Journal of Data Science 23 (4), 578–591
October 2025
DOI: 10.6339/25-JDS1165
Statistical Data Science

Mortgage Prepayment Modeling via a Smoothing Spline State Space Model

Haoran Lu[1], Huimin Cheng[2], Ye Wang[1], Yaoguo Xie[3], Huan Yan[3], Xindong Wang[4], Ping Ma[1,*], and Wenxuan Zhong[1,*]

[1]*Department of Statistics, University of Georgia, Athens, GA 30602, United States*
[2]*Department of Biostatistics, Boston University, Boston, MA 02118, United States*
[3]*Model Risk Management, Wells Fargo, Charlotte, NC 28202, United States*
[4]*Model Risk Management, Wells Fargo, McLean, VA 22102, United States*

Abstract

Loan behavior modeling is crucial in financial engineering. In particular, predicting loan prepayment based on large-scale historical time series data of massive customers is challenging. Existing approaches, such as logistic regression or nonparametric regression, could only model the direct relationship between the features and the prepayments. Motivated by extracting the hidden states of loan behavior, we propose the smoothing spline state space (QuadS) model based on a hidden Markov model with varying transition and emission matrices modeled by smoothing splines. In contrast to existing methods, our method benefits from capturing the loans' unobserved state transitions, which not only increases prediction performances but also provides more interpretability. The overall model is learned by EM algorithm iterations, and within each iteration, smoothing splines are fitted with penalized least squares. Simulation studies demonstrate the effectiveness of the proposed method. Furthermore, a real-world case study using loan data from the Federal National Mortgage Association illustrates the practical applicability of our model. The QuadS model not only provides reliable predictions but also uncovers meaningful, hidden behavior patterns that can offer valuable insights for the financial industry.

Keywords *hidden Markov model; mortgage prepayment; nonparametric model; smoothing spline ANOVA*

1 Introduction

Over the past few decades, advances in science and technology have led to the routine generation of vast amounts of data. The term "big data" has become a part of everyday language. Researchers across various fields are striving to harness these massive datasets to make predictions and uncover patterns and anomalies. These efforts have not only fueled emerging areas like artificial intelligence but have also revitalized traditional sectors such as finance. Among existing financial models, loan prepayment behavior modeling (LPBM) has consistently been a key focus. LPBM plays a crucial role in forecasting market trends and preparing for reinvestment. Further, an effective LPBM can help manage risks in the home mortgage market, as demonstrated by events like the subprime mortgage crisis of 2007–2008 and the severe impact of the COVID-19 outbreak in 2019 (Van Deventer et al., 2013; Fuster et al., 2021; Agarwal et al.,

*Corresponding author. Email: pingma@uga.edu or wenxuan@uga.edu.

Received August 25, 2024; Accepted January 6, 2025

2020). For example, during the COVID-19 period from early 2020 to late 2021, the interest rate and mortgage rate dropped to an unprecedented low level. The 30-year primary mortgage rates dropped to a lower than 3% level during that period (Freddie Mac, 2024), which drove a huge historical wave of mortgage refinance with the monthly prepayment rate increasing from less than 10% to around 40% (Federal Housing Finance Agency, 2024). This means nearly 40% of the loans are expected to be repaid from the bank's mortgage servicing portfolio. This kind of behavior introduces a significant challenge to prepayment modeling since the underlying borrow behavior may be different due to rate and policy change.

Modeling loan prepayment behavior presents significant challenges that arise from both the scale of the data and the complexity of the underlying behaviors. Financial institutions often work with datasets of enormous magnitude, such as records for 30 million loans spanning over two decades, amounting to hundreds of gigabytes of information. Extracting meaningful insights from such large-scale data requires sophisticated computational methods that can handle the volume without compromising on accuracy. Furthermore, loan prepayment behavior is influenced by a myriad of factors, including market conditions, borrower characteristics, and economic policies, which interact in complex, often nonlinear ways. Traditional modeling techniques, such as logistic regression and basic time series models, frequently fail to capture the full extent of these interactions, leading to inaccurate predictions. This complexity calls for the development of advanced models that not only integrate financial and computational knowledge but also remain interpretable. This interpretability is crucial, as stakeholders must understand the rationale behind model predictions to design effective mortgage products and policies that enhance risk management and contribute to the overall stability of the financial market (Johnson et al., 2019).

Existing quantitative methods of loan behavior modeling can be broadly categorized into two approaches: statistical approaches and machine learning-artificial intelligence (ML-AI) approaches. Statistical analysis of loan behavior includes various approaches with a long history. One major category consists of regression models such as logistic regression and non-parametric regression (Kung et al., 2010; Maxam and LaCour-Little, 2001). Another important category includes stochastic process models such as hidden Markov model (HMM) (Lai et al., 2014), input-output HMM (IOHMM) (Bengio and Frasconi, 1995, 1996) and autoregressive models (Berger et al., 2018). Recently, the rise of big data modeling has enabled the application of ML-AI approaches to loan behavior analysis. Compared with classic statistical approaches, they have achieved significant success, especially in the prediction accuracy (Sirignano et al., 2016; Aldridge and Avellaneda, 2019; Ozbayoglu et al., 2020). However, their success has been met with skepticism from the financial community, including both industry professionals and regulatory bodies (Johnson et al., 2019). The primary concern is that ML-AI models often operate as black boxes (Guidotti et al., 2018; Fang et al., 2024), making them difficult to interpret and challenging to integrate with financial insights during model design. As a result, these models may not be suitable for high-fidelity decision-making.

In this paper, we aim to overcome the aforementioned challenges by developing a powerful and reliable statistical framework for loan behavior modeling. Specifically, we focus on the time series loan prepayment data of multiple loans from the Federal National Mortgage Association. Our goal is to build a precise and interpretable model for underlying loan behavior patterns by incorporating both statistical and financial insights. Classic regression approaches model the prepayment behavior directly as a function of features. However, the underlying pattern of loan behavior does not necessarily have a direct relationship with features. With the insights from the financial industry, it is better to model the loan behavior with a hidden layer of states (Lai et al., 2014). That is, the borrower has a hidden state of either "active" (more likely to prepay) or

"passive" (less likely to prepay) at each time point. The challenge is how to model the relationship of features, hidden states, and observed prepayment behavior. The general smoothing spline model has been a powerful method to model complicated nonlinear relationships (Gu, 2013; Helwig and Ma, 2015; Ma et al., 2015; Gu and Ma, 2005). Recent developments in smoothing spline enlarge the model's capability in large and complicated data and expand its application into more scientific fields (Sun et al., 2021; Meng et al., 2020). We propose the smoothing spline state space (QuadS) model, which defines the hidden states using a modified HMM framework with varying transition and emission matrices modeled as general smoothing splines of loan features. Unlike classic HMM with only the response, our proposed method introduces the features into the model. It is able to capture the complicated nonlinear relationship of features, hidden states, and emitted loan behaviors. The overall model is estimated by EM algorithm, and within each iteration, smoothing splines are fitted by penalized least squares.

