# MARKOV CHAINS, CLUSTERING, AND REINFORCEMENT LEARNING: APPLICATIONS IN CREDIT RISK ASSESSMENT AND SYSTEMIC RISK REDUCTION

RICHARD LE

A DISSERTATION SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN MATHEMATICS AND STATISTICS
YORK UNIVERSITY
TORONTO, ONTARIO

OCTOBER 2023

# Abstract

In this dissertation we demonstrate how credit risk assessment using credit rating transition matrices can be improved, as well as present a novel reinforcement learning (RL) model capable of determining a multi-layer financial network configuration with reduced levels of systemic risk. While in this dissertation we treat credit risk and systemic risk independently, credit risk and systemic risk are two sides of the same coin. Financial systems are highly interconnected by their very nature. When a member of this system experiences distress such as default, a credit risk event, this distress is often not felt in isolation. Due to the highly interconnected nature of financial systems, these shocks can spread throughout the system resulting in catastrophic failure, a systemic risk event.

The treatment of credit risk begins with the introduction of our first-order Markov model augmented with sequence-based clustering (SBC). Once we established this model, we explored its ability to predict future credit rating transitions, the transition direction of the credit ratings, and the default behaviour of firms using historical credit rating data. Once validated, we then extend this model using higher-order Markov chains. This time around, focusing more on the absorbing behaviour of Markov chains, and hence, the default behaviour under this new model. Using higher-order Markov chains, we also enjoy the benefit of capturing a phenomenon known as rating

momentum, characteristic of credit rating transition behaviour. Other than the credit rating data set, this model was also applied to a Web-usage mining data set, highlighting its generalizability.

Finally, we shift our focus to the treatment of systemic risk. While methods exist to determine optimal interbank lending configurations, they only treat single-layer networks. This is due to technical optimization challenges that arise when one considers additional layers and the interactions between them. These layers can represent lending products of different maturities. To consider the interaction between layers, we extend the DebtRank (DR) measure to track distress across layers. Next, we develop a constrained deep-deterministic policy gradient (DDPG) model capable of reorganizing the interbank lending network structure, such that the spread of distress is better mitigated.

# Acknowledgements

It is not an understatement when I say the existence of this dissertation would not be possible without the support of the people around me. First and foremost, I would like to give acknowledgement, my thanks, and express my deep gratitude for my supervisor, Professor Hyejin Ku. Your belief, patience, and words of encouragement allowed me to pursue the daunting task of exploring the unexplored with confidence. Your guidance and ability to consistently ask the right questions ensured I was always stepping in the right direction. Throughout my studies, you emphasized the importance of being critical about my work and taught me to always question whether the experimental outcomes were within expectations or reason. It has been inspiring working under your supervision, and I sincerely appreciate the time and energy you've dedicated towards my growth. I hope to carry these experiences and lessons throughout the rest of my life as a scientist and mathematician.

To my committee members, thank you for volunteering your time, energy, and your support. To Professor Hongmei Zhu, thank you for your kindness and selfless accommodations throughout my studies. Professor Shengyuan (Michael) Chen, your enthusiasm towards my projects very much served as a motivating factor and was encouraging. To my internal committee member, Professor Kee-Hong Bae, thank you for highlighting the importance of practical relevance. To my external committee

member, Professor David Saunders, thank you for your thorough review of my dissertation and your big picture questions.

A portion of the research conducted in this dissertation was made possible thanks to the credit rating data set provided by Professor Doobae Jun and his team. To Professor Sungjune Park, thank you for collecting and processing the Wikispeedia data set, as well as offering your time for meetings during our collaboration.

The math department is full of brilliant minds, and constantly served to motivate me. To Professor Alexey Kuznetsov, it was great having someone around to share our accomplishments in the gym with. I also appreciate your willingness to share stories of your beautiful family. What you managed to achieve in your life serves as an inspiration to me. Vishal Siewnarine, I appreciate you dropping by my office unannounced and for the laughs we shared. Thank you Neda Aminnejad, for your kindness and helping out with the MSGSA. Thank you Yohana Solomon, for taking over the MSGSA and doing an outstanding job bringing the department together. Laura Keanne, it was always a blast running into you in the gym, thank you for convincing me to take some time for myself. To Kelvin Chan, you are an amazing asset to the department and coordinating the math help centre would not have been possible without your help. Andrew Fleck, you are an amazing story teller and have never failed to make me laugh. Thank you Kaiyan Lin, for your constant support and being so accommodating with swapping our TA duties. To Morgan Wang, thank you for being a great office neighbour, your words of encouragement, and entertaining our discussions about our projects. I hope I was as helpful to you as you were to me. To our department allumnus, your incredible mastery over mathematics never failed to impress me. Allysa Lumley, you are a brilliant mathematician and greater human being. To Bransilav Nikolic, whether it's through personal growth, or new accomplishments, you constantly improve your life in some way, encouraging me to

constantly better myself. Thank you Kathleen Eng, for your words of encouragement and reminding me how close I was to finishing regardless of how close I was to finishing. To Nathan Gold, I am grateful to have met you. Your enthusiasm and the great number of things you accomplished during your study. You were a great inspiration to me from the beginning.

I would also like to dedicate a moment to my friends outside of York University. Vy Ngo, not only was your defense magnificent, but it also greatly influenced how I approached mine. What you managed to accomplish for yourself was truly inspiring. Shawn Lee, you were like an older brother to me; the countless laughs and company were very much needed throughout my studies. Paprika Chu, your work ethic is incredible, and you are wise beyond your years; you always had the perfect response to my life questions. All three of you were amazing role models to have around.

How often do you work 4 jobs within 2 years

Finally, to my family, throughout the ups and downs of my study, I could always count on you all for being present despite my frequent times of being absent. Thank you Raymond, Jessica, Thomas, Joseph, Victor, Philip, Angelina, and my countless other cousins who have been a source of support and company throughout my life. A special acknowledgement goes to the people I've spent all my life with, my brothers. To Charlie, thank you for bearing with my constant hovering whenever I got bored around the house. To Brian, you have one of the sharpest minds I know. Your knack for always asking "why" is endearing. To John, witnessing you tirelessly work on your art serves as reminder that hard work pays off. To my youngest brother, Jayden, I can tell you'll be the best of us. Finally, this dissertation would certainly not have been possible without the support of my mom, Mai Bach Thi Le. There's nothing I can say to express my deep gratitude for all the sacrifices you've had to endure for us. Truly, thank you for your love and support, not only for me but also for my brothers.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

We regularly assess and manage the risks that follows us as we navigate our daily lives. From the more mundane activities such as driving through the streets of Toronto, to academic endeavors such as embarking on a research project, one is constantly weighing the value of realizing your goals against the uncertainty of failure. Understanding the likelihood of adverse events and their consequences is not only essential to our daily lives but also plays an important role in management and decision making in the foundational systems of society. In this dissertation, we will explore how machine learning techniques can improve the assessment of risk in financial systems.

In the context of finance there are many different types of risk one might be concerned with. There's interest rate risk (the risk of a change in interest rates leading to the change in bond prices, affecting bond holders), exchange-rate risk (the risk that the price of a currency may change with respect to an asset you currently hold), and model risk (the risk of loss due to errors and inaccuracies in models used) to name a few. The types of risk discussed in this dissertation will be credit risk and

systemic risk.

Credit risk measures the potential for a debtor (the borrower) failing to repay their loan, leading to the creditor (the lender) failing to receive their owed money amounts. The realization of a debtor failing to meet their debt obligation is also known as default. A popular measure of credit risk is credit ratings. Credit ratings can be used assess the credit risk for corporations, government bodies, or financial institutions and their ability to satisfy their loan obligations. These credit ratings are assigned by credit rating agencies such as Standard & Poor's, Moody's, and Fitch. The ratings are represented by letter grades ranging from AAA (representing the best credit quality) to D (representing default, the worst credit quality). For investors and portfolio managers, credit ratings are crucial as they directly influence bond prices, interest rates offered by the issuers, and credit portfolios (Trueck & Rachev, 2005). The significance of credit ratings extends to banks as a downgrade in credit ratings can lead to an increase in excess cash holdings (Khieu & Pyles, 2012). For professionals in risk management and regulatory compliance, the role of credit ratings in assessing credit risk is not missed. This significance is especially highlighted with the introduction of the Basel II accords and subsequently, the Basel III accords. This regulatory framework allows for the use of internal and external credit ratings of banks in calculating risk-weighted assets which directly affects the calculation of their capital requirements for credit risk (Bank for International Settlements, 2023; Trueck & Rachev, 2009).

To measure the likelihood of an upgrade or downgrade, credit rating transition matrices (also referred to as migration matrices) are used. These matrices explicitly express the probability of transitioning from one credit rating to another. Typically, transition matrices are modelled under a Markov assumption where the transition behaviour is dependent only on the current credit rating. Empirically, credit rating

transitions display various non-Markovian behaviour, for example, downward rating momentum (the likelihood of downgrades are increased if the previous transition was a downgrade), rating persistence (the tendency for firms to main its current rating), and time non-homogeneity (Altman & Kao, 1992; Baena-Mirabete & Puig, 2018; D'Amico et al., 2019; Frydman & Schuermann, 2008; Lando & Skodeberg, 2002). By not considering these characteristics of credit ratings, this could lead to an assessment that does not accurately reflect the true level of credit risk present. Indeed, Güttler and Raupach (2008) highlight the tendency of more naive models to underestimate the Value-at-Risk (VaR) of credit portfolios by 8% of the correct value while momentum sensitive models estimate higher VaR in comparison. Therefore, banks that do not consider rating momentum may hold insufficient capital to buffer against unexpected losses. Baena-Mirabete and Puig (2018) also found similar observations when incorporating downward momentum and rating persistence. In this dissertation, we improve on traditional first-order transition matrices by capturing characteristics such as rating momentum by using high-order Markov models.

The second type of risk we explore in this dissertation is systemic risk. In the context of financial systems, systemic risk is the the risk of financial collapse due to the failure of some portion of the financial network leading to economic decline. While the realization of a systemic risk event can directly impact the general economy, elevated levels of systemic risk can also affect various parts of the financial system before any catastrophe is realized. Using the treasury/eurodollar spread (TED) as a measure of systemic risk, Bianchi et al. (2010) investigates the relationship between systemic risk and hedge fund returns. They note that rising levels of systemic risk can result in behavioural changes of managers resulting in a reduction of equity and stock momentum exposures and for short sellers, they increase their short exposure. Another consequence of increasing levels of systemic risk is the weakening of the

beneficial effects of diversification Busse et al. (2014). Strobl (2016) supports this idea by finding a large correlation between systemic risk and idiosyncratic risk. Using marginal expected shortfall and residual volatility of returns as proxies for measures of systemic risk and idiosyncratic risk respectively, their results suggest that regulations reducing systemic risk will reduce the overall riskiness of financial institutions.

The most popular example of a systemic risk event is the global financial crisis of 2008. Not only did the financial crisis affect the general economy, but it also heavily influenced subsequent regulatory policy making. Shortly after the crisis, the Basel III accord was introduced to address the inefficiencies of the pre-crisis regulatory framework. These accords introduced various changes, such as placing greater emphasis on loss-absorbing capital in the form of Common Equity Tier 1 capital, the increase in the level of capital requirements that needed to be met, and revising deficient risk-capturing frameworks, among others (Bank for International Settlements, 2018). While minimum capital requirements certainly play a role in mitigating some aspects of systemic risk, the topology of the network itself also contributes to systemic risk (Allen & Gale, 2000; Boss et al., 2004; Gai & Kapadia, 2010; Nier et al., 2007). The importance of considering the network structure has recently been emphasized with works by authors like Poledna et al. (2017), who has demonstrated that policies focusing on the interbank network structure may prove to be more effective than capital requirements. Furthermore, by not considering different layers of the financial system, the level of systemic risk can be heavily underestimated (Poledna et al., 2015). In this dissertation, we contribute to the literature by developing a novel reinforcement learning (RL) method to reorganize multi-layer financial networks, producing network configurations with reduced systemic risk. Moreover, by extending the popular DebtRank (DR) measure of systemic risk developed by Battiston et al. (2012), we introduce a novel DebtRank-like measure to account for how distress might

propagate through different layers of a financial network.

While the treatment of credit risk and systemic risk in this dissertation is separate, it is important to note the strong interdependence between these two types of risk. A notable relationship can be established by considering sovereign credit ratings and the sovereign ceiling effect. The sovereign ceiling effect describes the tendency for the credit rating of banks to not exceed their sovereign's credit rating. A study by Sehgal et al. (2018) finds evidence suggesting that increased levels of systemic risk can lower sovereign credit ratings. Consequently, in the presence of sovereign downgrades, Huang and Shen (2015) suggests that the rating of banks far from the ceiling are still affected due to the deterioration of government assets they may hold. Conversely, the rating of banks near the ceiling are mainly determined by the sovereign rating regardless of the asset type they hold. Individually, the insolvency of a few participants of the financial system can negatively impact the rest of the financial system. While at the same time, the organization of the system can exacerbate the individual fall outs. In other words, better microprudential policies work to provide safer financial institutions and thus a safer financial system as a whole but whenever such spillovers are sufficiently strong, the financial system as a whole may be worse off, even though the behaviour of banks individually may be perfectly rational (Freixas et al., 2015). Therefore, addressing both credit risk and systemic risk in this dissertation is essential in improving the overall function of a financial system.

