Latent Similarity-Enhanced Credit Risk Prediction

Hongzhe Zhang
School of Management and Economics, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), China
zhanghongzhe@cuhk.edu.cn

Wei Qian*
Department of Applied Economics and Statistics, University of Delaware, Newark, Delaware
weiqian@udel.edu

Xiao Fang*
Lerner College of Business and Economics, University of Delaware, Newark, Delaware
xfang@udel.edu

Given the sheer size of the consumer credit market and the huge number of consumer credit users, credit risk prediction, or how to predict delinquent (or default) probabilities of consumer credits to aid financial institutions in granting and managing them, has become a critical problem in the consumer credit industry. While it is desirable to employ both users' intrinsic characteristics and similarities among them for effective credit risk prediction, existing studies rely solely on similarities derived from their observed characteristics and fail to account for unobserved similarities among them. To address this challenge, we propose a latent similarity-enhanced credit risk prediction model, which operationalizes the similarity between a pair of users as a combination of the observed and latent similarities between them. We then design a new method that estimates the model parameters, learns latent similarities among users, and integrates both observed and latent similarities among users with their intrinsic characteristics for credit risk prediction. We further extend our method to the multiclass and numerical credit risk prediction problems. Extensive empirical evaluations with real world data demonstrate the superior predictive power of our method over benchmark methods for a broad spectrum of credit risk prediction problems. We also show substantial economic value generated from the superiority of our method through a case study.

Key words: credit risk prediction, latent similarity, machine learning, information systems: enabling technologies

---

* Corresponding Authors

1. Introduction

Consumer credit refers to an amount of money loaned to a consumer by a financial institution (Hand and Henley 1997). Over the years, the number of consumer credit users as well as the size of consumer credit market have grown tremendously. Taking the U.S. as an example, its consumer credit market, including credit cards, auto and student loans, as well as mortgages, reached $14.6 trillion or 69.5% of its GDP at the end of 2020[1] and more than 90% of U.S. adults held at least one credit card in 2020.[2] Given the huge number of consumer credit users and the sheer size of the consumer credit market, credit risk prediction, or how to predict delinquent (or default) probabilities of consumer credits to aid financial institutions in granting and managing them, has become a critical problem in the consumer credit industry (Altman et al. 2005, Chatterjee et al. 2007, De Almeida Filho et al. 2010). Answers to this problem could significantly affect consumers, financial institutions, and even national and international economies. Effective credit risk prediction benefits consumers by giving valued consumers access to much-needed credits as well as financial institutions by understanding credit risk levels and reducing default losses. Consequently, effective credit risk prediction has a positive impact on national and international economies. On the other hand, ineffective credit risk prediction hurts both consumers and financial institutions, which in turn damages the economies. For example, the global financial crisis of 2007–2008 was partially caused by ineffective credit risk management (Thomas et al. 2017), and the estimated cost of the crisis ranges from $6 trillion to $14 trillion in the U.S. alone (Atkinson et al. 2013).

There are two types of credit risk prediction: application scoring and behavioral scoring. Application scoring targets credit applicants and aims at predicting their individual probability of defaulting a loan if granted (e.g., how likely a loan applicant will fail to repay the loan as scheduled). Behavioral scoring, on the other hand, focuses on existing borrowers and attempts to predict and monitor their individual probability of delinquency (e.g., how likely an existing borrower will fail to

[1] See https://www.cnbc.com/2021/02/17/household-debt-rises-to-14point6-trillion-due-to-record-breaking-rise-in-mortgage-loans.html (last accessed on March 26, 2022).

[2] See https://www.experian.com/blogs/ask-experian/consumer-credit-review (last accessed on March 26, 2022).

repay the minimum payment due). Both application scoring and behavioral scoring are important to financial institutions. The former assists financial institutions to identify and filter out risky credit applicants whereas the latter helps them mitigate risks of existing loans. Moreover, with the advent of the Basel Accord banking regulations, financial institutions are required to estimate the risks for all of their existing loan portfolios and set aside regulatory capital accordingly (Thomas et al. 2017). As a result, behavioral scoring has become an essential tool for financial institutions and is the focus of this study.

Both application scoring and behavioral scoring can be formulated as a classification problem and solved using a classification method although training data for behavioral scoring are richer with additional information such as borrowers' repayment and billing data (Thomas 2000). Logistic regression remains the most commonly used classification method for credit risk prediction (Thomas et al. 2017). Ensemble methods, which combine multiple classifiers through bagging or boosting (Breiman 1996, Freund and Schapire 1997), have also been employed for credit risk prediction (West et al. 2005, Paleologo et al. 2010). Among the ensemble methods, XGBoost (or extreme gradient boosting) has gained popularity with its excellent predictive performance (Chen and Guestrin 2016). In addition, recent studies have proposed deep learning-based methods, such as recurrent neural network, for credit risk prediction (Babaev et al. 2019).

The methods discussed above employ borrowers' intrinsic characteristics for credit risk prediction. Prior studies have proposed credit risk prediction methods that utilize similarities among borrowers, in addition to their intrinsic characteristics. For example, Henley and Hand (1996) and Kennedy et al. (2013) identify a borrower's nearest neighbors based on their similarities with other borrowers and apply the K-Nearest Neighbor (KNN) classifier to predict their credit risk. Lee et al. (2021) compute similarities among borrowers to construct a borrower network and then employ a graph convolutional network approach (Kipf and Welling 2017) to extract network features for credit risk prediction. While it is beneficial to consider both similarities among borrowers and their intrinsic characteristics for credit risk prediction, existing studies rely solely on similarities computed from observed characteristics. For example, Lee et al. (2021) calculate similarities

among borrowers based on their observed characteristics, such as credit history and demographic information.

Similarities among borrowers manifest in various aspects beyond those derived from observed characteristics. Therefore, we operationalize the similarity between a pair of borrowers as a combination of two components: the observed similarity, computed from their observed intrinsic characteristics, and the latent similarity (unobserved similarity), derived from unobserved intrinsic characteristics. Building on this, we propose a credit risk prediction model that integrates both observed and latent similarities among borrowers, as well as their intrinsic characteristics. We then design a method to estimate the model parameters, learn latent similarities, and combine observed and latent similarities among borrowers with their intrinsic characteristics for credit risk prediction. Therefore, our method differs from prior credit risk prediction methods that focus solely on observed similarities among borrowers. Moreover, unlike traditional credit risk prediction methods that rely exclusively on borrowers' intrinsic characteristics, our method leverages the predictive power of a borrower's similar peers' credit risks to enhance the prediction of the borrower's credit risk.

2. Related Work

2.1. Methods for Credit Risk Prediction

In general, credit risk prediction, including application scoring and behavioral scoring, can be modeled as a classification problem, which is then solved with a classification method (Thomas 2000). Training data for application scoring typically contain loan applicants' demographic data (e.g., age), financial data (e.g., income), and credit history data (e.g., credit score). In addition, training data for behavioral scoring also consist of existing borrowers' repayment and spending data (Thomas 2000). Once training data are constructed, common classification methods can be applied to the data for application scoring or behavioral scoring.

The most widely used classification method for credit risk prediction is logistic regression (Thomas et al. 2017), which models a binary dependent variable (e.g., delinquency or not) using a linear combination of independent variables transformed through a sigmoid function. Aside from

logistic regression, other classification methods, such as discriminant analysis, Naive Bayes and nearest neighbor, have also been applied to credit risk prediction (Hand 2001). Fostered by the advancement of artificial intelligence in recent years, credit risk prediction methods have also been developed based on machine learning algorithms including multilayer perceptron (Malhotra and Malhotra 2003, Sinha and May 2004), decision tree (Lessmann et al. 2015), and support vector machine (Yao et al. 2015). Meanwhile, it is well known that combining multiple classifiers through bagging or boosting can outperform a single one in predictive performance (Freund and Schapire 1997, Breiman 1996). Accordingly, West et al. (2005) investigate different ensemble strategies for the neural network model and find that the generalization ability of the ensembled model is superior to the single-best model in credit risk prediction. Paleologo et al. (2010) adopt an ensemble technique, namely subagging, to improve the performance of base classifiers for credit risk prediction. Because of its excellent performance, XGBoost is one of the most commonly used ensemble methods for credit risk prediction (Chen and Guestrin 2016); the method keeps the scheme of gradient boosting machine (Friedman 2001) but introduces a more advanced regularization mechanism to avoid model overfitting. In addition, recent studies have employed deep learning methods, such as recurrent neural network and graph convolutional network, for credit risk prediction (Babaev et al. 2019, Lee et al. 2021). Homophily theory suggests that individuals with similar characteristics tend to exhibit similar behaviors (McPherson et al. 2001). Therefore, another way to improve the performance of credit risk prediction is leveraging similarities among users, which we review next.

2.2. Similarity-based Credit Risk Prediction

An early study that employs similarities among borrowers for credit risk prediction is Henley and Hand (1996), which applies the KNN classifier to predict borrowers' credit risks. Specifically, a borrower's nearest neighbors are identified based on similarities among borrowers, which are calculated using their characteristics provided in their loan application forms. Due to its good performance and interpretability, the KNN-based method has been widely adopted in credit risk prediction (Hand and Vinciotti 2003, Twala 2010, Kennedy et al. 2013). Another prevalent approach predicts

a borrower's credit risk using the weighted average of other borrowers' credit risks, where weights are determined based on similarities among borrowers. For instance, Guo et al. (2016) propose a method that employs kernel weights to predict a loan's return rate by calculating a weighted average of the return rates of similar loans. Other methods construct a network of borrowers based on similarities among them and then utilize network features for credit risk prediction. For example, Giudici et al. (2020) build a network by treating each borrower as a node and assigning the weight of an edge between two borrowers based on the similarity between them. Here, the similarity is calculated as the reciprocal of the Euclidean distance between the borrowers' characteristics. Centrality measures are then derived from the network for credit risk prediction. Lee et al. (2021) also construct a network of borrowers based on similarities among them but employ a graph convolutional network approach (Kipf and Welling 2017) to extract network features for credit risk prediction. Additional data have been explored to measure similarities among borrowers. For example, Fernandes and Artes (2016) and Medina-Olivares et al. (2022) compute spatial similarities among borrowers using their geographical data and demonstrate that incorporating these similarities can significantly improve the performance of credit risk prediction.

Our literature review suggests the following research gaps. First, traditional credit risk prediction methods ignore the predictive power of similar borrowers' repayment behaviors in predicting a borrower's credit risk. Second, while recent studies have highlighted the benefits of considering similarities among borrowers for more effective credit risk prediction, they focus only on similarities derived from borrowers' observed characteristics but fail to account for unobserved similarities among them. To address these gaps, we propose a latent similarity-enhanced credit risk prediction model, which operationalizes the similarity between a pair of borrowers as a combination of observed and latent similarities between them. The former is measured based on borrowers' observed intrinsic characteristics and the latter is gauged from their unobserved intrinsic characteristics. We then develop a novel method to estimate the model parameters and predict borrowers' probabilities of delinquency. Therefore, the novelty of our study lies in its novel credit risk prediction model and a new method, which estimates the model parameters and incorporates both observed and latent similarities to predict borrowers' credit risks.

3. Problem Formulation and Model

Let $V = \{v_1, v_2, \cdots, v_n\}$ denote a set of n users of a credit product (e.g., credit card). They are granted credit by a financial institution and need to repay statement balances or installments periodically. For user v_i, let $\mathbf{x}_{i,t} \in \mathbb{R}^p$ represent p observed intrinsic characteristics of the user in billing cycle t, $i = 1, 2, \ldots, n$. A billing cycle is typically a month. Besides, we denote the credit delinquency outcome for user v_i in billing cycle t as a binary variable $y_{i,t}$, where $y_{i,t} = 1$ represents delinquency and $y_{i,t} = 0$ means no delinquency. More specifically, in the credit card example, $\mathbf{x}_{i,t}$ describes v_i's information available at the end of billing cycle t such as income level, education level, statement balance at the end of the billing cycle, and number of transactions in the billing cycle, whereas the outcome variable $y_{i,t}$ represents whether the user repays at least the minimum payment due within 21 days after the closing date of the billing cycle, i.e., $y_{i,t} = 0$ if the user repays the minimum payment due or more and $y_{i,t} = 1$ otherwise. Now we can formally define the credit risk prediction problem studied in this paper.

Credit Risk Prediction Problem. Let Ψ denote the current billing cycle. Given a set of n credit product users $V = \{v_1, v_2, \cdots, v_n\}$, their observed intrinsic characteristics $\mathbf{x}_{i,t}$ and delinquency outcomes $y_{i,t}$ in each of the previous L billing cycles t for $t = \Psi - L, \cdots, \Psi - 1$, as well as their characteristics $\mathbf{x}_{i,\Psi}$ in the current billing cycle Ψ, $i = 1, 2, \cdots, n$, we aim to predict these users' delinquency outcomes $y_{i,\Psi}$ in the current billing cycle.

We next describe our model for credit risk prediction. As a distinguishing feature, we take advantage of similarities among users for credit risk prediction beyond the sole use of their observed intrinsic characteristics. For the sake of simplicity, we drop the time subscript t from $\mathbf{x}_{i,t}$ and $y_{i,t}$, since our model is applicable to each billing cycle. Given any user pair $\{v_i, v_j\}$ from V with $i \neq j$, we denote the similarity between them as s_{ij}, which is modeled as follows:

$$s_{ij} = \alpha_i + \alpha_j + \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j + \mathbf{z}_i^T \mathbf{z}_j. \tag{1}$$

Here, $\tilde{\mathbf{x}}_i = \mathbf{x}_i / \|\mathbf{x}_i\|_2$ is the normalization of $\mathbf{x}_i$, where $\mathbf{x}_i$ represents the vector of observed intrinsic characteristics of user v_i. Thus, the term $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j$ corresponds to the observed similarity between v_i

and v_j, which is measured based on their observed intrinsic characteristics (e.g., those recorded in the data). The term $\alpha_i + \alpha_j + \mathbf{z}_i^T \mathbf{z}_j$ models the unobserved (latent) similarity between v_i and v_j, where α_i and $\mathbf{z}_i$ represent unobserved intrinsic characteristics of v_i. Using both the additive term $(\alpha_i + \alpha_j)$ and the product term $(\mathbf{z}_i^T \mathbf{z}_j)$ to model the unobserved similarity enriches the flexibility of our model, which is in line with the literature (Miller et al. 2009, Menon and Elkan 2011). According to Equation (1), the similarity between a pair of users consists of both the observed and unobserved similarities between them.

As similar users are likely to exhibit similar repayment behaviors, it follows that for a user $v_i \in V$, the user's delinquent probability p_i is implied by the user's observed intrinsic characteristics $\mathbf{x}_i$ as well as delinquent probabilities of other users who are similar to the user. Therefore, we model p_i as the probability of delinquency, conditioning on v_i's intrinsic characteristics $\mathbf{x}_i$, similarity information $\mathcal{S}_i = \{s_{ij} : v_j \in V, v_j \neq v_i\}$, and delinquent probabilities of other users $\mathcal{P}_{j|j\neq i} = \{p_j : v_j \in V, v_j \neq v_i\}$:

$$p_i = \Pr(y_i = 1 \,|\, \mathbf{x}_i, \mathcal{S}_i, \mathcal{P}_{j|j\neq i}). \tag{2}$$

Let θ_i denote v_i's log-odds of delinquency and $\theta_i = \log \dfrac{p_i}{1-p_i}$. Following the same idea of modeling p_i, we model θ_i as:

$$\theta_i = \mathbf{x}_i^T \boldsymbol{\beta} + c_i \sum_{v_j \in V:\, j \neq i} s_{ij} \theta_j, \tag{3}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the coefficient vector and θ_j denotes v_j's log-odds of delinquency, $j \neq i$. Here θ_i is expressed as the addition of two components: the first component $\mathbf{x}_i^T \boldsymbol{\beta}$ is a linear combination of v_i's observed intrinsic characteristics; the second component $\sum_{v_j \in V:\, j \neq i} s_{ij} \theta_j$ is related to similarity information and is the sum of other users' log-odds of delinquency weighted by the similarity between each of them and v_i. User-specific parameter c_i regulates the relative importance between the user's intrinsic component (first component) and similarity component (second component) in predicting θ_i. In sum, Equations (1) and (3) constitute our credit risk prediction model, the parameters of which are $\boldsymbol{\beta}$, c_i, α_i, and $\mathbf{z}_i$, $i = 1, 2, \ldots, n$. Our model distinguishes from existing models for credit risk prediction in its operationalization of similarity, i.e., the model of s_{ij} in Equation (1) and the similarity component in Equation (3).

4. Solution Method

In this section, we propose how to estimate our model parameters and predict users' delinquent probabilities based on the estimated parameters.

4.1. Objective Function

We define vectors $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)^T$, $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)^T$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^T$, and matrices $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)^T$, $\tilde{X} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_n)^T$, $Z = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n)^T$, $C = \text{diag}(\boldsymbol{c})$, $G = [s_{ij}]_{n\times n}$, where C is a $n \times n$ diagonal matrix with $c_1, c_2, \cdots, c_n$ as its diagonal elements and diagonal elements in G are defined to be 0. With these definitions, by the model of θ_i in Equation (3), it can be seen that

$$\boldsymbol{\theta} = X\boldsymbol{\beta} + CG\boldsymbol{\theta}. \tag{4}$$

Moreover, by the model of s_{ij} in Equation (1), we have

$$G = \mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T - 2\text{diag}(\boldsymbol{\alpha}) + \tilde{X}\tilde{X}^T + ZZ^T - \sum_{i=1}^{n}(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i + \mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T, \tag{5}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is an all-ones vector, $\text{diag}(\boldsymbol{\alpha})$ is a $n \times n$ diagonal matrix with $\alpha_1, \alpha_2, \cdots, \alpha_n$ as its diagonal elements, and $\mathbf{e}_i \in \mathbb{R}^n$ is a standard unit vector with its i^{th} element being 1 and other elements being 0. The derivations of Equations (4) and (5) are given in Appendices A.1 and A.2, respectively.

By Equation (4), we have

$$\boldsymbol{\theta} = (E_n - CG)^{-1}X\boldsymbol{\beta} =: M^{-1}X\boldsymbol{\beta}. \tag{6}$$

where $M = E_n - CG$ and E_n is an identity matrix of size n. Interestingly, Equation (6) reveals an explicit form for each user's log-odds of delinquency, which depends on the user's intrinsic characteristics in X, coefficient vector $\boldsymbol{\beta}$, and similarities among users captured in M (that involves Z, $\boldsymbol{\alpha}$, and $\boldsymbol{c}$).