Contributions Our methodological contribution is proposing the QuadS model with an applicable estimation procedure, incorporating financial insight by modeling loan behavior hidden states. Extensive simulations and case studies showcase the privilege of QuadS in loan prepayment modeling over existing methods. The QuadS model is a powerful tool for the financial industry to better understand and manage loan portfolios, anticipate market trends, and develop effective risk management strategies.

The rest of the paper is organized as follows. In Section 2, we describe the proposed QuadS model and the EM algorithm estimation procedure. In Section 3, the simulation study shows the performance of the proposed model. A detailed case study on loan data follows in Section 4. In Section 5, we conclude the article.

2 Methodology

In the following sections, we first define the key terminologies and notations before presenting the detailed setup of our proposed model, including its structural components and underlying assumptions. We then develop the estimation method, and algorithms used to fit the model to the data.

2.1 Model Specification

Let R denote the number of loans and T_r denote the number of time points of the rth loan for $r = 1, \ldots, R$. We observe the prepayment indicators $\mathbf{Y} = \{Y^{\langle r,t\rangle}\}$, where $Y^{\langle r,t\rangle} \in \{$"unprepaid"=0, "prepaid"=1$\}$ represents whether the rth loan is prepaid at time t. Note that a loan record ends when it is paid, so $\{Y^{\langle r,1\rangle}, Y^{\langle r,2\rangle}, \ldots, Y^{\langle r,T_r\rangle}\}$ will always have the form of $\{0, 0, 0, \ldots, 0, 1\}$. We also observe a p-dimensional feature variable $\mathbf{Z} = \{\mathbf{Z}^{\langle r,t\rangle}\}$, $\mathbf{Z}^{\langle r,t\rangle} \in \mathbb{R}^p$. Within the p features at time point t, some features represent some market factor, e.g., the unemployment rate, and some represent factors of individuals, e.g., the borrower's credit score. In our setting, we further assume that each loan has a hidden state at each time point. Denote the hidden states as $\mathbf{X} = \{X^{\langle r,t\rangle}\}$, where $X^{\langle r,t\rangle} \in \{$"passive"=0, "active"=1$\}$. The passive state means the loan has a small probability of being prepaid, and the active state means the loan has a relatively large probability of being prepaid.

We propose the smoothing spline state space (QuadS) model to analyze the relationship of the observed features $\{Z^{\langle r,t\rangle}\}$, observed prepayment indicators $\{Y^{\langle r,t\rangle}\}$ and unobserved hidden states $\{X^{\langle r,t\rangle}\}$. Figure 1 gives an illustration of the QuadS model structure. Analogous to classic

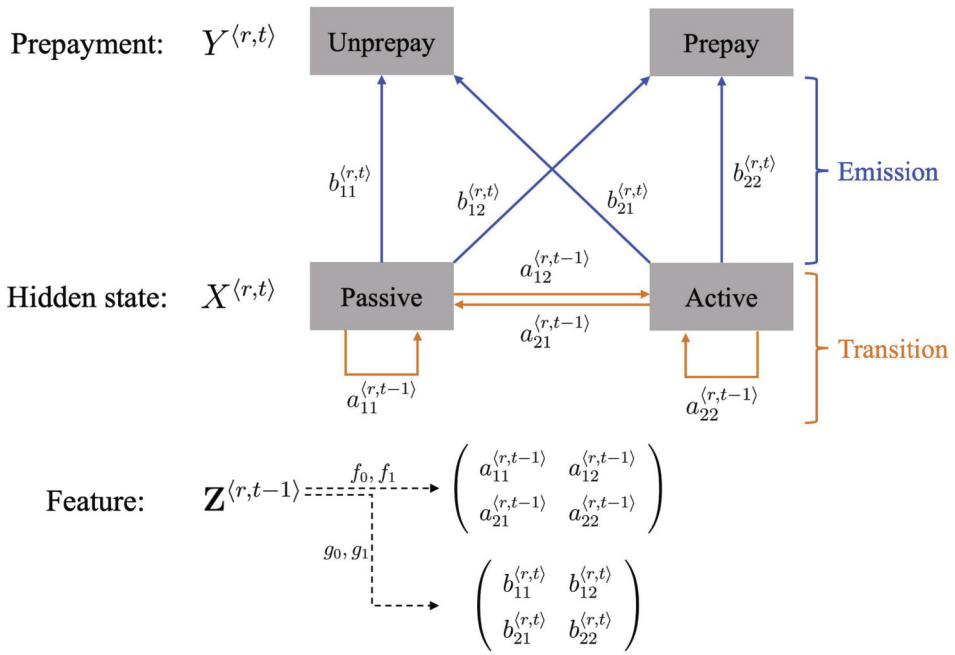

Figure 1: Illustration of the QuadS model structure.

HMM, we assume $Y^{\langle r,t\rangle}$ only depends on $X^{\langle r,t\rangle}$. The probability of observing $Y^{\langle r,t\rangle}$ given $X^{\langle r,t\rangle}$ is specified by a fixed emission probability $\mathbf{B}^{\langle r,t\rangle}$ which varies for different r and t, where

$$\mathbf{B}^{\langle r,t\rangle} = \begin{pmatrix} P(Y^{\langle r,t\rangle}=0|X^{\langle r,t\rangle}=0) & P(Y^{\langle r,t\rangle}=1|X^{\langle r,t\rangle}=0) \\ P(Y^{\langle r,t\rangle}=0|X^{\langle r,t\rangle}=1) & P(Y^{\langle r,t\rangle}=1|X^{\langle r,t\rangle}=1) \end{pmatrix} = \begin{pmatrix} b_{11}^{\langle r,t\rangle} & b_{12}^{\langle r,t\rangle} \\ b_{21}^{\langle r,t\rangle} & b_{22}^{\langle r,t\rangle} \end{pmatrix}.$$

We further assume $X^{\langle r,t\rangle}$ only depends on $X^{\langle r,t-1\rangle}$. Then, the probability of $X^{\langle r,t-1\rangle}$ changing to $X^{\langle r,t\rangle}$ given $X^{\langle r,t-1\rangle}$ is specified by a transition matrix $\mathbf{A}^{\langle r,t-1\rangle}$ which varies with different r and t, where

$$\begin{aligned} \mathbf{A}^{\langle r,t-1\rangle} &= \begin{pmatrix} P(X^{\langle r,t\rangle}=0|X^{\langle r,t-1\rangle}=0) & P(X^{\langle r,t\rangle}=1|X^{\langle r,t-1\rangle}=0) \\ P(X^{\langle r,t\rangle}=0|X^{\langle r,t-1\rangle}=1) & P(X^{\langle r,t\rangle}=1|X^{\langle r,t-1\rangle}=1) \end{pmatrix} \\ &= \begin{pmatrix} a_{11}^{\langle r,t-1\rangle} & a_{12}^{\langle r,t-1\rangle} \\ a_{21}^{\langle r,t-1\rangle} & a_{22}^{\langle r,t-1\rangle} \end{pmatrix}. \end{aligned}$$

