}|\beta_j-\beta_{j-1}|.$$

The first penalty shrinks the coefficient β_j toward zero, and the second penalty encourages neighboring coefficients β_j and β_{j-1} to be similar. The fused LASSO can achieve feature selection and grouping effects simultaneously (Petersen et al. 2016). Nevertheless, the prerequisite is that we need to order the features properly.

2.1.5. The L_0-Based Approaches. There is a direct way to conduct feature selection—that is, using L_0 norm, which is free from bias. The optimization problem is defined as follows (Hazimeh and Mazumder 2020):

$$\min \frac{1}{2}\sum_{1}^{N}(y_i-\beta_0-\boldsymbol{x}_i^T\boldsymbol{\beta}_j)^2,\ \text{s.t.}\ ||\boldsymbol{\beta}||_0 \le k,$$

where $\|\boldsymbol{\beta}\|_0$ is the L_0 norm—that is, the number of the nonzero entries in vector $\boldsymbol{\beta}$. It is an NP-hard problem. Recent progress on solving the above problem is to use approximation methods, such as convex semi-infinite programming (Won et al. 2020), primal dual active sets (Zheng et al. 2022), fractional mixed-integer optimization (Gómez and Prokopyev 2021), and the ConCave-Convex Procedure (Shi et al. 2018). Although L_0 is free from bias, its solving methods introduce approximation errors and are more computationally inefficient than L_1 (Shi et al. 2018).

2.1.6. Applications of the Above Regularizations to Generalized Linear Models. The regularization methods previously introduced are all applicable to other generalized linear models, such as LR and L-SVM (Won et al. 2020). The characteristics and disadvantages of the above methods are summarized in Table 1. Given the volume of the credit data, we argue that ElasticNet is suitable to conduct feature selection and grouping effect analysis in credit evaluation.

2.2. Distance Metric Learning

DML is a technique for calibrating distance computation. The basic idea of DML is to make similar data closer and dissimilar data farther apart in the space.

2.2.1. Basic Theory of DML. Most existing DML models utilize a triplet to encapsulate a data point such that an optimization problem can be formulated easily (Weinberger and Saul 2009). A triplet is defined as $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$, in which $\boldsymbol{x}_j$ is a data point, $\boldsymbol{x}_i$ is the closest data point to $\boldsymbol{x}_j$ with the same label, and $\boldsymbol{x}_k$ is the closest data point to $\boldsymbol{x}_j$ with a different label. Let matrix $A \in \mathbb{R}^{m\times m}$ denote a distance metric, and a distance function in DML can be presented as follows (Xing et al. 2002):

$$d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T A(\boldsymbol{x}_i - \boldsymbol{x}_j)},$$

where A is a Positive Semi-Definite (PSD) matrix. The objective of DML is to learn a distance metric A that minimizes the summed $d_A(\boldsymbol{x}_i, \boldsymbol{x}_j)$ or maximizes the summed $d_A(\boldsymbol{x}_j, \boldsymbol{x}_k)$ (Cakir et al. 2019). The DML problem can also be formulated as an equivalent linear space transformation:

$$d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T A(\boldsymbol{x}_i - \boldsymbol{x}_j)} = \sqrt{(P\boldsymbol{x}_i - P\boldsymbol{x}_j)^T (P\boldsymbol{x}_i - P\boldsymbol{x}_j)}.$$

With a projection matrix P, a transformed space can be obtained via $\boldsymbol{x} \to P\boldsymbol{x}$, where $P^T P = A$. Generally, P can be solved via Cholesky decomposition or Eigen decomposition toward A. Besides, the linear space transformation with P can also be generalized as a nonlinear space transformation: $\boldsymbol{x} \to \sigma(\boldsymbol{x})$ (Cakir et al. 2019).

The general task of DML is to learn the parameter matrix A, the linear transformation P, or the nonlinear transformation $\sigma(\cdot)$. There are many well-established DML models, and the following subsections review two popular types.

2.2.2. Full-Matrix-Based DML. The first matrix-based DML model was proposed by Xing et al. (2002), which formulated the DML as an optimization problem:

$$\min_A \sum_{(x_i, x_j)\in S} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_A^2, \text{ s.t. } \sum_{(x_j, x_k)\in D} \|\boldsymbol{x}_j - \boldsymbol{x}_k\|_A^2 \geq 1, A \geq 0,$$

where $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in S$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the same class and $(\boldsymbol{x}_j, \boldsymbol{x}_k) \in D$ if $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are in different classes. $A \geq 0$ is a positive definite matrix constraint, and the constraint ensures that A does not collapse the data set into a single point and avoids negative distances.

Table 1. Characteristics and Disadvantages of the Regularization Methods

Model	Characteristic	Disadvantage
LASSO	Sparse coefficient; feature selection.	Only keep one feature in a correlated feature group.
ElasticNet	Grouping effect; feature selection.	\
Group LASSO	Grouping effect; select all or none of the features in a group.	Need to know the groups of the features in advance.
Fused LASSO	Feature selection; consequent features have similar coefficients.	Need to present the features in an appropriate order.
L_0	Feature selection; no selection bias.	NP-hard problem to solve.

Another popular DML approach is "Large Margin Nearest Neighbors" (LMNN) (Weinberger and Saul 2009). The optimization problem was formulated as follows:

$$\begin{aligned}&\min(1-v)\sum(\boldsymbol{x}_i-\boldsymbol{x}_j)^T A(\boldsymbol{x}_i-\boldsymbol{x}_j)+v\sum(1-y)\xi_{ijk},\\ &\text{s.t.}\quad (\boldsymbol{x}_j-\boldsymbol{x}_k)^T A(\boldsymbol{x}_j-\boldsymbol{x}_k)-(\boldsymbol{x}_i-\boldsymbol{x}_j)^T A(\boldsymbol{x}_i-\boldsymbol{x}_j)\geq 1-\xi_{ijk},\xi_{ijk},A\geq 0,\end{aligned}$$

where $v\in[0,1]$ is a trade-off parameter. It is a Semidefinite Programming (SDP) problem. Its advantage is that it learns a distance metric using local pairwise constraints, which makes it suitable for local models like k-nearest neighbors (k-NN) and k-Means. The limitation is that it is computationally expensive because it keeps a PSD projection to the PSD core at each step of the gradient descent procedure (Der and Saul 2012).

Other famous matrix-based DML approaches include DML-eig (Ying and Li 2012), Information-Theoretic Metric Learning (ITML) (Davis et al. 2007), and DMLMJ (Nguyen et al. 2017). The merit of DML-eig is that it only computes the largest eigenvector. ITML introduced LogDet divergence regularization into DML, and its key feature is that the optimization process provides an automatic and computationally cheap way to preserve the PSD A (Davis et al. 2007). DMLMJ used Jeffrey divergence as the loss function (Nguyen et al. 2017), which is free from intensive matrix decomposition (Nguyen et al. 2017).

A common limitation of the matrix-based DML approaches is that they all keep a PSD matrix A, which results in a time-consuming SDP problem (Xing et al. 2002). Thus, DML techniques were rarely applied to large-scale data sets. Besides, the full-matrix DML is equivalent to a linear transformation and causes the transformed features to be hardly interpretable. Thus, the applied value of the full-matrix DML approaches in credit evaluation is limited.

2.2.3. Diagonal-Matrix-Based DML. To avoid SDP, our former research decomposed the DML as a two-stages learning process, in which the first stage learned a full matrix using unsupervised method to represent features well, and the second stage learned a diagonal matrix to rescale the features to accomplish DML (Li et al. 2021). Suppose there is a diagonal matrix P_D, $P_D\in\mathbb{R}^{m\times m}$. The diagonal transformation $X\rightarrow P_D X$ can be denoted as follows:

$$\boldsymbol{y}_i=P_D\boldsymbol{x}_i=\begin{bmatrix}p_1&&&\\&p_2&&\\&&\dots&\\&&&p_m\end{bmatrix}\begin{bmatrix}x_{i1}\\x_{i2}\\\dots\\x_{im}\end{bmatrix}=\begin{bmatrix}y_{i1}\\y_{i2}\\\dots\\y_{im}\end{bmatrix},$$

where p_i is a diagonal element in P_D. This way, the diagonal DML matrix will only have a scaling effect on the data set—assigning weights to the original features without altering the original meanings of the features. Then, the diagonal DML matrix P_D can be solved via the following problem (Li et al. 2021):

$$\begin{aligned}&\min_{P_D}\sum_{a=1}^{t}\|P_D\boldsymbol{x}_a-P_D\boldsymbol{x}_b\|_2^2+\mu\sum\xi_j,\\ &\quad\text{s.t. }\|P_D\boldsymbol{x}_a-P_D\boldsymbol{x}_c\|_2^2-\|P_D\boldsymbol{x}_a-P_D\boldsymbol{x}_b\|_2^2\geq\boldsymbol{\tau}-\xi_j,\\ &\qquad\xi_j\geq 0,P_D=diag(\sqrt{p_1},\sqrt{p_2},\dots,\sqrt{p_m}),p_1,p_2,\dots,p_m\geq 0,\end{aligned}\tag{1}$$

where ξ_j are nonnegative slack variables, and t is the number of triplets. The above problem can be rewritten as a standard linear programming problem:

$$\min_{W}\boldsymbol{C}^T\boldsymbol{W},\ \text{ s.t. }\ \boldsymbol{FW}\geq\boldsymbol{\tau},\ w_i\geq 0,\tag{2}$$

where $\boldsymbol{W}$ is a vector containing variables, and w_i denotes the element in $\boldsymbol{W}$. $\boldsymbol{W}$ consists of two types of variables: the square of the elements in P_D and the slack variables in Equation (1). The inner structures of $\boldsymbol{W}$, $\boldsymbol{C}$, and F are shown as follows:

$$\begin{aligned}\boldsymbol{W}&=[\,p_1^2\quad\dots\quad p_m^2\quad\xi_1\quad\dots\quad\xi_t\,],\\ \boldsymbol{C}&=[\,c_1\quad\dots\quad c_m\quad\mu_1\quad\dots\quad\mu_1\,],\\ F&=\begin{bmatrix}f_{11}&\dots&f_{1m}&\xi_1&&\\ \dots&\dots&\dots&&\dots&\\ f_{t1}&\dots&f_{tm}&&&\xi_t\end{bmatrix}=[\,\hat{F}_{t\times m}\quad\xi_t\,].\end{aligned}$$

We found that the diagonal DML proposed by us is potentially an ideal form for feature selection because it does not recombine the features and grantees interpretability. But it has not been used specially for such a task, and its mathematical properties remain unstudied as well. The major differences between this study and our former research (Li et al. 2021) include: (1) This study proved mathematically why the diagonal DML model can be used for feature selection and grouping effects. By contrast, the contribution of Li et al. (2021) was to approximate a full DML matrix with a diagonal matrix multiplied by an orthogonal matrix, such that the optimization problem can be solved more rapidly than full-matrix-based SDP. (2) This study proposed an ADMM-based parallel solver to overcome the large-scale constraints problem of DML, which has been a long-standing computational difficulty of DML.