To estimate parameters $\boldsymbol{\beta}$, Z, $\boldsymbol{\alpha}$ and $\boldsymbol{c}$, we obtain the negative log-likelihood $h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ from a training data set of users' observed intrinsic characteristics and their delinquency outcomes $\{\mathbf{x}_i, y_i\}_{i=1}^n$:

$$h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = \frac{1}{n}\sum_{i=1}^{n} -y_i\theta_i + \log(1 + e^{\theta_i}). \tag{7}$$

The derivation of $h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ is given in Appendix A.3. To avoid overfitting, we impose ridge-type regularization (Hastie et al. 2009, p. 61) on the parameters and use a penalized objective function $q(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ for parameter estimation,

$$q(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) + \frac{\lambda_Z}{2}\|Z\|_F^2 + \frac{\lambda_{\boldsymbol{\alpha}}}{2}\|\boldsymbol{\alpha}\|_2^2 + \frac{\lambda_{\boldsymbol{c}}}{2}\|\boldsymbol{c}\|_2^2 + \frac{\lambda_{\boldsymbol{\beta}}}{2}\|\boldsymbol{\beta}\|_2^2 \tag{8}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_2$ is the ℓ_2-norm, and λ_Z, $\lambda_{\boldsymbol{\alpha}}$, $\lambda_{\boldsymbol{c}}$ and $\lambda_{\boldsymbol{\beta}}$ are positive hyper-parameters.

4.2. Parameter Estimation

Given objective function (8), we propose an alternating minimization approach formally presented in Algorithm 1, which learns parameters $\boldsymbol{\beta}$, Z, $\boldsymbol{\alpha}$, and $\boldsymbol{c}$ of the credit risk prediction problem via blockwise minimization iteratively until convergence. Owing to its computational efficiency and excellent performance, alternating minimization has been widely adopted to solve nonconvex problems (Goodfellow et al. 2016) As shown, the algorithm takes a training data set of users' observed intrinsic characteristics and their delinquency outcomes $\{\mathbf{x}_i, y_i\}_{i=1}^n$ as inputs and produces estimations for the parameters. It first randomly initializes these parameters. In our implementation, we initialize each element in $\hat{Z}^{(0)}$ and $\hat{\boldsymbol{\beta}}^{(0)}$ by randomly sampling from a standard normal distribution, and set each element in $\hat{\boldsymbol{\alpha}}^{(0)}$ and $\hat{\boldsymbol{c}}^{(0)}$ to 0 and $\frac{1}{n}$, respectively, where $\hat{Z}^{(0)}$, $\hat{\boldsymbol{\beta}}^{(0)}$, $\hat{\boldsymbol{\alpha}}^{(0)}$, and $\hat{\boldsymbol{c}}^{(0)}$ denote the initializations of the parameters. The algorithm then iteratively updates one set of parameters while fixes the others until convergence (lines 4–11 of the algorithm). For example, in line 5, it minimizes the objective function with respect to the parameters in Z while fixing the rest of the parameters as constants.

Algorithm 1 An alternating minimization algorithm for parameter estimation.

Input: $\varepsilon > 0$: convergence threshold

$\{\mathbf{x}_i, y_i\}_{i=1}^n$: training data

Output: $\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}}$: parameter estimations

1: Set iteration count $r = 0$.
2: Initialize $\hat{Z}^{(r)}$, $\hat{\boldsymbol{\alpha}}^{(r)}$, $\hat{\boldsymbol{c}}^{(r)}$, $\hat{\boldsymbol{\beta}}^{(r)}$.
3: Compute $q^{(r)} = q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$ according to Equation (8).
4: **repeat**
5: $\quad \hat{Z}^{(r+1)} = \arg\min_Z q(\hat{\boldsymbol{\beta}}^{(r)}, Z, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.
6: $\quad \hat{\boldsymbol{\alpha}}^{(r+1)} = \arg\min_{\boldsymbol{\alpha}} q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \boldsymbol{\alpha}, \hat{\boldsymbol{c}}^{(r)})$.
7: $\quad \hat{\boldsymbol{c}}^{(r+1)} = \arg\min_{\boldsymbol{c}} q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \boldsymbol{c})$.
8: $\quad \hat{\boldsymbol{\beta}}^{(r+1)} = \arg\min_{\boldsymbol{\beta}} q(\boldsymbol{\beta}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r+1)})$.
9: $\quad$ Compute $q^{(r+1)} = q(\hat{\boldsymbol{\beta}}^{(r+1)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r+1)})$ according to Equation (8).
10: $\quad r = r + 1$.
11: **until** $|q^{(r)} - q^{(r-1)}| < \varepsilon$.
12: **return** $(\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}}) = (\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.

Although Algorithm 1 is built upon the general framework of alternating minimization, there are problem-specific obstacles to be conquered. Particularly, we need to find an efficient way to solve the optimization problems in lines 5–8 of the algorithm with stable numerical performance. In response, we analyze the properties of our problem in Propositions 1–3, propose an efficient algorithm to solve these optimization problems, and show the convergence of Algorithm 1 in Theorem 1. Specifically, we solve the optimization problems with gradient descent and derive the gradients in Proposition 1.

PROPOSITION 1. The gradients of the objective $q(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ w.r.t. Z, $\boldsymbol{\alpha}$, $\boldsymbol{c}$, and $\boldsymbol{\beta}$ are

$$\begin{aligned}
\frac{\partial q}{\partial Z} &= (M^{-1}X\boldsymbol{\beta})\Big(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}CZ\Big) + \big(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big)(\boldsymbol{\beta}^T X^T (M^{-1})^T Z) \\
&\quad - 2Z \odot \Big((M^{-1}X\boldsymbol{\beta}) \odot \big(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big)\mathbf{1}_d^T\Big) + \lambda_Z Z, \\
\frac{\partial q}{\partial \boldsymbol{\alpha}} &= (M^{-1}X\boldsymbol{\beta})\Big(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{1}_n\Big) + \big(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big)(\boldsymbol{\beta}^T X^T (M^{-1})^T \mathbf{1}_n) \\
&\quad - 2(M^{-1}X\boldsymbol{\beta}) \odot \big(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big) + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha}, \\
\frac{\partial q}{\partial \boldsymbol{c}} &= \Big((M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\Big) \odot \big(GM^{-1}X\boldsymbol{\beta}\big) + \lambda_{\boldsymbol{c}}\boldsymbol{c},
\end{aligned}$$

$$\frac{\partial q}{\partial \boldsymbol{\beta}} = X^T (M^{-1})^T \frac{\partial h}{\partial \boldsymbol{\theta}} + \lambda_{\boldsymbol{\beta}} \boldsymbol{\beta},$$

where $\frac{\partial h}{\partial \boldsymbol{\theta}} = \frac{1}{n}(-\mathbf{y} + \boldsymbol{\pi}_\theta)$, $\mathbf{y} = (y_1, \cdots, y_n)^T$, $\boldsymbol{\pi}_\theta = (\pi(\theta_1), \cdots, \pi(\theta_n))^T$, $\pi(\theta_i) = \frac{1}{1+e^{-\theta_i}}$, $\frac{\partial h}{\partial \boldsymbol{\theta}^T} = (\frac{\partial h}{\partial \boldsymbol{\theta}})^T$, $\odot$ denotes the element-wise product, and $\mathbf{1}_n \in \mathbb{R}^n$ is an all-ones vector.

Proof. See Appendix B.1.

With these explicit forms of gradients, it seems straightforward to solve these optimization problems with gradient descent. Nevertheless, the credit risk prediction problem typically involves a large number of users and it is computationally prohibitive to calculate the inverse of the $n \times n$ matrix M in the gradients, where n is the number of users. Computing the inverse of a $n \times n$ matrix can have a large time complexity of $\mathcal{O}(n^3)$, except for some special matrices (e.g., diagonal matrix).[3] To overcome this technical difficulty, we analyze the special structure of matrix M in Proposition 2 and propose an efficient way to compute its inverse in Proposition 3 by leveraging its special structure. We note that existing alternating minimization algorithms, such as the ALS algorithm (Hastie et al. 2015), are designed to overcome the computational challenge of conducting the singular value decomposition operation on a large and sparse matrix. Thus, they are not applicable to the specific technical difficulty encountered in our problem.

PROPOSITION 2. Define $D_n = E_n + C\big(2\text{diag}(\boldsymbol{\alpha}) + \sum_{i=1}^n (\tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_i + \mathbf{z}_i^T \mathbf{z}_i) \mathbf{e}_i \mathbf{e}_i^T\big)$, $Q_1 = (\mathbf{1}_n, \boldsymbol{\alpha}, \tilde{X}, Z)$ and $Q_2 = (\boldsymbol{\alpha}, \mathbf{1}_n, \tilde{X}, Z)$, where Q_1 is the column-wise concatenation of $\mathbf{1}_n, \boldsymbol{\alpha}, \tilde{X}, Z$ and Q_2 is the column-wise concatenation of $\boldsymbol{\alpha}, \mathbf{1}_n, \tilde{X}, Z$. By the definition of M in Equation (6), we have

$$M = D_n - C Q_1 Q_2^T.$$

Proof. See Appendix B.2.

It is noted that D_n is a diagonal matrix of size n and both Q_1 and Q_2 are $n \times (p+d+2)$ matrices, where p and d respectively denote the number of observed and unobserved characteristics of a user. In our intended applications, we have $p+d+2 \ll n$ as the combined number of observed and unobserved characteristics of a user is much less than the number of users. Moreover, the rank

[3] Although some efforts have been made to improve the computational efficiency of matrix inversion, LU decomposition remains a widely used method to compute the inverse of a matrix, which has a time complexity of $\mathcal{O}(n^3)$(Press et al. 2007).

of $CQ_1Q_2^T$ is at most $p+d+2$, since C is a diagonal matrix of size n and the rank of both Q_1 and Q_2 cannot exceed $p+d+2$. According to Proposition 2, M can be expressed as a low-rank correction (i.e., $CQ_1Q_2^T$) to a diagonal matrix of size n (i.e., D_n). Employing the special structure of M discovered in Proposition 2, we can efficiently compute M^{-1} by Proposition 3.

PROPOSITION 3. Based on Proposition 2, we have

$$M^{-1} = D_n^{-1} + D_n^{-1}CQ_1\left(E_{p+d+2} - Q_2^T D_n^{-1} CQ_1\right)^{-1} Q_2^T D_n^{-1}.$$

where E_{p+d+2} is an identity matrix of size $p+d+2$.

Proof. Given the special structure of M in Proposition 2, Proposition 3 follows from a direct application of the Sherman-Morrison-Woodbury formula (Press et al. 2007, p. 80).

By Proposition 3, computing the inverse of the large-sized matrix M can be equivalently converted to an expression that involves inverting matrices D_n and $E_{p+d+2} - Q_2^T D_n^{-1} CQ_1$, both of which can be efficiently inverted. Matrix D_n is a diagonal matrix, whose inverse can be directly and efficiently obtained by replacing each diagonal element with its reciprocal. Matrix $E_{p+d+2} - Q_2^T D_n^{-1} CQ_1$ is a square matrix of size $p+d+2$ (typically much smaller than n); thus it is more time-efficient to invert $E_{p+d+2} - Q_2^T D_n^{-1} CQ_1$ than inverting M. Equipped with Proposition 3, our proposed Algorithm 1 is particularly suitable for real-world credit risk prediction problems with a large number of users.

Based on Propositions 1–3, we propose how to solve the optimization problems in lines 5–8 of Algorithm 1. In particular, Algorithm 2 solves the optimization problem in line 5 of Algorithm 1 and updates parameter estimations $\hat{Z}$. Algorithms for solving the other optimization problems are similar to Algorithm 2 and thus omitted for reasons of space. As shown, Algorithm 2 updates $\hat{Z}$ via gradient descent, where the gradient can be computed using the formulas given in Propositions 1 and 3. It employs the backtracking line search with the Armijo rule (Nocedal and Wright 2006, p. 33) to choose step size (lines 6-9 in Algorithm 2), which ensures that the objective function keeps going downhill and converges (Bertsekas 1999). Next, we show the convergence of Algorithm 1.

THEOREM 1. The proposed Algorithm 1 for parameter estimation converges.

Proof. See Appendix B.3.

Taken together, Algorithm 1 iteratively updates estimations for parameters $\boldsymbol{\beta}$, Z, $\boldsymbol{\alpha}$ and $\boldsymbol{c}$ via blockwise minimization. Each optimization step in the algorithm is solved using Algorithm 2 or a similar procedure, which are developed based on Propositions 1–3. Theorem 1 guarantees the convergence of Algorithm 1.

Algorithm 2 A gradient descent algorithm for updating Z in line 5 of Algorithm 1.

Input: ε^Z: convergence threshold for updating Z, $\varepsilon^Z > 0$

γ, ζ: Armijo parameters, $0 < \gamma \leq 1/2$, $0 < \zeta < 1$

Output: $\hat{Z}^{(r+1)}$

1: Set iteration count $\rho = 0$.
2: Initialize $\breve{Z}^{(\rho)} = \hat{Z}^{(r)}$, $\breve{q}^{(\rho)} = q(\hat{\boldsymbol{\beta}}^{(r)}, \breve{Z}^{(\rho)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.
3: Define $q(Z) = q(\hat{\boldsymbol{\beta}}^{(r)}, Z, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.
4: **repeat**
5: Compute the gradient of $q(Z)$ at $Z = \breve{Z}^{(\rho)}$, denoted by $\frac{\partial q}{\partial Z}(\breve{Z}^{(\rho)})$, with the formulas given in Propositions 1 and 3.
6: Set stepsize $u = 1$.
7: **while** $q(\breve{Z}^{(\rho)} - u\frac{\partial q}{\partial Z}(\breve{Z}^{(\rho)})) > q(\breve{Z}^{(\rho)}) - \gamma u \|\frac{\partial q}{\partial Z}(\breve{Z}^{(\rho)})\|_F^2$ **do**
8: $u = \zeta u$. (choose step size u via a backtracking line search)
9: **end while**
10: $\breve{Z}^{(\rho+1)} = \breve{Z}^{(\rho)} - u\frac{\partial q}{\partial Z}(\breve{Z}^{(\rho)})$.
11: Compute $\breve{q}^{(\rho+1)} = q(\hat{\boldsymbol{\beta}}^{(r)}, \breve{Z}^{(\rho+1)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$ according to Equation (8).
12: $\rho = \rho + 1$.
13: **until** $|\breve{q}^{(\rho)} - \breve{q}^{(\rho-1)}| < \varepsilon^Z$.
14: **return** $\hat{Z}^{(r+1)} = \breve{Z}^{(\rho)}$.

4.3. Credit Risk Prediction

We use training data in the previous billing cycles to estimate the parameters. Specifically, we employ training data of users' observed intrinsic characteristics $\mathbf{x}_{i,t}$ and delinquency outcomes $y_{i,t}$ in each of the previous L billing cycles t, where $t = \Psi - L, \cdots, \Psi - 1$, $i = 1, 2, \cdots, n$, and Ψ denotes the current billing cycle. We subsequently construct the negative log-likelihood $h_t(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ for each of the previous billing cycle the same way as that of Equation (7) and form the composite

negative log-likelihood objective function $\sum_{t=\Psi-L}^{\Psi-1} h_t(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ (Varin et al. 2011). We then substitute $h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ in Equation (8) with the composite negative log-likelihood objective function and the proposed Algorithm 1 remains applicable for parameter estimation. In doing so, our model can effectively learn parameters $\mathbf{z}_i$ and α_i based on users' diverse historical repayment behaviors and time-variant intrinsic characteristics. With the estimated parameters $(\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}})$ and users' observed intrinsic characteristics $\mathbf{x}_{i,\Psi}$ in the current billing cycle Ψ, $i = 1, 2, \cdots, n$, we can compute their log-odds of delinquency $\boldsymbol{\theta}_\Psi = (\theta_{1,\Psi}, \theta_{2,\Psi}, \cdots, \theta_{n,\Psi})^T$ using Equation (6). The predicted individual delinquent probabilities are then given by $p_{i,\Psi} = \pi(\theta_{i,\Psi}) = \frac{1}{1+e^{-\theta_{i,\Psi}}}$, $i = 1, 2, \cdots, n$.

In sum, our proposed credit risk prediction method is distinct from existing methods in its operationalization of similarity. Specifically, our method models similarity in Equation (1) and incorporates it into credit risk prediction in Equation (3). It then estimates the model parameters using Algorithms 1 and 2, based on the findings in Propositions 1–3 and Theorem 1. These algorithms, propositions, and theorem constitute the methodological novelty of our study.

5. Extensions

We next extend our model to solve the multiclass credit risk prediction problem in Section 5.1 and the numerical credit risk prediction problem in Section 5.2.

5.1. Multiclass Credit Risk Prediction Problem

For the multiclass credit risk prediction problem, credit risk outcome y_i takes one of the more than two categories, i.e., $y_i \in \{0, 1, \cdots, K\}$ with $K \geq 2$. For example, credit card companies make a significant portion of their revenues from interests charged on cards with partially paid statement balances.[4] Therefore, it is reasonable to further categorize non-delinquent users into two subgroups: those who pay off their statement balances (i.e., no interests charged) and those who make partial payments (i.e., interests charged). In this example, y_i can be one of the three outcomes defined below:

$$y_i = \begin{cases} 2 & \text{if user } v_i \text{ is delinquent (failing to pay the minimum due),} \\ 1 & \text{if user } v_i \text{ is non-delinquent but only makes partial payment,} \\ 0 & \text{if user } v_i \text{ is non-delinquen and pays off the statement balance.} \end{cases}$$

[4] See https://www.experian.com/blogs/ask-experian/how-do-credit-card-companies-make-money (last accessed on March 26, 2022).

Let $p_{k,i}$ be the probability that the credit risk outcome of user v_i is k. To define the log-odds of being a credit risk outcome, we set credit risk outcome 0 as the reference. Accordingly, for user v_i, we define the user's log-odds $\theta_{k,i}$ of being credit risk outcome k as $\theta_{k,i} = \log \frac{p_{k,i}}{p_{0,i}}$ (i.e., the logarithm of the probability of being credit risk outcome k to that of being outcome 0), for $k = 1, 2, \cdots, K$. Following the same reasoning of modeling log-odds for the binary credit risk prediction problem, we model $\theta_{k,i}$ as

$$\theta_{k,i} = \mathbf{x}_i^T \boldsymbol{\beta}_k + c_i \sum_{v_j \in V: j \neq i} s_{ij} \theta_{k,j} \qquad \text{for } k = 1, 2 \ldots, K, \tag{9}$$

where $\boldsymbol{\beta}_k \in \mathbb{R}^p$ is the coefficient vector associated with credit risk outcome k and s_{ij} is the similarity between v_i and v_j, which is modeled by Equation (1). According to Equation (9), $\theta_{k,i}$ depends on user v_i's observed intrinsic characteristics $\mathbf{x}_i$ as well as log-odds of other users who are similar to the user. Parameter c_i denotes the relative importance between the component of the user's intrinsic characteristics and the similarity component in determining $\theta_{k,i}$.