With the transition and emission matrices, we are able to define the following conditional distributions of $X^{\langle r,t\rangle}$ and $Y^{\langle r,t\rangle}$ by

$$\begin{aligned} \big(Y^{\langle r,t\rangle}=j|X^{\langle r,t\rangle}=i\big) &\sim \mathrm{Ber}\big(b_{ij}^{\langle r,t\rangle}\big), \quad \text{for } i=0,1,\ j=0,1, \quad \text{and} \\ \big(X^{\langle r,t\rangle}=j|X^{\langle r,t-1\rangle}=i\big) &\sim \mathrm{Ber}\big(a_{ij}^{\langle r,t-1\rangle}\big), \quad \text{for } i=0,1,\ j=0,1. \end{aligned}$$

We define the initial probability $\boldsymbol{\pi} = (\pi_0, \pi_1)$ by

$$P\big(X^{\langle r,1\rangle}=0\big)=\pi_0, \quad P\big(X^{\langle r,1\rangle}=1\big)=\pi_1, \quad \text{where } \pi_0+\pi_1=1.$$

Define the functions f_0, f_1, g_0, g_1 by

$$\mathrm{logit}\big(b_{12}^{\langle r,t\rangle}\big) = g_0\big(\mathbf{Z}^{\langle r,t-1\rangle}\big), \quad b_{11}^{\langle r,t\rangle} = 1 - b_{12}^{\langle r,t\rangle},$$

$$\begin{aligned}
\operatorname{logit}\big(b_{22}^{\langle r,t\rangle}\big) &= g_1\big(\mathbf{Z}^{\langle r,t-1\rangle}\big), \quad b_{21}^{\langle r,t\rangle} = 1 - b_{22}^{\langle r,t\rangle}, \\
\operatorname{logit}\big(a_{12}^{\langle r,t-1\rangle}\big) &= f_0\big(\mathbf{Z}^{\langle r,t-1\rangle}\big), \quad a_{11}^{\langle r,t-1\rangle} = 1 - a_{12}^{\langle r,t-1\rangle}, \\
\operatorname{logit}\big(a_{22}^{\langle r,t-1\rangle}\big) &= f_1\big(\mathbf{Z}^{\langle r,t-1\rangle}\big), \quad a_{21}^{\langle r,t-1\rangle} = 1 - a_{22}^{\langle r,t-1\rangle}.
\end{aligned}$$

Denote the four unknown functions (f_0, f_1, g_0, g_1) by $\mathscr{F}$. By using different specific function classes of $\mathscr{F}$, QuadS can accommodate various application scenarios. Note that classic HMM is a special case of QuadS with f_0, f_1, g_0, g_1 being constant functions.

2.2 Estimation

We estimate $\boldsymbol{\pi}$ and $\mathscr{F}$ by minimizing the following penalized complete data log-likelihood functional

$$\begin{aligned}
l(\boldsymbol{\pi}, \mathscr{F}) = &- \sum_{r=1}^{R} \Big\{ \log P\big(X^{\langle r,1\rangle} | \boldsymbol{\pi}\big) + \sum_{t=2}^{T_r} \log P\big(X^{\langle r,t\rangle} | X^{\langle r,t-1\rangle}, \mathbf{Z}, \mathscr{F}\big) + \sum_{t=1}^{T_r} \log P\big(Y^{\langle r,t\rangle} | X^{\langle r,t\rangle}, \mathscr{F}\big) \Big\} \\
&+ \sum_{\substack{f \in \\ \{f_0, f_1, g_0, g_1\}}} \frac{1}{2} \lambda_f J(f), \qquad (1)
\end{aligned}$$

where the first term is the negative log-likelihood, $J(f) = J(f, f)$ is a quadratic functional that quantifies the roughness of f, and $\{\lambda_f\}$ are smoothing parameters that control the trade-off between the goodness of fit and the smoothness. Note that in (1), $\mathbf{Z}$ and $\mathbf{Y}$ are observed data, and $\mathbf{X}$ are unobserved.

To discuss the assumptions, we use a general symbol f to represent either one of f_0, f_1, g_0, g_1, and a general symbol $\mathbf{Z} = (z_{\langle 1\rangle}, z_{\langle 2\rangle}, \ldots, z_{\langle p\rangle})^T$ to represent $\mathbf{Z}^{\langle r,t\rangle}$. Each entry $z_{\langle k\rangle}$ takes values in $\mathbb{R}$, and we assume f to be a smooth function defined on $\mathbb{R}^p$. The functional ANOVA decomposition of f is

$$f(\mathbf{Z}) = f_0^* + \sum_{j=1}^{p} f_j^*(z_{\langle j\rangle}) + \sum_{j=1}^{p} \sum_{k=j+1}^{p} f_{jk}^*(z_{\langle j\rangle}, z_{\langle k\rangle}) + \cdots + f_{1,\ldots,d}^*(z_{\langle 1\rangle}, \ldots, z_{\langle d\rangle}), \qquad (2)$$

where f_0^* is a constant, f_j^*'s are main effects, f_{jk}^*'s are two-way interactions, and so on. In practical data analysis, one usually includes only the main effects, with the possible addition of a few lower-order interactions. Higher-order interactions are less interpretable yet more difficult to estimate, so they are often excluded in practical estimation to control model complexity (Gu, 2013).

To minimize (1), we consider smooth functions in the space $\{f : J(f) < \infty\}$ or a subspace therein. As an abstract generalization of the vector spaces used extensively in multivariate analysis, Hilbert spaces inherit many nice properties of the vector spaces. However, Hilbert spaces are not strict enough for our model because it cannot ensure the functional (1) to be continuous in f. Therefore, we need a constrained Hilbert space in which the evaluation functional is continuous. Such a Hilbert space is referred to as a reproducing kernel Hilbert space (RKHS). For example, the space of functions with square-integrable second derivatives is an RKHS if it is equipped with appropriate inner products (Gu, 2013). For the evaluation functional $[x](\cdot)$, by the Riesz representation theorem, there exists a nonnegative definite bivariate function $R(x, y)$, the reproducing kernel, that satisfies $\langle R(x, \cdot), f(\cdot)\rangle = f(x)$, called the representer of $[x](\cdot)$, in an RKHS. Given an RKHS, we can derive the reproducing kernel from the Green function

associated with the quadratic functional $J(\cdot)$. (The construction of the reproducing kernel is beyond the scope of this article. See Gu, 2013 for details.)