2.2.4. Neural Network-Based DML. A neural network could exert arbitrary nonlinear transformations $\sigma(\cdot)$ toward the data sets. For instance, Hoffer and Ailon (2015) proposed a neural network called TripletNet, whose loss function was defined as "Triplet Loss":

$$L_{Triplet} = \max(0, \|G_W(X) - G_W(X^S)\|_2 - \|G_W(X) - G_W(X^D)\|_2 + \alpha),$$

where α is the margin value, G_W is the linear or nonlinear transformation with neural network, X is an inputted data point, X^S is a data point similar to the inputted X, and X^D is a dissimilar data point to the inputted X. Although neural-network-based approaches have better performance in most cases, they recombine original features, which make the interpretability of the transformed features even harder than the matrix-based ones.

2.3. Feature Selection and Grouping Effect Analysis in Finance and Related Domains

Many works studied feature selection in financial and other business applications. Piramuthu (1999) used LASSO to conduct feature selection for credit risk evaluation. Han et al. (2016) developed information collection algorithms to learn effective factors for nonprofit fundraising. Keshanian et al. (2022) proposed a feature selection approach based on Nash-Bargaining and second-order cone programming to solve the practical feature selection problems in online advertising and information systems. Zhang et al. (2021) proposed a review selection method for finding informative subset samples from online reviews based on a heuristic method. Gómez and Prokopyev (2021) proposed a subset selection approach based on linear regression that involved solving a sequence of mixed-integer quadratic optimization problems.

Feature group effect refers to the interactions of the features, which are determined by the inherent group structures of the features (Jiang et al. 2021). Feature selection models tend to select only one feature from a group of highly correlated features (Yuan and Lin 2006). As a result, many correlated credit risk features, which contribute equally to credit risks, may be omitted. However, this phenomenon has not been well recognized. Feature grouping effect analysis is still underdeveloped and needs to be further studied in credit evaluation. Maldonado et al. (2017) incorporated a group penalty function in the SVM formulation to penalize the variables that belong to the same group. Cui et al. (2021) developed a multiple structural interacting elastic net model for feature selection.

3. The Data Set and the Statistical Evidence of Linear Inseparability

This section describes the credit data set used in the experiment and analyzes the possible reasons for the linear inseparability of credit data.

3.1. The Credit Data Set Used in This Study

We used a real-life loan credit data set from a Chinese bank. The data set had about 96,000 instances, and the details of the features are summarized in Table 2.

After transforming the nominal features into numeric ones using the one-hot encoding method (Goodfellow et al. 2016), there were 59 features.

3.2. Statistical Evidence and Possible Reasons of Linear Inseparability

Credit evaluation can be considered a binary classification problem. To make a data set linearly separable, the ideal scenario is that data from each class follow a distinct multivariate Gaussian distribution, and the two Gaussian hyperellipsoids can be separated by a hyperplane. Nevertheless, real-life credit data hardly follow such distributions. Credit data usually have many subpatterns, and data points from different credit classes are intertwined in local areas.

We argue that the inherent heterogeneity of customers is one of the reasons why credit data are linearly inseparable and have multiple subpatterns. For instance, in our credit evaluation data, we found that the credit patterns of customers from Beijing and Gansu province were different. As a result, we developed different linear

Table 2. The Details of the Features

Type	Feature	Value range	Description
Numeric	Age	14–86	The age of the customer
	Saving deposits	7E3–2.4E6	The saving deposits of the customer
	Monthly income	300–15,000	The monthly income of the customer
	Deposit account num.	0–163	The number of deposit accounts
	Credit card num.	0–140	The number of credit cards
	Overdue interest	1–6.0	Overdue interest on the credit card
	Loan account num.	0–95	The number of loan accounts
	Longest delayed days	0–65	Longest days delayed from the repayment date
	Delayed repayment times	0–99	Times of delay for repayment
	Credit limit change rate	−6–38	The rate of change in credit card limit
	Credit card usage times	0–2,500	Usage times of the credit card
	Outstanding debt	0–5,000	The remaining debt to be paid
	Equated Monthly Instalment (EMI) debt	0–80,000	The remaining EMI debt to be paid
	Credit utilization rate	20–50	The utilization rate of credit card
	Account age	1–450	The age of the account
	Investment deposits	0–1,800	The investment deposits of the customer
	Average monthly balance	0–1,600	The average monthly balance of the customer
Nominal	Overdue month	12 types	The most frequent overdue month
	Occupation	16 types	The occupation of the customer
	Gender	4 types	The gender of the customer
	If minimal payment	3 types	If paid the minimal amount
	Payment type	7 types	The frequently used payment type

models for the customers from the two places. But afterward, we realized that, in addition to geographical location, other factors (such as gender, education level, and consumption time) can also cause heterogeneity and, hence, subpatterns in the data as well.

To validate the heterogeneity, we conducted an empirical investigation on how the categorical values of nominal features affected the distribution of numeric features. Figure 1 shows the distributions of six typical nominal and numeric features.

It can be seen from Figure 1 that the data in the same numeric feature were divided into different groups, according to the categorical values of another nominal feature, and the distributions of the grouped data in the same numeric feature varied. This phenomenon implied that there were many heterogeneous subpatterns in these features, and the intertwining of these subpatterns can result in linear inseparability. The heterogeneity of customers was also observed in other applications, such as personalized search engines (Yoganarasimhan 2020) and nonmortgage fintech lending (Kelley et al. 2022).

We conducted an analysis of variance (ANOVA) on all the pairwise combinations of the nominal and numeric features. The results showed that 26 numeric features had significant heterogeneous subpatterns under the groups divided by the values of the nominal features. The three pairs corresponding to Figure 1 were recorded in Table 3.

It can be seen from Table 3 that the *F* values were substantially larger than the critical values (95% confidence), indicating that the heterogeneity among the data distributions under different groups was statistically significant, which was also reflected by the *p*-values.

Figure 1. (Color online) The Comparisons of the Distributions of the Grouped Values in Some Numeric Features

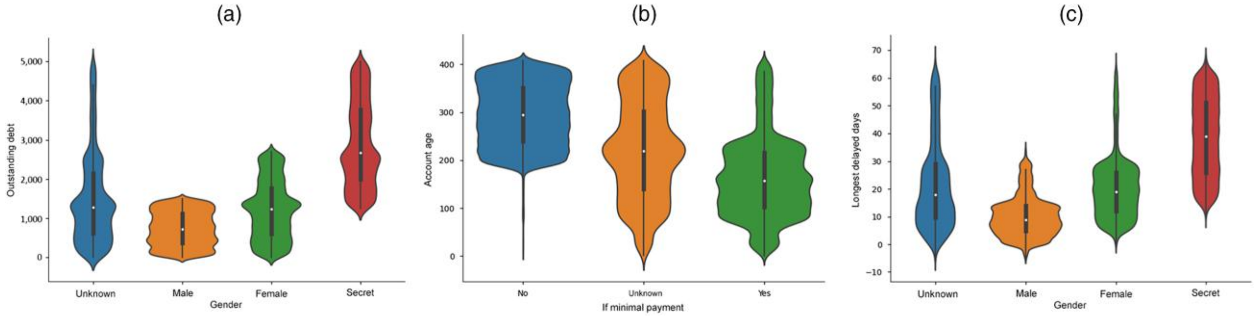


Table 3. The ANOVA Results of Six Pairwise Features (Grouping According to the Values of the Nominal Features)

Nominal feature	Numeric feature	*F* value	Critical value	*p*-value
Gender	Outstanding debt	27,165	2.605	<1.0E-6
If minimal payment	Account age	27,052.45	2.9958	<1.0E-6
Gender	Longest delayed days	22,310.79	2.605	<1.0E-6

In credit evaluation practice, heterogeneity is ubiquitous, such as different repaying habits, educational backgrounds, preferences, and consuming behaviors. Another reason for the linear inseparability of credit data is their social nature. They are often influenced by multiple complex latent factors and are easily affected by the constantly evolving external social environments. Furthermore, defaulters usually take countermeasures based on the antidefault actions of financial institutions.

4. The Proposed Feature Selection and Grouping Effect Analysis Models

This section introduces our ideas, models, and the algorithm.

4.1. Problem Formulation

Based on the analysis of the characteristics of the credit evaluation in Section 3 and the limitations of the existing linear models in Section 2.1, we need a new feature selection model that can handle linearly inseparable credit data. The model also needs to take into consideration the correlated features that indicate credit risks equally. Inspired by the potentials of the diagonal DML matrix in Li et al. (2021), we propose to conduct feature selection and grouping effect analysis by adding an ElasticNet regularization—that is, a combination of the L_1(also known as (a.k.a.) LASSO) and L_2(a.k.a. Ridge) regularization terms—to the objective function of Equation (2) as follows:

$$\min \boldsymbol{C}^T\boldsymbol{W} + \lambda[\alpha\|\boldsymbol{W}\|_1 + (1-\alpha)\|\boldsymbol{W}\|_2^2],\ \text{s.t. } \boldsymbol{FW} \geq \boldsymbol{\tau},\ w_i \geq 0, \tag{3}$$

where α is the trade-off coefficient between L_1 and L_2, $\alpha \in [0,1]$, and λ denotes the regularization coefficient ($\lambda \geq 0$). The scalar vector $\boldsymbol{\tau}$ on the right side is arbitrary because all elements in P_D can be scaled up and down with identical ratios with no impacts on applications such as k-NN. However, the specific value of $\boldsymbol{\tau}$ can have an influence on Lipschitz continuity and then the convergence speed of gradient-descent solvers. Further discussion on this issue is out of the scope of the study.

Now, two problems arise from the above equation:

Problem 1. How to verify that the model in Equation (3) can be used to conduct feature selection while keeping the correlated features such that no important credit risks are neglected and prove mathematically why the model can accomplish such an effect, given that financial applications need a very solid and reliable theory background.

Problem 2. How to solve the ElasticNet regularized optimization problem efficiently. Originally, the diagonal DML problem could be solved efficiently with linear programming. However, when a quadratic term is added to the objective function, traditionally, we need to resort to gradient-descent-based solvers, which are very inefficient with abundant slack variables generated by the constraints in Equation (3).

We aim to handle the two problems in the following sections.

4.2. Grouping Effect Analysis and Feature Selection via Regularizations

This section explains how to conduct grouping effect analysis and feature selection via regularizing the diagonal DML model.

4.2.1. The ElasticNet Regularized Diagonal DML as an Unconstrained Problem. To handle the constraints, we can add an punishment function $\phi(W)$ to the objective function and form a new unconstrained problem:

$$L_{\alpha,\beta,\lambda}(W) = \boldsymbol{C}^T\boldsymbol{W} + \phi(\boldsymbol{W}) + \lambda[\alpha\|\boldsymbol{W}\|_1 + (1-\alpha)\|\boldsymbol{W}\|_2^2]. \tag{4}$$

The solution to the above problem will be analyzed in Section 4.3.