Throughout this dissertation, we will be making use of different elements from various disciplines, including finance, mathematics, and, most notably, machine learning. By leveraging techniques from the field of machine learning, we improve credit rating transition matrix estimation methods and hence, credit risk assessment. Moreover, we develop a RL framework capable of reducing systemic risk in a multi-layer financial system. Therefore, ideas of this dissertation can be separated into

treatments of two types of risk, credit risk, and systemic risk. The issues of credit migration are addressed in Chapters 2 and 3. The reduction of systemic risk is addressed in 4.

Finally, we conclude this dissertation in Chapter 5. In this chapter, we discuss the implications and significance of the work achieved in this dissertation. We believe that the augmented Markov chain models, enhanced using SBC, offer practitioners additional approaches to consider the various non-Markovian properties of credit rating sequences while providing superior predictive performance compared to traditional transition matrices. While the combination of our RL framework and extended multi-layer DR measure offers a novel and flexible method to determine the configuration with lower levels of systemic risk in a multi-layer financial system. This approach offers additional insight into the structure of low systemic risk systems. Furthermore, the RL framework introduced in this dissertation is general enough to be applied in other contexts where the goal is network reorganization, opening up new paths for research in other fields.

# Chapter 2

# First-Order Markov Chains and Sequence-Based Clustering

In this chapter, we study the effectiveness of estimating credit rating transition matrices by augmenting first-order Markov chains using sequence-based clustering (SBC) to better characterize companies using solely historical credit rating sequences. By better utilizing historical credit rating sequences, we can improve the quality of estimated credit rating transition matrices, a critical component in credit risk assessment models. We group firms together based on their sequence matrices and hence, with similar transition behaviour together. The research presented in this chapter has been published in Le et al. (2021).

## 2.1 Introduction

Credit ratings and their revisions can lead to a number of major decisions and hence, consequences. It is in one's best interest to invest in better forecasting techniques to

mitigate any credit rating dependent losses. In this chapter, we will be adapting the general clustering methodology described in Park et al., 2008 and apply a transition matrix estimation method to predict future credit behaviours solely from historical credit ratings. Park et al. (2008) developed a sequence representation scheme based on Markov models, enabling sequences of web usage activities to be clustered using vector based distances. This method is known as SBC. As far as we can tell, we are the first to study the application of SBC using K-means strictly on historical credit rating sequences. The majority of models observed in literature use a snapshot of a company's financial statement and fewer models use a historical sequence of financial statements (Chen et al., 2013).

Markov chains are commonly used in modelling the behaviour of credit rating transitions over time. Jarrow et al. (1997) were one of the first to model the term structure of credit risk spreads using Markov chains in both the discrete and continuous time case. They estimated the transition probability matrix from historical data by first estimating the generator matrix from a 1-year estimate of transition probabilities provided from a credit rating agency. The generator matrix can be estimated either implicitly from bond market prices or from historical bond transition rating changes. Thomas et al. (2002) extends the Jarrow–Turnbull model by introducing a hidden Markov model for the term structure of credit risk spreads. Kiefer and Larson (2004) tested the effectiveness of using a time-homogeneous Markov model to describe the credit rating transitions of municipal bonds, commercial paper, and sovereign debt. They have found that the time-homogeneous Markov model can adequately describe credit rating transitions of municipal bonds over a period of 5 years and commercial paper over a period of 6 months. Credit rating transitions for sovereign debt are also adequately described by Markov models but this conclusion may be the result of the low number of data samples. Dharmaraja et al. (2017) introduces a hybrid Markov

model where they incorporate the asset value of the firm in the transition probabilities of credit ratings. Sharma et al. (2018) investigates the financial performance of insurance companies by using credit rating transition matrices under a Markov model, noting that less risky rating grades result in more rating stability.

Studies have shown the promising results that clustering can produce in the context of credit risk and credit rating predictions. In a study by Guo et al. (2012), they compared their proposed support vector domain description (SVDD) combined with fuzzy clustering model with other kinds of support vector machine learning techniques in the context of corporate credit rating classification. The performance of each model was evaluated based on the hit-ratio, the ratio of the number of correct classifications and the overall number of classifications. The variables used as the input of the model are bond-rating data sets from the Korean and Chinese markets. The variables range from shareholder's equity to cash flow from operating activities. Chen et al. (2013) use a trajectory clustering procedure consisting of two consecutive self-organizing maps (SOM) processes. Their method allows for the visualizations of the bankruptcy trajectories of companies enabling a unique perspective and insight on bankruptcy influences. Their model clusters financial statements containing 29 financial ratios of companies spanning 3 years. Morales et al. (2015) applied different fuzzy classification methods for the use in rating classifications. They use both credit ratings and financial statement ratios in their model. Irmatova (2016) introduces a relative attribute rating model (RELARM) based on relative PCA attributes and K-means clustering. Using 9 financial and economic parameters, their model assigns ratings based off the ranked projections of the cluster centres onto a rating vector. In the case of long-term credit rating prediction, the true rating that a firm receives in the future will not be known until that future date arrives. During this period, new credit rating information may become available. Kuncheva and Sánchez (2008)

terms this type of problem as delayed labelling and investigates the effectiveness of online nearest neighbour classifiers for treating delayed labelling problems. Plasse and Adams (2016) developed an online linear discriminant analysis algorithm which was applied to a real world consumer credit data set where delayed label information was introduced synthetically. Montiel et al. (2017) proposed two over-indebtedness risk prediction frameworks, one of which treats over-indebtedness as a streaming learning problem. Although not done in this study, we may be able to extend our model to consider delayed labelling by treating credit rating sequences as a streaming data problem.

The remainder of this chapter is organized into 5 sections. In Section 2.2 we discuss the methodology and theory behind our proposed model. In Section 2.3 we provide an overview of the three different classification scenarios that our model will undertake. In Section 2.4 we introduce the data to be used, and describe the specific methods of the experiments. In Section 2.5 we present the results and discussions of our proposed model. Finally we conclude the study with a discussion of the results in Section 2.6.

## 2.2 Methodology

In this section, we describe the methods for the first-order Markov model augmented with SBC. We begin by defining the sequence matrices and their properties. The sequence matrices will be the main objects that are being clustered. We then describe credit rating transition matrices as these will be used in the classification algorithm. Finally, we go over the details of the K-means clustering method.

### 2.2.1 Sequence Matrices

We introduce sequence matrices in order to measure the distance between the historical patterns of credit ratings of firms. Consider an $n$-state time-homogeneous Markov chain where each state represents a particular credit rating. In order to begin clustering these objects, we utilize the representation of sequence vectors and sequence matrices introduced in Park et al. (2008).

**Definition 2.2.1.** Let $m \in \mathbb{N}$, $X_1^m, X_2^m, ..., X_T^m$ be a sequence of random variables, and $\mathcal{S}$ be the state space. Then the $m$th sequence vector of length $T_m \in \mathbb{N}$ is defined by the vector $\mathbf{x}_m(T_m) = (X_1^m, X_2^m, ..., X_{T_m}^m)$ with states $X_t^m \in \mathcal{S}$.

**Definition 2.2.2.** Let $N_{ij}$ be the number of transitions from state $i$ to $j$ for the $m$th firm with the sequence vector $\mathbf{x}_m(T_m)$. The corresponding sequence matrix $\mathbf{S}_m$ is then an $n \times n$ matrix whose entries are denoted by

$$S_m[i,j] = \begin{cases} \frac{N_{ij}}{\sum_{j'} N_{ij'}} & \text{if } N_{ij} > 0, \\ 0 & \text{if } N_{ij} = 0, \end{cases} \tag{2.1}$$

and so, the entries represent the relative frequency of transitions from state $i$ to $j$.

Therefore, given a sequence vector $\mathbf{x}_m(T_m)$ we can generate the corresponding sequence matrix $\mathbf{S}_m$. This sequence matrix describes the frequency of transitions of the given sequence vector.

In the context of credit ratings, there tends to be few credit rating transitions away from the current rating over the period of $T_m$ leading to sparse credit rating sequence matrices. An extreme example of this observation would be one where a firm takes only one rating for the entire period of $T_m$. Suppose $X_t = 2$ for $t \leq T_m$,

| Notation | Definition |
|---|---|
| $\mathcal{S}$ | State space of the Markov chain |
| $n$ | Number of states in the state space |
| $X_t^m$ | State of the $m$th sequence at time $t$ |
| $T_m$ | Full length of the $m$th sequence |
| $\mathbf{x}_m(t)$ | Sequence vector up to time $t$, defined by $(X_0^m, ..., X_t^m)$ |
| $C$ | Total number of clusters used in the K-means algorithm |
| $c$ | The $c$th cluster among $C$ clusters. |
| $\mathbf{S}_m$ | Sequence matrix of the $m$th sequence |
| $\Delta t$ | Transition period defined by $t_{l+1} - t_l$ |
| $\mathbf{P}_c(\Delta t)$ | Representative $\Delta t$-year first-order transition matrix of the $c$th cluster |
| $\mathbf{Q}_c$ | Sub-matrix of the $\mathbf{P}_c$ containing the transition probabilities from and to transient states |
| $\mathbf{R}_c$ | Vector of $\mathbf{P}_c$ containing the transition probabilities from a transient state to an absorbing state |
| $\tau$ | Number of time steps, indicating how far into the future we want to predict |
| $\mathbf{r}_c^\tau$ | Vector whose entries are the probability of default within $\tau$ time steps |
| $\hat{d}_m(t, t')$ | The transition direction for a transition from $X_t^m$ to $X_{t'}^m$ |
| $\mathcal{L}$ | Set of labels used for classification |
| $\theta$ | Threshold used to compare against the absorption probabilities |
| $\hat{y}_m(\theta)$ | Label or prediction made for the $m$th sequence given for some threshold $\theta$ |
| $K$ | Number of folds used in $K$-fold cross validation |
| $S_{\text{size}}$ | Sample size, the number of sequences used in each experiment |
| $d_{xy}$ | Somers' D |

Table 2.1: A table to reference for some notation used throughout this chapter.

then the resulting sequence matrix would contain a single entry at $S[2,2] = 1$ and $S[i,j] = 0$ everywhere else.

Now we present some general properties of sequence matrices when using the Euclidean distance measure for the comparison of different sequence matrices. The Euclidean distance measure is used to measure the distance between the historical patterns of credit ratings for the firms. We consider a time-homogeneous Markov chain $X$ with state space $\mathcal{S} = \{1, 2, ..., n\}$. For a sequence matrix, the sum of the entries in a nonzero row is 1, i.e., $S_m[i,j] = 0$ for all $1 \leq j \leq n$ or,

$$\sum_{j=1}^{n} S_m[i,j] = 1.$$

**Lemma 2.2.1.** Consider a vector $(a_1, a_2, ..., a_n)$ that satisfies $\sum_{i=1}^{n} a_i = 1$.

(a) The minimum value of $\sum_{i=1}^{n} a_i^2$ is $\frac{1}{n}$ and it is achieved at $(\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n})$.

(b) If $0 \leq a_i \leq 1$ for all $i$, then the maximum value of $\sum_{i=1}^{n} a_i^2$ is 1, that is,

$$\frac{1}{n} \leq \sum_{i=1}^{n} a_i^2 \leq 1.$$

*Proof.* By the Cauchy-Schwarz inequality,

$$\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} 1^2\right) \geq \left(\sum_{i=1}^{n} a_i\right)^2$$

Since $\sum_{i=1}^{n} a_i = 1$,

$$\sum_{i=1}^{n} a_i^2 \geq \frac{1}{n}$$

where the equality holds when $a_i = \frac{1}{n}$, $i = 1, ..., n$. For the maximum value, we

consider

$$\sum_{i=1}^{n} a_i^2 = \left(\sum_{i=1}^{n} a_i\right)^2 - \sum_{i \neq j}^{n} a_i a_j.$$

Since $0 \leq a_i, a_j \leq 1$,

$$\sum_{i=1}^{n} a_i^2 \leq \left(\sum_{i=1}^{n} a_i\right)^2 = 1.$$

□

Next we consider two sequence matrices $\mathbf{S}_{m_1}$ and $\mathbf{S}_{m_2}$ that represent Markov chains $X^{m_1}$ and $X^{m_2}$ where each state refers to credit ratings of firm $m_1$ and firm $m_2$. Let $\mathcal{S}_1$ be the set of states where Markov chain $X^{m_1}$ has ever visited for $t < T$, i.e.,

$$\begin{aligned}\mathcal{S}_1 &= \{i : S_{m_1}[i,j] > 0 \text{ for some } j\} \\ &= \{i : N_{ij} > 0 \text{ for some } j\} \subset \mathcal{S}\end{aligned}$$

Similarly, we let $\mathcal{S}_2 = \{i : S_{m_2}[i,j] > 0 \text{ for some } j\}$. The corresponding sequence matrices $\mathbf{S}_{m_1}$ and $\mathbf{S}_{m_2}$ for Markov chains $X^{m_1}$ and $X^{m_2}$ have the following property.