Denoting $\boldsymbol{\theta}_k = (\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,n})^T$ and using notations X, C, G, and E_n defined in Section 4.1, we can derive the following equation from Equation (9).

$$\boldsymbol{\theta}_k = X\boldsymbol{\beta}_k + CG\boldsymbol{\theta}_k, \tag{10}$$

where G is given in Equation (5). The derivation of Equation (10) is similar to that of Equation (4) shown in Appendix A.1 and thus omitted. By Equation (10), we obtain the explicit form for the log-odds $\boldsymbol{\theta}_k$:

$$\boldsymbol{\theta}_k = (E_n - CG)^{-1} X\boldsymbol{\beta}_k =: M^{-1} X\boldsymbol{\beta}_k, \tag{11}$$

where $M = E_n - CG$. For the simplicity of notation, we denote the coefficients set $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_K)$ as $\boldsymbol{\beta}_{\{1:K\}}$. To estimate parameters $\boldsymbol{\beta}_{\{1:K\}}$, Z, $\boldsymbol{\alpha}$, and $\boldsymbol{c}$ for the multiclass credit risk prediction problem, we derive the negative log-likelihood $h^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ from a training data set of users' observed intrinsic characteristics and their multiclass credit risk outcomes $\{\mathbf{x}_i, y_i\}_{i=1}^n$:

$$h^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = \frac{1}{n} \sum_{i=1}^{n} \Big((\sum_{k=1}^{K} -\delta_{k,i} \theta_{k,i}) + \log(1 + \sum_{k=1}^{K} e^{\theta_{k,i}}) \Big), \tag{12}$$

where $\delta_{k,i} = 1$ if $y_i = k$ and $\delta_{k,i} = 0$ otherwise. The derivation of the equation is given in Appendix C.1. Consequently, the regularized objective function is given by,

$$q^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = h^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) + \frac{\lambda_Z}{2}\|Z\|_F^2 + \frac{\lambda_{\boldsymbol{\alpha}}}{2}\|\boldsymbol{\alpha}\|_2^2 + \frac{\lambda_{\boldsymbol{c}}}{2}\|\boldsymbol{c}\|_2^2 + \sum_{k=1}^{K}\frac{\lambda_{\boldsymbol{\beta}}}{2}\|\boldsymbol{\beta}_k\|_2^2, \tag{13}$$

where λ_Z, $\lambda_{\boldsymbol{\alpha}}$, $\lambda_{\boldsymbol{c}}$ and $\lambda_{\boldsymbol{\beta}}$ are positive hyper-parameters.

We estimate the parameters by optimizing objective function (13) with an alternating minimization algorithm. The algorithm is similar to Algorithm 1, except that it uses the gradients given in Proposition 4 and updates $\boldsymbol{\beta}_{\{1:K\}}$ instead of $\boldsymbol{\beta}$ in each iteration. We give the algorithm in Appendix C.2 for the sake of space.

PROPOSITION 4. The gradients of the objective $q^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ w.r.t. Z, $\boldsymbol{\alpha}$ $\boldsymbol{c}$, and $\boldsymbol{\beta}_{\{1:K\}}$ are

$$\begin{aligned}
\frac{\partial q^m}{\partial Z} &= \sum_{k=1}^{K}[(M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}CZ) + (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^T X^T(M^{-1})^T Z) \\
&\quad - 2Z \odot \big((M^{-1}X\boldsymbol{\beta}_k) \odot (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})\mathbf{1}_d^T\big)] + \lambda_Z Z, \\
\frac{\partial q^m}{\partial \boldsymbol{\alpha}} &= \sum_{k=1}^{K}[(M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{1}_n) + (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^T X^T(M^{-1})^T\mathbf{1}_n) \\
&\quad - 2(M^{-1}X\boldsymbol{\beta}_k) \odot (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})] + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha}, \\
\frac{\partial q^m}{\partial \boldsymbol{c}} &= \sum_{k=1}^{K}[\big((M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}\big) \odot \big(GM^{-1}X\boldsymbol{\beta}_k\big)] + \lambda_{\boldsymbol{c}}\boldsymbol{c}, \\
\frac{\partial q^m}{\partial \boldsymbol{\beta}_k} &= X^T(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k} + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}_k \quad \text{for} \quad k = 1, 2, \ldots, K,
\end{aligned}$$

where $\frac{\partial h^m}{\partial \boldsymbol{\theta}_k} = \frac{1}{n}(-\boldsymbol{\delta}_k + \boldsymbol{\pi}_k^m)$, $\boldsymbol{\delta}_k = (\delta_{k,1}, \cdots, \delta_{k,n})^T$, $\boldsymbol{\pi}_k^m = (\pi_{k,1}^m, \cdots, \pi_{k,n}^m)^T$, $\pi_{k,i}^m = \frac{e^{\theta_{k,i}}}{1+\sum_{\tilde{k}=1}^{K} e^{\theta_{\tilde{k},i}}}$, and $\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T} = (\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})^T$.

Proof. See Appendix C.3.

It is noted that Propositions 2 and 3 remain applicable to efficiently compute the inverse of the large-sized matrix M in the gradients given in Proposition 4. As a result, Algorithm 2 is still applicable to the multiclass credit risk prediction problem, with a simple modification of using the gradients in Proposition 4 for the multiclass case. Therefore, this subsection generalizes the credit risk prediction problem defined in Section 3 to the multiclass scenario and maintains the convenient computational scale-up.

Applying the algorithm developed in this subsection to the training data of users' observed intrinsic characteristics and credit risk outcomes in the previous L billing cycles, we obtain parameter estimates $(\hat{\boldsymbol{\beta}}_{\{1:k\}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}})$. With these parameter estimates and users' observed intrinsic characteristics $\mathbf{x}_{i,\Psi}$ in the current billing cycle Ψ, $i = 1, 2, \ldots, n$, we employ Equation (11) to compute the log-odds $\boldsymbol{\theta}_{k,\Psi} = (\theta_{k,1,\Psi}, \theta_{k,2,\Psi}, \cdots, \theta_{k,n,\Psi})^T$ for each credit risk outcome k, $k = 1, 2, \ldots, K$. Recall that $\theta_{k,i,\Psi} = \log \frac{p_{k,i,\Psi}}{p_{0,i,\Psi}}$ and $\sum_{k=0}^{K} p_{k,i,\Psi} = 1$. We thus can calculate the predicted probability of each credit risk outcome for user v_i in the current billing cycle Ψ as

$$p_{0,i,\Psi} = \frac{1}{1 + \sum_{\tilde{k}=1}^{K} e^{\theta_{\tilde{k},i,\Psi}}}, \quad p_{k,i,\Psi} = \frac{e^{\theta_{k,i,\Psi}}}{1 + \sum_{\tilde{k}=1}^{K} e^{\theta_{\tilde{k},i,\Psi}}} \quad \text{for} \quad k = 1, 2, \cdots, K.$$

5.2. Numerical Credit Risk Prediction Problem

Numerical credit risk indicators, such as repayment rate (i.e., the proportion of repaid statement balance), uncover more granular information about users' credit risks than categorical credit risk outcomes. For example, non-delinquent users who repay their minimum payment dues can be further differentiated according to their repayment rates. We thus extend our model to predict numerical credit risk indicators. Let μ_i denote the estimation of user v_i's numerical credit risk indicator y_i. Similar to the binary credit risk prediction problem, we model μ_i as the addition of two components: one component is defined on v_i's observed intrinsic characteristics $\mathbf{x}_i$ and the other one is based on similarity information. Accordingly, we have

$$\mu_i = \mathbf{x}_i^T \boldsymbol{\beta} + c_i \sum_{v_j \in V : j \neq i} s_{ij} \mu_j, \tag{14}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ denotes the coefficient vector, c_i represents the relative importance between the two components, and s_{ij} is defined in Equation (1). We denote $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_n)^T$ and use notations X, C, G, and E_n defined in Section 4.1. Similar to the derivation of Equation (4) in Appendix A.1, we obtain the following equation from Equation (14):

$$\boldsymbol{\mu} = X\boldsymbol{\beta} + CG\boldsymbol{\mu}, \tag{15}$$

which implies that

$$\boldsymbol{\mu} = (E_n - CG)^{-1} X\boldsymbol{\beta} =: M^{-1} X\boldsymbol{\beta}, \tag{16}$$

where $M = E_n - CG$.

To estimate parameters $\boldsymbol{\beta}$, Z, $\boldsymbol{\alpha}$, and $\boldsymbol{c}$, we employ a penalized ordinary least squares (OLS) approach and obtain the OLS loss function h^n from a training data set of users' observed intrinsic characteristics and their numerical credit risk indicators $\{\mathbf{x}_i, y_i\}_{i=1}^n$:

$$h^n(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = \frac{1}{2n}(Y - \boldsymbol{\mu})^T(Y - \boldsymbol{\mu}), \tag{17}$$

where $Y = (y_1, y_2, \cdots, y_n)^T$. The penalized objective $q^n(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ is then given by

$$q^n(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = h^n(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) + \frac{\lambda_Z}{2}\|Z\|_F^2 + \frac{\lambda_{\boldsymbol{c}}}{2}\|\boldsymbol{c}\|_2^2 + \frac{\lambda_{\boldsymbol{\alpha}}}{2}\|\boldsymbol{\alpha}\|_2^2 + \frac{\lambda_{\boldsymbol{\beta}}}{2}\|\boldsymbol{\beta}\|_2^2, \tag{18}$$

where λ_Z, $\lambda_{\boldsymbol{\alpha}}$, $\lambda_{\boldsymbol{c}}$ and $\lambda_{\boldsymbol{\beta}}$ are positive hyper-parameters.

An alternating minimization algorithm is proposed to estimate the parameters by optimizing objective function (18). Note that given $\hat{Z}$, $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{c}}$, estimating parameters $\boldsymbol{\beta}$ reduces to the parameter estimation for a ridge linear regression problem and has a closed-form solution (e.g., Hastie et al. 2009, p.64):

$$\hat{\boldsymbol{\beta}} = \left(X_{\hat{M}}^T X_{\hat{M}} + \lambda_{\boldsymbol{\beta}} E_p\right)^{-1} X_{\hat{M}}^T Y, \tag{19}$$

where $X_{\hat{M}} = \hat{M}^{-1}X$, $\hat{M} = E_n - \hat{C}\hat{G}$ according to Equation (16), $\hat{C} = \mathrm{diag}(\hat{\boldsymbol{c}})$, $\hat{G}$ is computed by plugging $\hat{Z}$ and $\hat{\boldsymbol{\alpha}}$ into Equation (5), and E_p is an identity matrix of size p. The parameter estimation algorithm described in Appendix D.1 is similar to Algorithm 1, except that it updates $\boldsymbol{\beta}$ according to Equation (19) and other parameters with the gradients given in Proposition 5. The algorithm enjoys the same advantages as demonstrated from Propositions 2 and 3 for the efficient computation of M^{-1} in the gradients given in Proposition 5. We can obtain parameter estimates $(\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}})$ by applying the algorithm to the training data of users' observed intrinsic characteristics and credit risk indicators in the previous L billing cycles. With the estimated parameters and users' observed intrinsic characteristics $\mathbf{x}_{i,\Psi}$ in the current billing cycle Ψ, $i = 1, 2, \ldots, n$, their predicted credit risk indicators, $\boldsymbol{\mu}_\Psi$, can be computed by Equation (16).

PROPOSITION 5. The gradients of the objective $q^n(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c})$ with respect to Z, $\boldsymbol{\alpha}$ and $\boldsymbol{c}$ are.

$$\frac{\partial q^n}{\partial Z} = (M^{-1}X\boldsymbol{\beta})\left(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}CZ\right) + \left(C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\right)(\boldsymbol{\beta}^T X^T (M^{-1})^T Z)$$
$$- 2Z \odot \left((M^{-1}X\boldsymbol{\beta}) \odot \left(C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\right)\mathbf{1}_d^T\right) + \lambda_Z Z,$$

$$\frac{\partial q^n}{\partial \boldsymbol{\alpha}} = (M^{-1}X\boldsymbol{\beta})(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{1}_n) + (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n)$$
$$- 2(M^{-1}X\boldsymbol{\beta}) \odot (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}) + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha},$$
$$\frac{\partial q^n}{\partial \boldsymbol{c}} = \big((M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\big) \odot \big(GM^{-1}X\boldsymbol{\beta}\big) + \lambda_{\boldsymbol{c}}\boldsymbol{c},$$

where $\frac{\partial h^n}{\partial \boldsymbol{\mu}} = \boldsymbol{\mu} - Y$ and $\frac{\partial h}{\partial \boldsymbol{\mu}^T} = (\frac{\partial h}{\partial \boldsymbol{\mu}})^T$.

Proof. See Appendix D.2.

6. Empirical Evaluation

6.1. Data and Evaluation Procedure

We obtained data from a non-U.S. commercial bank's provincial branch. One data set contains 11,761 credit card users' repayment information over 22 billing cycles/months. For every user in each billing cycle, the data set shows the user's minimum payment due as well as the repayment amount within 21 days after the closing date of the billing cycle. A user v_i is delinquent for billing cycle t if the user's repayment amount is less than the minimum payment due and we set $y_{i,t} = 1$; otherwise, the user is considered non-delinquent and $y_{i,t} = 0$. On average, there are 3,932 delinquent users (or 33.43% of the credit card users in our data set) in a billing cycle.[5] The delinquency rate ranges from from 28.89% to 35.07% across 22 billing cycles with a median of 33.81%.

Another data set depicts the billing information of these users over the same time period. In particular, it includes the spending amount and number of transactions by a user in a billing cycle as well as the user's statement balance, accumulated interest owed by the end of the billing cycle, and the number of historical delinquencies. The third data set contains these users' demographic information, including the type of residence (e.g., rent or own), income level, account holding branch, education level.[6] User v_i's billing and demographic information in billing cycle t was used to construct the user's observed intrinsic characteristics $\mathbf{x}_{i,t}$.

[5]The delinquency rate calculated from our dataset is higher than that reported by the industry because our dataset excludes data on inactive users who do not have spending data for a long time period (e.g., six consecutive months). On the other hand, when a financial institution calculates its delinquency rate, inactive users are included in the calculation and treated as "no delinquency".

[6]In this country, it is prohibited to employ protected attributes such as gender, age, and marital status for credit risk prediction. Hence, these attributes are not included in the demographic information.

We next detail the evaluation procedure. Let Ψ be the current billing cycle. We constructed training data using users' intrinsic characteristics $\mathbf{x}_{i,t}$ and delinquency outcomes $y_{i,t}$ in each of the previous 12 billing cycles t, $t = \Psi - 12, \cdots, \Psi - 1$. Using the training data, we then trained our and each benchmark method to predict delinquent probability $p_{i,\Psi}$ for every user in the current billing cycle based on $\mathbf{x}_{i,\Psi}$.[7] The performance of a method was evaluated using common metrics for credit risk prediction that include precision, recall, F1 score, and AUC (Khandani et al. 2010, Lu et al. 2023). Specifically, for a method, top-Q users refer to the group of Q users with the highest delinquent probabilities predicted by this method and deemed as delinquent by the method. Let TP_Ψ denote the number of true delinquent users among the top-Q users and n_Ψ^d be the number of true delinquent users in billing cycle Ψ. Precision is the fraction of predicted delinquent users who are actually delinquent:

$$\text{precision} = \frac{TP_\Psi}{Q}.$$

Recall is the fraction of true delinquent users who are correctly predicted as delinquent:

$$\text{recall} = \frac{TP_\Psi}{n_\Psi^d}.$$

F1 score is the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

In our evaluation, AUC reveals the probability that a randomly chosen delinquent user will be predicted to have a higher delinquent probability than a randomly chosen nondelinquent user (Fang et al. 2013). The value of AUC is in the range of [0,1] and a higher AUC indicates better performance of credit risk prediction.

We carefully chose representative existing methods for credit risk prediction as benchmarks. Logistic regression is the most widely used method and industry standard for credit risk prediction (Thomas et al. 2017). Moreover, the key methodological difference between our latent similarity-enhanced method (LSE) and ridge logistic regression (RLR, Le Cessie and Van Houwelingen 1992)

[7] The objective of behavioral scoring is to employ borrowers' historical billing and demographic data as well as their historical delinquency outcomes to predict their delinquency probabilities in the current billing cycle. Therefore, we split our training and test data on the temporal domain.

is the consideration of similarity by LSE, where the similarity between a pair of users consists of both the observed and latent similarities between them. Therefore, RLR was chosen as a useful benchmark. The performance difference between our method and RLR not only reveals the practical value of our method but also demonstrates the performance contribution from its methodological novelty. Besides, we benchmarked against two representative methods for credit risk prediction: multilayer perceptron (MLP) and support vector machine (SVM). Both methods have been shown excellent performance for credit risk prediction (Baesens et al. 2003). We also compared our method with a popular ensemble method for credit risk prediction – extreme gradient boosting (XGBoost, Chen and Guestrin 2016). Recent studies have proposed deep learning-based methods for credit risk prediction. A representative method in this category is developed by Babaev et al. (2019), who propose a recurrent neural network (RNN) model based on gated recurrent unit (GRU) for credit risk prediction. We adapted this method for our data set and named it as RNN-GRU. In addition, Wang et al. (2021) propose a temporal-aware Graph Neural Network (TemGNN) method for credit risk prediction over dynamic graphs. To apply this method, we first utilized users' observed intrinsic characteristics to construct a similarity network among them for each billing cycle and then adapted TemGNN to these constructed dynamic similarity networks for credit risk prediction. We designate this benchmark method as SIM-TemGNN. Table 1 summarizes the methods compared in our evaluation.[8]

Table 1 Summary of Methods Compared in the Evaluation

Method	Note
LSE	Latent Similarity-Enhanced method, our method
RLR	Ridge Logistic Regression, benchmark
MLP	Multilayer Perceptron, benchmark
SVM	Support Vector Machine, benchmark
XGBoost	eXtreme Gradient Boosting, benchmark
RNN-GRU	Recurrent Neural Network based on Gated Recurrent Unit, benchmark
SIM-TemGNN	Temporal-aware GNN Applied to Dynamic Similarity Networks, benchmark

To tune hyperparameters for each compared method, we trained the method with users' observed intrinsic characteristics and delinquency outcomes in billing cycles 1 to 12 to predict each user's

[8]The proposed LSE was implemented in R version 3.6. All other benchmarks were implemented in Python version 3.8.

delinquent probability in billing cycle 13 ($\Psi = 13$). Values of hyperparameter that enabled the method to achieve its best performance (measured with the metrics elaborated above) were chosen. For our method, we set its hyperparameters $\lambda_Z = 0.01$, $\lambda_{\boldsymbol{c}} = 0.5$, $\lambda_{\boldsymbol{\alpha}} = 1$, $\lambda_{\boldsymbol{\beta}} = 0.001$, and $d = 2$, and set convergence thresholds in Algorithms 1 and 2 to 0.001. Each user has a connection score α_i that needs to be learned whereas regression coefficients $\boldsymbol{\beta}$ are shared by all users. Therefore, a large $\lambda_{\boldsymbol{\alpha}}$ was chosen to reduce the model complexity while a small $\lambda_{\boldsymbol{\beta}}$ was selected to help learn the relation between users' observed intrinsic characteristics and delinquency outcomes. The Armijo parameters γ and ζ in Algorithm 2 were set to 0.1 and 0.5, respectively. For RLR, its ridge parameter was set to 10^{-4}. MLP had 3 hidden layers of sizes 128, 64, and 32, respectively. SVM employed the RBF kernel (Burges 1998) with its regularization parameter being set to 10. XGBoost used tree boosters with a maximum depth of 7 and the number of boosting rounds was set to 400. RNN-GRU was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 256, and the size of its hidden states was set to 128. For SIM-TemGNN, to construct a similarity network for each billing cycle, we calculated the Euclidean distance between each pair of users based on their observed intrinsic characteristics $\mathbf{x}_{i,t}$ in each billing cycle. For each user, the top 10 closest users (with the smallest distances) were selected as adjacent nodes in the similarity network for that billing cycle. The MLP component of SIM-TemGNN had 3 hidden layers with sizes of 128, 64, and 32, respectively, its GNN component consisted of 2 layers, each with an embedding size of 32, and the size of each hidden state of its LSTM component was set to 64.