4.2.2. The Basic Theory of the Grouping Effect. We propose that the grouping effect of two features can be measured by the difference of their coefficients, as defined in Theorem 1.

Theorem 1. *Given that all the features are standardized or normalized, the grouping effect of two features i and j can be measured as follows:*

$$|\hat{w}_i - \hat{w}_j| = \left|\frac{(c_i - c_j) + (\boldsymbol{F}_i - \boldsymbol{F}_j)^T \boldsymbol{\beta}}{2\lambda(1-\alpha)}\right|, \quad \textit{if} \quad \alpha \neq 1, \tag{5}$$

where w_i and w_j are the coefficients of the two features, $\boldsymbol{\beta}$ are the Lagrangian coefficients, and $\alpha \neq 1$.

The proof of the above equation can be found in Online Appendix A.

Lemma 1. *Suppose the correlation coefficient of the features i and j is $\sigma_{i,j}$. If feature i and j are highly correlated ($\sigma_{i,j} \rightarrow 1$), the coefficients of the two features in the ElasticNet regularized DML model tend to be equal, and the two features have a grouping effect:*

$$\lim_{\sigma_{i,j}\rightarrow 1} |\hat{w}_i - \hat{w}_j| = \lim_{\sigma_{i,j}\rightarrow 1}\left|\frac{(c_i - c_j) + (\boldsymbol{F}_i - \boldsymbol{F}_j)^T \boldsymbol{\beta}}{2\lambda(1-\alpha)}\right| = 0, \quad \textit{if} \quad \alpha \neq 1. \tag{6}$$

The proof of the above equation can be found in Online Appendix B.

Lemma 2. *If only L_1 regularization is used, the diagonal DML model does not have a feature grouping effect.*

We can see that if only L_1 regularization is used—that is, $\alpha = 1$—Theorem 1 does not hold. From the proof of the theorem, we know that $\frac{\partial L}{\partial w_i} = c_i + \boldsymbol{F}_i^T \boldsymbol{\beta} + \lambda\alpha 2\lambda(1-\alpha)\hat{w}_i = 0$. If $\alpha = 1$, the equation will be irrelevant to the coefficient of the feature i—that is, $\hat{w}_i$. Therefore, theoretically, we cannot guarantee any feature grouping effect with L_1 regularization.

4.2.3. Quantitative Evaluation of the Grouping Effect.

From Equation (6), we know that the coefficients of highly correlated features tend to be similar in values. However, the coefficients of some features can be similar coincidently, and the grouping effect analysis may not be robust. To eliminate such coincidences, we need to conduct several rounds of learning with different combinations of α and λ. In general, a grid search of the optimal hyperparameters α and λ will generate such combinations.

We see that the coefficient difference $|\hat{w}_i - \hat{w}_j|$ can be influenced by the values of α and λ. Suppose we change the hyperparameter α and λ from α^0 and λ^0 to α^1 and λ^1; the influence to the coefficient difference can be evaluated as follows:

$$\frac{|\hat{w}_i{}^1 - \hat{w}_j{}^1|}{|\hat{w}_i{}^0 - \hat{w}_j{}^0|} = \left|\frac{(c_i - c_j) + (\boldsymbol{F}_i - \boldsymbol{F}_j)^T \boldsymbol{\beta}^1}{2\lambda^1(1-\alpha^1)}\right| \cdot \left|\frac{2\lambda^0(1-\alpha^0)}{(c_i - c_j) + (\boldsymbol{F}_i - \boldsymbol{F}_j)^T \boldsymbol{\beta}^0}\right| \approx \frac{\lambda^0(1-\alpha^0)}{\lambda^1(1-\alpha^1)}. \tag{7}$$

To calculate more accurately the group effects using different combinations of λ and α, we need to mitigate the influence of different λ and α to $|\hat{w}_i - \hat{w}_j|$ first. Given a pair of initial values of λ and α—that is, λ^0 and α^0—we propose the following correction factor to amend the coefficient difference in grid search: $\alpha^{s\rightarrow 0} = [\lambda^s(1-\alpha^s)]/[\lambda^0(1-\alpha^0)]$, such that $|w_i^0 - w_j^0| = \alpha^{s\rightarrow 0} \cdot |w_i^s - w_j^s|$, where s denotes sth step of grid search. Then, the average difference between w_i and w_j in the grid search can be calculated as: $\overline{diff}_{i,j} = \frac{1}{t}\sum_{s=1}^{t} \alpha^{s\rightarrow 0} \cdot |w_i^s - w_j^s|$.

4.2.4. Feature Selection with the Regularized Diagonal DML.

As put formerly, feature selection can be viewed as an L_0 optimization problem:

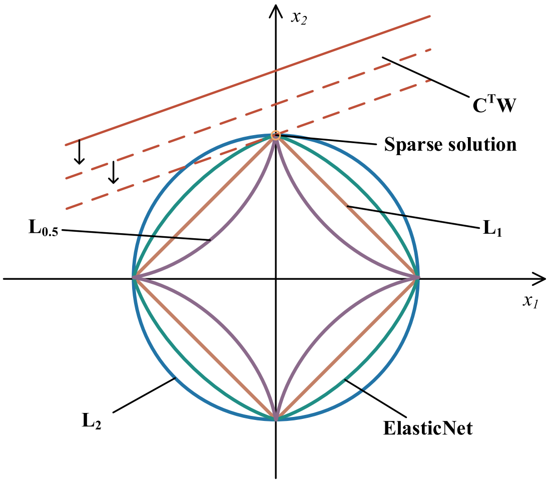
$$\min f(\boldsymbol{W}), \text{ s.t. } \|\boldsymbol{W}\|_0 \leq t_0, \tag{8}$$

where t_0 is a positive integer. However, $\|\boldsymbol{W}\| \leq t_0$ is a nonconvex constraint region, and it is an NP-hard problem in optimization. Theoretically, any L_q constraint where $0 \leq q \leq 1$ has an effect of feature selection. As shown in Figure 2, when $0 \leq q \leq 1$, the constraint region has sharp corners on the coordinate axes, and the objective function is very likely to be tangent to the constraint region at the corners, resulting in a sparse solution—that is, many zero coefficients. It is known that the only convex region in $L_q(0 \leq q \leq 1)$ is L_1—that is, the best convex approximation of the L_0 constraint is the L_1 constraint. The L_1 regularized $\lambda\alpha\|W\|_1$ term can be viewed as a Lagrangian term, and it is equivalent to the following optimization problem:

$$\min \boldsymbol{C}^T\boldsymbol{W} + \phi(\boldsymbol{W}) + \lambda(1-\alpha)\|\boldsymbol{W}\|_2^2, \text{ s.t. } \|\boldsymbol{W}\|_1 \leq t_1. \tag{9}$$

By Lagrangian duality, there is a one-to-one correspondence between the constrained problem in Equation (9) and the Lagrangian Form Equation (4). Depending on whether $\alpha = 1$, we have two conditions in feature selection.

Figure 2. (Color online) The Objective Function and the Constraint Regions



Condition 1. $\alpha = 1$. *In this condition, the L_2 regularization term is eliminated from the objective function. According to Lemma 2, the selected features have no grouping effect. When heavily punished by the L_1 term, the model tends to select as fewer features as possible. If some features are highly correlated, the model selects only one of them (Piramuthu 1999, Zou and Hastie 2005, Yuan and Lin 2006).*

Condition 2. $0 < \alpha < 1$. *In this condition, both the L_1 and the L_2 regularization terms are kept (ElasticNet regularization). Because of the L_1 term, the model still has a feature selection effect. According to Lemma 2, if $\alpha \neq 1$, the model also has a grouping effect. That is to say, if some features are highly correlated, the model will select all of them, leaving the coefficients of the correlated features almost the same.*

We consider that the properties of the model in Condition 2 are beneficial to credit evaluation. It is very important to capture all the risk sources that may cause damage to financial systems. Using only the L_1 term in Condition 1, we probably neglect the important risk sources that are highly correlated to other risk sources.

4.2.5. Feature Group Detection.

Based on the above analysis, we outline two rules to detect feature groups: (a) The coefficients of the highly correlated variables are almost identical to each other in the regularization path. We can use $\overline{diff}_{i,j}$ to evaluate if two features have a grouping effect. (b) The features kept by the L_1 regularization ($\alpha = 1$) are hardly possible to be correlated. Thus, we can remove the pairwise combinations of the kept features from grouping effect analysis. We outline the above rules in Algorithm 1 to detect feature groups.

Algorithm 1 (Feature Group Detection)

```
Input: A data set $< X, L >$
Output: Feature groups
1. Calculate the ElasticNet regularization paths using Equation (4) with different regularization coefficients $\lambda$ and $\alpha$ (including $\alpha = 1$).
2. Build a Pairwise Comparison Matrix (PCM) whose element $s_{i,j}$ denotes the similarity of the pairwise features.
3. Set the values of the entries in PCM to $s_{i,j} = 1 - \overline{diff}_{i,j}$, where $i, j = 1, 2, .., m$ and $i \neq j$.
4. Set the values of the entries corresponding to the pairwise features kept by regularization ($\alpha = 1$) to 0.
5. Adjust the element $s_{i,j}$ in PCM. For $\forall s_{i,j}$ in PCM, $\hat{s}_{i,j} = \begin{cases} 1, s_{i,j} \geq 1 - \epsilon \\ 0, i = j \\ \epsilon, s_{i,j} < 1 - \epsilon \end{cases}$ ; //$\epsilon$ is a small value such as 0.05, and $\epsilon > 0$.
6. Use the adjusted PCM as the adjacency matrix of a graph $G$, whose nodes can be considered as features, and cluster the nodes using community detection algorithms (such as FastUnfolding) or spectral clustering.
7. Return the detected clusters $C = \{c_1, c_2, \ldots, c_g\}$ as feature groups and interpret the features in each cluster $c_i$.
```

4.3. The Solution to the Proposed Model

This section handles the nondifferentiable issue of the L_1 term and presents a parallel solver for the aforementioned optimization problems based on ADMM.

4.3.1. The Approximation of the L_1 Regularization Term. Because the L_1 term is not differentiable at $w_i = 0$, we propose to use a proximal function $p(w_i)$ to approximate the absolute value of w_i in L_1 term near $w_i = 0$. $p(w_i)$ is defined as: $p(w_i) = \frac{w_i^2}{2c} + \frac{c}{2}$, where c is a constant and $c > 0$. It is easy to deduce that $y = |w_i|$ and $p(w_i)$ are tangent at $\pm c$, as shown in Figure 3.