**Theorem 2.2.1.** *Suppose that there is no intersection between $\mathcal{S}_1$ and $\mathcal{S}_2$, i.e., $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, implying two Markov chains $X^{m_1}$ and $X^{m_2}$ have not visited the same state. Then the Euclidean distance $||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}||$ between $\mathbf{S}_{m_1}$ and $\mathbf{S}_{m_2}$ is*

$$\sqrt{2} \leq ||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}|| \leq \sqrt{n}$$

*Proof.* Let $k = |\mathcal{S}_1| < n$. Without loss of generality, we may assume that

$$\begin{aligned}\mathcal{S}_1 &= \{1, 2, ..., k\} \\ &= \{i : S_{m_1}[i,j] > 0 \text{ for some } j\}\end{aligned}$$

Then[1] the sequence matrix $\mathbf{S}_{m_1}$ is of the form

$$\begin{pmatrix} \mathbf{S'}_{m_1} & \mathrm{O} \\ \mathrm{O} & \mathrm{O} \end{pmatrix}$$

where $\mathbf{S'}_{m_1}$ is a $k \times k$ subsection of $\mathbf{S}_{m_1}$ and O is the zero matrix. We have the Euclidean square distance between $\mathbf{S'}_{m_1}$ and the zero matrix is

$$||\mathbf{S'}_{m_1}||^2 \geq \frac{1}{k} + \frac{1}{k} + \cdots + \frac{1}{k} = \frac{k}{k} = 1$$

by using Lemma 2.2.1. Applying the same argument, the Euclidean square distance between $\mathbf{S}_{m_2}$ and the zero matrix is greater than or equal to 1. Since $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, we get

$$||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}||^2 \geq 1 + 1 = 2$$

therefore, $||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}|| \geq \sqrt{2}$. On the other hand, by Lemma 2.2.1,

$$\sum_{j=1}^{k} S'_{m_1}[i,j]^2 \leq 1$$

for each $1 \leq i \leq k$ where $S'_{m_1}[i,j]$ are entries of $\mathbf{S'}_{m_1}$. Thus

$$||\mathbf{S}_{m_1}||^2 = ||\mathbf{S'}_{m_1}||^2 \leq 1 + 1 + \cdots + 1 = k.$$

Since $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$, we have $|\mathcal{S}_2| \leq n - k$ and $||\mathbf{S}_{m_2}||^2 \leq n - k$. Then

$$||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}||^2 = ||\mathbf{S}_{m_1}||^2 + ||\mathbf{S}_{m_2}||^2 \leq k + (n - k) = n.$$

[1]This might not be the case when $X_T^{m_1} = j$ and $X_t^{m_1} \neq j$ for all $t < T$. We exclude this sequence here. In this special case, we have a slightly different bound depending on $c$.

Therefore, $||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}|| \leq \sqrt{n}$. □

**Theorem 2.2.2.** *Suppose that two Markov chains $X^{m_1}$ and $X^{m_2}$ have not made the same transition, i.e.,*

$$\{[i,j] : S_{m_1}[i,j] > 0\} \cap \{[i,j] : S_{m_2}[i,j] > 0\} = \emptyset.$$

*Then the Euclidean distance $||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}||$ between $\mathbf{S}_{m_1}$ and $\mathbf{S}_{m_2}$ is*

$$\sqrt{2} \leq ||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}|| \leq \sqrt{2n}.$$

*Proof.* The argument is essentially the same as in Theorem 2.2.1. For the upper bound,

$$\sum_{j=1}^{n} S_{m_1}[i,j]^2 \leq 1$$

for each $1 \leq i \leq n$, so $||\mathbf{S}_{m_1}||^2 \leq n$. Since $S_{m_1}[i,j] \times S_{m_2}[i,j] = 0$ for $1 \leq i, j \leq n$, we get

$$||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}||^2 = ||\mathbf{S}_{m_1}||^2 + ||\mathbf{S}_{m_2}||^2 \leq 2n.$$

Thus, $||\mathbf{S}_{m_1} - \mathbf{S}_{m_2}|| \leq \sqrt{2n}$. □

**Definition 2.2.3.** For a sequence vector $\mathbf{x}_m(T_m) = (X_1^m, X_2^m, ..., X_{T_m}^m)$ of length $T_m$ for some firm $m$, it is said to be *ascending* if $X_{t+1}^m \leq X_t^m$ for all $t < T_m$. The sequence vector $\mathbf{x}_m(T_m)$ is said to be *descending* if $X_{t+1}^m \geq X_t^m$ for all $t < T_m$.

Note that there are at most two nonzero entries in each nonzero row for sequence matrices corresponding to ascending or descending sequence vectors. An ascending vector indicates the credit ratings of a firm have been upgraded while a descending vector implies the credit ratings have been downgraded. Credit rating sequences with

ascending or descending sequences are examples of sequences that exhibit rating drift behaviour. As noted in D'Amico et al. (2019), rating drift is more pronounced in downgrades rather than upgrades.

### 2.2.2 Markov Chains

Credit rating sequences can be modelled as Markov processes. We consider a discrete $n$-state time-homogeneous Markov chain. A transition probability is the conditional probability of a stochastic process transitioning to one state given its current state, that is,

$$Pr\{X_t = j | X_{t-1} = i\}, \tag{2.2}$$

where $X_t \in \mathcal{S}$ is the credit rating at time $t \in \mathbb{N}$ with state space $\mathcal{S} = \{1, 2, ..., n\}$. A Markov chain must satisfy the Markov property. The Markov property is stated as the following

$$Pr\{X_t = j \mid X_0 = i_t, ..., X_{t-1} = i\} = Pr\{X_t = j \mid X_{t-1} = i\}, \tag{2.3}$$

$$= p_{ij}. \tag{2.4}$$

### 2.2.3 Transition Matrices

The conditional probabilities $p_{ij}$ is called the one-step transition probability and can be arranged in a matrix $\mathbf{P}$ called a transition matrix with entries $P[i, j] = p_{ij}$. With

$n = |\mathcal{S}|$, the transition matrix satisfies the following properties

$$p_{ij} \geq 0 \quad \forall i, j \leq n, \tag{2.5}$$

$$\sum_{j=1}^{n} p_{ij} = 1 \quad \forall i \leq n. \tag{2.6}$$

The Markov property implies that the transition probabilities only depend on its current state. To calculate the probability of transitions $\tau$ step into the future we use Theorem 2.2.3 (see for instance, Taylor and Karlin (1998)).

**Theorem 2.2.3.** *Let $p_{ij}^{(\tau)} = Pr\{X_{t+\tau} = j \mid X_t = i\}$ and $p_{ij}$ represent the entry of a transition matrix $\mathbf{P}$. Then, the $\tau$-step transition probability $p_{i,j}^{(\tau)}$ of transitioning from state $i$ to $j$ satisfies*

$$p_{ij}^{(\tau)} = \sum_{k=0}^{\infty} p_{ik} p_{kj}^{(\tau-1)}, \tag{2.7}$$

*where we define*

$$p_{ij}^{(0)} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Note that Equation (2.7) represents matrix multiplication and using transition matrices $\mathbf{P}$ we have the equivalent representation $\mathbf{P}^{(\tau)} = \mathbf{P} \times \mathbf{P}^{(\tau-1)}$. Given the assumption of time-homogeneity we can then write more generally

$$\mathbf{P}^{(\tau)} = \mathbf{P} \times \mathbf{P} \times ... \times \mathbf{P} = \mathbf{P}^{\tau}. \tag{2.8}$$

Therefore, using Equation (2.8), one can obtain the transition probabilities for any $\tau$-step transition.

Given the clusters of credit rating sequences formed from the K-means algorithm one can generate transition matrices based on the members of the respective clusters.

The industry standard for estimating transition matrices from credit rating sequences is the cohort approach (Christensen et al., 2004; Gunnvald, 2014).

**Definition 2.2.4. (Cohort Approach)** Let $n = |\mathcal{S}|$, $\{\mathbf{x}_m(T_m) \mid m \leq M_c \in \mathbb{N}, c = 1, 2, ..., C\}$ be the set of credit rating sequences in cluster $c$ with a total of $M_c$ members, $\mathcal{T} = \{t_l \mid 0 \leq l \leq T \text{ with } t_l < t_{l+1}\}$ be the set of equally spaced observed time points of the credit ratings $\mathbf{x}_m(T_m)$ used in constructing the representative transition matrix. Then we define transition period $\Delta t$ by $t_{l+1} - t_l$. Then the representative $\Delta t$-year transition matrix of the $c$th cluster, $\mathbf{P}_c(\Delta t)$ is an $n \times n$ matrix whose entries are denoted by

$$P_c[i, j] = \frac{\sum\limits_{t_l \in \mathcal{T}} N_{ij}(t_l)}{\sum\limits_{t_l \in \mathcal{T}} N_i(t_l)}, \tag{2.9}$$

where $N_{ij}(t_l)$ is the number of companies that had transitioned from state $i$ to $j$ in the $\Delta t$ period, $N_i(t_l)$ is the total number of companies whose current state was $i$ at time $t_l$.

To generate longer period $\Delta t$-year transition matrices one can redefine $\Delta t$ by $t_{l+2} - t_l$ instead. Consequently, a drawback of the cohort approach is the possibility of completely missing the existence of an intermediate credit rating in time if we choose to sample points when $\Delta t$ is large. For example, suppose we have a credit rating sequence $X = (1, 5, 1, 1, 1, 1, 1)$ with times $t_0, t_1, t_2, t_3, t_4, t_5, t_6$ corresponding to years 2000, 2001, 2002, 2003, 2004, 2005, 2006. In our example, we will generate our transition matrices by sampling time points $t_0, t_2, t_4$, and $t_6$. The observed credit rating sequence used in the estimation of the transition matrix is then $X_{\text{obs}} = (1, 1, 1, 1)$. Therefore, the transition to and from credit rating 5 will be completely missed using the estimated 2-year transition matrix.

An alternative approach that captures the intermediate transitions is the duration

approach. The duration approach first estimates the transition matrix by taking the matrix exponential of an estimated generator matrix (Gunnvald, 2014; Lando & Skodeberg, 2002). The difference between the cohort and duration approach has been intensively studied by Jafry and Schuermann (2004). They have noted that the cohort approach overestimates default probabilities (the last column of the transition matrix) for less risky rating categories and underestimates default probabilities for the most risky rating categories. By generating a bond portfolio of 400 exposures, they have also concluded that ignoring the efficiency gain in the duration approach is more damaging.

### 2.2.4 Clustering Algorithm: K-means

In our model, we will be making use of a variant of the K-means algorithm to cluster our data set. The purpose of clustering is to partition the firms into groups that share similar transition behaviours in their respective credit rating sequences. The base K-means algorithm is a popular choice in many applications due to its ease of implementation, simplicity, efficiency, and empirical success (Jain, 2010). Given the assumption that credit rating sequences can be modelled by an $n$-state time-homogeneous Markov chain, it may be natural to immediately consider clustering transition matrices of the individual firms. Unfortunately, using Euclidean distance to compare transition matrices leads to the mis-clustering of firms who do not experience any transitions in their credit rating sequence but belong to different credit ratings at the same time. For example, let $\mathbf{x}_1 = (2, 2, 2, 2, 2)$ and $\mathbf{x}_2 = (5, 5, 5, 5, 5)$ then the respective transition matrices $\mathbf{P}_1$, and $\mathbf{P}_2$ are

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

As the above matrices are transition matrices they must satisfy the propoerties (2.5) and (2.6) and hence, the resulting matrix is the identity matrix. Calculating the Euclidean distance we have $\|\mathbf{P}_1 - \mathbf{P}_2\| = 0$ despite being two firms with completely different credit rating sequences. If instead we generated the sequences matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ for $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively

$$\mathbf{S}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

using the Euclidean distance between $\mathbf{S}_1$ and $\mathbf{S}_2$ results in $\|\mathbf{S}_1 - \mathbf{S}_2\| = \sqrt{2}$. Hence by using sequence matrices, we circumvent this problem as two firms having experienced no transitions over some period $T$ will be considered "far" from each other in Euclidean norm. Therefore, by using the Euclidean distance measure on sequence matrices instead of transition matrices, the differences between the resulting

clusters will have a more intuitive interpretation.

In the conventional K-means clustering algorithm the initial cluster centroids are chosen completely randomly. Because the resulting clusters are highly dependent on the initial cluster centroids, the initialization of the cluster centroids is an important question to answer. To improve the initialization of the clustering process we choose to instead use PCA-guided K-means (Xu et al., 2015). The idea of PCA-guided K-means is that the optimal solution to the minimization problem lies in space known as the PCA-subspace, a smaller space than the original space. To implement this algorithm, we first cluster our data in the PCA-subspace and then initialize our cluster centroids in the feature space based on the cluster membership in the PCA-subspace. Although the resulting solution is not guaranteed to be the global optimal solution, the resulting solution tends to be better (in terms of within cluster variance) than the solutions obtained by just searching within the full data space. Therefore, using the PCA-guided K-means algorithm we intend to partition $S_{\text{size}}$ firms into $C$ clusters such that firms in each cluster share similar credit rating transition behaviour. A diagram of the first-order SBC model can be found in Figure 2.1.

## 2.3 Long-Term Credit Rating Model

We will be testing the performance of the model against three different classification scenarios:

1. The prediction of future credit rating.
2. The prediction of the direction of future credit rating transitions.
3. The classification of risky firms most likely to default.