The minimization of (1) is performed in an RKHS $\mathcal{H} \subseteq \{f : J(f) < \infty\}$ in which $J(f)$ is a square seminorm. To incorporate (2) in estimating multivariate functions, we consider $f_j \in \mathcal{H}_{\langle j \rangle}$, where $\mathcal{H}_{\langle j \rangle}$ is a RKHS with tensor sum decomposition $\mathcal{H}_{\langle j \rangle} = \mathcal{H}_{0\langle j \rangle} \oplus \mathcal{H}_{1\langle j \rangle}$, where $\mathcal{H}_{0\langle j \rangle}$ is the finite-dimensional "parametric" subspace consisting of parametric functions, and $\mathcal{H}_{1\langle j \rangle}$ is the "nonparametric" subspace consisting of smooth functions. The induced tensor product space is

$$\mathcal{H} = \bigotimes_{j=1}^{d} \mathcal{H}_{\langle j \rangle} = \bigoplus_{\mathcal{S}} \left[\left(\bigotimes_{j \in \mathcal{S}} \mathcal{H}_{1\langle j \rangle} \right) \otimes \left(\bigotimes_{j \notin \mathcal{S}} \mathcal{H}_{0\langle j \rangle} \right) \right] = \bigoplus_{\mathcal{S}} \mathcal{H}_{\mathcal{S}}, \tag{3}$$

where the summation runs over all subsets $\mathcal{S} \subseteq \{1, \ldots, p\}$. The subspaces $\mathcal{H}_{\mathcal{S}}$ form two large subspaces, $\mathcal{N}_J = \{\eta : J(\mu) = 0\}$ and $\mathcal{H} \ominus \mathcal{N}_J$ with the reproducing kernel $R_J(\cdot, \cdot)$. We have completed specifying the tensor product RKHS for f.

We aim to find $\hat{\boldsymbol{\pi}}$ and $\hat{\mathscr{F}}$ minimizing $l(\boldsymbol{\pi}, \mathscr{F})$, with $\hat{\boldsymbol{\pi}} \in [0, 1]^2$, and $\hat{f}_0$, $\hat{f}_1$, $\hat{g}_0$, $\hat{g}_1$ are from the aforementioned tensor product RKHS. However, $\mathbf{X}$ in $l(\boldsymbol{\pi}, \mathscr{F})$ is not observed, so we cannot directly solve $\hat{\boldsymbol{\pi}}$ and $\hat{\mathscr{F}}$. Thus, we employ the EM algorithm (McLachlan and Krishnan, 2007) in the following Algorithm (1), which iteratively performs an expectation step (E-step) and a maximization step (M-step).

We introduce the EM algorithm for QuadS in the following, with the detailed derivations in the Appendix. We introduce two indicator variables $x_i^{\langle r,t \rangle} = I(X^{\langle r,t \rangle} = i)$ and $y_i^{\langle r,t \rangle} = I(Y^{\langle r,t \rangle} = i)$, $i = 0, 1$, where I is the indicator function. We denote

$$\begin{aligned}
\gamma_i^{\langle r,t \rangle} &= E\left[x_i^{\langle r,t \rangle} | \mathbf{Y}, \mathbf{Z}, \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}\right] = P\left(X^{\langle r,t \rangle} = i | \mathbf{Y}, \mathbf{Z}, \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}\right), \\
\xi_{ij}^{\langle r,t-1 \rangle} &= E\left[x_j^{\langle r,t \rangle} x_i^{\langle r,t-1 \rangle} | \mathbf{Y}, \mathbf{Z}, \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}\right] = P\left(X^{\langle r,t \rangle} = j, X^{\langle r,t-1 \rangle} = i | \mathbf{Y}, \mathbf{Z}, \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}\right).
\end{aligned}$$

In the E-step, we take the expectation of l with respect to $\mathbf{X}$. We obtain that expected penalized log-likelihood $Q(\boldsymbol{\pi}, \mathscr{F} | \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}) = E[l(\boldsymbol{\pi}, \mathscr{F}) | \mathbf{Y}, \mathbf{Z}, \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}]$ could be written as

$$\begin{aligned}
Q(\boldsymbol{\pi}, \mathscr{F} | \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}) = -\sum_{r=1}^{R} \Bigg\{ & \sum_{i=0}^{1} \gamma_i^{\langle r,1 \rangle} \log \pi_i + \sum_{t=2}^{T_r} \sum_{i=0}^{1} \sum_{j=0}^{1} \xi_{ij}^{\langle r,t-1 \rangle} \log a_{ij}^{\langle r,t-1 \rangle} \\
& + \sum_{t=1}^{T_r} \sum_{i=0}^{1} \sum_{j=0}^{1} \gamma_i^{\langle r,t \rangle} y_j^{\langle r,t \rangle} \log b_{ij}^{\langle r,t \rangle} \Bigg\} + \sum_{\substack{f \in \\ \{f_0, f_1, g_0, g_1\}}} \frac{1}{2} \lambda_f J(f).
\end{aligned} \tag{4}$$

To compute $\gamma_i^{\langle r,t \rangle}$ and $\xi_{ij}^{\langle r,t \rangle}$, we use a forward-backward procedure similar to the classical HMM. Details of this procedure can be found in the Appendix.

In the M-step, we update $(\hat{\boldsymbol{\pi}}, \hat{\mathscr{F}}) = \operatorname{argmin}_{\boldsymbol{\pi}, \mathscr{F}} Q$. By generalizing the backward-forward method in classic HMM, minimizing (1) is equivalent to the following five separate problems:

$$\operatorname*{argmin}_{\pi_i} -\sum_{r=1}^{R} \sum_{i=0}^{1} \gamma_i^{\langle r,1 \rangle} \log \pi_i, \quad \text{such that } \sum_{i=0}^{1} \pi_i = 1, \tag{5}$$

$$\operatorname*{argmin}_{f_0} \left[-\sum_{r=1}^{R} \sum_{t=2}^{T_r} \left\{ \xi_{11}^{\langle r,t-1 \rangle} \log \frac{1}{1 + e^{f_0(\mathbf{Z}^{\langle r,t-1 \rangle})}} + \xi_{12}^{\langle r,t-1 \rangle} \log \frac{e^{f_0(\mathbf{Z}^{\langle r,t-1 \rangle})}}{1 + e^{f_0(\mathbf{Z}^{\langle r,t-1 \rangle})}} \right\} + \frac{\lambda_{f_0}}{2} J(f_0) \right], \tag{6}$$

$$\underset{f_1}{\operatorname{argmin}}\left[-\sum_{r=1}^{R}\sum_{t=2}^{T_r}\left\{\xi_{21}^{\langle r,t-1\rangle}\log\frac{1}{1+e^{f_1(\mathbf{Z}^{\langle r,t-1\rangle})}}+\xi_{22}^{\langle r,t-1\rangle}\log\frac{e^{f_1(\mathbf{Z}^{\langle r,t-1\rangle})}}{1+e^{f_1(\mathbf{Z}^{\langle r,t-1\rangle})}}\right\}+\frac{\lambda_{f_1}}{2}J(f_1)\right], \tag{7}$$

$$\underset{g_0}{\operatorname{argmin}}\left[-\sum_{r=1}^{R}\sum_{t=2}^{T_r}\left\{\gamma_0^{\langle r,t\rangle}y_1^{\langle r,t\rangle}\log\frac{1}{1+e^{g_0(\mathbf{Z}^{\langle r,t-1\rangle})}}+\gamma_0^{\langle r,t\rangle}y_2^{\langle r,t\rangle}\log\frac{e^{g_0(\mathbf{Z}^{\langle r,t-1\rangle})}}{1+e^{g_0(\mathbf{Z}^{\langle r,t-1\rangle})}}\right\}+\frac{\lambda_{g_0}}{2}J(g_0)\right], \tag{8}$$

$$\underset{g_1}{\operatorname{argmin}}\left[-\sum_{r=1}^{R}\sum_{t=2}^{T_r}\left\{\gamma_1^{\langle r,t\rangle}y_1^{\langle r,t\rangle}\log\frac{1}{1+e^{g_1(\mathbf{Z}^{\langle r,t-1\rangle})}}+\gamma_1^{\langle r,t\rangle}y_2^{\langle r,t\rangle}\log\frac{e^{g_1(\mathbf{Z}^{\langle r,t-1\rangle})}}{1+e^{g_1(\mathbf{Z}^{\langle r,t-1\rangle})}}\right\}+\frac{\lambda_{g_1}}{2}J(g_1)\right]. \tag{9}$$