The absolute value of each variable in the L_1 term can be approximated as:

$$f_1(w_i) = \begin{cases} w_i^2/2c + c/2, \ -c < w_i < c \\ w_i, w_i \geq c \\ -w_i, w_i \leq -c. \end{cases}$$

$f_1(w_i)$ is differentiable at all points. The upper error bound of the approximation can be solved with the following optimization problem: $\max \frac{w_i^2}{2c} + \frac{c}{2} - w_i$, s.t. $0 \leq w_i \leq c$. The solution is $w_i = 0$, and the maximal value is $c/2$—that is, the upper error bound is: $f_1(w_i) - |w_i| \leq c/2$. Suppose the dimension of the data is m; then, the total upper error bound is $cm/2$. As long as the value of c is small enough, the approximation error is trivial. With the above approximation to the L_1 term, the whole loss function becomes:

$$L_{\alpha,\beta,\lambda}(\boldsymbol{W}) = \boldsymbol{C}^T\boldsymbol{W} + \phi(\boldsymbol{W}) + \lambda[\alpha f_1(\boldsymbol{W}) + (1-\alpha)\|\boldsymbol{W}\|_2^2]. \tag{10}$$

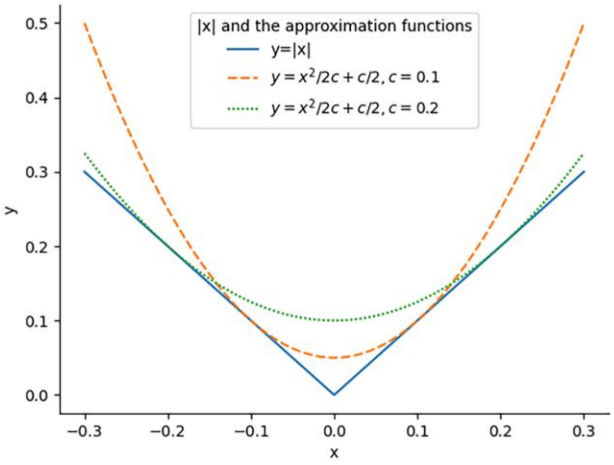
4.3.2. Parallel Computation as a Consensus Problem. Because every data point generates a constraint along with a slack variable in Equation (1), the optimization of DML with credit data usually concerns a large number of variables, and the solving of the optimization problem can be very slow using traditional gradient-descent methods. This section aims to decompose the optimization problem into smaller linear programming problems, which can be solved efficiently and parallelly, and avoid large-scale gradient-descent optimization. Suppose we can separate the credit data into several blocks and conduct a DML on each block; then, a challenge is how to coordinate the parameters learned from different blocks. As shown in Equation (10), the objective function of our unconstrained optimization problem contains two types of terms:

a. The first type is the ones that can be calculated separately in each block, including the distance between two nearest instances of the same label and the punishment that keeps the distance of each instance to its nearest neighbor of different label greater than the distance to its nearest neighbor of the same label. Suppose that we divide the instances into N blocks, and there are N_t instances in the tth block. This type of terms from Equation (10) can be formulated as follows:

$$L_t(W_t) = \boldsymbol{C}^T\boldsymbol{W}_t + \phi(\boldsymbol{W}_t), \tag{11}$$

where $\boldsymbol{W}_t$ means the parameters learned from the tth block. The above problem is equivalent to Equation (1) and Equation (2), which results in a linear programming problem. Thanks to the sparse structure of the coefficients, Problem (2) can be solved efficiently using tools such as HIGHS, regardless of abundant slack variables generated by the constraints.

Figure 3. (Color online) The L_1 Regularization and the Approximation Functions



b. The second type is the ones that cannot be calculated separately on each block, mainly referring to the ElasticNet regularization. This type can be formulated as follows:

$$g(\boldsymbol{Z}) = \lambda[\alpha f_1(\boldsymbol{Z}) + (1-\alpha)\|\boldsymbol{Z}\|_2^2]. \tag{12}$$

Then, the consensus problem can be formulated as the following optimization problem:

$$\min \sum_{t=1}^{N} L_t(\boldsymbol{W}_t) + g(\boldsymbol{Z}), \text{ s.t. } \boldsymbol{W}_{tm} - \boldsymbol{Z} = \boldsymbol{0},\ t = 1, \dots, N, \tag{13}$$

where $\boldsymbol{W}_{mt}$ represents the main variables excluding slack variables generated by the constraints in each block, and $\boldsymbol{Z}$ is the global variable, representing a consensus of $\boldsymbol{W}_{mt}$ from different blocks. Now, the abundant slack variables generated by the constraints in Equation (2) only exist in each L_t and are not involved in Equation (12).

The ADMM problem can be written as the augmented Lagrangian (Boyd et al. 2011):

$$L_\rho(\boldsymbol{W}_{t1}, \dots, \boldsymbol{W}_{tN}, \boldsymbol{Z}, \boldsymbol{y}) = \sum_{t=1}^{N} L_t(\boldsymbol{W}_t) + y_t^T(\boldsymbol{W}_{mt} - \boldsymbol{Z}) + \frac{\rho}{2}\|\boldsymbol{W}_{mt} - \boldsymbol{Z}\|_2^2, \tag{14}$$

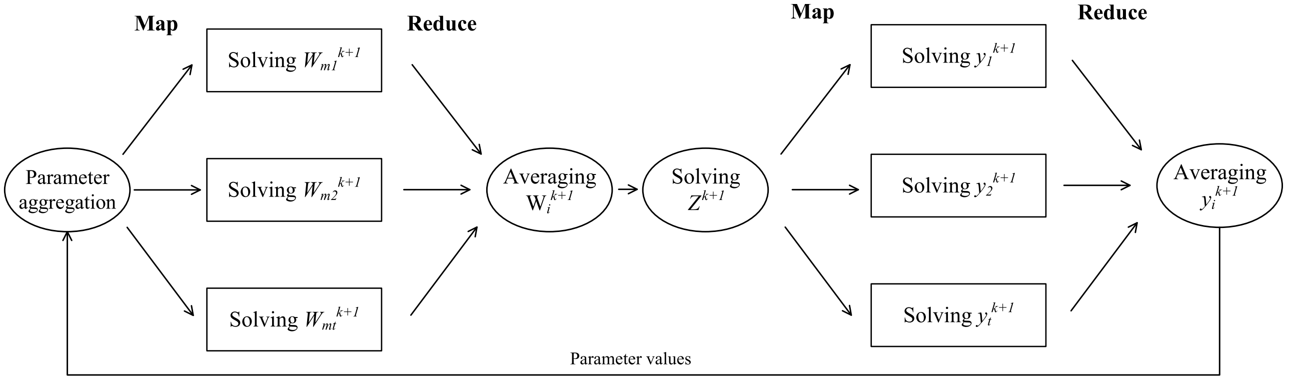
and $L_\rho(\boldsymbol{W}_{t1}, \dots, \boldsymbol{W}_{tN}, \boldsymbol{Z}, \boldsymbol{y})$ can be solved by the following ADMM updates:

$$\begin{cases} \boldsymbol{W}_{mt}^{k+1} := \arg\min\limits_{W_{mt}}(L_t(\boldsymbol{W}_t) + \rho\|\boldsymbol{W}_{mt} - \boldsymbol{Z}^k + \boldsymbol{y}_t^k\|_1) \\ \boldsymbol{Z}^{k+1} := \arg\min\limits_{Z}(g(\boldsymbol{Z}) + (N\rho/2)\|\boldsymbol{Z} - \overline{\boldsymbol{W}_{mt}}^{k+1} - \overline{\boldsymbol{y}_t}^{k}\|_2^2) \\ \boldsymbol{y}_t^{k+1} := \boldsymbol{y}_t^k + \boldsymbol{W}_{mt}^{k+1} - \boldsymbol{Z}^{k+1}, \end{cases} \tag{15}$$

where $\overline{\boldsymbol{W}_{mt}}$ and $\overline{\boldsymbol{y}}$ denote the average of the N parallel blocks. During ADMM iterations, the punishment terms in Equation (15) force W_{mt} from different parallel blocks to be similar to each other, and the original ElasticNet regularization—that is, $g(\boldsymbol{Z})$—can be realized simultaneously in the consensus-reaching process. Specially, suppose that $\sum_{i=1}^{m} \rho|w_i - zy_i| = \rho\|\boldsymbol{W}_{mt} - \boldsymbol{Z}^k + \boldsymbol{y}_t^k\|_1$), which is the punishment term in the first function; then, the punishment is equivalent to a constraint $|w_i - zy_i| \le \tau_2/\rho$, where τ_2 is an arbitrary positive constant and its value has no effect on DML because τ_2/ρ becomes close to zero as ρ increases in iterations. Thus, the first line in Equation (15) is still a linear programming problem with $2m$ more linear constraints compared with Equation (2) and can be solved efficiently. As for the second line in Equation (15), it is a quadratic problem with a small number of variables (only 59 variables corresponding to the weights of the features in this study) and can be solved easily with the quasi-Newton method. The computation logic is graphically outlined in Figure 4.

As shown in Figure 4, the update of $\boldsymbol{W}_t^{k+1}$ and $\boldsymbol{y}_t^{k+1}$ is an ideal scenario for distributed computation models, such as Fork-Join and Map-Reduce.

Figure 4. The Distributed Computation Framework of the ADMM Solver



5. Experiments

The proposed models were implemented using Julia language. All code and data for the experiments can be found in an accompanying GitHub repository (Li et al. 2024). All the experiments were conducted on a Lenovo server, which had two Xeon 8168 CPUs (96 threads) and 256G RAM.

The experiments were designed to validate and observe three issues: (a) the impact of the proposed diagonal DML model on the performance of distance-based classifiers on credit evaluation; (b) the difference between the proposed feature selection model and other benchmark feature selection models; and (c) the feature grouping effects with coefficient evolution on the regularization path and the differences with other correlation analysis methods.

5.1. Evaluation of the Proposed Diagonal DML Model

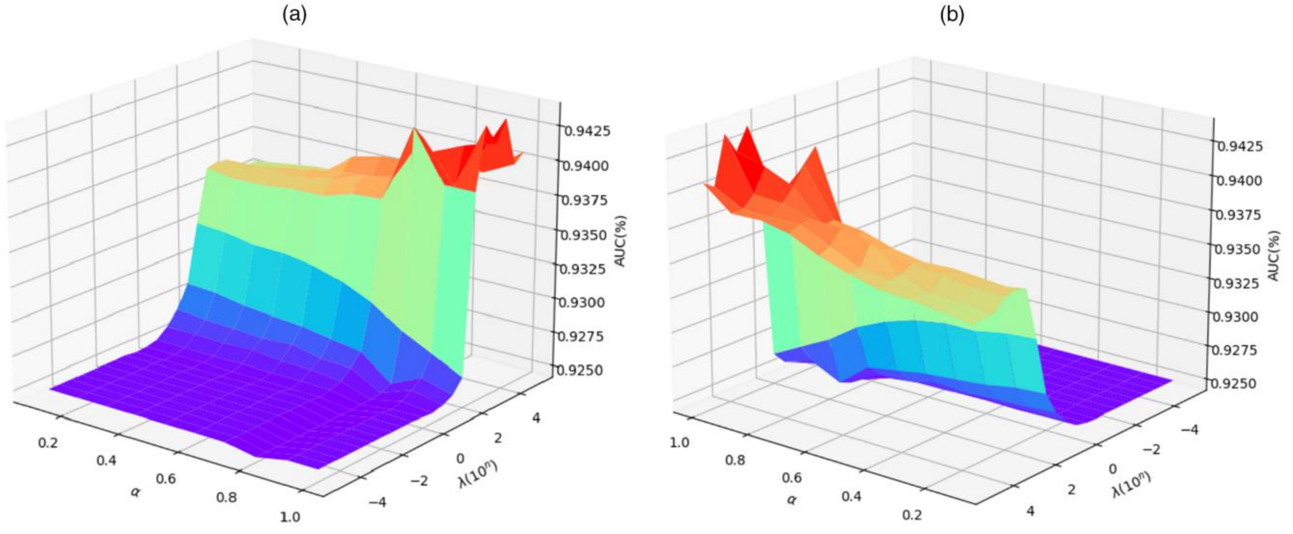
This section reports the parameter setting and training process of the proposed model and evaluates its performance. The credit data were divided into 96 groups, and each group had about 1,000 samples. The ADMM solver ran parallelly on the 96 groups to solve the linear programming problems corresponding to the first line in Equation (15) and then conducted a quadratic consensus-reaching optimization corresponding to the second line in Equation (15). These subtask groups were implemented with Julia multithreads. Further performance evaluation of the ADMM solver can be found in Online Appendix C.