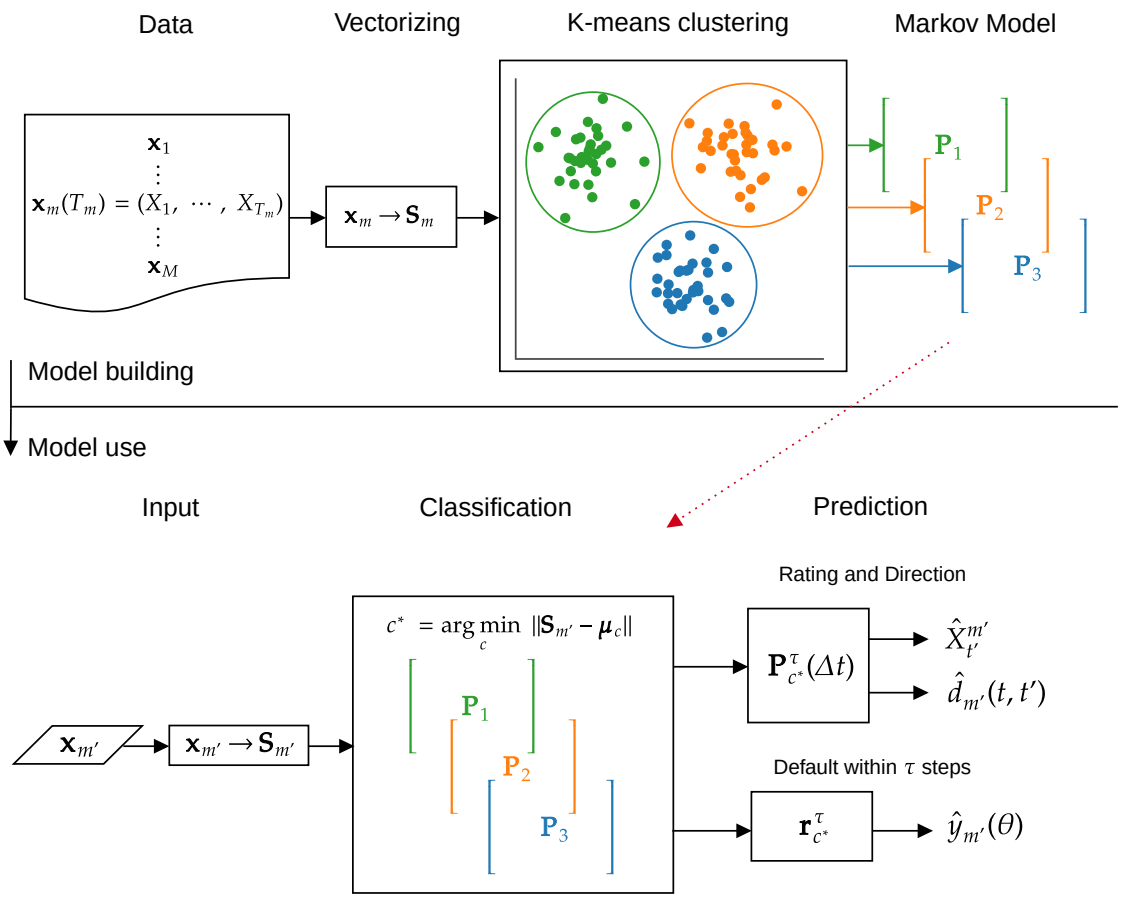



Figure 2.1: A diagram detailing the first-order Markov model combined with SBC.

For each classification scenario we first split the entire data set into a training set and a test set. The model is trained using the training set and is evaluated based on its classification performance using the test set. García et al. (2015) highlights the importance of experimental design in credit scoring and bankruptcy prediction. They note that the choice of data splitting method is dependent on the nature of the classifiers and complexity of the problem. In our study, we found that the $K$-fold cross-validation method suited our goals well. We forgo the use of the single holdout

method as this results in our model producing a single set of representative transition matrices. For similar reasons, we forgo the use of leave-one-out cross-validation as the set of representative transition matrices may remain relatively unchanged by removing a single sequence matrix from the training set. By using $K$-fold cross-validation we test the effectiveness of our model in the case of a variety of different clusters and the predictive power of their representative transition matrices. After clustering the training set we generate the representative transition matrix $\mathbf{P}_c(\Delta t)$ for each cluster. This is done using the cohort approach.

For each scenario we let $\mathbf{x}_m(T_m)$ be the credit rating sequence of the $m$th firm from the test set. For each firm, we generate the sequence matrix $\mathbf{S}_m$ based on $\mathbf{x}_m(T_m)$. The sequence matrices of the training set are then partitioned into $C$ different clusters. Using the testing set, we assign a single firm to one of the $C$ clusters based on the Euclidean distance between $\mathbf{S}_m$ and the clusters' centroid $\boldsymbol{\mu}_c$. That is, the assigned cluster $c^*$ is chosen by

$$c^* = \arg\min_c \|\mathbf{S}_m - \boldsymbol{\mu}_c\| . \tag{2.10}$$

After assigning the firm to a cluster, we can estimate the future behaviour of the firm by using the cluster's representative transition matrix.

### 2.3.1 Credit Rating and Transition Direction Prediction

In the credit rating prediction scenario, we intend to determine the most likely credit rating a firm will take at time $t'$ in the future given the current credit rating $X_t$ at time $t$. We will consider $n$ class labels for an $n$-state homogeneous Markov chain, i.e. $\mathcal{S} = \{1, 2, ..., n\}$. Let $\tau$ be the difference $t' - t$, then, using the representative transition matrix $\mathbf{P}_c(\Delta t)$, we calculate the $\tau$-step transition matrix $\mathbf{P}_c^\tau(\Delta t)$. Because we assume time-homogeneity, the $\tau$-step transition matrix can be calculated by using

Equation (2.8), that is

$$\mathbf{P}_c^{\tau}(\Delta t) = \mathbf{P}_c(\Delta t)^{(t'-t)}. \tag{2.11}$$

The prediction of the future credit rating $\hat{X}_{t'}^m$ for some firm $m$ is then

$$\hat{X}_{t'}^m = \arg\max_j P_c^{\tau}[X_t^m, j]. \tag{2.12}$$

For the evaluation of multi-class classification performance we will be considering the number of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). In the binary case, we can organize these counts using a confusion matrix

| | | Predicted | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | TP | FN |
| | Negative | FP | TN |

where our classes are the "Positive" and "Negative" classes. Given the confusion matrix it can then be observed that the count of TP represent the number of correct prediction of the positive class, FP represents the number of incorrect predictions of the positive class, FN represents the number of incorrect predictions of the negative class, and TN represents the number of correct prediction for the negative class. For the multi-class classification scenario with classes A, B, C, and D we can generate the following confusion matrix:

| | | Predicted classes | | | |
|---|---|---|---|---|---|
| | | A | B | C | D |
| Actual classes | A | $c_{1,1}$ | $c_{1,2}$ | $c_{1,3}$ | $c_{1,4}$ |
| | B | $c_{2,1}$ | $c_{2,2}$ | $c_{2,3}$ | $c_{2,4}$ |
| | C | $c_{3,1}$ | $c_{3,2}$ | $c_{3,3}$ | $c_{3,4}$ |
| | D | $c_{4,1}$ | $c_{4,2}$ | $c_{4,3}$ | $c_{4,4}$ |

Then, the number of TP, FP, FN, and TN can be calculated each individual class in a similar manner to the binary example by treating one of our classes as the positive class and everything else as the negative. Generally, let $\mathcal{L} = \{L_l \mid 1 \leq l \leq L\}$ be the set of class labels, then to calculate the number of TP, FP, FN, and TN for class label $L_l \in \mathcal{L}$ we must consider the confusion matrix for classes "$L_l$" and "Non-$L_l$"

| | | Predicted | |
|---|---|---|---|
| | | $L_l$ | Non-$L_l$ |
| Actual | $L_l$ | $\text{TP}_l$ | $\text{FN}_l$ |
| | Non-$L_l$ | $\text{FP}_l$ | $\text{TN}_l$ |

where each cell of the above confusion matrix can be calculated by the following

equations

$$\mathrm{TP}_l = c_{l,l} \tag{2.13}$$

$$\mathrm{FP}_l = \sum_{i=1}^{L} c_{i,l}, \text{ for } i \neq l \tag{2.14}$$

$$\mathrm{FN}_l = \sum_{j=1}^{L} c_{l,j}, \text{ for } j \neq l \tag{2.15}$$

$$\mathrm{TN}_l = \sum_{i=1}^{L}\sum_{j=1}^{L} c_{i,j} - (\mathrm{TP}_l + \mathrm{FP}_l + \mathrm{FN}_l). \tag{2.16}$$

To evaluate the wellness of the estimates made in the multi-class classification scenario we will be using the average accuracy, denoted by AA, and the micro-averaged F1 score, denoted by $\mathrm{F1}_\mu$ (Sokolova & Lapalme, 2009).

$$\mathrm{AA} = \frac{\sum_{l=1}^{L} \frac{\mathrm{TP}_l + \mathrm{TN}_l}{\mathrm{TP}_l + \mathrm{FN}_l + \mathrm{FP}_l + \mathrm{TN}_l}}{L} \tag{2.17}$$

$$\mathrm{F1}_\mu = 2\frac{\mathrm{Pr}_\mu \cdot \mathrm{Re}_\mu}{\mathrm{Pr}_\mu + \mathrm{Re}_\mu} \tag{2.18}$$

where

$$\mathrm{Pr}_\mu = \frac{\sum_{l=1}^{L} \mathrm{TP}_l}{\sum_{l=1}^{L} (\mathrm{TP}_l + \mathrm{FP}_l)} \tag{2.19}$$

$$\mathrm{Re}_\mu = \frac{\sum_{l=1}^{L} \mathrm{TP}_l}{\sum_{l=1}^{L} (\mathrm{TP}_l + \mathrm{FN}_l)} \tag{2.20}$$

where $\text{Pr}_\mu$ and $\text{Re}_\mu$ are the micro-averaged precision and recall respectively. It should also be noted that when using micro-averaging for multi-class classification the micro-averaged recall, micro-average precision, and micro-average F1 score are equal to each other. Micro-averaging is used instead of macro-averaging (that is, averaging the precision, recall, and F1 across the $L$ classes respectively) because macro-averaging weights each class's precision, recall, and F1 score equally across the classes while micro-averaging takes into consideration the size of each of the classes for the respective measure. This prevents the smaller classes from over contributing in the averaging of the F1 score (Sokolova & Lapalme, 2009).

When using $K$-fold cross-validation, a total confusion matrix is calculated by summing up the $K$ confusion matrices that were generated at each fold. This total confusion matrix is then used to calculate $\text{TP}_l, \text{FP}_l, \text{FN}_l$ and $\text{TN}_l$ as this is the most unbiased method in computing the F1 score when there is a high class imbalance (Forman & Scholz, 2010).

In the transition direction prediction scenario, we intend to determine which direction a firm's credit rating will move in by time $t'$ in the future, given the current credit rating $X_t^m$ at time $t$. We define this set of class labels as $\mathcal{L} = \{-1, 0, 1\}$ where -1, 0, and 1 represent the downgrade, stay, and upgrade classes, respectively. Using the representative transition matrix $\mathbf{P}_c(\Delta t)$, we calculate the $\tau$-step transition matrix $\mathbf{P}_c^\tau(\Delta t)$. The prediction of the direction that firm $m$'s credit rating will change at time $t'$ is then estimated by $\hat{d}_m(t, t')$ where

$$\hat{d}_m(t,t') = \begin{cases} 1 & \text{if} \quad Pu = \max(Pu, Ps, Pd) \\ 0 & \text{if} \quad Ps = \max(Pu, Ps, Pd) \\ -1 & \text{if} \quad Pd = \max(Pu, Ps, Pd) \end{cases} \tag{2.21}$$

where

$$Pu = \sum_{j<X_t^m} P_c^\tau[X_t^m, j]$$
$$Ps = P_c^\tau[X_t^m, X_t^m]$$
$$Pd = \sum_{X_t^m<j} P_c^\tau[X_t^m, j].$$

The predicted change in the direction of the credit rating $\hat{d}_m(t, t')$ is then compared to the true change in direction $d_m(t, t')$, calculated by

$$d_m(t, t') = \begin{cases} 1 & \text{if} \quad X_{t'}^m < X_t^m \\ 0 & \text{if} \quad X_{t'}^m = X_t^m \\ -1 & \text{if} \quad X_{t'}^m > X_t^m. \end{cases} \tag{2.22}$$

To evaluate the wellness of the estimates made using the test set, we again calculate a total confusion matrix from the $K$-fold cross-validation and calculate the average accuracy using Equation (2.17) and micro-average F1 score using Equation (2.18).

### 2.3.2 Prediction of Default Behaviour

In the default behaviour prediction scenario, we intend to determine whether a firm will be in default within $\tau$ quarters based on their current credit rating. We do so by checking the probability of default of a firm against an appropriate threshold enabling the classification of the firm's default behavior. That is, whether the firm will be in default or not within $\tau$ quarters. The class labels that we will consider are binary. We define the class label set as $\mathcal{L} = \{1, 0\}$ where 1 represents a firm having defaulted

within $\tau$ quarters and 0 for a firm not defaulting within $\tau$ quarters.

Using the representative transition matrix $\mathbf{P}_c(\Delta t)$ of the $c$th cluster we calculate the probability of defaulting within the next $\tau$ quarters. To calculate the probability of default within $\tau$ time steps we first define the following

**Definition 2.3.1.** Given state space $\mathcal{S}$ and $n = |\mathcal{S}|$, let $\mathbf{P}_c(\Delta t)$ be the $n \times n$ representative $\Delta t$-year transition matrix of cluster $c$. Then $\mathbf{Q}_c$ is the $(n-1) \times (n-1)$ subsection of $\mathbf{P}_c(\Delta t)$ containing entries $P_c[i, j]$ for $1 \leq i, j < n$ and $\mathbf{R}_c$ is a vector of size $n-1$ containing entries $P_c[i, j]$ for $1 \leq i < n$ and $j = n$.