First, we use the method of Lagrange multipliers to solve (5). Denoting the Lagrange multiplier by λ, we have $\sum_{r=1}^{R}\sum_{i=0}^{1}\{\gamma_i^{\langle r,1\rangle}\log\pi_i-\lambda(\sum_{j=0}^{1}\pi_i-1)\}$. Setting the derivative of the Lagrangian equal to zero, we have

$$0=\frac{\partial}{\partial\pi_i}\left\{\sum_{r=1}^{R}\sum_{i=0}^{1}\gamma_i^{\langle r,1\rangle}\log\pi_i-\lambda\left(\sum_{j=0}^{1}\pi_i-1\right)\right\}=\sum_{r=1}^{R}\left\{\frac{\gamma_i^{\langle r,1\rangle}}{\pi_i}-\lambda\right\}.$$

Thus, we have $\pi_i=\frac{\sum_{r=1}^{R}\gamma_i^{\langle r,1\rangle}}{R\lambda}$. Since $\lambda=\lambda\sum_{j=0}^{1}\pi_j$, we have $\lambda=\frac{\sum_{j=0}^{1}\sum_{r=1}^{R}\gamma_j^{\langle r,1\rangle}}{R}$. Therefore, we can update $\hat{\pi}_i$ using the following equation

$$\hat{\pi}_i=\frac{\sum_{r=1}^{R}\gamma_i^{\langle r,1\rangle}}{\sum_{j=0}^{1}\sum_{r=1}^{R}\gamma_j^{\langle r,1\rangle}}. \tag{10}$$

Note that the expression from of (6)–(9) implies that they are four separated penalized weighted logistic regression problems. Taking (6) as an example, for the aforementioned tensor product RKHS (3), the subspaces $\mathcal{H}_{\mathcal{S}}$ form two large subspaces, $\mathcal{N}_J=\{f:J(f)=0\}$ which is the null space of $J(f)$, and $\mathcal{H}\ominus\mathcal{N}_J$ with the reproducing kernel $R_J(\cdot,\cdot)$. The solution to (6) has the following form

$$f_0(z)=\sum_{v=1}^{m}d_v\phi_v(z)+\sum_{i=1}^{T}c_iR_J(s_i,z), \tag{11}$$

where $\{\phi_v\}_{v=1}^{m}$ is a basis of $\mathcal{N}_J$, d_v and c_i are the coefficients, and $\mathbf{s}=(s_1,\ldots,s_T)$ is a distinct combination of all z_{ij} $(i=1,\ldots,n,\ j=1,\ldots,n_i)$. For (7)–(9), the solutions are in the same form as in (11).

The algorithm of QuadS is summarized in the Algorithm 1. The selection of smoothing parameters is an important issue in the QuadS algorithm, and we select them by general cross validation (GCV) method (Gu and Wahba, 1991). In particular, for our proposed EM-algorithm procedure for computing QuadS, multiple functions are estimated by the general smoothing splines (GSS) models in each M-step, and the smoothing parameter needs to be selected for all of them. For each function estimation in each M-step, we use the fully iterative GCV tuning to estimate the smoothing parameters, which is realized by the R package gss (Gu, 2014).

3 Simulation

3.1 Simulation 1

To assess the performance of the proposed method, we carried out simulation studies to compare QuadS with some existing methods. In this simulation study, we considered a two-dimensional

Algorithm 1: EM Algorithm for QuadS.

Input: $\mathbf{Y}$, $\mathbf{Z}$

Output: $\hat{\boldsymbol{\pi}}$, $\hat{\mathscr{F}} = (\hat{f}_0, \hat{f}_1, \hat{g}_0, \hat{g}_1)$

Initialization: set $\hat{\boldsymbol{\pi}} = \hat{\boldsymbol{\pi}}_0$, $\hat{\mathscr{F}} = \hat{\mathscr{F}}_0$ from some random initial.

while $|l^{\text{new}} - l^{\text{old}}| \geqslant 0.0001$ **do**

  E-step:

  - Conditional on $\hat{\boldsymbol{\pi}}$, $\hat{\mathscr{F}}$, update γ and ξ.
  - Calculate the expected penalized log-likelihood $Q(\boldsymbol{\pi}, \mathscr{F} | \hat{\boldsymbol{\pi}}, \hat{\mathscr{F}})$.

  M-step:

  - Conditional on γ, update $\hat{\boldsymbol{\pi}}$ by (5)
  - Conditional on γ and ξ, update $\mathscr{F}$ by (6)–(9).

end

variable $\mathbf{Z} = \{(z_1^{\langle r,t\rangle}, z_2^{\langle r,t\rangle})\}$, where $z_1^{\langle r,t\rangle} \sim \text{Unif}(-0.5, 1.5)$, $z_2^{\langle r,t\rangle} \sim \text{Unif}(2.5, 3)$, $\{z_1^{\langle r,t\rangle}\}$ and $\{z_2^{\langle r,t\rangle}\}$ are mutually independent. Let the initial probability $\boldsymbol{\pi} = (\pi_0, \pi_1) = (0.9, 0.1)$, and we consider two scenarios of the underlying functions:

$$
\begin{aligned}
g_0 &= \big(0.3\big(10^6 z_1^{11}(1 - z_1)^6 + 10^4 z_1^3 (1 - z_1)^{10}\big) - 2\big) z_2, \\
g_1 &= \big(-0.3\big(10^6 z_1^{11}(1 + z_1)^6 + 10^4 z_1^3 (1 + z_1)^{10}\big) - 2\big) z_2, \\
f_0 &= \big(0.3\big(10^6 (z_1 + 0.5)^{11}(0.5 - z_1)^6 + 10^4 (0.5 + z_1)^3 (0.5 - z_1)^{10}\big) - 2\big) z_2, \quad \text{and} \\
f_1 &= f_0.
\end{aligned}
$$

We set the number of loans R to be 100, 200, and 300. For all loans, We set the minimum number of time points $T_{\min} = 10$, and the maximum number of time points $T_{\max} = 50$. For the rth loan ID, we simulate the prepayment behavior with the following steps for $t = 1, \ldots, T_{\max}$.