5.1.1. Grid Search of the Optimal Parameters. The punishment parameter μ in Equation (1) was set as 5,000. It turned out that the results were not very sensitive to this punishment parameter, and 5,000 was a satisfying value based on intensive grid search results. We also conducted a grid search on the optimal parameter combination of α and λ and recorded the performances of 3-NN on the data sets transformed by the diagonal DML model with the corresponding α and λ.

In Figure 5, the best area under the receiver operating characteristic curve (AUC) performance of 3-NN on the whole data set was achieved around $\alpha = 0.8$ and $\lambda = 10^5$. The grid search results showed that the parameter combinations of $\alpha \in [0.7, 1.0]$ and $\lambda \in [10^{2.5}, 10^5]$ were satisfactory for the credit evaluation. The AUC surface of the proposed model with different parameter combinations was approximately smooth and convex, indicating that the proposed model is robust, and the parameters were easy to determine with the grid search method in practice.

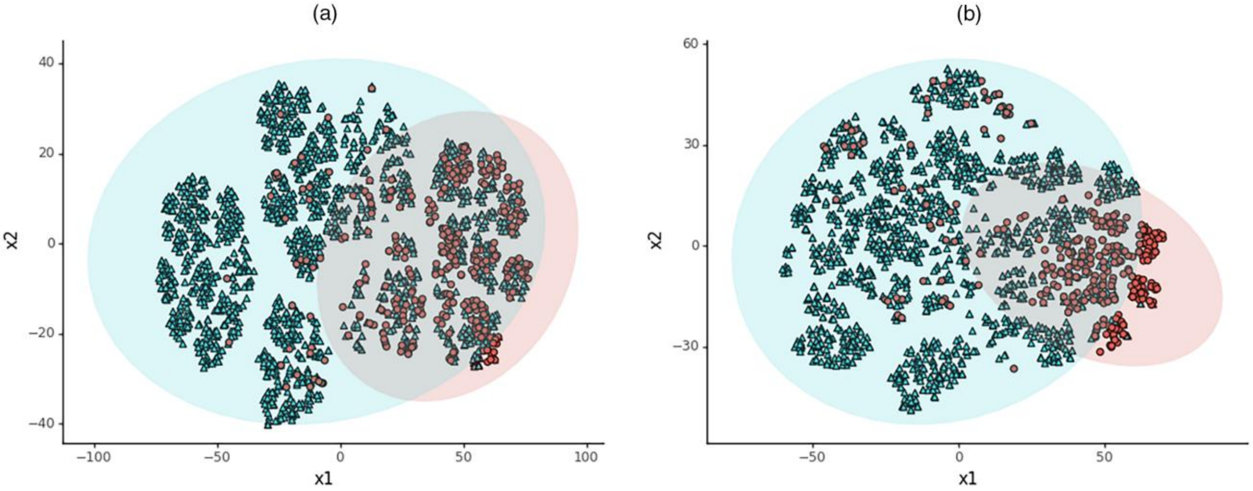
5.1.2. Effects of the Proposed Model on Data Distribution. To illustrate how the proposed diagonal DML transformation would affect the distributions of the credit data set, we used the t-distributed stochastic neighbor embedding technique (Van der Maaten and Hinton 2008) to reduce the dimensions of the data set to two and visualized in Figure 6, with different shapes of data points representing different credit classes.

Figure 5. The 3-NN's AUC Distribution Using Different α and λ Parameter Values



Notes. (a) AUC distribution from angle A. (b) AUC distribution from angle B.

Figure 6. (Color online) The Distribution of the Credit Data Set Before and After the DML



Notes. (a) Before DML. (b) After DML.

It can be seen from Figure 6(b) that after conducting DML, the data points of the same class were pushed closer in the local areas, while data points of different classes were pulled farther apart, compared with Figure 6(a). The data transformation from Figure 6, (a) and (b) should be beneficial to distance-based models like *k*-NN and *k*-Means.

5.1.3. The Performance Evaluation. Based on the optimal parameters obtained by the grid search, we transformed the original data set using the proposed diagonal DML model and evaluated the impacts of the DML on credit evaluation. To quantify the impact, we used four classical distance-based classifiers—that is, *k*-NN, RBFClassifier, KStar, and Locally weighted learning (LWL). For comparative purposes, we also included a representative linear model, L-SVM. The accuracy, true positive rate (TPR), AUC, F-Measure, and Matthews correlation coefficient were selected as the evaluation criteria. The results were obtained with three-fold cross-validation, as recorded in Table 4. The standard deviations between folds were recorded after the symbol "±."

It can be seen from Table 4 that the proposed diagonal DML approach had a significant effect on improving *k*-NN's classification results. The AUC of *k*-NN, which is a well-balanced measure to validate the overall performance of classifiers, was improved by about 14%. It showed that the proposed DML approach greatly benefited distance-based models, as they themselves cannot automatically assign weights to features.

The performance improvement of the linear model L-SVM was also evident. Generally speaking, the diagonal DML approach should not affect linear models because they can assign weights to features by themselves. In practice, the DML eliminated some irrelevant features and, thus, helped the linear models avoid overfitting to some extent.

Because the proposed model only learns the diagonal parameters in the DML matrix, it is also necessary to validate if there is a significant performance decline compared with other full-matrix and nonlinear models. We

Table 4. Performance of the Diagonal DML Approach with Distance-Based Classifiers and a Linear Model

Classifier	Dataset	Accuracy	TPR	AUC	F-measure	MCC
k-NN	Orig.	80.80 ± 0.30	46.01 ± 0.54	80.21 ± 0.41	46.88 ± 0.68	35.18 ± 0.88
	Trans.	91.97 ± 0.07	81.53 ± 0.49	94.27 ± 0.17	92.08 ± 0.23	74.01 ± 0.27
RBFCla.	Orig.	90.46 ± 0.08	86.42 ± 0.11	95.71 ± 0.05	79.85 ± 0.18	74.43 ± 0.23
	Trans.	92.40 ± 0.16	85.06 ± 0.31	96.02 ± 0.24	80.48 ± 0.27	75.95 ± 0.34
KStar	Orig.	84.39 ± 0.25	84.35 ± 0.26	88.37 ± 0.37	84.38 ± 0.29	48.40 ± 0.41
	Trans.	86.80 ± 0.18	86.83 ± 0.20	90.64 ± 0.28	86.72 ± 0.26	55.32 ± 0.33
LWL	Orig.	81.57 ± 0.36	81.63 ± 0.48	71.01 ± 0.42	81.87 ± 0.30	40.72 ± 0.45
	Trans.	90.72 ± 0.07	90.67 ± 0.26	84.43 ± 0.15	90.70 ± 0.27	69.19 ± 0.28
L-SVM	Orig.	73.90 ± 9.95	26.16 ± 35.75	55.42 ± 7.74	14.88 ± 18.67	8.17 ± 12.18
	Trans.	82.21 ± 0.07	75.78 ± 0.04	79.72 ± 0.04	70.21 ± 0.14	65.63 ± 0.17

Notes. The *k* in *k*-NN was set as three. "Trans." means the data sets transformed by the regularized D-DML method.

Table 5. Performance Comparison with Other Full-Matrix and Nonlinear DML Models

Model	Time (s)	Accuracy	TPR	AUC	F-measure	MCC
D-DML	24.1	90.71 ± 0.07	90.70 ± 0.42	92.43 ± 0.18	90.77 ± 0.12	69.64 ± 0.32
D-DML-R	25.4	91.97 ± 0.07	81.53 ± 0.49	94.27 ± 0.17	92.08 ± 0.23	74.01 ± 0.27
ITML	111.7	80.06 ± 0.25	36.10 ± 8.53	74.47 ± 0.14	40.00 ± 0.87	28.46 ± 1.00
DMLMJ	6,960.9	88.78 ± 0.06	72.01 ± 0.66	91.12 ± 0.16	70.27 ± 0.27	63.39 ± 0.31
LMNN	6.7E5	82.67 ± 0.07	70.03 ± 0.47	84.63 ± 0.14	63.22 ± 0.25	55.86 ± 0.37
TripletNet	1,933.2	92.93 ± 0.18	82.06 ± 0.17	94.45 ± 0.30	95.01 ± 0.11	74.92 ± 0.55

Note. D-DML refers to diagonal DML without regularization; D-DML-R refers to diagonal DML with regularizations.

used three full-matrix DML models (ITML, DMLMJ, and LMNN) and a nonlinear model (TripletNet) for comparison purpose. The time costs of these models were recorded in the second column of Table 5, and the classification performance of 3-NN based on the data sets transformed by these models was recorded in other columns of Table 5.

It can be seen from Table 5 that there was only slight performance disadvantage compared with the nonlinear model, TripletNet. We believe that it is necessary to keep a diagonal matrix when the DML model is used to conduct feature selection; otherwise, the DML model will recombine the original features and make the results noninterpretable, which is unacceptable in credit evaluation.

Compared with other full-matrix methods, the proposed model reduced the time complexity greatly, and the regularization terms also helped the proposed model avoid overfitting. One possible disadvantage of the proposed model was that it reduced the number of parameters, which may cause underfitting. Table 5 shows that no performance decline was observed compared with other matrix-based DML models, indicating that the proposed model avoided underfitting.

Above all, the experiments showed that the proposed diagonal DML model has positive impacts on distance-based models, such as *k*-NN. It is highly suitable for improving the performance of distance-based models when the credit data are not linearly separable.

5.2. Feature Selection with L_1 Regularization

To validate the feature selection performance of the diagonal DML (two conditions, as introduced in Section 4.2.4), we compared it with 11 other feature selection models: SVM-LASSO, LR-LASSO, LR-ElasticNet, LR-fused-LASSO, LR-group-LASSO, LR-L_0, Correlation-based Feature Selection (CFS), Principal Components (PC), InfoGain, GainRatio, and Symmetrical Uncertainty (SU). The number of features kept by these models, the parameter setting, and the classifiers used were recorded in Table 6.