Given the subsections $\mathbf{Q}_c$ and $\mathbf{R}_c$ as defined above, we then denote $\mathbf{r}_c^\tau$ as the $(n-1) \times 1$ vector whose entries are the probability of default within $\tau$ quarters for a firm assigned to cluster $c$ and is calculated by

$$\mathbf{r}_c^\tau = (\mathbf{I} + \mathbf{Q}_c + \mathbf{Q}_c^2 + ... + \mathbf{Q}_c^{(\tau-1)})\mathbf{R_c} \tag{2.23}$$

where $\mathbf{I}$ is the identity matrix. The entries of $\mathbf{r}_c^\tau$ are denoted by $r_c^\tau(i)$ for $1 \leq i < n$. Given a firm $m$ that was assigned to cluster $c$, the probability of default within $\tau$ quarters based on the firm's current credit rating $X_t^m$ is then $r_c^\tau(X_t^m)$. Once a firm has been assigned a probability of default, we will refer to the associated probability as a "risk score". At every step of the $K$-fold cross-validation process we assign all the firms in each of the fold's respective test set a risk score. By the $K$th fold of the cross-validation process, all of the firms in the data set will be assigned a risk score.

To measure the quality of the classification of the firms' default state, we will assess our model based on two different evaluation measures. The first evaluation is done using the measure Somers' Delta (Somers' D). The measure Somers' D is an asymmetric measure of association between an independent ($x$) and dependent

variable ($y$) (Somers, 1962; Trueck & Rachev, 2009). Somers' D measures the association between the independent and dependent variables by considering the number of concordant pairs, the number of discordant pairs, and the number of tied pairs on the dependent variable.

**Definition 2.3.2. (Somers' D)** Let $C$ be the number of concordant pairs, $D$ be the number of discordant pairs, and $Y_0$ be the number of tied pairs on the dependent variable. A pair $(x_i, y_i)$ and $(x_j, y_j)$ is concordant when both $x_i > x_j$ and $y_i > y_j$. A pair is discordant when $x_i > x_j$ and $y_i < y_j$. A pair is tied on the dependent variable when $y_i = y_j$. Somers' D is then calculated as

$$d_{yx} = \frac{C - D}{C + D + Y_0} \tag{2.24}$$

so the value of $d_{yx}$ ranges from $-1$ to 1.

The operational interpretation of Somers' D is the measure of the proportionate excess of concordant over discordant pairs among the number of pairs not tied on the independent variable. Somers' D can be applied in two types of applications (Newson, 2006). To measure the effect of the independent variable on the dependent variable, treating $d_{yx}$ as a measure of "effect size", or, to measure the performance of the independent variable as a predictor of the dependent variable, treating $d_{yx}$ as "predictor performance indicator". Given the context of credit risk we let the estimated risk score be the independent variable and the true default status (whether a firm has indeed defaulted within the next $\tau$ time steps) as the dependent variable.

The second evaluation is done by setting a threshold value $\theta$ and then classifying a firm as being in default or not based on whether their risk score exceeds the chosen threshold value. This comparison and classification is done using threshold values

from a discretized interval ranging from 0 and 1. Hence, for each $\theta$ chosen, the $m$th firm can be assigned a predicted default behaviour $y_m$ based on $r_c^\tau(X_t^m)$

$$\hat{y}_m(\theta) = \begin{cases} 1 & \text{if} \quad r_c^\tau(X_t^m) > \theta, \\ 0 & \text{if} \quad r_c^\tau(X_t^m) \leq \theta. \end{cases} \tag{2.25}$$

A convenient method for presenting the performance of a classification model is through the use of a receiver operating characteristic (ROC) curve. A ROC curve is a plot of the true positive rate, TPR (also known as the recall) against the false positive rate, FPR. To construct this curve we select a threshold $\theta$, estimate $\hat{y}_m$ using Equation (2.25), generate a confusion matrix, and then calculate the FPR and TPR by

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.26}$$

and

$$\text{TPR} = \text{Re} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.27}$$

This process is done for all thresholds from 0 to 1. At the same time, the precision and F1 score value can be calculated from the confusion matrix at every threshold.

In practice, it is more useful to choose an optimal threshold or "cut-off" point for binary classification. By choosing a threshold we set the rate for Type I and Type II errors. In the context of credit risk, a Type I error can result in opportunity costs and lost potential profits from lost interest income, while a Type II error can result in the lost interest and principal through defaults (Trueck & Rachev, 2009). Y. Liu (2002) calculates the optimal threshold by taking the line tangent to the ROC curve. This tangent line has a slope that is proportional to the ratio of "good" and "bad" cases, and inversely proportional to the cost ratio of the Type I and Type II errors.

In general it is difficult to use costs to evaluate models as different institutions have different cost and pay-off structures and so, it would be challenging to present a single cost function and provide a general framework for optimal decision making of a financial institution (Trueck & Rachev, 2009). Instead, we will be using the methods described in Sanchez (2016) to determine the optimal threshold in the worst-case scenario for the purpose of model evaluation. By using game theory and treating the classifier and "nature" as players, we choose the optimal threshold at the point where the ROC curve and the descending diagonal line (i.e. the line $\text{TPR} = 1 - \text{FPR}$) intersect.

## 2.4 Data and Experimental Methods

In this section we describe the data and methods used in this study. First, we present the data, its characteristics, and how the data was processed before classification. Next, we define the model parameters and present the algorithm used to evaluate the three classification scenarios described in Section 2.3.

### 2.4.1 Data

The data set we will be using was collected and provided by National Information & Credit Evaluation Inc., a major bond-rating company in Korea. The data set consisted of monthly corporate credit ratings from 1986-09-01 to 2018-09-01 for 1899 firms in Korean indices such as the KOSDAQ and KOSPI. Firms in this data set can

take any rating from the following set of 22 credit ratings

$$\{\text{AAA, AA+, AA, AA}-\text{, A+, A, A}-\text{, BBB+, BBB, BBB}-,$$
$$\text{BB+, BB, BB}-\text{, B+, B, B}-\text{, CCC+, CCC, CCC}-\text{, CC, C, D}\}.$$

The firms that take the "D" rating are considered to be in default. Some firms were "closed" after some time and are considered to be in default. Firms that were missing credit rating sequences, made for sale, or were merged with another firm were removed from the data set. After pruning the data set, there are 1648 firms remaining in the data set. The distribution of the remaining 1648 firms' credit rating classes for selected dates can be found in Figure 2.2.

From Figure 2.2, it can be observed that there are very few samples in credit class CCC+, CCC, CCC−, CC, and C for the selected dates. We mitigate the negative effects of this imbalanced data set by combining similar categories together reducing the number of credit rating classes from 22 to 7 classes. Doing so will minimize the number of classes that contain low instances of that minority class. The particular mapping of old classes to new classes can be found in Table 2.2. The distribution

| New Ratings | Old Ratings |
|---|---|
| AAA | {AAA} |
| AA | {AA+, AA, AA−} |
| A | {A+, A, A−} |
| BBB | {BBB+, BBB, BBB−} |
| BB | {BB+, BB, BB−} |
| B | {B+, B, B−, CCC+, CCC, CCC−} |
| C | {CC, C, D} |

Table 2.2: The 7 aggregated classes.

of the new aggregated classes can be found in Figure 2.3. By reducing the number

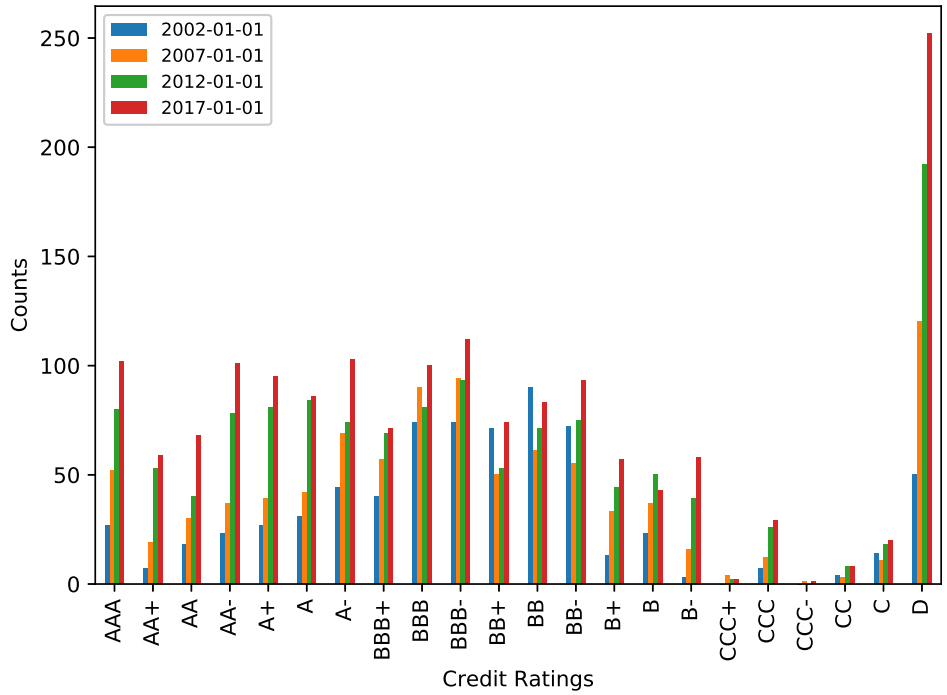



Figure 2.2: The frequency distribution of the 22 class credit ratings for the years 2002, 2007, 2012, 2017

of classes to 7 we diminish the degree of imbalance that was present in the data set. Something to note is that the distribution of credit ratings appears to change dramatically from year to year. This is in part due to the fact that a number of firms were not rated or did not exist at that time. For example, only 712 firms were rated on 2002-01-01 where as 1617 firms were rated on 2017-01-01. For the credit rating prediction and transition direction classification scenarios, we will be using the relabelled credit rating sequences as outlined in Table 2.2. For the default prediction classification scenario we will move the old ratings CC and C to the new B rating group leaving the last class to represent default by containing exclusively D ratings.

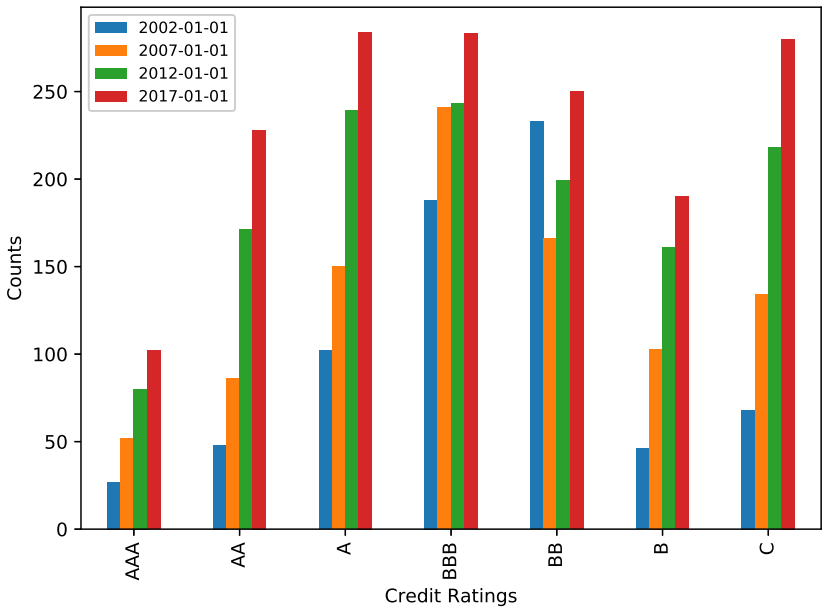



Figure 2.3: The frequency distribution of the 7 class credit ratings for the years 2002, 2007, 2012, 2017

### 2.4.2 Experimental Method

We treat the credit rating sequences as a Markov process with a state space $\mathcal{S} = \{1, 2, 3, 4, 5, 6, 7\}$ with the numbers 1 representing the least risky credit class AAA and 7 representing the most risky credit class C. The total number of clusters $C$ was set to 15.

For each classification scenario, we set the input date $t$ where $t \in \{2000, 2001, 2002, 2003\}$. The input date represents the initial point in time we will begin making our prediction from. For quarterly transition matrices we set $\Delta t$ to be 0.25. The predictions will be made $\tau$ quarters into the future where $\tau \in \{20, 40, 60\}$. Credit rating sequences with fewer than 5 years of credit rating data will not be used and excluded from the analysis. Therefore, of the remaining 1648 firms, the number of valid firms was reduced to the amounts indicated in Table 2.3.

| Input Date | Number of Valid Firms |
|---|---|
| 2000-01-01 | 542 |
| 2001-01-01 | 590 |
| 2002-01-01 | 712 |
| 2003-01-01 | 752 |

Table 2.3: The number of valid firms used in the 5-fold cross-validation based on the input date.

Each firm is assigned a sequence matrix $\mathbf{S}_m$ based on its credit rating sequence $\mathbf{x}_m(T_m)$ as described in section 2.2.1. Using 5-fold cross-validation we split the number of valid firms into two groups, a training set, and test set. 15 clusters are then generated based off the training set using PCA-guided K-means. A representative $\Delta t$-transition matrix $\mathbf{P}_c(\Delta t)$ was estimated for each cluster where $\Delta t = 0.25$. The transition matrices are generated as described in Section 2.2.3. Given $\mathbf{P}_c(\Delta t)$ we can then calculate $\mathbf{P}_c^{\tau}(\Delta t)$ where $\tau \in \{20, 40, 60\}$ using Equation (2.11).