6.2. Evaluation Results and Analysis

Following the evaluation procedure, we conducted experiments to evaluate the performance of each method in Table 1 by varying current billing cycle from $\Psi = 14$ to $\Psi = 22$. Given the average number of delinquent users in a billing cycle being 3,932, we set $Q = 3,900$ in all the experiments. Take the experiment with $\Psi = 14$ as an example. In this experiment, each method was trained with users' observed intrinsic characteristics and delinquency outcomes in billing cycles 2 to 13 to predict each user's delinquent probability in billing cycle 14 and the top-$3,900$ users with the highest delinquent probabilities were predicted to be delinquent by the method.

Table 2 Performance Comparison for Credit Risk Prediction Problem.

Metric	Method	$\Psi=14$	$\Psi=15$	$\Psi=16$	$\Psi=17$	$\Psi=18$	$\Psi=19$	$\Psi=20$	$\Psi=21$	$\Psi=22$	Mean	Std.
Precision ($Q=3,900$)	LSE (Our Method)	0.829	0.837	0.843	0.848	0.851	0.843	0.839	0.841	0.836	0.841	0.006
	RLR	0.579	0.573	0.569	0.556	0.581	0.563	0.563	0.566	0.568	0.569	0.008
	MLP	0.660	0.677	0.634	0.674	0.693	0.700	0.643	0.664	0.695	0.671	0.023
	SVM	0.613	0.624	0.631	0.618	0.627	0.628	0.614	0.612	0.613	0.620	0.008
	XGBoost	0.770	0.775	0.778	0.775	0.792	0.785	0.771	0.775	0.774	0.777	0.007
	RNN-GRU	0.747	0.752	0.751	0.754	0.757	0.751	0.753	0.747	0.749	0.751	0.003
	SIM-TemGNN	0.711	0.734	0.721	0.715	0.724	0.739	0.728	0.717	0.728	0.724	0.009
Recall ($Q=3,900$)	LSE (Our Method)	0.810	0.809	0.823	0.852	0.819	0.832	0.842	0.858	0.832	0.831	0.017
	RLR	0.566	0.553	0.556	0.559	0.559	0.555	0.565	0.578	0.564	0.562	0.008
	MLP	0.644	0.654	0.619	0.678	0.667	0.690	0.645	0.678	0.691	0.663	0.024
	SVM	0.598	0.603	0.616	0.622	0.604	0.620	0.616	0.625	0.609	0.612	0.009
	XGBoost	0.752	0.749	0.760	0.780	0.762	0.775	0.773	0.790	0.770	0.768	0.013
	RNN-GRU	0.729	0.726	0.733	0.758	0.729	0.741	0.755	0.762	0.744	0.742	0.014
	SIM-TemGNN	0.694	0.710	0.704	0.719	0.697	0.729	0.730	0.731	0.724	0.715	0.015
F1 Score ($Q=3,900$)	LSE (Our Method)	0.819	0.823	0.833	0.850	0.834	0.837	0.841	0.849	0.834	0.840	0.007
	RLR	0.572	0.563	0.562	0.558	0.570	0.559	0.564	0.572	0.566	0.564	0.005
	MLP	0.652	0.666	0.627	0.676	0.680	0.695	0.644	0.671	0.693	0.669	0.025
	SVM	0.605	0.613	0.623	0.620	0.615	0.624	0.615	0.618	0.611	0.618	0.005
	XGBoost	0.761	0.762	0.769	0.777	0.777	0.780	0.772	0.782	0.772	0.776	0.005
	RNN-GRU	0.738	0.739	0.742	0.756	0.743	0.746	0.754	0.755	0.747	0.749	0.006
	SIM-TemGNN	0.703	0.722	0.713	0.717	0.710	0.734	0.729	0.724	0.726	0.722	0.009
AUC	LSE (Our Method)	0.931	0.934	0.938	0.944	0.937	0.941	0.943	0.942	0.937	0.939	0.004
	RLR	0.752	0.746	0.744	0.747	0.756	0.742	0.752	0.761	0.754	0.750	0.006
	MLP	0.806	0.817	0.787	0.829	0.822	0.840	0.804	0.826	0.838	0.819	0.017
	SVM	0.770	0.781	0.786	0.790	0.782	0.789	0.787	0.793	0.784	0.785	0.007
	XGBoost	0.894	0.891	0.896	0.906	0.902	0.909	0.901	0.908	0.899	0.901	0.006
	RNN-GRU	0.880	0.878	0.883	0.893	0.881	0.888	0.890	0.892	0.887	0.886	0.005
	SIM-TemGNN	0.855	0.865	0.844	0.852	0.859	0.863	0.849	0.856	0.863	0.856	0.007

Table 2 summarizes the performance of the investigated methods, and LSE is shown to outperform each benchmark across different evaluation billing cycles in all four performance metrics. Averaged across Ψ, the average precision of LSE is 0.841; that is, on average, 84.1% of the users predicted to be delinquent by our method are truly delinquent. In comparison, the average precision of XGBoost (the best performing benchmark) is 0.777, which is 8.18% lower than that of our method. In addition, LSE surpasses XGBoost by 8.19% in average recall, 8.28% in average F1 score, and 4.20% in average AUC. By applying the paired t-test to the experimental results in Table 2, the performance improvement by LSE over each benchmark in every evaluation metric is statistically significant ($p < 0.001$). The observed performance advantage of our method over the benchmarks can be attributed to its methodological novelty, i.e., our new model of user similarity for credit risk prediction. This is particularly evidenced by the performance improvements of our

method over RLR (i.e., 47.87% in average precision, 47.89% in average recall, 48.79% in average F1 score, and 25.06% in average AUC), as our method is largely reduced to RLR if we drop the methodological novelty from it. Therefore, these performance improvements reveal the performance contributions by the methodological novelty of our method. Besides, the outperformance of LSE over SIM-TemGNN (i.e., 16.10% in average precision, 16.11% in average recall, 16.32% in average F1 score, and 9.62% in average AUC) also sheds light on the value of latent similarity for credit risk prediction, even when temporal information is explicitly taken into account. Unlike SIM-TemGNN which relies on observed similarities among users for credit risk prediction, our method is built on both observed and unobserved similarities among users.

Moreover, we conducted an ablation study to empirically investigate the contribution of factors $\mathbf{z}$ and α to the performance of our method. In the ablation study, we only dropped the term involving factor $\mathbf{z}$ from Equation (1) but kept Equation (3) intact. We called this simplified method without factor $\mathbf{z}$ as LSE-Z. The performance difference between our method LSE and LSE-Z reveals the contribution of factor $\mathbf{z}$ to the predictive accuracy of our method. Similarly, we only dropped the term involving factor α from Equation (1), and referred to the resulted method without factor α as LSE-α. We also removed both the term involving factor $\mathbf{z}$ and the term involving factor α from Equation (1), and referred to the resulted method without both factors as LSE-Z-α. We compared the AUC of the four methods for billing cycle $\Psi = 14$. As reported in Table 3, LSE outperforms LSE-Z by 3.79% because of the consideration of factor $\mathbf{z}$, surpasses LSE-α by 2.87% due to the inclusion of factor α, and exceeds LSE-Z-α by 10.05% owing to the incorporation of both factors.

Table 3 Ablation Study ($\Psi = 14$)

Method	AUC	AUC Improvement by LSE
LSE (Our Method)	0.931	
LSE-Z	0.897	3.79%
LSE-α	0.905	2.87%
LSE-Z-α	0.846	10.05%

In addition, we conducted an evaluation by reducing the number of training billing cycles. This allows us to investigate how the performance benefits of the proposed method are affected when

a bank intends to predict users' credit risks at an earlier stage and has limited observations of users' historical behaviors. Averaged across test billing cycles $\Psi = 14, 15, ..., 22$, Table 4 presents the average AUC for LSE (our method) and XGBoost (the best-performing benchmark) respectively, with the number of training billing cycles reduced from 12 to 6. As expected, while our method continues to outperform the best-performing benchmark, its performance improvement declines as the number of training billing cycles is reduced. The main advantage of our method lies in leveraging unobserved similarities among users to enhance credit risk prediction. When fewer training billing cycles are employed, less amount of historical data is available for learning unobserved similarities among users. Consequently, it becomes more challenging for our method to effectively learn these unobserved similarities, thereby diminishing the performance advantage of our method. Further analyses in Appendix E showcase the performance of our method across different subgroups of users, illustrate its robustness by varying the values of Q, and demonstrate its effectiveness in reconstructing latent similarities among users. In particular, the subgroup analyses in Appendix E.1 indicate that our method is particularly effective in classifying users for whom the similarity component plays a more significant predictive role, as well as those who exhibit strong similarities with a larger number of individuals. These observations align with the rationale of our method. That is, if a user shows strong similarities with a greater number of individuals, our method can harness additional predictive power from the behaviors of these similar counterparts, leading to improved predictive performance.

Table 4 Performance Comparison on Average AUC: Reducing Number of Training Billing Cycles

# of Training Billing Cycles	12	11	10	9	8	7	6
LSE (Our Method)	0.939	0.938	0.935	0.931	0.928	0.922	0.913
XGBoost	0.901	0.902	0.899	0.896	0.897	0.894	0.893
Improvement by LSE	4.20%	3.99%	4.00%	3.90%	3.45%	3.13%	2.24%

6.3. Case Study

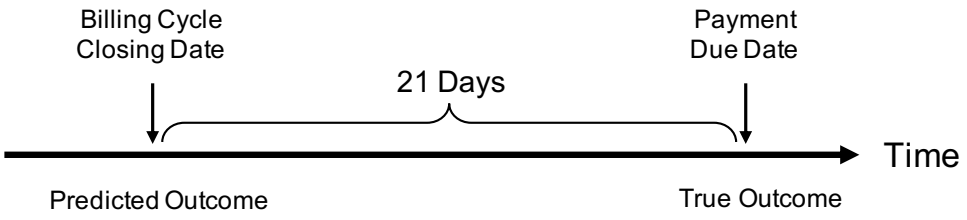
Having demonstrated the superiority of our method in predicting delinquency outcomes, we now show the economic value generated from its superior predictive power using a case study. Consider

the following profit matrix for user v_i, given the predicted and actual outcomes for the current billing cycle Ψ. Each entry of the matrix, $P_{i,\Psi}(a,b)$, denotes the profit resulting from predicting the user as outcome a while the true outcome is b for the billing cycle Ψ, where $a,b \in \{0,1\}$ with 0 being "No Delinquency" and 1 being "Delinquency". A negative value of $P_{i,\Psi}(a,b)$ means cost. Given a method's prediction and the true outcome for user v_i, we can derive the profit by the method for v_i from the profit matrix. The overall profit generated from the method is the sum of the individual profits across all users. Next, we define each entry of the profit matrix.

Profit Matrix for User v_i Given Prediction for Billing Cycle Ψ

	Actual "No Delinquency"	Actual "Delinquency"
Predict as "No Delinquency"	$P_{i,\Psi}(0,0)$	$P_{i,\Psi}(0,1)$
Predict as "Delinquency"	$P_{i,\Psi}(1,0)$	$P_{i,\Psi}(1,1)$

Figure 1 Timeline of predicted and true delinquency outcomes.



As shown in Figure 1, a user's delinquency outcome is predicted at the closing date of a billing cycle, and the true outcome is observed on the payment due date, 21 days after the closing date. Therefore, to specify each entry in the profit matrix, we need to determine the impact of the delinquency outcome prediction over the 21-day period between the closing date and the payment due date of a billing cycle. The bank from which we collected data has two primary sources of profit from its credit card business: interchange fees and interest. The former refers to the fee charged by the bank to merchants who process credit card transactions and the latter is the interest charged by the bank to users who do not pay their bills in full. Following the literature (Elkan 2001), we measure each entry in the profit matrix against the same baseline, which is the state of the bank

before it makes a decision regarding a user. Specifically, if a user is predicted as "Delinquency", the bank suspends the user's credit card immediately. As a result, the bank makes no profit and suffers no loss from the user in comparison to the baseline. Thus, we have

$$P_{i,\Psi}(1,0) = P_{i,\Psi}(1,1) = 0. \tag{20}$$

If a user is predicted as "No Delinquency", the user will be able to use the credit card as usual. Let $Amt_{i,\Psi}$ denote user v_i's credit card spending amount during the 21-day period immediately following the closing date of billing cycle Ψ. Importantly, even if a user is currently non-delinquent, the bank may still incur a loss if the user later becomes a charge-off and permanently fails to repay $Amt_{i,\Psi}$. Let p^L denote the probability that a currently non-delinquent user will eventually become a charge-off. Then, we have

$$P_{i,\Psi}(0,0) = Amt_{i,\Psi} \times r^f + Amt_{i,\Psi} \times r^n \times (1 - p^L) - Amt_{i,\Psi} \times p^L. \tag{21}$$

Here, $Amt_{i,\Psi} \times r^f$ is the interchange fee collected from the merchant, where r^f is the rate of interchange fee. The term $Amt_{i,\Psi} \times r^n \times (1 - p^L)$ represents the expected interest income, conditional on the currently non-delinquent user not charging off and eventually repaying $Amt_{i,\Psi}$ (with probability $1 - p^L$). r^n denotes the expected amount of interest earned per dollar of credit card spending. The term $Amt_{i,\Psi} \times p^L$ reflects the expected loss due to the future charge-off risk.

Similarly, for users who are currently delinquent, not all will ultimately charge off; some may still repay their overdue amounts in future billing cycles. Accordingly, we define p^{LD} as the probability that a currently delinquent user will become a charge-off. As a result, for users predicted as non-delinquent but who are actually delinquent, we have

$$P_{i,\Psi}(0,1) = Amt_{i,\Psi} \times r^f + Amt_{i,\Psi} \times r^n \times (1 - p^{LD}) - Amt_{i,\Psi} \times p^{LD}, \tag{22}$$

where $Amt_{i,\Psi} \times r^n \times (1 - p^{LD})$ represents the expected interest income if the delinquent user eventually repays and $Amt_{i,\Psi} \times p^{LD}$ reflects the expected loss from the potential charge-off. According to the bank, the rate of interchange fee r^f is 0.45% and the expected amount of interest generated from one dollar of credit card spending r^n is \$0.0117.[9] The probabilities of charge-off for

[9] All monetary amounts in the case study have been converted to U.S. dollars.

non-delinquent and delinquent users are 0.0046 and 0.0641, respectively; that is, $p^L = 0.0046$ and $p^{LD} = 0.0641$.

Next, we determine the prediction decision based on the expected profit from classifying the user as "No Delinquency." Specifically, because the realized profits $P_{i,\Psi}(0,0)$ and $P_{i,\Psi}(0,1)$ depend on the actual spending $Amt_{i,\Psi}$, which has not yet been observed at the moment of decision-making, we instead estimate these profits as follows:

$$\overline{P}_i(0,0) := \overline{Amt}_i \times r^f + \overline{Amt}_i \times r^n \times (1 - p^L) - \overline{Amt}_i \times p^L,$$

and

$$\overline{P}_i(0,1) := \overline{Amt}_i \times r^f + \overline{Amt}_i \times r^n \times (1 - p^{LD}) - \overline{Amt}_i \times p^{LD},$$

where $\overline{Amt}_i$ is the average spending of user v_i across several historical 21-day grace periods. Given the predicted delinquency probability $p_{i,\Psi}$, the expected profit of classifying the user as non-delinquent is computed as:

$$(1 - p_{i,\Psi}) \times \overline{P}_i(0,0) + p_{i,\Psi} \times \overline{P}_i(0,1).$$

If this expected profit is positive, the user is classified as "No Delinquency" and allowed to continue using the card; otherwise, the user is classified as "Delinquency" and their card is suspended.

Once decisions have been made for all users, the total realized profit is computed as the sum of $P_{i,\Psi}(a,b)$ over all users v_i. Table 5 reports the profit generated by each method for billing cycle $\Psi = 14$. As reported, the profit of LSE is \$70,581, which surpasses that of XGBoost (the best performing benchmark) by 19.95% and that of RLR (the worst performing benchmark) by 120.33%. The results for other billing cycles are qualitatively similar.

Table 5 Profit Generated by Each Method ($\Psi = 14$).

Method	Profit	Improvement by LSE
LSE (Our Method)	\$70,581	
RLR	\$32,034	120.33%
MLP	\$37,311	89.17%
SVM	\$35,919	96.50%
XGBoost	\$58,844	19.95%
RNN-GRU	\$49,407	42.86%
SIM-TemGNN	\$40,158	75.76%

6.4. Evaluation Results for Multiclass and Numerical Credit Risk Prediction Problems

For the multiclass credit risk prediction problem, we consider a case of three credit risk outcomes: user v_i being delinquent in billing cycle t (failing to pay the minimum payment due), user v_i being non-delinquent in billing cycle t but only making partial payment, or user v_i being non-delinquent in billing cycle t and paying off his statement balance. Accordingly, we set $y_{i,t} = 2, 1$, and 0 for these three outcomes respectively. Averaged across 22 billing cycles in our data, 33.43% of the users (or 3,932 users) are delinquent, 11.17% of the users (or 1,314 users) are non-delinquent with partial payment, and 55.39% of the users (or 6,515 users) are non-delinquent with full payment. We still used user v_i's billing and demographic information in billing cycle t to construct $\mathbf{x}_{i,t}$, as described in Section 6.1.