1. Generate $X^{\langle r,t\rangle}$. If $t = 1$, sample $X^{\langle r,t\rangle}$ from $Ber(\pi_1)$. Otherwise sample $X^{\langle r,t\rangle}$ from $Ber(p^{\langle r,t\rangle})$, where $p^{\langle r,t\rangle} = f_0(\mathbf{Z}^{\langle r,t-1\rangle})$ if $X^{\langle r,t-1\rangle} = 0$, $p^{\langle r,t\rangle} = f_1(\mathbf{Z}^{\langle r,t-1\rangle})$ if $X^{\langle r,t-1\rangle} = 1$.
2. Generate $Y^{\langle r,t\rangle}$. Sample $Y^{\langle r,t\rangle}$ from $Ber(p^{\langle r,t\rangle})$, where $p^{\langle r,t\rangle} = g_0(\mathbf{Z}^{\langle r,t-1\rangle})$ if $X^{\langle r,t\rangle} = 0$, $p^{\langle r,t\rangle} = g_1(\mathbf{Z}^{\langle r,t-1\rangle})$ if $X^{\langle r,t\rangle} = 1$.
3. If $Y^{\langle r,t\rangle} = 1$ for some $t \leqslant T_{\max}$, we terminate the procedure, and label the rth loan ID as prepaid ID. Otherwise, we label this ID as unprepaid ID.

Further, we generate R loan ID, among which the proportion of not prepaid ID is r_0, and proportion of prepaid ID is r_1. By specifying r_0, r_1 at different values, we have the following three scenarios.

- Scenario 1 (low proportion of prepaid ID): set $r_0 = 0.8$, $r_1 = 0.2$.
- Scenario 2 (medium proportion of prepaid ID): set $r_0 = 0.5$, $r_1 = 0.5$.
- Scenario 3 (high proportion of prepaid ID): set $r_0 = 0.2$, $r_1 = 0.8$.

Figure 2 shows the number of time points of one replicate of the generated data with $R = 100$.

We randomly divide the generated R loans into a training set (50%) and a testing set (50%). The performance is quantified by the AUC of the testing data set. In comparison, we consider the proposed QuadS method, a linear simplified version of QuadS (Linear-QuadS), logistic regression (LR), general smoothing spline (GSS), and neural networks (NN) as competitors. We adopt a five-layer fully connected NN with five nodes in each layer. Note that Linear-QuadS has the same model structure as QuadS, but the functions f_0, f_1, g_0, g_1 are all taken as linear functions and estimated by a logistic regression within the M-step. The simulation is replicated 100 times.

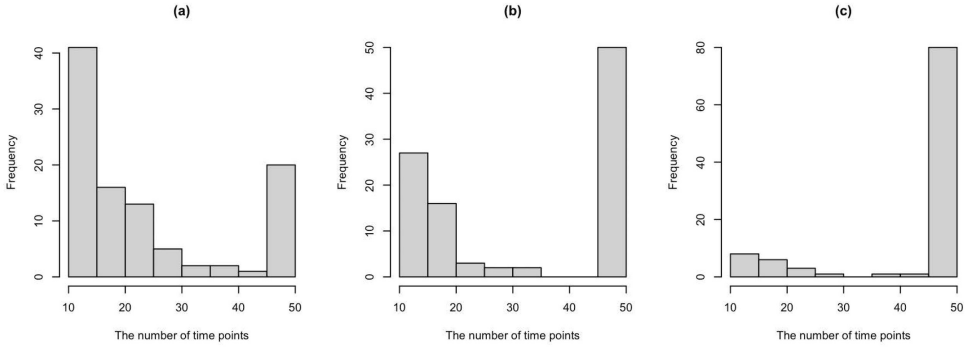

Figure 2: Training data of simulation 1. Three histograms show the number of time points of 3 scenarios: (a) Scenario 1: $r_0 = 0.2$, $r_1 = 0.8$. (b) Scenario 2: $r_0 = 0.5$, $r_1 = 0.5$. (c) Scenario 3: $r_0 = 0.8$, $r_1 = 0.2$.

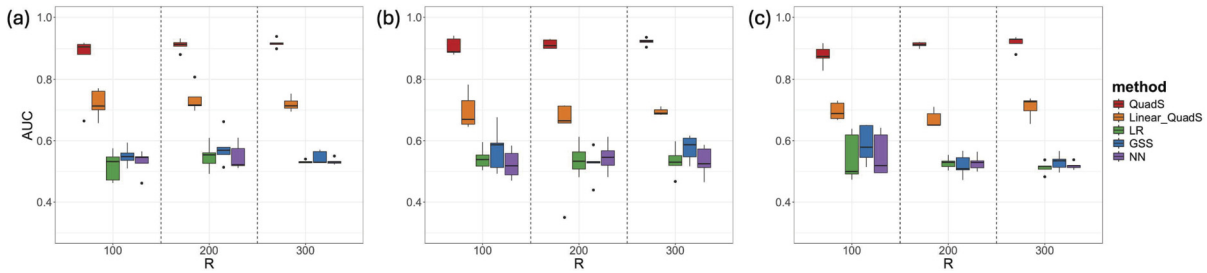

Figure 3: Results of simulation 1: testing AUC of QuadS, Linear-QuadS, logistic regression, general smoothing splines, and neural network with the number of loans $R = 100$, 200, or 300. (a) Scenario 1: $r_0 = 0.2$, $r_1 = 0.8$. (b) Scenario 2: $r_0 = 0.5$, $r_1 = 0.5$. (c) Scenario 3: $r_0 = 0.8$, $r_1 = 0.2$.

Figure 3 shows the average testing AUC results of scenarios 1-3, respectively. QuadS has the best performances in AUC for all 3 scenarios and all choices of R, and we also have the following observations. Firstly, our method was not sensitive to the proportion of prepaid loan IDs. Secondly, the AUC of QuadS increased as R increased, which makes sense as we had more training data. Overall, QuadS performed best and had the AUCs higher than 0.9 in all scenarios. This indicates that QuadS precisely captures the data-generating process, and thus, it is able to give highly accurate predictions. Linear-QuadS performs the second best with AUCs around 0.7. The comparison with Linear-QuadS supports that QuadS has more representation power by using non-parametric functions to model f_0, f_1, g_0, g_1. On the other hand, the three methods (LR, GSS, and NN) that do not consider the underlying states perform poorly with AUCs around 0.5, which is due to the model misspecification, which illustrates the necessity of the state space structure of QuadS in loan prepayment modeling.