Each firm from the test set is then assigned to a cluster and assigned a predicted class from a class label set based on the classification scenario as shown in Table 2.4. This process is done for all of the test sets at each fold. The result is that each valid firm is assigned to a predicted class by the end of the 5-fold cross-validation process.

| Scenario | Class Label Set ($\mathcal{L}$) |
|---|---|
| Credit Rating Prediction | {1,2,3,4,5,6,7} |
| Transition Direction | {-1, 0, 1} |
| Default | {1, 0} |

Table 2.4: The class labels for each classification scenario.

The effectiveness of our clustering model against the benchmark model will be based on the performance measures described in Section 2.3. The benchmark model uses a single representative transition matrix $\mathbf{P}(\Delta t)$ estimated from all of the credit rating sequences, in the absence of clustering. This single transition matrix is then used for classification purposes. The results in section 2.5 were calculated by averaging 1000 shuffled 5-fold cross-validation results.

The algorithm used in the different classification scenarios can be grouped into 2 main algorithms found in A.1. The credit rating prediction and transition direction prediction classification scenario can be found in Algorithm 1. The default prediction scenario algorithm can be found in Algorithm 2

## 2.5 Results and Discussion

The brackets beside the performance measures in the following tables are the standard deviation of the respective measures. The low standard deviation for the benchmark model is the result of the diagonally dominant matrices produced by the benchmark

model. From Table 2.5, the clustering model appears to outperform the benchmark model in terms of both the AA and the $F1_\mu$ score in the credit rating prediction scenario. Predictions were made using representative transition matrices where $\tau \in \{20, 40, 60\}$. Due to the imbalanced nature of the data set, the significance of the micro-averaged F1 score should have higher precedence over the averaged accuracy. It is a more accurate representation of the model's performance as it takes into account the size of the individual classes in $\mathcal{S}$.

| $\tau$ | Input Date | Predicted Date | (C) AA | (B) AA | (C) $F1_\mu$ | (B) $F1_\mu$ |
|---|---|---|---|---|---|---|
| 15 | 2000-01-01 | 2015-01-01 | **0.8993 (0.0036)** | 0.8585 (0.0027) | **0.6475 (0.0126)** | 0.5047 (0.0095) |
| | 2001-01-01 | 2016-01-01 | **0.9013 (0.0033)** | 0.8705 (0.0011) | **0.6546 (0.0115)** | 0.5467 (0.0038) |
| | 2002-01-01 | 2017-01-01 | **0.9148 (0.0028)** | 0.8712 (0.0020) | **0.7018 (0.0097)** | 0.5491 (0.0070) |
| | 2003-01-01 | 2018-01-01 | **0.9178 (0.0026)** | 0.8816 (0.0016) | **0.7125 (0.0090)** | 0.5855 (0.0058) |
| 10 | 2000-01-01 | 2010-01-01 | **0.9012 (0.0029)** | 0.8751 (0.0001) | **0.6542 (0.0102)** | 0.5627 (0.0004) |
| | 2001-01-01 | 2011-01-01 | **0.9039 (0.0026)** | 0.8833 (0.0001) | **0.6636 (0.0092)** | 0.5915 (0.0003) |
| | 2002-01-01 | 2012-01-01 | **0.9126 (0.0021)** | 0.8748 (0.0002) | **0.6942 (0.0075)** | 0.5617 (0.0007) |
| | 2003-01-01 | 2013-01-01 | **0.9202 (0.0022)** | 0.8860 (0.0003) | **0.7205 (0.0076)** | 0.6009 (0.0011) |
| 5 | 2000-01-01 | 2005-01-01 | **0.9207 (0.0024)** | 0.9135 (0.0001) | **0.7225 (0.0084)** | 0.6974 (0.0003) |
| | 2001-01-01 | 2006-01-01 | **0.9183 (0.0023)** | 0.9138 (0.0000) | **0.7140 (0.0081)** | 0.6983 (0.0000) |
| | 2002-01-01 | 2007-01-01 | **0.9244 (0.0021)** | 0.9097 (0.0000) | **0.7355 (0.0075)** | 0.6840 (0.0000) |
| | 2003-01-01 | 2008-01-01 | **0.9264 (0.0018)** | 0.9179 (0.0000) | **0.7425 (0.0062)** | 0.7128 (0.0000) |

Table 2.5: Results from the credit rating prediction scenario. Results under the column label (C) and (B) represent the results from clustering and the benchmark model respectively.

Similar to the credit rating prediction scenario, the results in Table 2.6 show that the clustering model outperforms the benchmark model. This is observed for both the AA and the $F1_\mu$ score when making predictions using representative transition matrices with $\tau \in \{20, 40, 60\}$.

The main diagonal of the transition matrices estimated from credit ratings tend to be diagonally dominant (Jafry & Schuermann, 2004). The main diagonal represents the probability that a firm maintains its current credit rating after a transition period.

| $\tau$ | Input Date | Predicted Date | (C) AA | (B) AA | (C) $F1_\mu$ | (B) $F1_\mu$ |
|---|---|---|---|---|---|---|
| 60 | 2000-01-01 | 2015-01-01 | **0.7992 (0.0089)** | 0.5671 (0.0053) | **0.6987 (0.0133)** | 0.3506 (0.0080) |
| | 2001-01-01 | 2016-01-01 | **0.8041 (0.0089)** | 0.5769 (0.0077) | **0.7062 (0.0134)** | 0.3654 (0.0115) |
| | 2002-01-01 | 2017-01-01 | **0.8293 (0.0084)** | 0.5746 (0.0044) | **0.7440 (0.0125)** | 0.3619 (0.0066) |
| | 2003-01-01 | 2018-01-01 | **0.8330 (0.0077)** | 0.5899 (0.0054) | **0.7495 (0.0116)** | 0.3848 (0.0081) |
| 40 | 2000-01-01 | 2010-01-01 | **0.7945 (0.0076)** | 0.6371 (0.0040) | **0.6917 (0.0114)** | 0.4557 (0.0060) |
| | 2001-01-01 | 2011-01-01 | **0.7962 (0.0065)** | 0.6647 (0.0078) | **0.6943 (0.0097)** | 0.4970 (0.0116) |
| | 2002-01-01 | 2012-01-01 | **0.8152 (0.0065)** | 0.6543 (0.0054) | **0.7228 (0.0097)** | 0.4814 (0.0080) |
| | 2003-01-01 | 2013-01-01 | **0.8270 (0.0066)** | 0.6772 (0.0059) | **0.7406 (0.0099)** | 0.5158 (0.0088) |
| 20 | 2000-01-01 | 2005-01-01 | **0.8246 (0.0060)** | 0.7983 (0.0000) | **0.7370 (0.0091)** | 0.6974 (0.0000) |
| | 2001-01-01 | 2006-01-01 | **0.8189 (0.0063)** | 0.7989 (0.0002) | **0.7284 (0.0095)** | 0.6983 (0.0003) |
| | 2002-01-01 | 2007-01-01 | **0.8338 (0.0052)** | 0.7893 (0.0000) | **0.7507 (0.0079)** | 0.6840 (0.0000) |
| | 2003-01-01 | 2008-01-01 | **0.8330 (0.0046)** | 0.8085 (0.0000) | **0.7495 (0.0069)** | 0.7128 (0.0000) |

Table 2.6: Results from the transition direction prediction scenario. Results under the column label (C) and (B) represent the results from clustering and the benchmark model respectively.

This observation can be commonly found in the benchmark model as it aggregates all of the training data before estimating the transition matrix. Because the transition matrix is used during the classification process, a consistently diagonally dominated transition matrix leads to similar predictions across the 1000 runs. Although the clustering model uses the same methods in generating the transition matrices, the results are more accurate. The difference is that the clustering model partitions firms with similar transition behaviours together and generates a representative transition matrix from this collection of firms. In other words, it can be said that the representative transition matrix of each cluster is custom-tailored to the distinct behaviour of each group of firms. Hence, the predictions are made on a test firm using a representative transition matrix that best characterizes it.

Judging from the values in Table 2.7, both the clustering and benchmark models demonstrate some ability in producing risk scores that function well as a predictor of the dependent variable (the default behaviour of the firms) as both models produce

$d_{yx} > 0$. However, despite both models performing well, it can be observed that the clustering model outperforms the benchmark model by producing "more effective" risk scores.

| $\tau$ | Input Date | Predicted Date | (C) $d_{yx}$ | (B) $d_{yx}$ |
|---|---|---|---|---|
| 60 | 2000-01-01 | 2015-01-01 | **0.2836 (0.0210)** | 0.1859 (0.0031) |
| | 2001-01-01 | 2016-01-01 | **0.2850 (0.0196)** | 0.1918 (0.0029) |
| | 2002-01-01 | 2017-01-01 | **0.3470 (0.0194)** | 0.2137 (0.0028) |
| | 2003-01-01 | 2018-01-01 | **0.3506 (0.0191)** | 0.2263 (0.0023) |
| 40 | 2000-01-01 | 2010-01-01 | **0.2649 (0.0188)** | 0.1863 (0.0028) |
| | 2001-01-01 | 2011-01-01 | **0.2658 (0.0175)** | 0.1943 (0.0026) |
| | 2002-01-01 | 2012-01-01 | **0.3206 (0.0181)** | 0.2051 (0.0028) |
| | 2003-01-01 | 2013-01-01 | **0.3392 (0.0184)** | 0.2223 (0.0023) |
| 20 | 2000-01-01 | 2005-01-01 | **0.2374 (0.0165)** | 0.1724 (0.0026) |
| | 2001-01-01 | 2006-01-01 | **0.2353 (0.0150)** | 0.1792 (0.0022) |
| | 2002-01-01 | 2007-01-01 | **0.2790 (0.0144)** | 0.2017 (0.0024) |
| | 2003-01-01 | 2008-01-01 | **0.2855 (0.0147)** | 0.2058 (0.0019) |

Table 2.7: Results of the Somers' D for the default behaviour prediction scenario. Results under the column label (C) and (B) represent the results from clustering and the benchmark model respectively.

From Table 2.8, it can be observed that the clustering model outperforms the benchmark model in all performance measures. Treating the false positive rate FPR as a measure of the Type I error and the false negative rate as a measure of the Type II error, we find that the clustering model has an overall lower proportion of FP and FN.

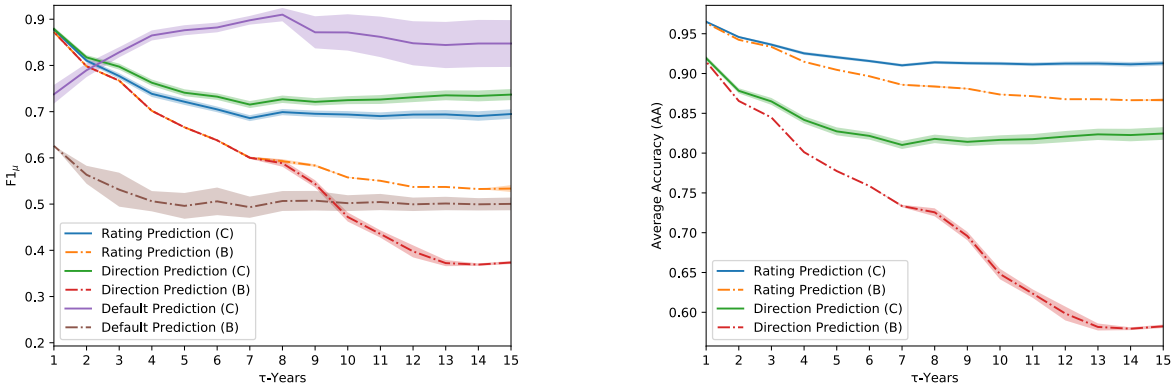
Plotting the F1 score and AA while varying the value of $\tau$ we can determine the effectiveness of each model for different prediction horizons. For practical purposes we take $\tau \in [1, 15]$ years, that is, prediction horizons ranging from one to fifteen years into the future. The transparent areas in the plots of Figure 2.4, represent the standard deviation of the respective results of each classification scenario. From Figure 2.4 it is obvious that the clustering model outperforms the benchmark model for the majority of the tested $\tau$ values in terms of both AA and $\text{F1}_\mu$ for all classification scenarios.