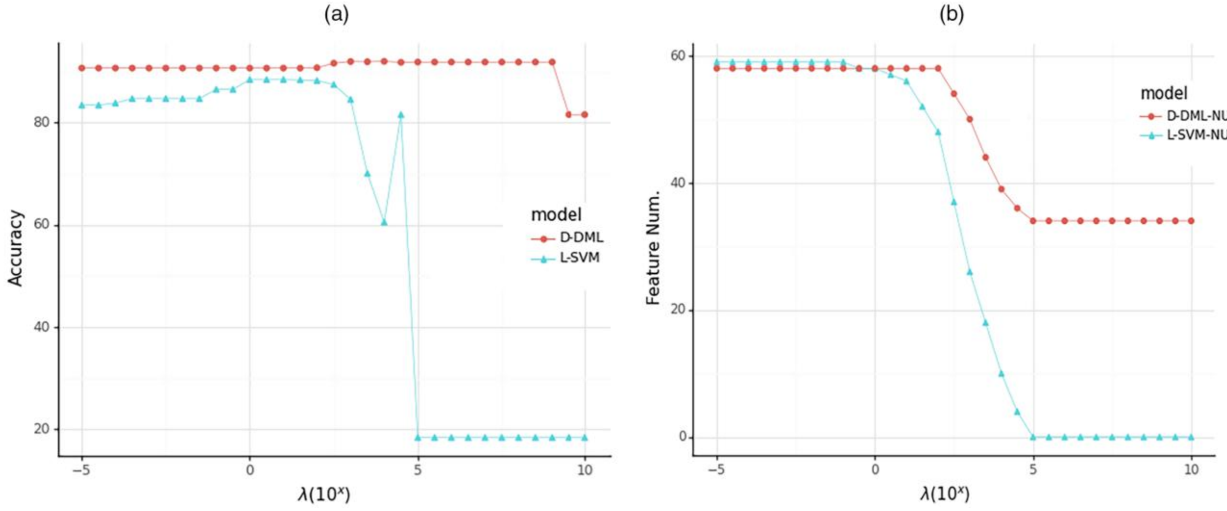
Table 6 shows that D-DML-ElasticNet and D-DML-L_1 achieved the best and second in terms of accuracy. As for the time cost, the ADMM solver helped the two versions of D-DML speed up to the same level as linear models. Traditionally, DML models were substantially slower than linear models because of large-scale constraints. Although the last five unsupervised methods were fast, their accuracies were lower than other methods.

Theoretically, the diagonal DML model (D-DML-L_1) and linear models like L-SVM use a similar mechanism to conduct feature selection—that is, L_1 regularization. Therefore, we used L-SVM-LASSO as a representative linear

Table 6. Performance Comparison of the Feature Selection Models

Model	Feature num. kept	Parameter setting	Base classifier	Time costs (s)	Accuracy
D-DML-L_1	34	$\lambda = 10^5$	3-NN	24.1	91.87 ± 0.07
D-DML-ElasticNet	45	$\lambda = 10^5$, $\alpha = 0.8$	3-NN	25.4	91.97 ± 0.07
SVM-LASSO	37	$\lambda = 10^{2.5}$	L-SVM	19.2	87.47 ± 0.12
LR-LASSO	37	$\lambda = 10^{2.5}$	LR	13.7	88.00 ± 0.10
LR-ElasticNet	41	$\lambda = 10^{2.5}$, $\alpha = 0.5$	LR	14.6	89.03 ± 0.11
LR-fused LASSO	40	$\lambda_1 = 10^{2.5}$, $\lambda_2 = 10^{2.5}$	LR	13.5	86.78 ± 0.18
LR-group LASSO	38	$\lambda = 10^{2.5}$	LR	13.8	85.34 ± 0.25
LR-L_0	36	$\lambda = 10^5$	LR	63.5	88.41 ± 0.13
CFS	12	/	/	4.5	63.25 ± 0.32
PC	48	varienceCovered = 0.95	/	5.6	86.52 ± 0.17
InfoGain	41	/	/	1.6	80.37 ± 0.26
GainRatio	41	/	/	3.2	80.37 ± 0.26
SU	41	/	/	3.5	80.37 ± 0.26

Figure 7. (Color online) Comparison of Feature Selection Between D-DML-L_1 and L-SVM-LASSO



Notes. (a) Accuracy. (b) Number of features.

model and conducted a comparison. Figure 7 outlines the accuracy and the different number of selected features when using the same punishment sequence (the values of λ) on the L_1 term.

It can be seen from Figure 7(a) that the accuracies of both the regularized D-DML-L_1 and regularized L-SVM-LASSO can be improved from the unregularized ones ($\lambda = 0 = 10^{-\infty}$), given the appropriate regularization parameters (such as $\lambda = 10^{2.5}$). The phenomenon proved again that the regularization mechanism can benefit the performance of the models. It can also be observed from Figure 7, (a) and (b) that with the increase of λ, the number of kept features and the accuracies decreased. If we want to keep a sparser model, we need to use a larger λ; however, the increase of λ to a certain level deteriorates the accuracy. In practice, we need to balance the model sparsity and accuracy. In this study, we proposed an "elbow" principle to determine the appropriate λ: if the accuracy curve in Figure 7(a) declined dramatically, we chose the first biggest inflection point on the right of the peak; if the accuracy curve was smooth and flat, we chose the last big inflection point on the curve in Figure 7(b). Based on the elbow principle, in terms of accuracy (Figure 7(a)), the most proper λ value for L-SVM was $10^{2.5}$, and the most proper λ value for the diagonal DML model was 10^5.

To compare the features kept by the D-DML-L_1 model and L-SVM-LASSO, we normalized the coefficients learned by the two models, respectively, such that the relative importance of the features can be compared with the same magnitude, as shown in Figure 8.

It can be seen from Figure 8 that the D-DML-L_1 and L-SVM-LASSO agreed that x9–x12, x36, x44, and x49 were important features for credit evaluation. The main difference was that the two models posed varying weights to the x9–x12, x17, x36, x41, x42, x46, and x51. In D-DML-L_1, x36 was the most important one, whereas in L-SVM-LASSO, x9–x12 were the most important ones.

5.3. Feature Grouping Effect Analysis via ElasticNet Regularization

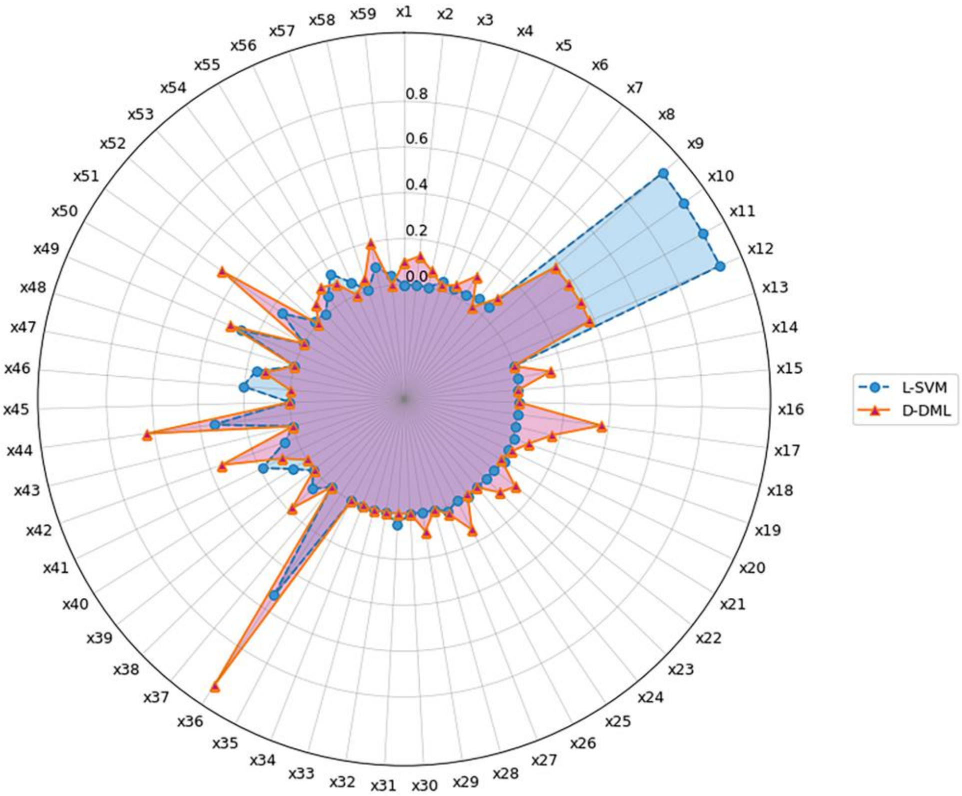
This section studied the characteristics of the D-DML-ElasticNet feature selection and validated the grouping effects between features.

5.3.1. Features Kept by ElasticNet Regularization.

As analyzed in Section 4.2.4, the D-DML-ElasticNet also has an effect of feature selection. Without loss of generality, we selected $\alpha = 0.8$ and $\lambda = 10^5$ as the parameters and compared its feature selection result with D-DML-L_1's results. The coefficients of the L_1 and ElasticNet regularized models are illustrated in Figure 9, (a) and (b), respectively.

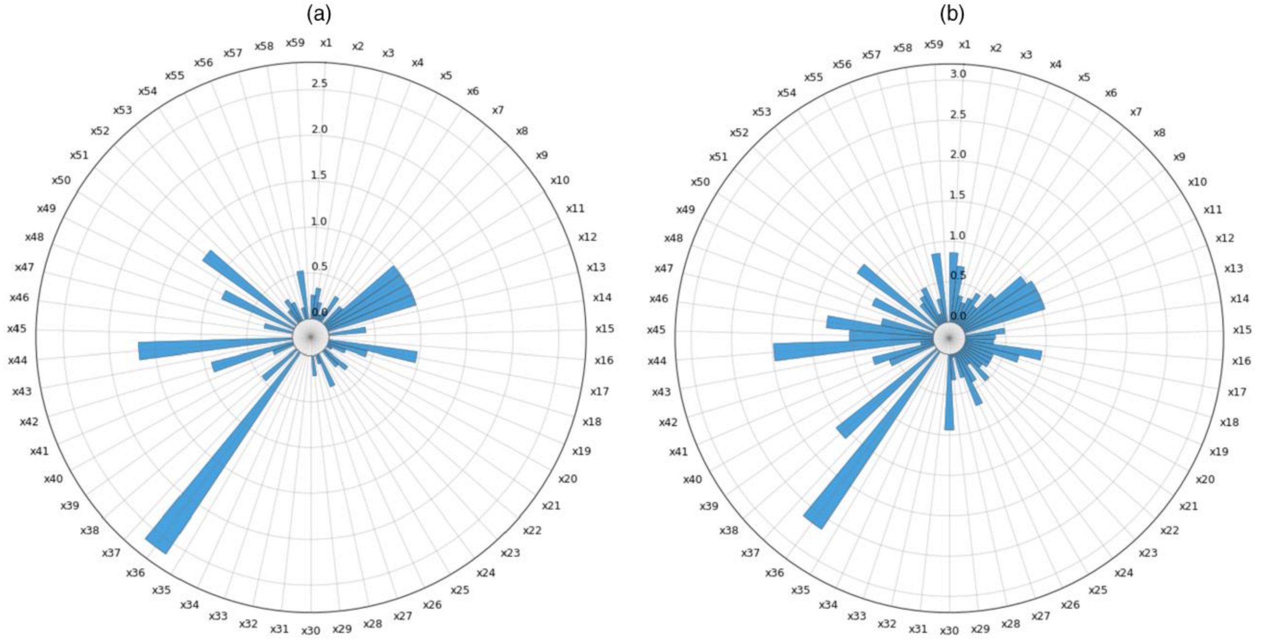
The most common influential features for customer credit were x36 and x44—that is, "Longest delayed days" and "Outstanding debt." We should take measures to control credit risk from "deferred payment" and "debt." Besides, we should also pay attention to x51, x17, x49, and x43, which were "Investment deposits," "Occupation=D" "Account age," "If minimal payment = 1," and "Gender=unknown," respectively. Other small influential factors were x27 (Occupation=N), x30 (Saving deposits), and "Overdue month" (x1–x12).