3.2 Simulation 2

In this simulation, we evaluate the robustness of QuadS when the true data generating process is logistic regression. Similar to Simulation 1, we considered a two-dimensional variable $\mathbf{Z} = \{(z_1^{\langle r,t\rangle}, z_2^{\langle r,t\rangle})\}$, where $z_1^{\langle r,t\rangle} \sim \text{Unif}(0,1)$, $z_2^{\langle r,t\rangle} \sim \text{Unif}(0,1)$, $\{z_1^{\langle r,t\rangle}\}$ and $\{z_2^{\langle r,t\rangle}\}$ are mutually

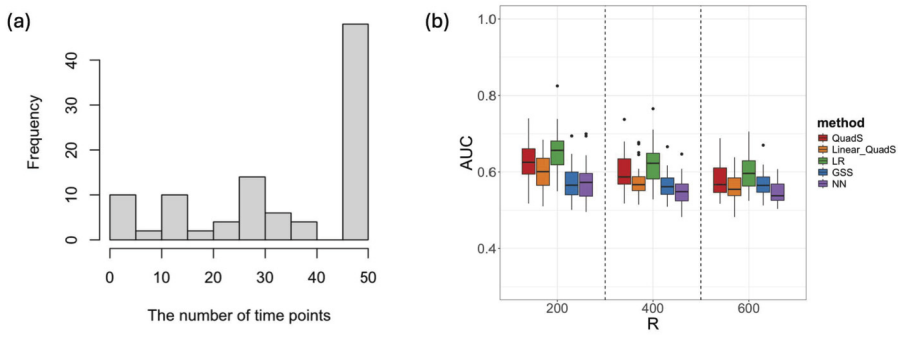

Figure 4: (a) Training data of simulation 2, where the histogram shows the number of time points of loans; (b) Results of simulation 2: testing AUC of QuadS, Linear-QuadS, logistic regression, general smoothing splines, and neural network.

independent. Let the initial probability $\boldsymbol{\pi} = (\pi_0, \pi_1) = (0.9, 0.1)$. Since there are no longer the hidden structures, we directly set a function $g = 0.1 + 0.1z_1 - 0.05z_2$. We set the number of loans R to be 200. For all loans, We set the minimum number of time points $T_{\min} = 10$, and the maximum number of time points $T_{\max} = 50$. For the rth loan ID, we simulate the prepayment behavior with the following steps for $t = 1, \ldots, T_{\max}$. Figure 4 (a) shows the number of time points of one replicate of the generated data with R = 100.

1. Generate $Y^{\langle r,t\rangle}$. Sample $Y^{\langle r,t\rangle}$ from $Ber(p^{\langle r,t\rangle})$, where $p^{\langle r,t\rangle} = g(\mathbf{Z}^{\langle r,t-1\rangle})$.
2. If $Y^{\langle r,t\rangle} = 1$ for some $t \leqslant T_{\max}$, we let it stop at time t, and label the rth loan ID as prepaid ID. Otherwise, we label this ID as an unprepaid ID.
3. We drop the loans with length smaller than $T_{\min}$.

We randomly divide the generated R loans into a training set (50%) and a testing set (50%). The performance is quantified by the AUC of the testing data set. We still consider Linear-QuadS, LR, GSS, and NN as competitors. The simulation is replicated 100 times.

The results are shown in Figure 4 (b). As expected, LR has the best performance since the data generating process is logistic regression. We observe that QuadS has the second-best performance by outperforming Linear-QuadS, GSS, and NN, which verifies the robustness of QuadS when the model is misspecified. We also noticed that all methods have test AUCs that are mostly below 0.7, which is because generated data mimics the loan data instead of being i.i.d. distributed. In particular, during the data generation procedure, the second step stops a loan when the loan is prepaid ($Y^{\langle r,t\rangle} = 1$), and the third step drops the loans that are too short. This kind of model misspecification poses a challenge for all compared methods.

4 Case Study

In this study, we analyze the single-family loan prepayment data from the Federal National Mortgage Association (FNMA). We obtain the single-family loan performance data that is available from Fannie Mae Data Dynamics.[1] This data set is about 40 gigabytes, which includes records of about 30 million loans. For each loan, we collected the month-by-month prepaid behaviors (y) from 2001 to the second quarter of 2019. Herein, y represents whether the loan has been prepaid,

[1] https://capitalmarkets.fanniemae.com/tools-applications/data-dynamics

i.e., $y = 1$ indicates prepaid, while $y = 0$ represents not prepaid. In addition, we collected the following features for each loan.

- Loan age (z_1) measures the age of the loan.
- Borrower's FICO score at origination (z_2) measures the quality of the borrower's credit. In general, it gives the likelihood an individual will timely pay his future obligations.
- Spread-at-origination, abbreviated as SATO (z_3), which is calculated as the difference between the borrower's original interest rate and the mortgage market rate at the origination date. Note that SATO is an additional index that indicates the credit quality of the borrower besides the FICO score. A high SATO value indicates low credit quality.
- Current Loan to value ratio, abbreviated as CLTV (z_4), is calculated by dividing the current balance by the property's current value.
- Refinance incentive (z_5) is calculated as the difference between the current mortgage market rate and the borrower's interest rate.

We preprocess the data using the following procedure: First, we deal with loans with missing data. Features like FICO score and OLTV are required in any mortgage contract. If they are missing, it must be due to a reporting error and not the borrower failing to provide this information. Those missing data points are a random subset of the data. Thus, in this paper, it is reasonable to assume our data is missing completely at random. We removed loans with missing data. Second, we removed loans that have less than five records. Third, to evaluate the model performance, we use a sliding window approach to split the data into the training and testing sets. As shown in Table 1, the time window for model training is set to be ten years, and the subsequent one year is employed for model testing. Table 1 shows the number of loans and proportion of prepaid loans in each training window and testing window.

The AUC is used to evaluate the prediction performance on the testing set. For model comparison, we employed three benchmark methods: logistic regression (LR), general smoothing spline (GSS), and neural networks (NN). We adopt a five-layer fully connected NN with five nodes in each layer, and the hyper-parameters are selected by cross-validation.

Table 2 demonstrates the results of different methods. The QuadS model consistently outperformed the other models on different training and testing years, with AUC scores ranging from 0.631 to 0.788. LR, GSS, and NN have a common drawback that they do not consider the

Table 1: Description of the training window and testing window. The training window has a length of ten years, and the testing window has a length of one year. P_a and P_b represent the proportion of prepaid loans in the training window and testing window, respectively.

Training	Testing	P_a	P_b
2001–2010	2011	0.689	0.074
2002–2011	2012	0.703	0.152
2003–2012	2013	0.700	0.077
2004–2013	2014	0.702	0.075
2005–2014	2015	0.712	0.072
2006–2015	2016	0.693	0.152
2007–2016	2017	0.709	0.107
2008–2017	2018	0.719	0.112
2009–2018	2019	0.709	0.304

Table 2: Testing AUC of QuadS, logistic regression, general smoothing spline, and neural network. Bold-faced numbers indicate the highest AUC in each row.