| $\tau$ | Input Date | (C) Re | (B) Re | (C) Pr | (B) Pr | (C) F1 | (B) F1 |
|---|---|---|---|---|---|---|---|
| 60 | 2000-01-01 | **0.9102 (0.0269)** | 0.7408 (0.0460) | **0.6583 (0.0873)** | 0.3445 (0.0230) | **0.7608 (0.0613)** | 0.4681 (0.0170) |
| | 2001-01-01 | **0.9179 (0.0225)** | 0.7486 (0.0421) | **0.6812 (0.0937)** | 0.3526 (0.0261) | **0.7783 (0.0633)** | 0.4774 (0.0188) |
| | 2002-01-01 | **0.9378 (0.0131)** | 0.7325 (0.0519) | **0.7856 (0.0793)** | 0.3941 (0.0272) | **0.8528 (0.0484)** | 0.5087 (0.0150) |
| | 2003-01-01 | **0.9420 (0.0133)** | 0.7628 (0.0354) | **0.7953 (0.0763)** | 0.4283 (0.0354) | **0.8605 (0.0464)** | 0.5465 (0.0222) |
| 40 | 2000-01-01 | **0.9275 (0.0283)** | 0.7655 (0.0410) | **0.6839 (0.0836)** | 0.3487 (0.0281) | **0.7841 (0.0594)** | 0.4767 (0.0220) |
| | 2001-01-01 | **0.9412 (0.0212)** | 0.7836 (0.0388) | **0.7289 (0.0843)** | 0.3742 (0.0382) | **0.8186 (0.0564)** | 0.5027 (0.0284) |
| | 2002-01-01 | **0.9425 (0.0129)** | 0.7428 (0.0501) | **0.7777 (0.0634)** | 0.3876 (0.0342) | **0.8508 (0.0398)** | 0.5014 (0.0189) |
| | 2003-01-01 | **0.9415 (0.0129)** | 0.7685 (0.0332) | **0.7853 (0.0690)** | 0.4358 (0.0389) | **0.8546 (0.0428)** | 0.5492 (0.0232) |
| 20 | 2000-01-01 | **0.9411 (0.0309)** | 0.7833 (0.0340) | **0.6864 (0.0554)** | 0.3351 (0.0383) | **0.7923 (0.0421)** | 0.4658 (0.0324) |
| | 2001-01-01 | **0.9595 (0.0237)** | 0.8111 (0.0303) | **0.7565 (0.0275)** | 0.3731 (0.0555) | **0.8456 (0.0200)** | 0.5058 (0.0450) |
| | 2002-01-01 | **0.9619 (0.0211)** | 0.7838 (0.0393) | **0.8033 (0.0119)** | 0.3958 (0.0345) | **0.8753 (0.0109)** | 0.5132 (0.0251) |
| | 2003-01-01 | **0.9598 (0.0100)** | 0.8088 (0.0220) | **0.7985 (0.0119)** | 0.4380 (0.0313) | **0.8717 (0.0080)** | 0.5583 (0.0246) |

Table 2.8: Results using the worst-case scenario thresholds for the default behaviour prediction scenario. Results under the column label (C) and (B) represent the results from clustering and the benchmark model respectively

As we vary $\tau$ there are two observations that can be made. The first observation: For decreasing values of $\tau$ the degree by which the clustering model outperforms the benchmark model also decreases. It is common for firms to maintain their current rating across shorter time periods, increasing the performance of the benchmark model as $\tau$ decreases. With longer time periods, firms are more likely to change ratings, and so the resulting benchmark transition matrix from Equation (2.11) with large $\tau$ does a poor job in catching all the different behaviours of every firm. The second observation: For increasing values of $\tau$, both the clustering and benchmark model performance decrease for the credit rating prediction and transition direction prediction scenarios. The rate at which the performance deteriorates, however, is higher in the benchmark model. The clustering model's performance decreases but then levels out eventually for $\tau \in [1, 15]$ in all classification scenarios. In the default prediction scenario the clustering model performance actually increases while the benchmark model performance decreases with increasing values of $\tau$. It can be stated that the clustering model's performance is more consistent over $\tau \in [1, 15]$. The

confusion matrices of each classification scenario and a breakdown of the performance of the model across each credit rating in terms of the F1 score is presented in Table A.2 and Figure A.3.




(a) $F1_\mu$ score results.

(b) AA measure results.

Figure 2.4: The results from the credit rating, rating transition direction, and default prediction scenarios. The labels (C) and (B) represent the clustering and benchmark models respectively. Note that the F1 score is not micro-averaged for the default prediction scenario.

## 2.6 Conclusion

The changes in the credit rating of a firm can have a substantial impact on bond pricing, valuation of credit derivatives, and management decisions of companies. In this chapter, we adapted the SBC technique used in web-usage mining to improve transition matrix estimation methods in the context of credit risk. The clustering algorithm used was the PCA-guided K-means algorithm and the analysis is in part possible due to the convenient cluster-ready representation of sequence matrices. Some properties of sequence matrices were presented which can prove to be beneficial

for future development and implementation in models that intend to utilize this sequence matrix representation. Credit rating prediction, credit rating transition direction prediction, and default behaviour prediction were the three classification scenarios that were used to test the performance of the clustering model.

The clustering model was compared against the benchmark model where clustering was absent. The results suggest that by clustering the sequence matrices of firms, the overall predictive power of the representative transition matrices is greater than just using a single transition matrix. The performance of the models in the credit rating prediction and transition direction prediction classification scenarios were evaluated in terms of the average accuracy and micro-averaged F1 score. The performance of the models in the default behaviour prediction classification scenario were evaluated in terms of the recall, precision, and F1 score. The worst-case scenario threshold provides a suitable means of evaluating our model against the benchmark model.

# Chapter 3

# High-Order Markov Chains and Sequence-Based Clustering

In this chapter, we extend the first-order SBC model introduced in Chapter 2. Instead of using first-order Markov models, we propose the use of high-order Markov models augmented with SBC. The efficacy of these models were evaluated under two different contexts: Default prediction for credit risk assessment and Web-usage mining where we explore the applicability of the higher-order SBC models in a more general setting.

## 3.1 Introduction

Many random processes can be modelled using Markov chains. To describe a process using first-order Markov chains, we assume that the future state of a random process is only dependent on the current state. Although in many cases this assumption is sufficient, the accuracy of the model can often be improved by considering additional information about the process's history. This can be done by using higher-order

Markov chains. High-order Markov chains assume that the future state of a random process is dependent not only on the current state, but also states previously visited by the process. A higher-order Markov model considers more information further into the past while a lower-order considers information more closer to the present. By considering additional information about the past, it has been shown that high-order Markov chains can appropriately model processes in a number of different applications including wind power distribution modelling (Carpinone et al., 2015), rear TV cover batch, DNA sequence reduction (Şahin et al., 2019), software reliability assessment of CubeSat nano satellites (Yakovyna & Symets, 2021), and secondary traffic collision due to an initial primary incident (Pugh & Park, 2021). While high-order Markov chains can be used in a variety of modelling problems, one should be cautioned that the effectiveness of high-order Markov chains is highly dependent on the particular application. For example, Şahin et al. (2019) noted that the first-order model outperformed the second-order model for predicting DNA sequences, noting that the next piece in a DNA sequence is highly correlated with the previous piece.

Absorbing Markov chains, a specific type of reducible Markov chain, are of particular interest. An absorbing Markov chain contains absorbing states, where once the process enters, it is unable to leave. Absorbing states often represent the case of failure or success, ending the process. Some examples of absorbing states include finding a service such as a taxi in taxi searching behaviour (Wong et al., 2005), and drop-out behaviour of students in educational institutions (Kuzilek et al., 2018; Nicholls, 2007). In this chapter, we investigate the performance of high-order absorbing Markov chains paired with SBC. As noted in Chapter 2, SBC was first introduced by Park et al. (2008) and is a clustering framework permitting the clustering of sequences based on their relative transition frequency. This approach forms clusters of sequences that express similar transition dynamics. This model

will be used in the application of corporate credit rating prediction and Web-usage mining.

In this chapter, we focus on two key applications of high-order Markov chains and sequence-based clustering. The first of these is corporate credit rating prediction, a critical aspect of financial risk assessment. Corporate credit ratings are assessments of the credit risk of a bank or financial institution and their ability to satisfy their loan obligations. These credit ratings are assigned by credit rating agencies such as Standard & Poor's, Moody's, and Fitch. The ratings are typically represented by letter grades ranging from AAA (representing the best credit quality) to D (representing default, the worst credit quality). In this context, the absorbing state for an institution would be default. Credit ratings play an important role in the finance industry as bond prices are dependent on risk associated with the issuing institution. An institution likely to default would need to compensate for this elevated risk by offering a higher premium. In addition, sovereign credit rating downgrades can also increase the risk of downgrades for credit ratings and negatively impact the performance of sovereign-bound firms, that is, firms whose credit ratings are near the rating of their sovereign (Almeida et al., 2017; To et al., 2022). By better modelling default behaviour of credit ratings, bonds and other financial products based on credit risk can be more accurately priced. This would lead to more fairly priced products for investors and additionally, would lead to better informed credit risk management decision making for portfolio managers. Furthermore, more accurate default behaviour modelling may contribute to a better understanding and modelling of the spillover effects of sovereign downgrades. The second application we are interested in is predicting a Web user's navigation on the Web, the primary Web-usage mining topic that has been studied for decades, but with few applications addressing the absorbing behaviors. In the Web-usage case, the absorbing state of a Web browsing sessions can be a Web

page with the user's desired information, or deciding on a movie after browsing and previewing a catalogue of streamable movies. Unlike in default prediction, there can be many absorbing states a user can end their browsing session on in this field.

A practical concern associated with high-order Markov chains is the exponential increase in number of transition probabilities to be estimated as the order and size of the state spaces increases. To treat this, Raftery (1985a) introduced a mixture transition distribution (MTD) model for modelling high-order Markov chains. Building on the work by Raftery (1985a), Ching et al. (2004) proposed and developed a more general higher-order Markov chain model for categorical data sequences (estimation of the lag parameters was done using linear programming). The categorical data sequences used included DNA sequences, sales demand data, and server log data. The MTD model was further extended by Nicolau (2014) who proposed a multivariate Markov chain model based on the MTD model. This extension is also capable of estimating the transition probabilities of higher-order Markov chains.

Remarkably, in the case of corporate credit ratings, there appears to be little research that addresses the efficacy of solely using high-order Markov chains, let alone in combination with machine learning techniques in forecasting default likelihood. Comparatively, high-order Markov chains were used significantly more in the Web-usage mining context than in the credit rating context. However, as far as we can tell there appears to be no literature on utilizing both high-order absorbing Markov chains and machine learning techniques, especially in the case where SBC is used, in the context of corporate credit rating and Web-usage mining. The advantage of using SBC is that we are not required to make restrictive assumptions about the data set as we use the K-means algorithm for clustering. We do however make the assumption that the data can be modelled using high-order absorbing Markov chains. SBC enables the capture of non-Markovian characteristics often observed in credit rating

sequences while the use of high-order Markov chains explicitly considers previous state dependencies. We also observe an unexpected benefit where the first-order Markov model paired with SBC offered a competitive advantage over using strictly high-order Markov chains in terms of parsimony. Hence, there is potential to use first-order Markov chains with SBC as an alternative to high-order Markov chains, given that SBC can sufficiently replace the role of high-order Markov chains.

The remainder of this chapter is organized into 6 sections. In Section 3.2 we present a review of related works. In Section 3.3, we outline the methodology of our work. We begin with outlining the theory of high-order Markov chain models, followed by describing how sequences can be represented using sequence matrices and how these objects can be used in clustering. The estimation of high-order transition matrices is also outlined in this section. In Section 3.4 we present the two classification scenarios our models will be tested against. In Section 3.5, we present the data used in this chapter. In Section 3.6, we present the results and discussion of the results. Finally, we conclude the chapter in Section 3.7.

## 3.2 Related Works

### 3.2.1 Corporate Credit Ratings

Over time the credit ratings of a firm can transition from one rating to another. This dynamic provides a natural application for Markov chains. Indeed, there are many studies that apply Markov chains to credit rating transitions (Jarrow et al., 1997; Sharma et al., 2018; Thomas et al., 2002). A crucial credit rating transition is the transition to the default state. An issuer of bonds that receives this rating has entered into bankruptcy and hence the repayment of the loan is effectively forfeited.

The default state can be considered an absorbing state and thus, appropriately modelled using absorbing Markov chains. In Parnes (2010), the complete probability distribution of time-to default for various credit ratings is presented. They also consider various credit rating transition sensitivities with the aim of recognizing developments of bond portfolio. They find it takes a little more than a decade for very low credit ratings to transition to a default rating while high credit ratings can take more than a century to default and the credit deterioration of a portfolio is highly sensitive to the number of bonds that will be upgraded from low investment grade to medium-high investment grade. Dharmaraja et al. (2017) presented the mean time to absorption and probability of absorption in closed form under a hybrid model of credit risk using discrete time first-order Markov chains.

Despite this convenient representation, credit rating transitions are known to also display a number of non-Markovian effects. D'Amico et al. (2019) lists these phenomena as downward rating momentum, duration, time non-homogeneity, ageing effect. To capture these phenomena they review several homogeneous and non-homogeneous semi-Markov models, including a model based on Markov regenerative process (Pasricha et al., 2017). Downward momentum specifically refers to the increase in probability that the next rating change will be a downgrade if the previous rating change was also a downgrade. In order to capture the idea of rating momentum, the rating before the current rating should be known (Altman & Kao, 1992; Dos Reis et al., 2020; Frydman & Schuermann, 2008; Lando & Skodeberg, 2002). Using a Markov mixture model, Frydman and Schuermann (2008) suggest that the credit rating a firm is assigned in the future depends not only on its current credit rating but is also influenced by the previous credit rating assignments. More recently, Dos Reis et al. (2020) proposed a method based on point processes, a generalization of Markov processes. They use this model to capture the non-Markovian effect of rating

momentum. They note their model yields higher probabilities of default for investment grades and lower default probabilities for some speculative grades. An additional characteristic observed in credit ratings is the effect of rating persistence. This is the observation that the longer a firm maintains a credit rating, the likelihood of a downgrade or upgrade decreases (Fuertes & Kalotychou, 2007; Lando & Skodeberg, 2002).