Figure 8. (Color online) Feature Selection Results of D-DML-L_1 and L-SVM-LASSO with L_1 Regularization



Compared with the L_1 regularization, ElasticNet kept more features. For instance, x15, x16, x21, x30, and x45 were kept by the ElasticNet, whereas L_1 posed zero weights to them. As analyzed in Section 4, the reason behind this phenomenon was that if the features were correlated, there were grouping effects between the features, and the ElasticNet tended to keep all of the correlated features, whereas L_1 tended to keep only one of them.

Figure 9. (Color online) The Coefficients with ElasticNet and L_1 Regularization



Notes. (a) L_1 ($\alpha = 1, \lambda = 10^5$). (b) ElasticNet ($\alpha = 0.8, \lambda = 10^5$).

5.3.2. Grouping Effect on the Regularization Path. Based on the definition in Equation (5), if there is a grouping effect between two features, the coefficients of them will always be similar and change parallelly on the ElasticNet regularization path. To validate this phenomenon, we illustrated the coefficients on the regularization path of both ElasticNet and L_1 in Figure 10.

It can be seen in Figure 10 that with the increase of the punishment, the coefficients (25 variables) in the L_1 regularized model decreased dramatically, and many of them reached the bottom, resulting in a much sparser model. By contrast, only a smaller part of the coefficients (8 variables) dived to zero in ElasticNet, and a larger part of the coefficients shrank slowly and parallelly. As analyzed in Section 4, the correlated features have a grouping effect—that is, the coefficients of the highly correlated features tended to be analogous—and such effect can be reflected partially by the coevolution of the coefficients in the ElasticNet regularization path.

5.3.3. Feature Group Detection Based on Algorithm 1. Based on the grid search result using different combinations of λ and α, we can quantify the group effects between pairwise features with Equation (7). As outlined in Algorithm 1, we used the pairwise grouping effects ($s_{i,j} = 1 - \overline{diff}_{i,j}$) as edges and the features as nodes to form a feature graph and then conducted community detection using the "unfolding" algorithm within the graph (the modularity was 0.579). The detection results were illustrated in Figure 11.

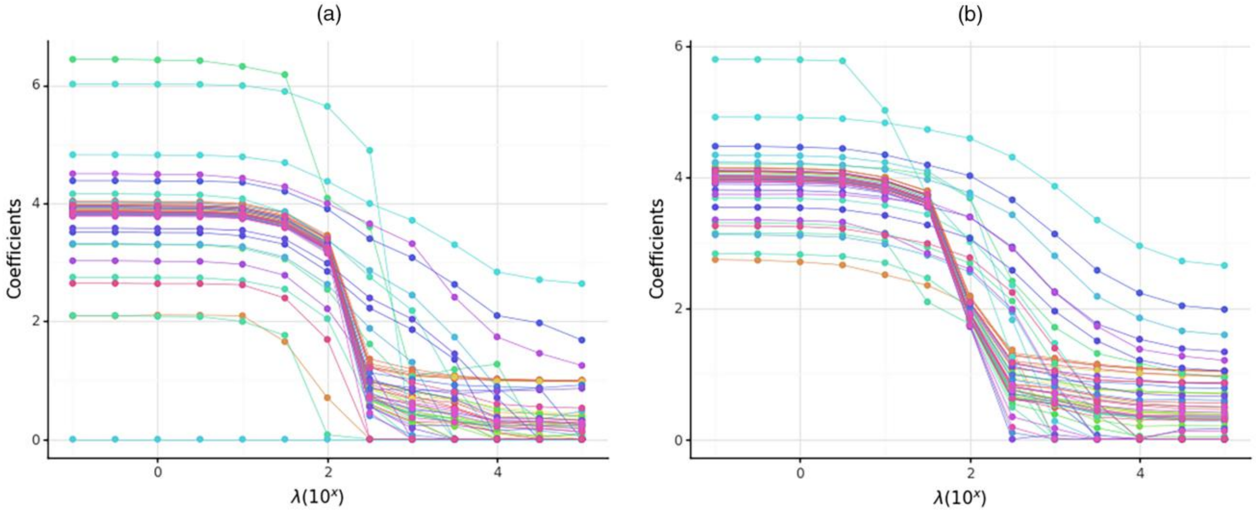
We can see from Figure 11 that many features had strong grouping effects. For instance, x21 and x53; x7 and x57; and x52 and x40 were highly correlated. The coefficients of these pairwise features were almost identical on the regularization path. As a result, these features were considered to have strong pairwise grouping effects. In credit risk analysis, we need to pay special attention to such grouping effects because these features represent similar credit risk sources equally. By contrast, if using the L_1 regularization, only one feature will be kept in each group. As a result, we may only notice the features kept by the L_1 regularized model and neglect the eliminated but correlated ones, which reflect potential credit risk sources.

Besides, there were three communities in the network. Based on the meanings of the features in the communities, we can speculate that the community on the left was mainly about the occupations and the related payment behavior (x53, payment=A), and the community in the middle was about the overdue month and the related payment type (x57, payment=B), whereas the community on the right was about gender (x40) and payment type (x52, payment=C).

To compare the grouping effects with traditional correlation analysis, we used Pearson's correlation coefficients and Variance Inflation Factor (VIF) (O'brien 2007) to test the correlations among the features. Pearson's correlation coefficients can reflect the pairwise correlations between features, whereas VIF's score indicates the multicollinearity among the features. The results are illustrated in Figure 12.

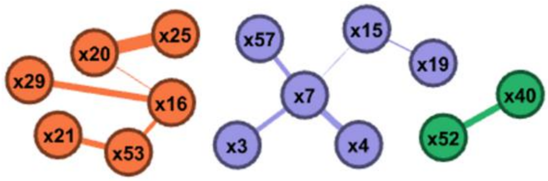
It can be seen from Figure 12(a) that the correlations reflected by Pearson's pairwise correlation coefficients and the VIF scores were not the same. For instance, x14–x20 were not severely correlated with other variables, but they had very high VIF scores, indicating that they can be represented by the linear combination of several other features—that is, multicollinearity.

Figure 10. Regularization Path of ElasticNet and L_1 Regularization



Notes. (a) L_1 regularization path ($\alpha = 1.0$). (b) ElasticNet regularization path ($\alpha = 0.8$).

Figure 11. (Color online) Feature Clusters Detected by Algorithm 1



It was also noteworthy that the recognized pairwise grouping effect results between the proposed model and Pearson's correlation were not the same. We can see from Equation (6) that the limit of the grouping effect $(|\overline{w}_i - \overline{w}_j| = 0)$ is equivalent to Pearson's correlation ($\sigma_{i,j} = 1$) only when two features are identical—that is, highly correlated. Nevertheless, there are dual variables β in Equation (6), and the specific solution of β depends on α, λ, c, F, and the labels. It indicates that the specific value of $|\overline{w}_i - \overline{w}_j|$ is partially influenced by the distribution of the labels, and this characteristic of grouping effect is very different from Pearson's correlation. Above all, the grouping effect in this study interacts with the labels, resulting in a supervised problem, whereas Pearson's correlation does not interact with the labels, resulting in an unsupervised problem.

5.4. Further Discussion of the Experimental Results

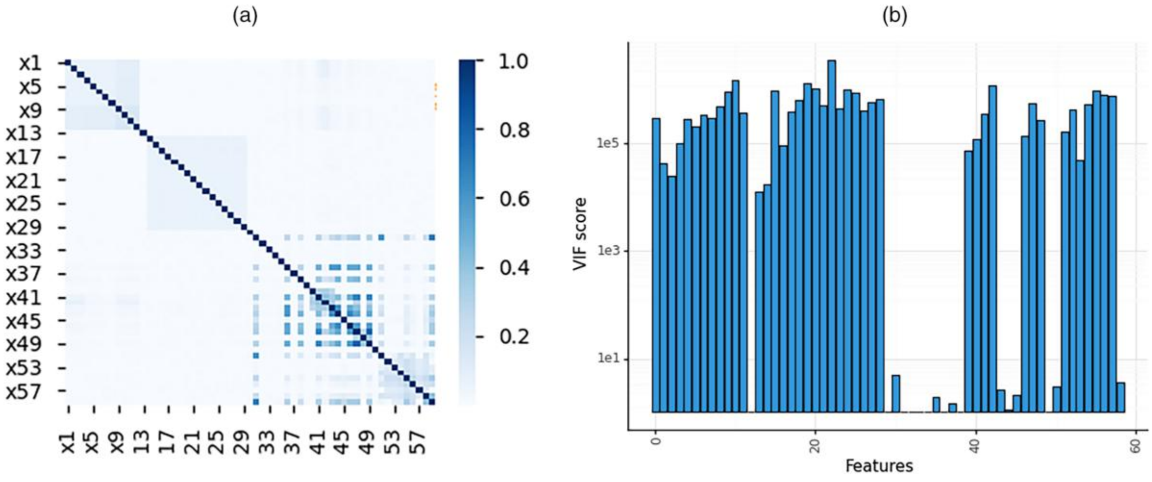
We further discuss the observations and implications based on the experiments:

5.4.1. The Theoretical Assumption of This Study. This study has a latent assumption: manifold learning, which assumes that neighboring data points tend to have the same labels. In credit evaluation, it is not crucial whether the original data follow this assumption; what matters is whether we can enforce such an assumption in an ideal space achieved through a diagonal linear transformation. In all, DML has a manifold assumption substituting linear models' "linear separability" assumption, and that is the theoretical difference between diagonal DML and linear models concerning feature selection and grouping effect analysis.