The evaluation procedure for multiclass credit risk prediction is similar to that for binary credit risk prediction described in Section 6.1. Let Ψ be current billing cycle. We constructed training data using users' observed intrinsic characteristics $\mathbf{x}_{i,t}$ and multiclass credit risk outcomes $y_{i,t}$ in each of the previous 12 billing cycles t, $t = \Psi - 12, \cdots, \Psi - 1$. Using the training data, we trained our method and each benchmark method to predict probability $p_{k,i,\Psi}$ for every user in billing cycle Ψ based on the user's observed intrinsic characteristics $\mathbf{x}_{i,\Psi}$, where $p_{k,i,\Psi}$ denotes the probability that the credit risk outcome of user v_i in billing cycle Ψ is k ($k = 0, 1, 2$). User v_i was then predicted to be the credit risk outcome associated with the highest predicted probability. The performance of a method was evaluated by comparing its predicted credit risk outcome against the true credit risk outcome for each user in billing cycle Ψ with commonly used multiclass classification metrics: accuracy, macro precision, macro recall and macro F1 score (Stab and Gurevych 2014, Narasimhan et al. 2016). Here, accuracy is the percentage of correctly classified users. To compute the other metrics, we first calculate binary precision, recall, and F1 score for each credit risk outcome by treating the outcome as positive and the rest of the outcomes as negative. Macro precision, recall, and F1 score are the average of binary precisions, binary recalls, and binary F1 scores across all the credit risk outcomes, respectively.

For the multiclass credit risk prediction case, we tuned hyperparameters for each method with $\Psi = 13$ and reported their performance from $\Psi = 14$ to $\Psi = 22$ in Table 6.[10] As shown in the table, our method significantly outperforms each benchmark method across billing cycles in each evaluation metric ($p < 0.001$). For example, averaged across Ψ, the mean accuracy of LSE is 0.805. That is, on average, 80.5% of credit risk outcomes predicted by our method turns out to be true, which outperforms that of RLR and that of XGBoost by 18.16% and 4.67%, respectively. The superiority of our method reported in Table 6 further confirms the value of our methodological novelty.

For the numerical credit risk prediction problem, we considered a case of predicting the repayment rate (or the proportion of repaid statement balance). In this case, we still used each user's billing and demographic information in every billing cycle to construct $\mathbf{x}_{i,t}$ but set $y_{i,t}$ as the user's repayment rate in the billing cycle. The average repayment rate in our dataset is 0.617. Like before, we used training data in the previous 12 billing cycles to train each method to predict repayment rate $\mu_{i,\Psi}$ for every user in current billing cycle Ψ. The performance of a method is measured using the mean absolute error (MAE), a widely adopted metric for numerical predictions (e.g., Yang et al. 2019):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\mu_{i,\Psi} - y_{i,\Psi}|,$$

where $\mu_{i,\Psi}$ and $y_{i,\Psi}$ denote user v_i's predicted and actual repayment rate in billing cycle Ψ, respectively, and n is the number of users. Since RLR and SVM are designed for classification, we employed their respective regression version: ridge regression (RR, Hastie et al. 2009, p. 61) and support vector regression (SVR, Smola and Schölkopf 2004), to predict the numerical repayment rate. The other benchmark methods are also applicable to regression problems. We tuned hyperparameters for each method with $\Psi = 13$ and conducted experiments to compare their performance

[10] In this case, we set hyperparameters of our method as: $\lambda_Z = 0.05$, $\lambda_c = 1$, $\lambda_{\boldsymbol{\alpha}} = 0.1$, $\lambda_{\boldsymbol{\beta}} = 0.001$, and $d = 2$. The ridge parameter of RLR was set to 10^{-4}. MLP had 3 hidden layers of sizes 128, 64, 32, respectively. SVM had the RBF kernel with regularization parameter being set to 500. For XGBoost, the number of boosting rounds and maximum depth of a tree booster were set to 500 and 8, respectively. For RNN-GRU, the size of hidden states was set to 128. For SIM-TemGNN, its MLP component had 3 hidden layers with sizes of 64, 64, and 16, respectively, its GNN component consisted of 2 layers, each with an embedding size of 16, and the size of each hidden state of its LSTM component was set to 32.

Table 6 Evaluation Results for Multiclass Credit Risk Prediction.

Metric	Method	$\Psi = 14$	$\Psi = 15$	$\Psi = 16$	$\Psi = 17$	$\Psi = 18$	$\Psi = 19$	$\Psi = 20$	$\Psi = 21$	$\Psi = 22$	Mean	Std.
Accuracy	LSE (Our Method)	0.800	0.804	0.799	0.805	0.809	0.812	0.810	0.808	0.801	0.805	0.005
	RLR	0.671	0.682	0.675	0.680	0.692	0.688	0.687	0.682	0.677	0.682	0.007
	MLP	0.713	0.726	0.709	0.724	0.728	0.734	0.741	0.733	0.721	0.725	0.010
	SVM	0.664	0.683	0.666	0.672	0.678	0.674	0.682	0.688	0.670	0.675	0.008
	XGBoost	0.757	0.763	0.767	0.770	0.775	0.780	0.782	0.765	0.766	0.769	0.008
	RNN-GRU	0.747	0.754	0.750	0.758	0.757	0.760	0.756	0.752	0.753	0.754	0.004
	SIM-TemGNN	0.723	0.737	0.741	0.731	0.719	0.738	0.732	0.736	0.732	0.732	0.007
Macro Precision	LSE (Our Method)	0.742	0.749	0.745	0.754	0.763	0.764	0.770	0.766	0.759	0.757	0.010
	RLR	0.577	0.600	0.601	0.590	0.609	0.603	0.595	0.609	0.602	0.598	0.010
	MLP	0.464	0.481	0.465	0.474	0.490	0.488	0.489	0.479	0.476	0.478	0.010
	SVM	0.571	0.580	0.579	0.564	0.595	0.571	0.596	0.601	0.572	0.581	0.013
	XGBoost	0.703	0.687	0.692	0.688	0.697	0.673	0.707	0.678	0.688	0.690	0.011
	RNN-GRU	0.502	0.497	0.497	0.495	0.500	0.498	0.497	0.495	0.490	0.497	0.003
	SIM-TemGNN	0.475	0.487	0.486	0.480	0.472	0.485	0.476	0.480	0.481	0.480	0.005
Macro Recall	LSE (Our Method)	0.682	0.709	0.703	0.715	0.708	0.702	0.705	0.715	0.705	0.705	0.010
	RLR	0.486	0.504	0.499	0.496	0.510	0.506	0.519	0.522	0.501	0.505	0.011
	MLP	0.518	0.523	0.519	0.521	0.528	0.530	0.534	0.538	0.523	0.526	0.007
	SVM	0.476	0.491	0.470	0.475	0.493	0.483	0.494	0.514	0.482	0.487	0.013
	XGBoost	0.624	0.631	0.632	0.643	0.640	0.651	0.653	0.631	0.624	0.637	0.011
	RNN-GRU	0.546	0.549	0.551	0.554	0.558	0.555	0.549	0.556	0.551	0.552	0.004
	SIM-TemGNN	0.527	0.536	0.545	0.533	0.526	0.536	0.529	0.541	0.534	0.534	0.006
Macro F_1 Score	LSE (Our Method)	0.693	0.721	0.717	0.729	0.725	0.718	0.721	0.725	0.721	0.719	0.010
	RLR	0.485	0.511	0.503	0.499	0.507	0.507	0.517	0.512	0.503	0.505	0.009
	MLP	0.489	0.498	0.490	0.495	0.503	0.506	0.509	0.506	0.497	0.499	0.007
	SVM	0.470	0.488	0.453	0.465	0.488	0.479	0.495	0.516	0.481	0.482	0.018
	XGBoost	0.643	0.648	0.649	0.659	0.657	0.659	0.670	0.643	0.639	0.652	0.010
	RNN-GRU	0.521	0.521	0.521	0.523	0.527	0.525	0.522	0.523	0.519	0.522	0.002
	SIM-TemGNN	0.499	0.509	0.513	0.505	0.497	0.509	0.501	0.509	0.506	0.505	0.005

from $\Psi = 14$ to $\Psi = 22$.[11] As shown in Table 7, LSE significantly outperforms each benchmark method in terms of MAE ($p < 0.001$). The average MAE of LSE is 0.181, which is lower than that of the benchmark methods by a range between 19.66% and 58.25%.

7. Conclusion

Our study belongs to the computational genre of design science research, which is concerned with developing computational methods to solve business and societal problems and aims at making methodological contributions (Rai 2017, Padmanabhan et al. 2022). In particular, our study solves

[11] The hyperparameters of our method were set as: $\lambda_Z = 0.01$, $\lambda_c = 0.1$, $\lambda_{\boldsymbol{\alpha}} = 1$, $\lambda_{\boldsymbol{\beta}} = 0.001$, and $d = 2$. For RR, its ridge parameter was set to 10^{-4}. MLP had 1 hidden layer of sizes 16. The regularization parameter of SVR was set to 100. For XGBoost, its number of boosting rounds and maximum depth of a tree booster were set to 800 and 6, respectively. The size of RNN-GRU's hidden states was set to 256. For SIM-TemGNN, its MLP component had 3 hidden layers with sizes of 64, 32, and 16, respectively, its GNN component consisted of 2 layers, each with an embedding size of 32, and the size of each hidden state of its LSTM component was set to 32.

Table 7 Evaluation Results for Numerical Credit Risk Prediction.

Metric	Method	$\Psi = 14$	$\Psi = 15$	$\Psi = 16$	$\Psi = 17$	$\Psi = 18$	$\Psi = 19$	$\Psi = 20$	$\Psi = 21$	$\Psi = 22$	Mean	Std.
MAE	LSE (Our Method)	0.187	0.179	0.182	0.177	0.183	0.180	0.176	0.183	0.185	0.181	0.004
	RR	0.358	0.360	0.359	0.359	0.357	0.359	0.359	0.361	0.367	0.360	0.003
	MLP	0.329	0.335	0.318	0.320	0.331	0.327	0.321	0.343	0.334	0.329	0.008
	SVR	0.305	0.298	0.291	0.293	0.290	0.294	0.307	0.309	0.302	0.299	0.007
	XGBoost	0.230	0.221	0.232	0.230	0.226	0.221	0.219	0.224	0.229	0.226	0.005
	RNN-GRU	0.403	0.396	0.414	0.403	0.408	0.398	0.417	0.403	0.407	0.405	0.007
	SIM-TemGNN	0.433	0.441	0.436	0.428	0.435	0.444	0.438	0.427	0.429	0.435	0.006

the problem of credit risk prediction and adds to extant literature with the following methodological contributions. First, we propose a novel latent similarity-enhanced credit risk prediction model, which operationalizes the similarity between a pair of users as a combination of the observed and latent similarities between them. In contrast, prior research on similarity-based credit risk prediction employs observed similarity solely. Second, we develop a new method that estimates the model parameters, learns latent similarities among users, and predicts users' probabilities of delinquency. Premised in our derived propositions and theorem, the method is computationally efficient and guarantees to converge. Thus, it is particularly suitable for solving real world credit risk prediction problems with a large number of users. In addition, we extend our model and method to the multiclass and numerical credit risk prediction problems. Consequently, we have developed methods that can solve a broad spectrum of credit risk prediction problems.

Our study offers several implications for business. First, our proposed method allows financial institutions to conduct credit risk prediction beyond the sole use of users' intrinsic data. This is realized through the similarity component of our method. As a result, the predictive power of our method substantially outperforms that of credit risk prediction methods widely used in the industry such as RLR, as we demonstrate in our empirical evaluation. Improved predictive power by our method enables financial institutions to more accurately pinpoint risky users and prevent losses and to better serve creditworthy users and harvest revenues, which, in turn, leads to improved profits. As we show in a case study (Section 6.3), our method expectedly produces 120.33% more profit than RLR, the industry standard for credit risk prediction (Thomas et al. 2017), and 19.95% more profit than XGBoost, a popular credit risk prediction method used by

financial institutions. Considering the enormous size of the consumer credit market, effective use of our method could potentially generate considerable extra profits for financial institutions compared to standard methods currently used in the industry. On the other hand, our method also benefits creditworthy users. By differentiating risky users from creditworthy users more accurately, our method reduces the chance of creditworthy users being misclassified as risky users, thereby giving creditworthy users wider access to much-needed consumer credits.

It is worth noting that the performance advantage of our method over the benchmarks stems from our novel model of user similarity for credit risk prediction. This underscores the predictive power of similar users' repayment behaviors in evaluating a focal user's credit risk. In other words, the repayment behaviors of users, who have a high degree of similarities with a focal user, could function as predictors for the focal user's credit risk. Thus, credit risk analysts could assess a user's credit status by analyzing the credit statuses of the user's similar peers, in addition to examining the user's repayment behavioral data. In addition, our study equips financial institutions with tools that can predict credit risks at a more refined level. The multiclass credit risk prediction method allows practitioners to further divide non-delinquent users into more business-relevant subgroups. This is useful as repayment behaviors of non-delinquent users are not homogeneous. For example, some repay full statement balance and incur no interest while some others merely repay the minimum due and have to pay interest. The numerical credit risk prediction method enables practitioners to predict individualized credit risk indicators such as repayment rate and future spending amount. Armed with these useful indicators, financial institutions can design personalized services for their users such as deciding whether to increase the credit limit for a user and, if so, how much to increase. Moreover, the proposed method could be applied to other business domains. For example, it could be extended to support personalized marketing. The objective of this task is to predict customers' future spending behaviors (e.g., spending amount) and design individualized marketing strategy (e.g., incentive) for each customer based on the prediction. Understandably, spending behaviors of a customer's similar peers could function as predictors for the customer's

spending behavior. Therefore, our method could improve the effectiveness of the prediction by considering not only a customer's past spending data but also spending behaviors of the customer's similar peers.

Our study can be extended in several directions. First, we develop a linear credit risk prediction model that considers similarities among users. It is interesting to investigate how to incorporate the similarity component into a non-linear model for credit risk prediction, such as a neural network model. Second, future work could extend our method to the application scoring problem, where no historical data are available to learn latent similarities. One viable approach to addressing this challenge is transfer learning, which learns a latent similarity model using users' historical data collected from other domains (e.g., spending data on e-commerce platforms) and adapts the model to the domain of credit risk prediction. Third, our method considers the number of historical delinquencies but overlooks the sequential information of historical delinquencies. Future work could integrate this information into our method to further improve its performance.

References

Altman E, Resti A, Sironi A (2005) *Recovery Risk: The Next Challenge in Credit Risk Management* (Risk Books, London).

Atkinson T, Luttrell D, Rosenblum H (2013) How bad was it? the costs and consequences of the 2007–09 financial crisis. Technical report, Federal Reserve Bank of Dallas, Dallas, TX.

Babaev D, Savchenko M, Tuzhilin A, Umerenkov D (2019) E.t.-rnn: Applying deep learning to credit loan applications. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York), 2183–2190.

Baesens B, Van Gestel T, Viaene S, Stepanova M, Suykens J, Vanthienen J (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society* 54(6):627–635.

Bertsekas DP (1999) *Nonlinear Programming,* 2nd ed. (Athena Scientific, Belmont, MA).

Breiman L (1996) Bagging predictors. *Machine Learning* 24(2):123–140.

Burges CJ (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Kowledge Discovery* 2(2):121–167.

Chatterjee S, Corbae D, Nakajima M, Ríos-Rull JV (2007) A quantitative theory of unsecured consumer credit with risk of default. *Econometrica* 75(6):1525–1589.

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York), 785–794.

Chen Y (2023) *Machine Learning and Optimization With Latent Variables.* Ph.D. thesis, Princeton University.

De Almeida Filho AT, Mues C, Thomas LC (2010) Optimizing the collections process in consumer credit. *Production and Operations Management* 19(6):698–708.

Elkan C (2001) The foundations of cost-sensitive learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2* (Morgan Kaufmann Publishers Inc., San Francisco), 973–978.

Fang X, Hu PJH, Li Z, Tsai W (2013) Predicting adoption probabilities in social networks. *Information Systems Research* 24(1):128–145.

Fernandes GB, Artes R (2016) Spatial dependence in credit risk and its improvement in credit scoring. *European Journal of Operational Research* 249(2):517–524.

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1):119–139.

Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 1189–1232.

Giudici P, Hadji-Misheva B, Spelta A (2020) Network based credit risk models. *Quality Engineering* 32(2):199–211.

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

Guo Y, Zhou W, Luo C, Liu C, Xiong H (2016) Instance-based credit risk assessment for investment decisions in p2p lending. *European Journal of Operational Research* 249(2):417–426.

Hand DJ (2001) Modelling consumer credit risk. *IMA Journal of Management mathematics* 12(2):139–155.

Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 160(3):523–541.

Hand DJ, Vinciotti V (2003) Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern recognition letters* 24(9-10):1555–1562.

Hastie T, Mazumder R, Lee JD, Zadeh R (2015) Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research* 16(1):3367–3402.

Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learning: Data mining, inference, and prediction,* 2nd ed. (Springer, New York).

Henley W, Hand DJ (1996) Ak-nearest-neighbour classifier for assessing consumer credit risk. *Journal of the Royal Statistical Society Series D: The Statistician* 45(1):77–95.

Kennedy K, Namee BM, Delany SJ (2013) Using semi-supervised classifiers for credit scoring. *Journal of the Operational Research Society* 64(4):513–529.

Khandani AE, Kim AJ, Lo AW (2010) Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance* 34(11):2767–2787.

Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. *Proceedings of the Fifth International Conference on Learning Representation (ICLR'17)* .

Le Cessie S, Van Houwelingen JC (1992) Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41(1):191–201.

Lee JW, Lee WK, Sohn SY (2021) Graph convolutional network-based credit default prediction utilizing three types of virtual distances among borrowers. *Expert Systems with Applications* 168:114411.

Lessmann S, Baesens B, Seow HV, Thomas LC (2015) Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* 247(1):124–136.

Lu T, Zhang Y, Beibei L (2023) Profit vs. equality? the case of financial risk assessment and a new perspective on alternative data. *MIS Quarterly* 47(4):1517–1556.

Malhotra R, Malhotra DK (2003) Evaluating consumer loans using neural networks. *Omega* 31(2):83–96.

McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27(1):415–444.

Medina-Olivares V, Calabrese R, Dong Y, Shi B (2022) Spatial dependence in microfinance credit default. *International Journal of Forecasting* 38(3):1071–1085.

Menon AK, Elkan C (2011) Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22* (Springer).

Miller K, Jordan M, Griffiths T (2009) Nonparametric latent feature models for link prediction. *Advances in neural information processing systems* 22.

Narasimhan H, Pan W, Kar P, Protopapas P, Ramaswamy HG (2016) Optimizing the multiclass f-measure via biconcave programming. *Proceeding of 2016 IEEE 16th International Conference on Data Mining* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 1101–1106.