Training	Testing	QuadS	LR	GSS	NN
2001–2010	2011	**0.688**	0.660	0.681	0.651
2002–2011	2012	**0.721**	0.680	0.661	0.548
2003–2012	2013	**0.754**	0.678	0.679	0.637
2004–2013	2014	**0.788**	0.637	0.729	0.719
2005–2014	2015	**0.697**	0.506	0.597	0.552
2006–2015	2016	**0.631**	0.567	0.587	0.556
2007–2016	2017	**0.694**	0.626	0.666	0.591
2008–2017	2018	**0.684**	0.600	0.680	0.557
2009–2018	2019	**0.702**	0.612	0.646	0.528

time structure in the model and this is also one major improvement of the QuadS model. From the perspective of stability point, QuadS and GSS have consistently high accuracy, while NN has unstable results in different training and testing years. This makes QuadS more reliable and suitable for high-fidelity finance modeling and decision-making in practice. Lastly, although LR has also shown stability across the years and has been a classic approach widely applied in the industry, it does not have enough representation power to model the loan prepayment data. The excellent performances of QuadS showcase that using this sophisticated statistical model can produce results that are both accurate and reliable for loan prepayment modeling.

5 Conclusion

In this paper, we introduced the Smoothing Spline State Space Model as a novel approach to modeling mortgage loan prepayment behavior, addressing the inherent challenges of large-scale financial data and the complex patterns of borrower behavior. By incorporating hidden Markov models with time-varying transition and emission matrices modeled through smoothing splines, the QuadS framework effectively captures the latent states of borrower behavior, which traditional models often overlook. Our method demonstrated superior predictive performance across various scenarios, as shown in both simulation studies and a case study using the Federal National Mortgage Association data.

The case study on FNMA data highlighted the practical value of the QuadS model, showcasing its ability to handle extensive datasets while providing interpretable results. The model's capacity to uncover hidden behavior patterns in loan data not only enhances predictive accuracy but also offers critical insights for financial institutions. These insights can inform the development of risk management strategies and the design of mortgage products, ultimately contributing to a more stable and efficient mortgage market.

In conclusion, the QuadS model represents a significant contribution to financial risk modeling, offering a powerful tool for understanding and predicting mortgage prepayment behavior. Its combination of accuracy, scalability, and interpretability makes it highly applicable to the financial industry, where it can play a vital role in managing loan portfolios and anticipating market trends.

Supplementary Material

Some details of the EM algorithm for QuadS are provided in Appendix A. The code and instructions of the QuadS method are available on GitHub (https://github.com/haoranlustat/QuadS). The dataset used in the case study is publicly available from Fannie Mae Data Dynamics (https://capitalmarkets.fanniemae.com/tools-applications/data-dynamics).

Funding

This research was partially supported by supported by the U.S. National Science Foundation under grants NSF DMS-1925066, DMS-1903226, DMS-2124493, DMS-2311297, DMS-2319279, DMS-2318809, the U.S. National Institute of Health under grant R01GM152814.

References

Agarwal S, Chomsisengphet S, Kiefer H, Kiefer LC, Medina PC (2020). Inequality during the COVID-19 pandemic: The case of savings from mortgage refinancing. *Available at SSRN 3750133*.

Aldridge I, Avellaneda M (2019). Neural networks in finance: Design and performance. *The Journal of Financial Data Science*, 1(4): 39–62. https://doi.org/10.3905/jfds.2019.1.4.039

Bengio Y, Frasconi P (1995). An input output HMM architecture. In: *Advances in Neural Information Processing Systems*, volume 7, 427–434.

Bengio Y, Frasconi P (1996). Input-output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7(5): 1231–1249. https://doi.org/10.1109/72.536317

Berger DW, Milbradt K, Tourre F, Vavra J (2018). Mortgage prepayment and path-dependent effects of monetary policy. Technical report, *National Bureau of Economic Research*.

Fang L, Chen Y, Zhong W, Ma P (2024). Bayesian knowledge distillation: A bayesian perspective of distillation with uncertainty quantification. In: *Forty-first International Conference on Machine Learning*, 12935–12956.

Federal Housing Finance Agency (2024). Prepayment Monitoring Report: First Quarter 2024. Technical report, *Federal Housing Finance Agency*.

Freddie Mac (2024). Primary Mortgage Market Survey (PMMS).

Fuster A, Hizmo A, Lambie-Hanson L, Vickery J, Willen PS (2021). How resilient is mortgage credit supply? evidence from the COVID-19 pandemic. Technical Report, *National Bureau of Economic Research*.

Gu C (2013). *Smoothing Spline ANOVA Models*, volume 297. Springer.

Gu C (2014). Smoothing spline ANOVA models: R package gss. *Journal of Statistical Software*, 58: 1–25. https://doi.org/10.18637/jss.v058.i05

Gu C, Ma P (2005). Optimal smoothing in nonparametric mixed-effect models. *The Annals of Statistics*, 33(3): 1357–1379. https://doi.org/10.1214/009053605000000110

Gu C, Wahba G (1991). Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. *SIAM Journal on Scientific and Statistical Computing*, 12(2): 383–398. https://doi.org/10.1137/0912021

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018). A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5): 1–42.

Helwig NE, Ma P (2015). Fast and stable multiple smoothing parameter selection in smoothing spline analysis of variance models with large samples. *Journal of Computational and Graphical Statistics*, 24(3): 715–732. https://doi.org/10.1080/10618600.2014.926819

Johnson K, Pasquale F, Chapman J (2019). Artificial intelligence, machine learning, and bias in finance: Toward responsible innovation. *Fordham Law Review*, 88: 499.

Kung J-Y, Wu C-C, Hsu S-Y, Lee S, Yang C-W (2010). Application of logistic regression analysis of home mortgage loan prepayment and default risk. *ICIC Express Letters*, 4(2): 325–331.

Lai TL, Su Y, Sun KH (2014). Dynamic empirical bayes models and their applications to longitudinal data analysis and prediction. *Statistica Sinica*, 24(4): 1505–1528.

Ma P, Huang JZ, Zhang N (2015). Efficient computation of smoothing splines via adaptive basis sampling. *Biometrika*, 102(3): 631–645. https://doi.org/10.1093/biomet/asv009

Maxam CL, LaCour-Little M (2001). Applied nonparametric regression techniques: Estimating prepayments on fixed-rate mortgage-backed securities. *Journal of Real Estate Finance and Economics*, 23(2): 139–160. https://doi.org/10.1023/A:1011102332025

McLachlan GJ, Krishnan T (2007). *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons.

Meng C, Zhang X, Zhang J, Zhong W, Ma P (2020). More efficient approximation of smoothing splines via space-filling basis selection. *Biometrika*, 107(3): 723–735. https://doi.org/10.1093/biomet/asaa019

Ozbayoglu AM, Gudelek MU, Sezer OB (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, 93: 106384. https://doi.org/10.1016/j.asoc.2020.106384

Sirignano J, Sadhwani A, Giesecke K (2016). Deep learning for mortgage risk. arXiv preprint: https://arxiv.org/abs/1607.02470

Sun X, Zhong W, Ma P (2021). An asymptotic and empirical smoothing parameters selection method for smoothing spline ANOVA models in large samples. *Biometrika*, 108(1): 149–166. https://doi.org/10.1093/biomet/asaa047

Van Deventer DR, Imai K, Mesler M (2013). *Advanced Financial Risk Management: Tools and Techniques for Integrated Credit Risk and Interest Rate Risk Management*. John Wiley & Sons.