5.4.2. The Difference Between the Proposed Model and Regularized Linear Models. The feature selection mechanism of the proposed method relies on the regularization applied to the loss function, which is similar to linear classifiers that utilize regularizations. Because the linear classifiers with regularization terms that can be used for feature selection mainly refer to L_1 and L_0, we compare the proposed method with them in terms of base models, theories, mathematical optimization, solvers, and applications and summarize the similarities and differences in Table 7.

5.4.3. Comparison with Other Distance Metric Learning Methods. The core similarity between the proposed method and other matrix-based DML methods lies in their shared theoretical foundation: transforming the feature space with a matrix to cluster data points sharing the same labels closer and pushing apart those belonging to different classes. Table 8 compares our method with other DML techniques in terms of matrix form, optimization problem, and solution techniques.

Figure 12. (Color online) Correlations Reflected by Mutual Correlation Coefficients and VIF Scores



Notes. (a) Mutual correlation coefficients. (b) VIF scores.

Table 7. Comparison of the Proposed Method with L_1 and L_0-Based Linear Classifiers

Aspects	L_1-based linear classifiers	L_0-based linear classifiers
Base models	**Similarities:** All the methods need to encapsulate the feature selection into mathematical optimization facilitating classification problems. **Differences:** The proposed method is based on diagonal DML, which aims to improve the performance of distance-based models such as k-NN. By contrast, L_1-based and L_0-based classifiers are based on linear models, such as LR, SVM, and LDA, which pursue linear separability.	
Theories	**Similarities:** All the methods rely on a mathematical property that a L_1 or L_0 regularization term is equivalent to a constraint term whose constraint region has sharp corners alongside the coordinate axes, and the original objective function is very likely to be tangent to the constraint region at the corners, resulting in a sparse solution—that is, many zero coefficients. **Differences:** The theoretical assumption behind the proposed method is "manifold learning." The assumption of linear classifiers is that data samples are independent and identically distributed, such that they can learn a universal model (hyperplanes) to handle all the data points.	
Optimization problems	**Similarities:** Both form the optimization as mathematically convex problems and add L_1 as extra terms, which introduce bias to the optimization problem but control overfitting. **Differences:** The proposed method forms the optimization as a large-scale linearly constrained problem. By contrast, the L_1-based linear classifier forms an unconstrained problem.	**Similarities:** Both use regularization terms, which are equivalent to constraint terms of optimization problems. **Differences:** The L_1 term is convex and easy to be approximated by other differentiable functions. The L_0-based regularization introduces no bias to the optimization problem, but the L_0-term is nonconvex and hard to handle.
Solvers	**Similarities:** Both need to handle the L_1 term, which is not differentiable at 0 point. **Differences:** The proposed method needs to transform the constraints into punishment terms of the objective function. The optimization problems of linear classifiers can be solved using plain gradient-descent methods.	**Similarities:** Both can utilize linear programming to facilitate the optimization. **Differences:** The proposed method can assign zero coefficients automatically, and its solving is based on gradient-descent methods or their variants. The L_0-based method needs to set the number of kept features in advance, and its solving involves mixed integer programming.
Applications	**Similarities:** All can be used in feature selection where a classification problem is concerned and the label information is leveraged. **Differences:** The proposed method does not have a prerequisite for the distribution of the data. It is especially suitable for scenarios where there are many different dominant influential factors on different parts of the data, and the patterns among the data change over time. By contrast, L_1-based and L_0-based linear classifiers require that the data points of different classes can be linearly separated, which is unrealistic for data sets with complex distributions.	

Table 8. Differences Between the Proposed Method and Other Linear DML Methods

	The proposed	ITML	DMLMJ	LMNN
Matrix forms	Diagonal	Full matrix		
Purposes	Feature selection and grouping effect analysis	Improve distance metric and benefit distance-based models		
Optimization problems	- Quadratic optimization - Introduce bias but avoid overfitting	- LogDet divergence regularization to avoid keeping a PSD matrix - Approximation method	- Jeffrey divergence as the loss function to avoid keeping a PSD matrix - Approximation method	- Semidefinite programming
Solvers	- ADMM-based solver - Linear programming is leveraged heavily during the optimization process	- Gradient-descent-based solver	- Gradient-descent-based solver	- Projection gradient method - Heavy Eigen decomposition is involved at each optimization step

6. Conclusions

To cope with the limitations of linear models, this study proposed a diagonal DML approach, to explore its L_1 and ElasticNet regularizations, which realize feature selection and grouping effect analysis in credit evaluation. Because every data point generates a constraint corresponding to a slack variable in DML, the optimization of DML with credit data usually involves a large number of variables, and the solving of the optimization problem can be difficult. To deal with this issue, we proposed a parallel solver based on ADMM to efficiently solve the optimization problems. The experiments showed that the AUC of k-NN was improved by 14% using the DML model. Besides, the feature selection and grouping results were also different from traditional models, such that novel credit risk sources (features) can be captured.

The grouping effect analysis in this study focused on pairwise correlations of the features. Nevertheless, there are many scenarios in which multicollinearities exist, as shown in Figure 12(b). One of the future research directions is to recognize such correlations and find out possible credit risks.

References

Basu R, Naughton JP (2020) The real effects of financial statement recognition: Evidence from corporate credit ratings. *Management Sci.* 66(4):1672–1691.

Bhat G, Ryan SG, Vyas D (2019) The implications of credit risk modeling for banks' loan loss provisions and loan-origination procyclicality. *Management Sci.* 65(5):2116–2141.

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations Trends® Machine Learn.* 3(1):1–122.

Cakir F, He K, Xia X, Kulis B, Sclaroff S (2019) Deep metric learning to rank. *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognition* (IEEE, Piscataway, NJ), 1861–1870.

Cui L, Bai L, Wang Y, Jin X, Hancock ER (2021) Internet financing credit risk evaluation using multiple structural interacting elastic net feature selection. *Pattern Recognition* 114:107835.

Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. *Proc. 24th Internat. Conf. Machine Learn.* (Association for Computing Machinery, New York), 209–216.

Der M, Saul L (2012) Latent coincidence analysis: A hidden variable model for distance metric learning. *Advances in Neural Information Processing Systems*, vol. 25 (Curran Associates, Inc., Red Hook, NY), 3230–3238.

Ferman B (2016) Reading the fine print: Information disclosure in the Brazilian credit card market. *Management Sci.* 62(12):3534–3548.

Gómez A, Prokopyev OA (2021) A mixed-integer fractional optimization approach to best subset selection. *INFORMS J. Comput.* 33(2):551–565.

Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press, Cambridge, MA).

Han B, Ryzhov IO, Defourny B (2016) Optimal learning in linear regression with combinatorial feature selection. *INFORMS J. Comput.* 28(4):721–735.

Hazimeh H, Mazumder R (2020) Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Oper. Res.* 68(5):1517–1537.

Hilscher J, Wilson M (2017) Credit ratings and credit risk: Is one measure enough? *Management Sci.* 63(10):3414–3437.

Hoffer E, Ailon N (2015) Deep metric learning using triplet network. *Proc. Similarity-Based Pattern Recognition Third Internat. Workshop SIMBAD 2015* (Springer, Cham, Switzerland), 84–92.

Hong LJ, Juneja S, Luo J (2014) Estimating sensitivities of portfolio credit risk using Monte Carlo. *INFORMS J. Comput.* 26(4):848–865.

Jiang H, Luo S, Dong Y (2021) Simultaneous feature selection and clustering based on square root optimization. *Eur. J. Oper. Res.* 289(1):214–231.

Kelley S, Ovchinnikov A, Hardoon DR, Heinrich A (2022) Antidiscrimination laws, artificial intelligence, and gender bias: A case study in nonmortgage fintech lending. *Manufacturing Service Oper. Management* 24(6):3039–3059.

Keshanian K, Zantedeschi D, Dutta K (2022) Features selection as a Nash-bargaining solution: Applications in online advertising and information systems. *INFORMS J. Comput.* 34(5):2485–2501.

Li T, Kou G, Peng Y, Yu PS (2021) A fast diagonal distance metric learning approach for large-scale datasets. *Inform. Sci.* 571:225–245.

Li T, Kou G, Peng Y, Yu PS (2024) Feature selection and grouping effect analysis for credit evaluation via regularized diagonal distance metric learning. http://dx.doi.org/10.1287/ijoc.2023.0322.cd, https://github.com/INFORMSJoC/2023.0322.

Maldonado S, Bravo C, López J, Pérez J (2017) Integrated framework for profit-based feature selection and SVM classification in credit scoring. *Decision Support Systems* 104:113–121.

Meier L, Van De Geer S, Bühlmann P (2008) The group lasso for logistic regression. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 70(1):53–71.

Nguyen B, Morell C, De Baets B (2017) Supervised distance metric learning through maximization of the Jeffrey divergence. *Pattern Recognition* 64:215–225.

O'brien RM (2007) A caution regarding rules of thumb for variance inflation factors. *Qual. Quant.* 41:673–690.

Petersen A, Witten D, Simon N (2016) Fused lasso additive model. *J. Comput. Graph. Statist.* 25(4):1005–1025.

Piramuthu S (1999) Feature selection for financial credit-risk evaluation decisions. *INFORMS J. Comput.* 11(3):258–266.

Shi Y, Miao J, Wang Z, Zhang P, Niu L (2018) Feature selection with $l_{2,1-2}$ regularization. *IEEE Trans. Neural Networks Learn. Systems* 29(10):4967–4982.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 58(1):267–288.

Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K (2005) Sparsity and smoothness via the fused lasso. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 67(1):91–108.

Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J. Machine Learn. Res.* 9(86):2579–2605.

Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. *J. Machine Learn. Res.* 10(9):207–244.

Won D, Manzour H, Chaovalitwongse W (2020) Convex optimization for group feature selection in networked data. *INFORMS J. Comput.* 32(1):182–198.

Xiao J, Tian Y, Jia Y, Jiang X, Yu L, Wang S (2023) Black-box attack-based security evaluation framework for credit card fraud detection models. *INFORMS J. Comput.* 35(5):986–1001.

Xing E, Jordan M, Russell SJ, Ng A (2002) Distance metric learning with application to clustering with side-information. *Proc. 15th Internat. Conf. Neural Inform. Processing Systems*, vol. 15 (MIT Press, Cambridge, MA), 521–528.

Ying Y, Li P (2012) Distance metric learning with eigenvalue optimization. *J. Machine Learn. Res.* 13(1):1–26.

Yoganarasimhan H (2020) Search personalization using machine learning. *Management Sci.* 66(3):1045–1070.

Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 68(1):49–67.

Zhang J, Wang C, Chen G (2021) A review selection method for finding an informative subset from online reviews. *INFORMS J. Comput.* 33(1):280–299.

Zheng Z, Zhang J, Li Y (2022) L_0-regularized learning for high-dimensional additive hazards regression. *INFORMS J. Comput.* 34(5):2762–2775.

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 67(2):301–320.