Nocedal J, Wright S (2006) *Numerical optimization,* 2nd ed. (Springer, New York).

Padmanabhan B, Fang X, Sahoo N, Burton-Jones A (2022) Machine learning in information systems research. *MIS Quarterly* 46(1):iii–xix.

Paleologo G, Elisseeff A, Antonini G (2010) Subagging for credit scoring models. *European Journal of Operational Research* 201(2):490–499.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) *Numerical recipes: The art of scientific computing,* 3rd ed. (Cambridge university press, Cambridge, UK).

Rai A (2017) Editor's comments: Diversity of design science research. *MIS Quarterly* 41(1):iii–xviii.

Sinha AP, May JH (2004) Evaluating and tuning predictive data mining models using receiver operating characteristic curves. *Journal of Management Information Systems* 21(3):249–280.

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. *Statistics and Computing* 14(3):199–222.

Stab C, Gurevych I (2014) Identifying argumentative discourse structures in persuasive essays. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Stroudsburg, PA), 46–56.

Thomas L, Crook J, Edelman D (2017) *Credit scoring and its applications,* 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia).

Thomas LC (2000) A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting* 16(2):149–172.

Twala B (2010) Multiple classifier application to credit risk assessment. *Expert systems with applications* 37(4):3326–3336.

Varin C, Reid N, Firth D (2011) An overview of composite likelihood methods. *Statistica Sinica* 21:5–42.

Wang D, Zhang Z, Zhou J, Cui P, Fang J, Jia Q, Fang Y, Qi Y (2021) Temporal-aware graph neural network for credit risk prediction. *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)* (SIAM).

West D, Dellana S, Qian J (2005) Neural network ensemble strategies for financial decision applications. *Computers & Operations Research* 32(10):2543–2559.

Yang M, Zheng Z, Mookerjee V (2019) Prescribing response strategies to manage customer opinions: A stochastic differential equation approach. *Information Systems Research* 30(2):351–374.

Yao X, Crook J, Andreeva G (2015) Support vector regression for loss given default modelling. *European Journal of Operational Research* 240(2):528–538.

Appendix A: Derivations of Equations (4), (5) and (7)

A.1. Derivation of Equation (4)

For the convenience of reading, we repeat notations defined in Section 4.1: $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_n)^T$, $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^T$, $\boldsymbol{c} = (c_1, \cdots, c_n)^T$, $C = \text{diag}(\boldsymbol{c})$, and $G = [s_{ij}]_{n\times n}$, where diagonal elements in G are defined to be 0. By the model of θ_i in Equation (3), we have

$$\theta_i = \mathbf{x}_i^T\boldsymbol{\beta} + c_i \sum_{v_j \in V:\, j\neq i} s_{ij}\theta_j.$$

Let $\mathbf{e}_i$ denote a standard unit vector with its i^{th} element being 1 and other elements being 0. It is easy to see that $\theta_i = \mathbf{e}_i^T\boldsymbol{\theta}$. Therefore, we have

$$\begin{aligned}
\theta_i = \mathbf{e}_i^T\boldsymbol{\theta} &= \mathbf{x}_i^T\boldsymbol{\beta} + c_i \sum_{v_j \in V:\, j\neq i} s_{ij}\theta_j \\
&= \mathbf{x}_i^T\boldsymbol{\beta} + c_i(s_{i1}, \ldots, s_{i(i-1)}, 0, s_{i(i+1)}, s_{in})\boldsymbol{\theta} \\
&= \mathbf{x}_i^T\boldsymbol{\beta} + c_i\mathbf{e}_i^T G\boldsymbol{\theta} \\
&= \mathbf{e}_i^T X\boldsymbol{\beta} + \mathbf{e}_i^T CG\boldsymbol{\theta} \\
&= \mathbf{e}_i^T(X\boldsymbol{\beta} + CG\boldsymbol{\theta}).
\end{aligned}$$

Equation (4) follows immediately. □

A.2. Derivation of Equation (5)

We use the notation in Appendix A.1 as well as $Z = (\mathbf{z}_1, \cdots, \mathbf{z}_n)^T$, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n)^T$. With the definition of $G = [s_{ij}]_{n\times n}$, it can be seen that $s_{ij} = \mathbf{e}_i^T G\mathbf{e}_j$. By Equation (1), we have

$$\begin{aligned}
s_{ij} = \mathbf{e}_i^T G\mathbf{e}_j &= \alpha_i + \alpha_j + \tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_j + \mathbf{z}_i^T\mathbf{z}_j. \\
&= \mathbf{e}_i^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{e}_j + \mathbf{e}_i^T\tilde{X}\tilde{X}^T\mathbf{e}_j + \mathbf{e}_i^T ZZ^T\mathbf{e}_j \\
&= \mathbf{e}_i^T\mathbf{1}_n\boldsymbol{\alpha}^T\mathbf{e}_j + \mathbf{e}_i^T\boldsymbol{\alpha}\mathbf{1}_n^T\mathbf{e}_j + \mathbf{e}_i^T\tilde{X}\tilde{X}^T\mathbf{e}_j + \mathbf{e}_i^T ZZ^T\mathbf{e}_j \\
&= \mathbf{e}_i^T(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T + \tilde{X}\tilde{X}^T + ZZ^T)\mathbf{e}_j,
\end{aligned}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is an all-ones vector.

Recall that the diagonal elements of G are defined to be 0. Specifically, the i^{th} diagonal element of G is given by

$$\begin{aligned}&\mathbf{e}_i^T(\mathbf{1}_n\boldsymbol{\alpha}^T+\boldsymbol{\alpha}\mathbf{1}_n^T+\tilde{X}\tilde{X}^T+ZZ^T)\mathbf{e}_i\\=&\boldsymbol{\alpha}^T\mathbf{e}_i+\mathbf{e}_i^T\boldsymbol{\alpha}+\mathbf{e}_i^T\tilde{X}\tilde{X}^T\mathbf{e}_i+\mathbf{e}_i^TZZ^T\mathbf{e}_i\\=&2\alpha_i+\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i+\mathbf{z}_i^T\mathbf{z}_i.\end{aligned}$$

To make the diagonal elements of G being 0, we have

$$G=\mathbf{1}_n\boldsymbol{\alpha}^T+\boldsymbol{\alpha}\mathbf{1}_n^T+\tilde{X}\tilde{X}^T+ZZ^T-2\text{diag}(\boldsymbol{\alpha})-\sum_{i=1}^n(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i+\mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T$$

This completes the derivation of Equation (5). □

A.3. Derivation of the Negative Log-likelihood Function (7)

Given a training data set of users' observed characteristics and their delinquency outcomes $\{\mathbf{x}_i,y_i\}_{i=1}^n$, we have

$$\begin{aligned}h(\boldsymbol{\beta},Z,\boldsymbol{\alpha},\boldsymbol{c})&=-\frac{1}{n}\log\Big(\prod_{i=1}^n p_i^{y_i}(1-p_i)^{(1-y_i)}\Big)\\&=-\frac{1}{n}\sum_{i=1}^n y_i\log(p_i)+(1-y_i)\log(1-p_i)\\&=-\frac{1}{n}\sum_{i=1}^n y_i\log\frac{p_i}{1-p_i}+\log(1-p_i).\end{aligned}$$

Recall that $\theta_i=\log\frac{p_i}{1-p_i}$ and we have

$$\begin{aligned}h(\boldsymbol{\beta},Z,\boldsymbol{\alpha},\boldsymbol{c})&=\frac{1}{n}\sum_{i=1}^n -y_i\theta_i-\log(1-\frac{e^{\theta_i}}{1+e^{\theta_i}})\\&=\frac{1}{n}\sum_{i=1}^n -y_i\theta_i+\log(1+e^{\theta_i}).\end{aligned}$$

This completes the derivation of the negative log-likelihood function (7). □

Appendix B: Proof of Propositions 1, 2 and Theorem 1

B.1. Proof of Proposition 1

We first derive the derivatives of matrix M^{-1} with respect to parameters z_{ij}, α_i, and c_i, where M is defined in Equation (6). These derivatives will then be used in the derivation of the gradients in

Proposition 1.

$$\begin{aligned}
\frac{\partial M^{-1}}{\partial z_{ij}} &= -M^{-1}\frac{\partial M}{\partial z_{ij}}M^{-1} = -M^{-1}\frac{\partial(E_n - CG)}{\partial z_{ij}}M^{-1} = -M^{-1}[-C\frac{\partial G}{\partial z_{ij}}]M^{-1} \\
&= M^{-1}C\frac{\partial(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T - 2\mathrm{diag}(\boldsymbol{\alpha}) + \tilde{X}\tilde{X}^T + ZZ^T - \sum_{\tilde{i}=1}^n(\tilde{\mathbf{x}}_{\tilde{i}}^T\tilde{\mathbf{x}}_{\tilde{i}} + \mathbf{z}_{\tilde{i}}^T\mathbf{z}_{\tilde{i}})\mathbf{e}_{\tilde{i}}\mathbf{e}_{\tilde{i}}^T)}{\partial z_{ij}}M^{-1} \\
&= M^{-1}C\frac{\partial(ZZ^T - \sum_{\tilde{i}=1}^n(\mathbf{z}_{\tilde{i}}^T\mathbf{z}_{\tilde{i}})\mathbf{e}_{\tilde{i}}\mathbf{e}_{\tilde{i}}^T)}{\partial z_{ij}}M^{-1} \\
&= M^{-1}C(Z\frac{\partial Z^T}{\partial z_{ij}} + \frac{\partial Z}{\partial z_{ij}}Z^T - \frac{\partial \mathbf{z}_i^T\mathbf{z}_i}{\partial z_{ij}}\mathbf{e}_i\mathbf{e}_i^T)M^{-1} \\
&= M^{-1}C\big(Z(\mathbf{e}_i\mathbf{e}_j^T)^T + (\mathbf{e}_i\mathbf{e}_j^T)Z^T - \frac{\partial(\sum_{\tilde{j}=1}^d z_{i\tilde{j}}^2)}{\partial z_{ij}}\mathbf{e}_i\mathbf{e}_i^T\big)M^{-1} \\
&= M^{-1}C\big(Z\mathbf{e}_j\mathbf{e}_i^T + \mathbf{e}_i(Z\mathbf{e}_j)^T - 2z_{ij}\mathbf{e}_i\mathbf{e}_i^T\big)M^{-1}
\end{aligned} \tag{A.1}$$

$$\begin{aligned}
\frac{\partial M^{-1}}{\partial \alpha_i} &= -M^{-1}\frac{\partial M}{\partial \alpha_i}M^{-1} = -M^{-1}\frac{\partial(E_n - CG)}{\partial \alpha_i}M^{-1} = -M^{-1}[-C\frac{\partial G}{\partial \alpha_i}]M^{-1} \\
&= M^{-1}C\frac{\partial(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T - 2\mathrm{diag}(\boldsymbol{\alpha}) + \tilde{X}\tilde{X}^T + ZZ^T - \sum_{\tilde{i}=1}^n(\tilde{\mathbf{x}}_{\tilde{i}}^T\tilde{\mathbf{x}}_{\tilde{i}} + \mathbf{z}_{\tilde{i}}^T\mathbf{z}_{\tilde{i}})\mathbf{e}_{\tilde{i}}\mathbf{e}_{\tilde{i}}^T)}{\partial \alpha_i}M^{-1} \\
&= M^{-1}C\frac{\partial(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T - 2\mathrm{diag}(\boldsymbol{\alpha}))}{\partial \alpha_i}M^{-1} \\
&= M^{-1}C(\mathbf{1}_n\mathbf{e}_i^T + \mathbf{e}_i\mathbf{1}_n^T - 2\mathbf{e}_i\mathbf{e}_i^T)M^{-1}.
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
\frac{\partial M^{-1}}{\partial c_i} &= -M^{-1}\frac{\partial M}{\partial c_i}M^{-1} = -M^{-1}\frac{\partial(E_n - CG)}{\partial c_i}M^{-1} = -M^{-1}[-\frac{\partial C}{\partial c_i}G]M^{-1} \\
&= M^{-1}\frac{\partial \mathrm{diag}(\boldsymbol{c})}{\partial c_i}GM^{-1} \\
&= M^{-1}(\mathbf{e}_i\mathbf{e}_i^TG)M^{-1}.
\end{aligned} \tag{A.3}$$

We also derive the partial derivative of the negative log-likelihood h defined in Equation (7) with respect to θ_i as follows:

$$\begin{aligned}
\frac{\partial h}{\partial \theta_i} &= \frac{\partial\big(\frac{1}{n}\sum_{\tilde{i}=1}^n -y_{\tilde{i}}\theta_{\tilde{i}} + \log(1 + e^{\theta_{\tilde{i}}})\big)}{\partial \theta_i} \\
&= \frac{1}{n}(-y_i + \frac{1}{1 + e^{-\theta_i}}).
\end{aligned}$$

Denoting $\pi(\theta_i) = \frac{1}{1+e^{-\theta_i}}$, we obtain

$$\frac{\partial h}{\partial \boldsymbol{\theta}} = \frac{1}{n}(-\mathbf{y} + \boldsymbol{\pi}_\theta), \tag{A.4}$$

where $\mathbf{y} = (y_1, \cdots, y_n)^T$ and $\boldsymbol{\pi}_\theta = (\pi(\theta_1), \cdots, \pi(\theta_n))^T$.

Given Equations (A.1), (A.4), (6) and (8), we have

$$\begin{aligned}
\frac{\partial q}{\partial z_{ij}} &= \frac{\partial h}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial \boldsymbol{\theta}}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial M^{-1}}{\partial z_{ij}}X\boldsymbol{\beta} + \lambda_Z z_{ij} \\
&= \frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C(Z\mathbf{e}_j\mathbf{e}_i^T + \mathbf{e}_i(Z\mathbf{e}_j)^T - 2z_{ij}\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta} + \lambda_Z z_{ij} \\
&= (\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}CZ\mathbf{e}_j)(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta}) + (\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{e}_i)((Z\mathbf{e}_j)^T M^{-1}X\boldsymbol{\beta}) \\
&\quad - 2z_{ij}(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{e}_i)(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta}) + \lambda_Z z_{ij} \\
&= (\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}CZ\mathbf{e}_j) + (\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\boldsymbol{\beta}^T X^T(M^{-1})^T Z\mathbf{e}_j) \\
&\quad - 2z_{ij}(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}) + \lambda_Z z_{ij} \\
&= \mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta}\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}CZ\big)\mathbf{e}_j + \mathbf{e}_i^T\big(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\boldsymbol{\beta}^T X^T(M^{-1})^T Z\big)\mathbf{e}_j \\
&\quad - 2\big(\mathbf{e}_i^T Z\mathbf{e}_j\big)\big(\mathbf{e}_i^T\big((M^{-1}X\boldsymbol{\beta})\odot(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})\big)\mathbf{1}_d^T\mathbf{e}_j\big) + \lambda_Z\mathbf{e}_i^T Z\mathbf{e}_j,
\end{aligned}$$

which immediately implies that

$$\begin{aligned}
\frac{\partial q}{\partial Z} &= (M^{-1}X\boldsymbol{\beta})(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}CZ) + (C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\boldsymbol{\beta}^T X^T(M^{-1})^T Z) \\
&\quad - 2Z\odot\big((M^{-1}X\boldsymbol{\beta})\odot(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})\mathbf{1}_d^T\big) + \lambda_Z Z.
\end{aligned}$$

Employing Equations (A.2), (A.4), (6) and (8), we have

$$\begin{aligned}
\frac{\partial q}{\partial \alpha_i} &= \frac{\partial h}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial \boldsymbol{\theta}}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial M^{-1}}{\partial \alpha_i}X\boldsymbol{\beta} + \lambda_{\boldsymbol{\alpha}}\alpha_i \\
&= \frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C(\mathbf{1}_n\mathbf{e}_i^T + \mathbf{e}_i\mathbf{1}_n^T - 2\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta} + \lambda_{\boldsymbol{\alpha}}\alpha_i \\
&= (\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{1}_n) + (\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n) \\
&\quad - 2(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}) + \lambda_{\boldsymbol{\alpha}}\alpha_i \\
&= \mathbf{e}_i^T(M^{-1}X\boldsymbol{\beta})(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{1}_n) + \mathbf{e}_i^T(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n) \\
&\quad - 2\mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta}\odot C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big) + \lambda_{\boldsymbol{\alpha}}\mathbf{e}_i^T\boldsymbol{\alpha},
\end{aligned}$$

which immediately implies that

$$\begin{aligned}
\frac{\partial q}{\partial \boldsymbol{\alpha}} &= (M^{-1}X\boldsymbol{\beta})(\frac{\partial h}{\partial \boldsymbol{\theta}^T}M^{-1}C\mathbf{1}_n) + (C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n) \\
&\quad - 2(M^{-1}X\boldsymbol{\beta})\odot(C(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}) + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha}.
\end{aligned}$$

Armed with Equations (A.3), (A.4), (6) and (8), we have

$$\begin{aligned}\frac{\partial q}{\partial c_i} &= \frac{\partial h}{\partial c_i} + \lambda_{\boldsymbol{c}} c_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial \boldsymbol{\theta}}{\partial c_i} + \lambda_{\boldsymbol{c}} c_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial c_i} + \lambda_{\boldsymbol{c}} c_i = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial M^{-1}}{\partial c_i}X\boldsymbol{\beta} + \lambda_{\boldsymbol{c}} c_i \\ &= \frac{\partial h}{\partial \boldsymbol{\theta}^T}\big(M^{-1}\mathbf{e}_i\mathbf{e}_i^T GM^{-1}\big)X\boldsymbol{\beta} + \lambda_{\boldsymbol{c}} c_i \\ &= (\mathbf{e}_i^T(M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}})(\mathbf{e}_i^T GM^{-1}X\boldsymbol{\beta}) + \lambda_{\boldsymbol{c}} c_i \\ &= \mathbf{e}_i^T\big(((M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}) \odot (GM^{-1}X\boldsymbol{\beta})\big) + \lambda_{\boldsymbol{c}}\mathbf{e}_i^T\boldsymbol{c},\end{aligned}$$

which immediately implies that

$$\frac{\partial q}{\partial \boldsymbol{c}} = \big((M^{-1})^T\frac{\partial h}{\partial \boldsymbol{\theta}}\big) \odot \big(GM^{-1}X\boldsymbol{\beta}\big) + \lambda_{\boldsymbol{c}}\boldsymbol{c}$$

Given Equations (A.4), (6) and (8), we have

$$\begin{aligned}\frac{\partial q}{\partial \beta_p} &= \frac{\partial h}{\partial \beta_p} + \lambda_{\boldsymbol{\beta}}\beta_p \\ &= \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial \boldsymbol{\theta}}{\partial \beta_p} + \lambda_{\boldsymbol{\beta}}\beta_p = \frac{\partial h}{\partial \boldsymbol{\theta}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial \beta_p} + \lambda_{\boldsymbol{\beta}}\beta_p = \frac{\partial l}{\partial \boldsymbol{\theta}^T}M^{-1}X\mathbf{e}_p + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}^T\mathbf{e}_p \\ &= \mathbf{e}_p^T(X^T(M^{-1})^T\frac{\partial l}{\partial \boldsymbol{\theta}} + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}),\end{aligned}$$

which immediately implies that

$$\frac{\partial q}{\partial \boldsymbol{\beta}} = X^T(M^{-1})^T\frac{\partial l}{\partial \boldsymbol{\theta}} + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}.$$

This completes the proof. □

B.2. Proof of Propositions 2

Define $D_n = E_n + C\big(2\text{diag}(\boldsymbol{\alpha}) + \sum_{i=1}^n(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i + \mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T\big)$, $Q_1 = (\mathbf{1}_n, \boldsymbol{\alpha}, \tilde{X}, Z)$, and $Q_2 = (\boldsymbol{\alpha}, \mathbf{1}_n, \tilde{X}, Z)$. By the definition of M in Equation (6) and the expression of G in Equation (5), we have

$$\begin{aligned}M &= E_n - CG \\ &= E_n - C(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T - 2\text{diag}(\boldsymbol{\alpha}) + \tilde{X}\tilde{X}^T + ZZ^T - \sum_{i=1}^n(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i + \mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T) \\ &= E_n + C\big(2\text{diag}(\boldsymbol{\alpha}) + \sum_{i=1}^n(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i + \mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T\big) - C(\mathbf{1}_n\boldsymbol{\alpha}^T + \boldsymbol{\alpha}\mathbf{1}_n^T + \tilde{X}\tilde{X}^T + ZZ^T) \\ &= E_n + C\big(2\text{diag}(\boldsymbol{\alpha}) + \sum_{i=1}^n(\tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_i + \mathbf{z}_i^T\mathbf{z}_i)\mathbf{e}_i\mathbf{e}_i^T\big) - C(\mathbf{1}_n, \boldsymbol{\alpha}, \tilde{X}, Z)(\boldsymbol{\alpha}, \mathbf{1}_n, \tilde{X}, Z)^T \\ &= D_n - CQ_1Q_2^T.\end{aligned}$$

This completes the proof of Proposition 2. □

B.3. Proof of Theorem 1

To prove the convergence of Algoritm 1, we first show that each iteration of the algorithm does not increase the objective (8); that is

$$q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)}) \geq q(\hat{\boldsymbol{\beta}}^{(r+1)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r+1)}). \tag{A.5}$$

Recall that each iteration of the algorithm consists of four minimization steps, each of which updates estimations for one subset of parameters while fixing the others. These minimization steps are implemented using gradient descent with the Armijo rule, which ensures that the objective (8) keeps going downhill after each minimization step. Therefore, we have

$$\begin{aligned} q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)}) &\geq q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)}) \\ &\geq q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r)}) \\ &\geq q(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r+1)}) \\ &\geq q(\hat{\boldsymbol{\beta}}^{(r+1)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \hat{\boldsymbol{c}}^{(r+1)}). \end{aligned}$$

Furthermore, the objective (8) is defined as

$$q(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) = h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) + \frac{\lambda_Z}{2}\|Z\|_F^2 + \frac{\lambda_{\boldsymbol{\alpha}}}{2}\|\boldsymbol{\alpha}\|_2^2 + \frac{\lambda_{\boldsymbol{c}}}{2}\|\boldsymbol{c}\|_2^2 + \frac{\lambda_{\boldsymbol{\beta}}}{2}\|\boldsymbol{\beta}\|_2^2.$$

where the negative log-likelihood function $h(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) \geq 0$ and and λ_Z, $\lambda_{\boldsymbol{\alpha}}$, $\lambda_{\boldsymbol{c}}$ and $\lambda_{\boldsymbol{\beta}}$ are positive hyper-parameters. It immediately implies $q(\boldsymbol{\beta}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) \geq 0$. Since Algorithm 1 monotonically decreases the lower-bounded objective (8), it is guaranteed to converge. □

Appendix C: Multiclass Credit Risk Prediction Problem

C.1. Derivation of Equation (12)

Given a training data set of users' observed characteristics and their multiclass credit risk outcomes $\{\mathbf{x}_i, y_i\}$, $y_i \in \{0, 1, \ldots, K\}$, $i = 1, 2, \ldots, n$, we have

$$\begin{aligned} h^m(\boldsymbol{\beta}_{\{1:K\}}, Z, \boldsymbol{\alpha}, \boldsymbol{c}) &= -\frac{1}{n}\log\Big(\prod_{i=1}^{n} p_{k,i}^{\delta_{k,i}}\Big) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K}\delta_{k,i}\log p_{k,i}, \end{aligned} \tag{A.6}$$

where $p_{k,i}$ is the probability that user v_i belongs to credit risk outcome k and $\delta_{k,i}=1$ if $y_i=k$ and $\delta_{k,i}=0$ otherwise.

Recall that $\theta_{k,i}=\log\frac{p_{k,i}}{p_{0,i}}$ and $\sum_{k=0}^{K}p_{k,i}=1$. Therefore, we have

$$p_{0,i}=\frac{1}{1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}}},\quad p_{k,i}=\frac{e^{\theta_{k,i}}}{1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}}},\quad k=1,\cdots,K. \tag{A.7}$$

Substitute the expressions of the probabilities in Equation (A.7) into Equation (A.6), we have

$$h^m(\boldsymbol{\beta}_{\{1:K\}},Z,\boldsymbol{\alpha},\boldsymbol{c})=-\frac{1}{n}\sum_{i=1}^{n}\big(-\delta_{0,i}\log(1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}})+\sum_{k=1}^{K}(\delta_{k,i}\theta_{k,i}-\delta_{k,i}\log(1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}})\big)$$

$$=\frac{1}{n}\sum_{i=1}^{n}\big((\sum_{k=1}^{K}-\delta_{k,i}\theta_{k,i})+(\sum_{k=0}^{K}\delta_{k,i})\log(1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}})\big)$$

Noting that $\sum_{k=0}^{K}\delta_{k,i}=1$ and replacing $\tilde{k}$ with k, we immediately have

$$h^m(\boldsymbol{\beta}_{\{1:K\}},Z,\boldsymbol{\alpha},\boldsymbol{c})=\frac{1}{n}\sum_{i=1}^{n}\big((\sum_{k=1}^{K}-\delta_{k,i}\theta_{k,i})+\log(1+\sum_{k=1}^{K}e^{\theta_{k,i}})\big).\quad\square$$

C.2. Parameter Estimation for the Multiclass Credit Risk Prediction Problem

Algorithm 3 An alternating minimization algorithm for parameter estimation (Multiclass Case).

Input: ε: convergence threshold, $\varepsilon>0$

$\{\mathbf{x}_i,y_i\}_{i=1}^{n}$: training data

Output: $\hat{\boldsymbol{\beta}}_{\{1:K\}},\hat{Z},\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{c}}$: parameter estimations

$r=0$. // r: iteration count

Initialize $\hat{Z}^{(r)}$, $\hat{\boldsymbol{\alpha}}^{(r)}$, $\hat{\boldsymbol{c}}^{(r)}$, $\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)}$.

Compute $q^{m,(r)}=q^m(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)},\hat{Z}^{(r)},\hat{\boldsymbol{\alpha}}^{(r)},\hat{\boldsymbol{c}}^{(r)})$ according to Equation (13).

repeat

  $\hat{Z}^{(r+1)}=\arg\min_Z q^m(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)},Z,\hat{\boldsymbol{\alpha}}^{(r)},\hat{\boldsymbol{c}}^{(r)})$.

  $\hat{\boldsymbol{\alpha}}^{(r+1)}=\arg\min_{\boldsymbol{\alpha}} q^m(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)},\hat{Z}^{(r+1)},\boldsymbol{\alpha},\hat{\boldsymbol{c}}^{(r)})$.

  $\hat{\boldsymbol{c}}^{(r+1)}=\arg\min_{\boldsymbol{c}} q^m(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)},\hat{Z}^{(r+1)},\hat{\boldsymbol{\alpha}}^{(r+1)},\boldsymbol{c})$.

  $\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r+1)}=\arg\min_{\boldsymbol{\beta}_{\{1:K\}}} q^m(\boldsymbol{\beta}_{\{1:K\}},\hat{Z}^{(r+1)},\hat{\boldsymbol{\alpha}}^{(r+1)},\hat{\boldsymbol{c}}^{(r+1)})$.

  Compute $q^{m,(r+1)}=q^m(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r+1)},\hat{Z}^{(r+1)},\hat{\boldsymbol{\alpha}}^{(r+1)},\hat{\boldsymbol{c}}^{(r+1)})$ according to Equation (13).

  $r=r+1$.

until $|q^{m,(r)}-q^{m,(r-1)}|<\varepsilon$.

return $(\hat{\boldsymbol{\beta}}_{\{1:K\}},\hat{Z},\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{c}})=(\hat{\boldsymbol{\beta}}_{\{1:K\}}^{(r)},\hat{Z}^{(r)},\hat{\boldsymbol{\alpha}}^{(r)},\hat{\boldsymbol{c}}^{(r)})$.

C.3. Proof of Proposition 4

We first derive the partial derivative of h^m (defined in Equation (12)) with respect to $\theta_{k,i}$:

$$\begin{aligned}\frac{\partial h^m}{\partial \theta_{k,i}} &= \frac{\partial\big(\frac{1}{n}\sum_{i=1}^{n}\big((\sum_{\tilde{k}=1}^{K}-\delta_{\tilde{k},i}\theta_{\tilde{k},i})+\log(1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}})\big)\big)}{\partial\theta_{k,i}}\\ &= \frac{1}{n}(-\delta_{k,i}+\frac{e^{\theta_{k,i}}}{1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}}}).\end{aligned}$$

Define $\boldsymbol{\delta}_k=(\delta_{k,1},\cdots,\delta_{k,n})^T$, $\boldsymbol{\pi}_k^m=(\pi_{k,1}^m,\cdots,\pi_{k,n}^m)^T$, and $\pi_{k,i}^m=\frac{e^{\theta_{k,i}}}{1+\sum_{\tilde{k}=1}^{K}e^{\theta_{\tilde{k},i}}}$. We have

$$\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}=\frac{1}{n}(-\boldsymbol{\delta}_k+\boldsymbol{\pi}_k^m). \tag{A.8}$$

Given Equations (A.1), (A.8), (11) and (13), we have

$$\begin{aligned}\frac{\partial q^m}{\partial z_{ij}} &= \frac{\partial h^m}{\partial z_{ij}}+\lambda_Z z_{ij}=\sum_{k=1}^{K}\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial \boldsymbol{\theta}_k}{\partial z_{ij}}+\lambda_Z z_{ij}=\sum_{k=1}^{K}\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial(M^{-1}X\boldsymbol{\beta}_k)}{\partial z_{ij}}+\lambda_Z z_{ij}=\sum_{k=1}^{K}\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial M^{-1}}{\partial z_{ij}}X\boldsymbol{\beta}_k+\lambda_Z z_{ij}\\
&=\sum_{k=1}^{K}[\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C(Z_j\mathbf{e}_i^T+\mathbf{e}_iZ_j^T-2z_{ij}\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta}_k]+\lambda_Z z_{ij}\\
&=\sum_{k=1}^{K}[(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}CZ_j)(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}_k)+(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{e}_i)(Z_j^TM^{-1}X\boldsymbol{\beta}_k)\\
&\quad-2z_{ij}(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{e}_i)(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}_k)]+\lambda_Z z_{ij}\\
&=\sum_{k=1}^{K}[(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}CZ\mathbf{e}_j)+(\mathbf{e}_i^TC(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^TX^T(M^{-1})^TZ\mathbf{e}_j)\\
&\quad-2z_{ij}(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}_k)(\mathbf{e}_i^TC(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})]+\lambda_Z z_{ij}\\
&=\sum_{k=1}^{K}[\mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta}_k\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}CZ\big)\mathbf{e}_j+\mathbf{e}_i^T\big(C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}\boldsymbol{\beta}_k^TX^T(M^{-1})^TZ\big)\mathbf{e}_j\\
&\quad-2\big(\mathbf{e}_i^TZ\mathbf{e}_j\big)(\mathbf{e}_i^T\big((M^{-1}X\boldsymbol{\beta}_k)\odot(C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})\big)\mathbf{1}_d^T\mathbf{e}_j)]+\lambda_Z\mathbf{e}_i^TZ\mathbf{e}_j,\end{aligned}$$

which implies that

$$\begin{aligned}\frac{\partial q^m}{\partial Z} &= \sum_{k=1}^{K}[(M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}CZ)+(C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^TX^T(M^{-1})^TZ)\\
&\quad-2Z\odot\big((M^{-1}X\boldsymbol{\beta}_k)\odot(C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})\mathbf{1}_d^T\big)]+\lambda_ZZ.\end{aligned}$$

With Equations (A.2), (A.8), (11) and (13), we have

$$\frac{\partial q^m}{\partial \alpha_i} = \frac{\partial h^m}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial \boldsymbol{\theta}_k}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial (M^{-1}X\boldsymbol{\beta}_k)}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial M^{-1}}{\partial \alpha_i}X\boldsymbol{\beta}_k + \lambda_{\boldsymbol{\alpha}}\alpha_i$$

$$= \sum_{k=1}^{K}[\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C(\mathbf{1}_n\mathbf{e}_i^T + \mathbf{e}_i\mathbf{1}_n^T - 2\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta}_k] + \lambda_{\boldsymbol{\alpha}}\alpha_i$$

$$= \sum_{k=1}^{K}[(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{1}_n) + (\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^T X^T(M^{-1})^T\mathbf{1}_n)$$

$$- 2(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta}_k)(\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})] + \lambda_{\boldsymbol{\alpha}}\alpha_i$$

$$= \sum_{k=1}^{K}[\mathbf{e}_i^T(M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{1}_n) + \mathbf{e}_i^T(C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^T X^T(M^{-1})^T\mathbf{1}_n)$$

$$- 2\mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta}_k \odot C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}\big)] + \lambda_{\boldsymbol{\alpha}}\mathbf{e}_i^T\boldsymbol{\alpha},$$

which implies that

$$\frac{\partial q^m}{\partial \boldsymbol{\alpha}} = \sum_{k=1}^{K}[(M^{-1}X\boldsymbol{\beta}_k)(\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}C\mathbf{1}_n) + (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\boldsymbol{\beta}_k^T X^T(M^{-1})^T\mathbf{1}_n)$$

$$- 2(M^{-1}X\boldsymbol{\beta}_k) \odot (C(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})] + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha}.$$

Employing Equations (A.3), (A.4), (6) and (8), we have

$$\frac{\partial q^m}{\partial c_i} = \frac{\partial h^m}{\partial c_i} + \lambda_{\mathbf{c}}c_i = \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial \boldsymbol{\theta}_k}{\partial c_i} + \lambda_{\mathbf{c}}c_i = \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial (M^{-1}X\boldsymbol{\beta}_k)}{\partial c_i} + \lambda_{\mathbf{c}}c_i$$

$$= \sum_{k=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial M^{-1}}{\partial c_i}X\boldsymbol{\beta}_k + \lambda_{\mathbf{c}}c_i = \sum_{k=1}^{K}[\frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\big(M^{-1}\mathbf{e}_i\mathbf{e}_i^T GM^{-1}\big)X\boldsymbol{\beta}_k] + \lambda_{\mathbf{c}}c_i$$

$$= \sum_{k=1}^{K}[(\mathbf{e}_i^T(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k})(\mathbf{e}_i^T GM^{-1}X\boldsymbol{\beta}_k)] + \lambda_{\mathbf{c}}c_i = \sum_{k=1}^{K}[\mathbf{e}_i^T\big(((M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}) \odot (GM^{-1}X\boldsymbol{\beta}_k)\big)] + \lambda_{\mathbf{c}}\mathbf{e}_i^T\mathbf{c},$$

which implies that

$$\frac{\partial q^m}{\partial \mathbf{c}} = \sum_{k=1}^{K}[\big((M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k}\big) \odot \big(GM^{-1}X\boldsymbol{\beta}_k\big)] + \lambda_{\mathbf{c}}\mathbf{c}$$

Given Equations (A.8), (11) and (13), we have

$$\frac{\partial q^m}{\partial \beta_{k,p}} = \frac{\partial h^m}{\partial \beta_{k,p}} + \lambda_{\boldsymbol{\beta}}\beta_{k,p} = \sum_{\tilde{k}=1}^{K} \frac{\partial h^m}{\partial \boldsymbol{\theta}_{\tilde{k}}^T}\frac{\partial \boldsymbol{\theta}_{\tilde{k}}}{\partial \beta_{k,p}} + \lambda_{\boldsymbol{\beta}}\beta_{k,p} = \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}\frac{\partial (M^{-1}X\boldsymbol{\beta}_k)}{\partial \beta_{k,p}} + \lambda_{\boldsymbol{\beta}}\beta_{k,p}$$

$$= \frac{\partial h^m}{\partial \boldsymbol{\theta}_k^T}M^{-1}X\mathbf{e}_p + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}_k^T\mathbf{e}_p = \mathbf{e}_p^T(X^T(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k} + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}_k),$$

which implies that

$$\frac{\partial q^m}{\partial \boldsymbol{\beta}_k} = X^T(M^{-1})^T\frac{\partial h^m}{\partial \boldsymbol{\theta}_k} + \lambda_{\boldsymbol{\beta}}\boldsymbol{\beta}_k.$$

This completes the proof of Proposition 4. □

Appendix D: Numerical Credit Risk Prediction Problem

D.1. Parameter Estimation for the Numerical Credit Risk Prediction Problem

Algorithm 4 An alternating minimization algorithm for parameter estimation (Numerical Case).

Input: ε: convergence threshold, $\varepsilon > 0$; $\{\mathbf{x}_i, y_i\}_{i=1}^n$: training data

Output: $\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}}$: parameter estimations

$r = 0$. // r: iteration count

Initialize $\hat{Z}^{(r)}$, $\hat{\boldsymbol{\alpha}}^{(r)}$, $\hat{\boldsymbol{c}}^{(r)}$.

Compute $\hat{\boldsymbol{\beta}}^{(r)}$ according to Equation (19).

Compute $q^{n,(r)} = q^n(\boldsymbol{\beta}^{(r)}, Z^{(r)}, \boldsymbol{\alpha}^{(r)}, \boldsymbol{c}^{(r)})$ according to Equation (18).

repeat

  $\hat{Z}^{(r+1)} = \arg\min_Z q^n(\hat{\boldsymbol{\beta}}^{(r)}, Z, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.

  $\hat{\boldsymbol{\alpha}}^{(r+1)} = \arg\min_{\boldsymbol{\alpha}} q^n(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \boldsymbol{\alpha}, \hat{\boldsymbol{c}}^{(r)})$.

  $\hat{\boldsymbol{c}}^{(r+1)} = \arg\min_{\boldsymbol{c}} q^n(\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r+1)}, \hat{\boldsymbol{\alpha}}^{(r+1)}, \boldsymbol{c})$.

  Compute $\hat{\boldsymbol{\beta}}^{(r+1)}$ according to Equation (19).

  Compute $q^{n,(r+1)} = q^n(\boldsymbol{\beta}^{(r+1)}, Z^{(r+1)}, \boldsymbol{\alpha}^{(r+1)}, \boldsymbol{c}^{(r+1)})$ according to Equation (18).

  $r = r + 1$.

until $|q^{n,(r)} - q^{n,(r-1)}| < \varepsilon$.

return $(\hat{\boldsymbol{\beta}}, \hat{Z}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{c}}) = (\hat{\boldsymbol{\beta}}^{(r)}, \hat{Z}^{(r)}, \hat{\boldsymbol{\alpha}}^{(r)}, \hat{\boldsymbol{c}}^{(r)})$.

D.2. Proof of Proposition 5

The partial derivative of loss function h^n (defined in Equation (17)) with respect to $\boldsymbol{\mu}$ is:

$$\frac{\partial h^n}{\partial \boldsymbol{\mu}} = \frac{\partial\left(\frac{1}{2}(Y-\boldsymbol{\mu})^T(Y-\boldsymbol{\mu})\right)}{\partial \boldsymbol{\mu}} = \frac{1}{2}\frac{\partial(Y^TY - 2\boldsymbol{\mu}^TY + \boldsymbol{\mu}^T\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \quad = \boldsymbol{\mu} - Y. \tag{A.9}$$

Given Equations (A.1), (A.9), (16) and (18), we have

$$\begin{aligned}
\frac{\partial q^n}{\partial z_{ij}} &= \frac{\partial h^n}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial \boldsymbol{\mu}}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial z_{ij}} + \lambda_Z z_{ij} = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial M^{-1}}{\partial z_{ij}}X\boldsymbol{\beta} + \lambda_Z z_{ij} \\
&= \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C(Z_j\mathbf{e}_i^T + \mathbf{e}_iZ_j^T - 2z_{ij}\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta} + \lambda_Z z_{ij} \\
&= (\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}CZ_j)(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}) + (\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{e}_i)(Z_j^TM^{-1}X\boldsymbol{\beta}) \\
&\quad - 2z_{ij}(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{e}_i)(\mathbf{e}_i^TM^{-1}X\boldsymbol{\beta}) + \lambda_Z z_{ij} \\
&= \mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta}\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}CZ\big)\mathbf{e}_j + \mathbf{e}_i^T\big(C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\boldsymbol{\beta}^TX^T(M^{-1})^TZ\big)\mathbf{e}_j \\
&\quad - 2\big(\mathbf{e}_i^TZ\mathbf{e}_j\big)\big(\mathbf{e}_i^T\big((M^{-1}X\boldsymbol{\beta})\odot(C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})\big)\mathbf{1}_d^T\mathbf{e}_j\big) + \lambda_Z\mathbf{e}_i^TZ\mathbf{e}_j,
\end{aligned}$$

which implies that

$$\frac{\partial q^n}{\partial Z} = (M^{-1}X\boldsymbol{\beta})(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}CZ) + (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\boldsymbol{\beta}^T X^T(M^{-1})^T Z)$$
$$- 2Z \odot \big((M^{-1}X\boldsymbol{\beta}) \odot (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})\mathbf{1}_d^T\big) + \lambda_Z Z.$$

With Equations (A.2), (A.9), (16) and (18), we have

$$\frac{\partial q^n}{\partial \alpha_i} = \frac{\partial h^n}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial \boldsymbol{\mu}}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial \alpha_i} + \lambda_{\boldsymbol{\alpha}}\alpha_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial M^{-1}}{\partial \alpha_i}X\boldsymbol{\beta} + \lambda_{\boldsymbol{\alpha}}\alpha_i$$
$$= \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C(\mathbf{1}_n\mathbf{e}_i^T + \mathbf{e}_i\mathbf{1}_n^T - 2\mathbf{e}_i\mathbf{e}_i^T)M^{-1}X\boldsymbol{\beta} + \lambda_{\boldsymbol{\alpha}}\alpha_i$$
$$= (\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{1}_n) + (\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n)$$
$$- 2(\mathbf{e}_i^T M^{-1}X\boldsymbol{\beta})(\mathbf{e}_i^T C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}) + \lambda_{\boldsymbol{\alpha}}\alpha_i$$
$$= \mathbf{e}_i^T(M^{-1}X\boldsymbol{\beta})(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{1}_n) + \mathbf{e}_i^T(C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n)$$
$$- 2\mathbf{e}_i^T\big(M^{-1}X\boldsymbol{\beta} \odot C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\big) + \lambda_{\boldsymbol{\alpha}}\mathbf{e}_i^T\boldsymbol{\alpha},$$

which implies that

$$\frac{\partial q^n}{\partial \boldsymbol{\alpha}} = (M^{-1}X\boldsymbol{\beta})(\frac{\partial h^n}{\partial \boldsymbol{\mu}^T}M^{-1}C\mathbf{1}_n) + (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\boldsymbol{\beta}^T X^T(M^{-1})^T\mathbf{1}_n)$$
$$- 2(M^{-1}X\boldsymbol{\beta}) \odot (C(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}) + \lambda_{\boldsymbol{\alpha}}\boldsymbol{\alpha}.$$

Employing Equations (A.3), (A.9), (16) and (18), we have

$$\frac{\partial q^n}{\partial c_i} = \frac{\partial h^n}{\partial c_i} + \lambda_{\boldsymbol{c}}c_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial \boldsymbol{\mu}}{\partial c_i} + \lambda_{\boldsymbol{c}}c_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial (M^{-1}X\boldsymbol{\beta})}{\partial c_i} + \lambda_{\boldsymbol{c}}c_i = \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\frac{\partial M^{-1}}{\partial c_i}X\boldsymbol{\beta} + \lambda_{\boldsymbol{c}}c_i$$
$$= \frac{\partial h^n}{\partial \boldsymbol{\mu}^T}\big(M^{-1}\mathbf{e}_i\mathbf{e}_i^T GM^{-1}\big)X\boldsymbol{\beta} + \lambda_{\boldsymbol{c}}c_i$$
$$= (\mathbf{e}_i^T(M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}})(\mathbf{e}_i^T GM^{-1}X\boldsymbol{\beta}) + \lambda_{\boldsymbol{c}}c_i$$
$$= \mathbf{e}_i^T\big(((M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}) \odot (GM^{-1}X\boldsymbol{\beta})\big) + \lambda_{\boldsymbol{c}}\mathbf{e}_i^T\boldsymbol{c},$$

which implies that

$$\frac{\partial q^n}{\partial \boldsymbol{c}} = \big((M^{-1})^T\frac{\partial h^n}{\partial \boldsymbol{\mu}}\big) \odot \big(GM^{-1}X\boldsymbol{\beta}\big) + \lambda_{\boldsymbol{c}}\boldsymbol{c}$$

This completes the proof of Proposition 5. □

Appendix E: Additional Analyses on Method Performance

E.1. Subgroup Analyses

We conducted subgroup analyses to further demonstrate the advantages of our proposed method. According to Equation (3), our proposed latent similarity-enhanced credit risk prediction model is given by:

$$\theta_i = \mathbf{x}_i^T \boldsymbol{\beta} + c_i \sum_{v_j \in V:\, j \neq i} s_{ij} \theta_j.$$

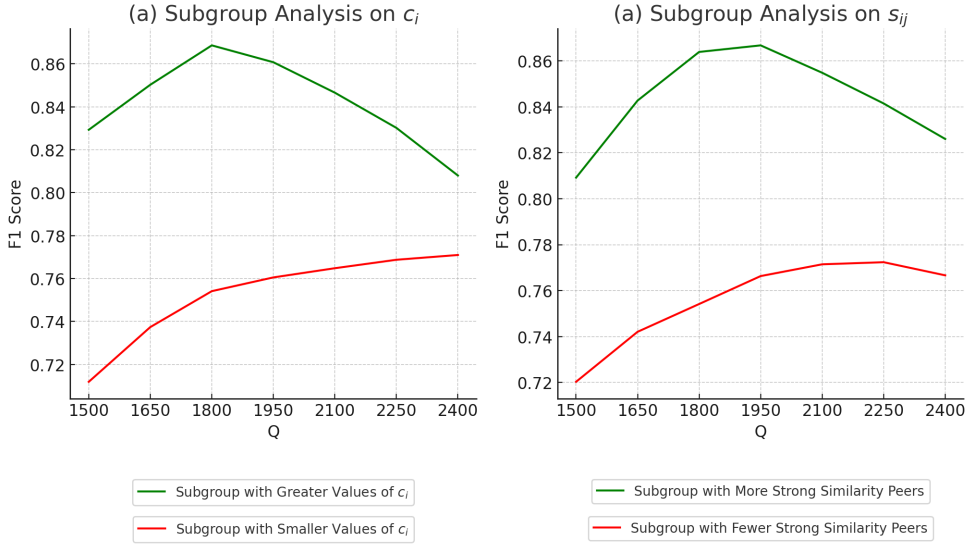
Here θ_i denotes user v_i's log-odds of delinquency. It is the addition of two components: the intrinsic component, represented by $\mathbf{x}_i^T \boldsymbol{\beta}$, and the similarity component $\sum_{v_j \in V:\, j \neq i} s_{ij} \theta_j$, which is the summation of other users' log-odds of delinquency weighted by the similarity between each of them and v_i (i.e., s_{ij}). User-specific parameter c_i regulates the relative importance between the intrinsic component and the similarity component in predicting θ_i.

First, we conducted a subgroup analysis with respect to c_i. We divided users into two equally sized subgroups based on their respective value of c_i. A user with a higher value of c_i indicates that the similarity component plays a more substantial role in predicting the user's delinquency. Averaged across all billing cycles, the AUC for the subgroup having smaller values of c_i stands at 0.905, while the AUC for the subgroup with greater values of c_i is 0.956. The result suggests that our method is more likely to correctly classify users, for whom the similarity component has a greater predictive role. Figure A.1(a) reports F1 scores for each subgroup across different values of Q. Similarly, F1 scores of the subgroup with higher values of c_i consistently surpass those of the subgroup with lower values of c_i These observations are in alignment with our expectation that if the individual behaves more independently, the benefit of considering the similarity component would be decreased, which in turn leads to the reduced advantage of the proposed method.

Next, we performed another subgroup analysis with respect to s_{ij}. For each user, we counted its number of user-pairs with strong similarities, where a strong similarity is defined to be above the median of all pair-wise similarities among users. We then constructed two equally sized subgroups of users based on their respective number of user-pairs with strong similarities. A user having

a greater number of user-pairs with strong similarities indicates that the user has more strong similarity peers. The average AUC for the subgroup of users with more strong similarity peers is 0.954; while for the subgroup of users with fewer such peers, it stands at 0.912. Figure A.1(b) plots the F1 score for each subgroup against the value of Q. As shown, F1 scores of the subgroup with more strong similarity peers consistently outperforms those of the subgroup with fewer such peers Again, these performance differences between the two subgroups are in alignment with the rationale of our method. When a user has strong similarities with more individuals, our method can potentially exploit more predictive power from these similar individuals' behaviors, resulting in improved predictive performance.

Figure A.1 Subgroup Analysis: F1 Score



E.2. Robustness Check

For robustness check, we further varied the number of predicted delinquent users from $Q = 3,000$ ($\sim 25\%$ of all users) to $Q = 4,800$ ($\sim 40\%$ of all users), with an increment of 300. Table A.1 reports the evaluation results for billing cycle $\Psi = 14$ in terms of precision, recall, and F1 score. Averaged across Q, the average precision of LSE is 0.824, outperforming that of benchmark methods by a range between 7.14% and 42.38%. Moreover, the average recall of LSE surpasses that of benchmark methods by a range between 7.32% and 42.53%, and the average F1 score improves by a range

between 7.23% and 42.46%. The performance improvement by our method over each benchmark in every evaluation metric, as reported in Table A.1, is also statistically significant ($p < 0.001$). Similarly, we obtained significant improvements by LSE over benchmarks for other billing cycles and omitted these evaluation results here for space consideration.

Table A.1 Performance Comparison for Credit Risk Prediction Problem: Varying Q ($\Psi = 14$).

Metric	Method	$Q=3000$	$Q=3300$	$Q=3600$	$Q=3900$	$Q=4200$	$Q=4500$	$Q=4800$	Mean	Std.
Precision	LSE (Our Method)	0.888	0.871	0.853	0.829	0.805	0.777	0.748	0.824	0.051
	RLR	0.634	0.617	0.598	0.579	0.560	0.541	0.524	0.579	0.040
	MLP	0.722	0.702	0.681	0.660	0.633	0.605	0.581	0.655	0.051
	SVM	0.669	0.648	0.630	0.613	0.595	0.576	0.553	0.612	0.041
	XGBoost	0.847	0.821	0.796	0.770	0.743	0.717	0.692	0.770	0.056
	RNN-GRU	0.836	0.806	0.775	0.747	0.720	0.695	0.670	0.750	0.060
	SIM-TemGNN	0.811	0.782	0.751	0.711	0.674	0.653	0.621	0.715	0.070
Recall	LSE (Our Method)	0.667	0.720	0.769	0.810	0.847	0.876	0.899	0.798	0.084
	RLR	0.476	0.510	0.539	0.566	0.589	0.610	0.630	0.560	0.055
	MLP	0.542	0.580	0.614	0.644	0.665	0.682	0.698	0.632	0.057
	SVM	0.503	0.536	0.568	0.598	0.626	0.649	0.664	0.592	0.060
	XGBoost	0.636	0.678	0.718	0.752	0.782	0.808	0.832	0.744	0.071
	RNN-GRU	0.628	0.666	0.698	0.729	0.757	0.783	0.806	0.724	0.064
	SIM-TemGNN	0.610	0.646	0.677	0.694	0.709	0.736	0.746	0.688	0.049
F1 Score	LSE (Our Method)	0.762	0.788	0.809	0.819	0.826	0.824	0.816	0.806	0.023
	RLR	0.544	0.558	0.567	0.572	0.574	0.573	0.572	0.566	0.011
	MLP	0.619	0.635	0.646	0.652	0.649	0.641	0.634	0.640	0.011
	SVM	0.574	0.587	0.597	0.605	0.610	0.610	0.603	0.598	0.013
	XGBoost	0.727	0.743	0.755	0.761	0.762	0.759	0.756	0.752	0.013
	RNN-GRU	0.717	0.729	0.735	0.738	0.738	0.737	0.732	0.732	0.007
	SIM-TemGNN	0.696	0.708	0.712	0.703	0.691	0.692	0.678	0.697	0.012

E.3. Latent Similarity Reconstruction

We conducted a simulation study to validate the meaningfulness of the unobserved (latent) similarity learned by our method. In the simulation study, we generated synthetic data to validate whether our method can effectively reconstruct unobserved similarities among users. Recall that the similarity s_{ij} between users v_i and v_j is defined as

$$s_{ij} = \alpha_i + \alpha_j + \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j + \mathbf{z}_i^T \mathbf{z}_j, \tag{A.10}$$

where $\tilde{\mathbf{x}}_i$ is the normalization of observed intrinsic characteristics of user v_i, $\mathbf{z}_i$ and α_i represent unobserved intrinsic characteristics of this user. Correspondingly, the unobserved similarity within s_{ij} is expressed as

$$s_{ij}^{\circ} := \alpha_i + \alpha_j + \mathbf{z}_i^T \mathbf{z}_j. \tag{A.11}$$

In the simulation study, we aimed to reconstruct the unobserved similarity s_{ij}° using our method.

To generate synthetic data, we first simulated each user with eight observed intrinsic characteristics and three unobserved intrinsic characteristics (two z's and one α). The value of each observed or unobserved intrinsic characteristic was generated from the standard normal distribution $N(0,1)$. Next, we need to generate the label (0 or 1) for each user, where 0 represents no delinquency and 1 denotes delinquency. For this purpose, with the generated observed and unobserved intrinsic characteristics of each user, we calculated the user's log-odds of delinquency θ_i using Equations (1) and (3). In this calculation, the regression coefficients $\boldsymbol{\beta}$ were simulated from a uniform distribution $U[5,20]$, and the parameter c_i was set to $1/50$. With a user's θ_i calculated, we computed the user's probability of delinquency p_i as $\frac{e^{\theta_i}}{1+e^{\theta_i}}$ and sampled the user's delinquency label y_i according to p_i. In our simulation study, we generated synthetic data for 50 users over 12 billing cycles.

Before executing our method, we removed the unobserved intrinsic characteristics from the generated synthetic data. Subsequently, we employed our method on the modified synthetic data to estimate these unobserved intrinsic characteristics. The estimated unobserved similarity $\hat{s}_{ij}^{\circ}$ between a pair of users was then calculated with their estimated unobserved intrinsic characteristics using Equation (A.11). Since we knew the true values of their unobserved intrinsic characteristics, we also computed the actual unobserved similarity s_{ij}° using these true values and Equation (A.11). It is common to validate the meaningfulness of a latent variable employed by a machine learning algorithm by showing how effective the variable can be reconstructed by the algorithm (Chen 2023). Accordingly, to validate the meaningfulness of the unobserved similarity in our study, we evaluated the effectiveness of reconstructing this latent variable by our method using the relative error (RE) measure defined below:

$$\mathrm{RE} = \frac{\sum_{i \neq j} (s_{ij}^{\circ} - \hat{s}_{ij}^{\circ})^2}{\sum_{i \neq j} (s_{ij}^{\circ})^2}. \tag{A.12}$$

The value of RE is in the range of $[0, +\infty]$, and a smaller relative error indicates a more effective reconstruction.

We conducted the simulation study for 100 times. The averaged relative error attained by our method across 100 simulation runs is 0.447. It is noted that our method achieves this reconstruction error without the knowledge of the distribution of unobserved intrinsic characteristics. In

comparison, let us consider a benchmark that has the knowledge of the distribution of unobserved intrinsic characteristics. Specifically, the benchmark sampled unobserved intrinsic characteristics from the standard normal distribution and then calculated the estimated unobserved similarity $\hat{s}_{ij}^{\circ}$ using Equation (A.11). The benchmark achieved an averaged relative error of 1.99 across 100 simulation runs. The significant outperformance of our method, operated without the distribution knowledge, over the benchmark equipped with the distribution knowledge demonstrates the effectiveness of our method in reconstructing the unobserved similarity.