The non-Markovian characteristic we aim to consider in this study is the downward momentum observation. This will be accomplished by using a combination of high-order Markov models and SBC. Regarding high-order Markov chains estimation, we find that the use of high-order Markov chains in credit rating transitions is a recent development beginning with the work done by Baena-Mirabete and Puig (2018) who proposed several parsimonious models for high-order Markov chains. They address the issue of persistent multistates (where the probability that an asset moves from one rating to another is small) in credit rating data where the sparseness of the data often leads to convergence failures in the estimation and unrealistic transition probabilities. Based on municipal rating data and the estimated transition probabilities, they further verify the findings in previous literature. Using second-order Markov chains, they observe downgrade momentum. Additionally, they also observe that the more a credit rating persists in its current rating, the lower the probability of a downgrade or upgrade in the future.

### 3.2.2 Web-Usage Mining

In the context of Web-usage mining an absorbing state is a natural notion representing the end of a Web session or the purchase of a product. This has led to a number of studies incorporating this concept using absorbing Markov chains. For example,

Massa and Puliafito (2002) proposed a graphical monitoring instrument based on absorbing Markov chains to analyze navigational paths of Web users. In the interest of journalism research, Vermeer and Trilling (2020) found absorbing Markov chains to be an effective approach in analyzing and discovering meaningful patterns in the sequence of Web pages users take to find the online news they are looking for.

While previously mentioned studies use first-order Markov chains, a number of research have made use of higher-order Markov chains while leveraging machine learning techniques to mine Web user behaviour. To improve Web user navigation analysis, Borges and Levene (2007, 2005) utilized K-means clustering to extend first-order Markov chains to variable-length Markov chains. They suggest that the efficacy of considering short or long term history is dependent on the specific website. That is, different sites require different amounts of history to understand. Awad and Khalil (2012) predicted Web user's navigation behavior using various combinations of Markov models and association rule mining models. In particular, they use an all-$K$th Markov model, that is, an ensemble of Markov chain models of varying order where a prediction is made on the highest feasible order Markov chain. They pair this model with association rule mining and find that this combination results in improved prediction accuracy. Pal et al. (2021) developed a real-time user click stream processing framework making use of the hybrid model that consists of the K-means clustering algorithm, and the expectation-maximization algorithm followed by the use of high-order Markov chains to analyze streams of Web user's browsing behaviour allowing for the prediction of future Web clicks. Although high-order Markov models have shown to be effective, in some cases, there is evidence that suggests this might not always be true in Web navigation behaviour (Piccardi et al., 2021). In particular, Deshpande and Karypis (2004) developed a method utilizing a pruning scheme in combination with the all-$K$th Markov model to reduce the state

complexity and improve prediction accuracy. The effectiveness of high-order Markov chains in the context of Web navigation pattern is also challenged in Singer et al. (2014). They note that while high-order Markov models are effective in general topic navigation prediction, they appear to underperform when considering page-to-page Web navigation prediction. While higher-order Markov models can prove to add additional predictive power, we also consider the possibility that for certain choices of $k$th-order Markov chains, SBC may be a viable alternative to circumvent the issue of the exponentially growing number of parameters in such models.

While Markov chains alone have shown to be quite effective in modelling Web-usage mining in contrast to the credit rating context, only a handful of studies have highlighted the effectiveness of incorporating clustering techniques in Markov chain modelling (Ansari et al., 2015; Borges & Levene, 2005; Mlika & Karoui, 2020). These studies have shown that the predictive capabilities of Markov chains can be bolstered by incorporating clustering techniques in the model. In the context of Web-usage mining, work done by Cadez et al. (2003) may be an example of the earliest use of clustering with Markov chains. They use a model-based approach to cluster sequences, as opposed to the distance-based approach in order to account for sequences with varying lengths. Park et al. (2008) developed a general SBC method for Web-usage mining using fuzzy ART-enhanced K-means clustering while proposing various sequence representation schemes in association with Markov models. Using their sequence representation schemes, pre-existing distance-based clustering algorithms can be used for identifying Web user groups of similar navigation behaviors. Employing similar methodologies, Park and Vasudev (2017) used first-order absorbing Markov chains to predict Web users' navigation behavior. Prediction of navigation behaviour was facilitated by the use of SBC techniques with absorbing Markov chains. Their results suggest that using absorbing Markov chains in this context may

strengthen the accuracy of the estimated remaining time of a Web user's session or the probability to purchase a specific product.

## 3.3 Methodology

### 3.3.1 High-Order Markov Chains

In this study we treat Web-usage mining sequences and credit rating transitions as discrete time-homogeneous high-order Markov chains. Let $\{X_t\}$ be a sequence of random variables where $X_t \in \mathcal{S}$ and $\mathcal{S}$ is defined to be the set $\{1, 2, ..., n\}$, representing the state space. The $k$th-order Markov property says that the observation at time $t$ is dependent only on the last $k$ observations. The transition probabilities of a $k$th-order Markov chain are written as

$$Pr\{X_t = i_0 \mid X_0 = i_t, ..., X_{t-1} = i_1\} \tag{3.1}$$

$$= Pr\{X_t = i_0 \mid X_{t-k} = i_k, ..., X_{t-1} = i_1\} \tag{3.2}$$

$$= p_{i_k ... i_0}. \tag{3.3}$$

When $k = 1$ we have a first-order Markov chain where the observation of $X_t$ at time $t$ is dependent only on the previous observation at time $t - 1$. The following is

| Notation | Definition |
|---|---|
| $\mathcal{S}$ | State space of the Markov chain |
| $n$ | Number of states in the state space |
| $\mathcal{A}$ | Set of absorbing states of a Markov chain |
| $k$ | Order of a Markov chain |
| $X_t^m$ | State of the $m$th sequence at time $t$ |
| $T_m$ | Full length of the $m$th sequence |
| $\mathbf{x}_m(t)$ | Sequence vector up to time $t$, defined by $(X_0^m, ..., X_t^m)$ |
| $\mathbf{x}_m(t \mid k)$ | High-order state vector at time $t$ defined by $(X_{t-k+1}^m, ..., X_t^m)$ for the $m$th sequence |
| $C$ | Total number of clusters used in the K-means algorithm |
| $c$ | The $c$th cluster among $C$ clusters. |
| $\mathbf{S}_m^k$ | $k$th-order sequence matrix of the $m$th sequence |
| $\mathbf{P}_c^k$ | Representative $k$th-order transition matrix of the $c$th cluster |
| $\lambda_g$ | $g$th lag parameter used in the MTDg model |
| $\mathbf{Q}_{\text{MTD},g}$ | An $n \times n$ transition matrix describing the relationship between the $g$th lag and the present |
| $q_{i_g i_0}^{(g)}$ | Non-negative elements of $\mathbf{Q}_{\text{MTD},g}$ |
| $\mathbf{Q}_c^k$ | Sub-matrix of the $\mathbf{P}_c^k$ containing the transition probabilities from and to transient states |
| $\mathbf{R}_c^k$ | Sub-matrix of $\mathbf{P}_c^k$ containing the transition probabilities from a transient state to an absorbing state |
| $\tau$ | Number of time steps, indicating how far into the future we want to predict |
| $\mathbf{B}_c^{k,(\tau)}$ | Matrix containing the probabilities of being absorbed within $\tau$ steps given the current state $\mathbf{x}_m(t \mid k)$ for the $c$th cluster |
| $\tilde{\mathbf{m}}_c^k$ | Vector containing the average number of steps before absorption given the current state $\mathbf{x}_m(t \mid k)$ for the $c$th cluster |
| $\mathcal{L}$ | Set of labels used for classification |
| $\theta$ | Threshold used to compare against the absorption probabilities |
| $\hat{y}_m(\theta)$ | Label or prediction made for the $m$th sequence given for some threshold $\theta$ |
| $\tilde{\mathbf{x}}_m(t)$ | Remaining length of the $m$th sequence, defined by $T_m - t$ |
| $K$ | Number of folds used in $K$-fold cross validation |
| $M$ | Number of $K$-fold cross validation experiments we run |
| $S_{\text{size}}$ | Sample size, the number of sequences used in each experiment |

Table 3.1: A table to reference for some notation used throughout this chapter.

an example of a transition matrix of a high-order Markov chain with $k = 2$ and $n = 3$

$$\mathbf{P} = \begin{array}{cc} & \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 3 & 1 \\ 3 & 2 \\ 3 & 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} p_{111} & p_{112} & p_{113} \\ p_{121} & p_{122} & p_{123} \\ p_{131} & p_{132} & p_{133} \\ p_{211} & p_{212} & p_{213} \\ p_{221} & p_{222} & p_{223} \\ p_{231} & p_{232} & p_{233} \\ p_{311} & p_{312} & p_{313} \\ p_{321} & p_{322} & p_{323} \\ p_{331} & p_{332} & p_{333} \end{pmatrix} \end{array}. \tag{3.4}$$

The size of a transition matrix for a $k$th-order Markov chain with a state space of size $n$ is $n^k \times n$.

To calculate the absorbing probabilities of a $k$th-order Markov chain, we use the language and analysis of $k$-dimensional matrices described by Elayat (1973). A $k$-dimensional matrix is denoted by

$$\mathbf{A} = [a_{i_k i_{k-1} \ldots i_1 i_0}], \tag{3.5}$$

where $i_j \in \mathcal{S}$ for $j = 0, 1, ..., k$. Therefore, the $(i_k i_{k-1} \ldots i_1 i_0)$-th coordinate of $\mathbf{A}$ is represented by $A[i_k, ..., i_0] = a_{i_k i_{k-1} \ldots i_1 i_0}$. For example, the transition matrix (3.4) is a $k$-dimensional matrix

$$\mathbf{P} = [p_{i_2 i_1 i_0}], \tag{3.6}$$

with $n = 3$ and $k = 2$. We also note that unlike first-order transition matrices, the

indices do not correspond to the row and column of the transition matrix.

In general, the transition probability matrices $\mathbf{P}$ of a $k$th-order Markov chain are of size $n^k \times n$. The elements of the transition matrix $\mathbf{P}$ are denoted by $p_{i_k i_{k-1} \ldots i_1 i_0}$ where

$$p_{i_k i_{k-1} \ldots i_1 i_0} = Pr\{X_t = i_0 \mid X_{t-k} = i_k, ..., X_{t-1} = i_1\}, \tag{3.7}$$

that is, the probability of transitioning to state $i_0$ is conditioned on the previous $k$ states for each row of the matrix. The transition probabilities must also satisfies the following constraints

$$0 \leq p_{i_k i_{k-1} \ldots i_1 i_0} \leq 1, \tag{3.8}$$

$$\sum_{i_0=1}^{n} p_{i_k i_{k-1} \ldots i_1 i_0} = 1 \quad \forall i_k, i_{k-1}, \ldots, i_1 \in \mathcal{S}. \tag{3.9}$$

Next, we define matrix multiplication for $k$-dimensional transition probability matrices. Let $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ be $n^k \times n$ matrices. Then we define the product $\mathbf{C} = \mathbf{AB}$ where the elements of $\mathbf{C}$ are given by

$$c_{i_k i_{k-1} \ldots i_1 i_0} = \sum_{j=1}^{n} a_{i_k i_{k-1} \ldots i_1 j} b_{i_{k-1} \ldots i_1 j i_0}. \tag{3.10}$$

For $k$-dimensional transition matrices, matrix multiplication as defined by equation (3.10) satisfies the following properties:

1. $\mathbf{AB} \neq \mathbf{BA}$
2. $(\mathbf{AB})\mathbf{C} \neq \mathbf{A}(\mathbf{BC})$
3. $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$
4. $(\mathbf{B} + \mathbf{C})\mathbf{A} = \mathbf{BA} + \mathbf{CA}$

The identity matrix $\mathbf{I}$ for any matrix $\mathbf{P}$ of size $n^k \times n$ is defined by

$$\mathbf{PI} = \mathbf{P}, \tag{3.11}$$

where the entries of $\mathbf{I}$ are $i_{i_k i_{k-1} \ldots i_1 i_0}$ defined by

$$i_{i_k i_{k-1} \ldots i_1 i_0} = \begin{cases} 1 & \text{if } i_1 = i_0, \\ 0 & \text{Otherwise.} \end{cases}$$

It should also be mentioned that, $\mathbf{IP} \neq \mathbf{P}$ by the matrix multiplication properties noted above. The Chapman-Kolmogorov equation for a transition matrix $\mathbf{P}$ is then

$$p^{(\tau)}_{i_k i_{k-1},\ldots,i_1 i_0} = \sum_{j=1}^{n} p_{i_k i_{k-1},\ldots,i_1 j} p^{(\tau-1)}_{i_{k-1},\ldots,i_1 j i_0}, \tag{3.12}$$

where $\tau$ is a positive integer greater than 1. Keeping in mind that the associative law no longer holds and thus, the order of operations is important. Therefore, the $\tau$-step transition probability is represented by

$$\mathbf{P}^{(\tau)} = \mathbf{P}\mathbf{P}^{(\tau-1)} \tag{3.13}$$

$$= \mathbf{P}(\mathbf{P}(\mathbf{P}\ldots(\mathbf{P}(\mathbf{PP}))\ldots)) \tag{3.14}$$

$$= \mathbf{P}^{\tau}, \tag{3.15}$$

where the elements are defined by equation (3.12).

For a time-homogenous absorbing Markov chain of order $k$ and state space $\mathcal{S} = \{1, 2, \ldots, n\}$, we assume $1, 2, \ldots, n-r$ are transient states and denote the set of absorbing states by $\mathcal{A} = \{n-r+1, \ldots, n\}$. Then the rows of the transition matrix $\mathbf{P}$

of size $n^k \times n$ can be re-arranged such that

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I_1} \end{bmatrix}, \tag{3.16}$$

where $\mathbf{Q}$ is an $(n-r)^k \times (n-r)$ matrix containing the probabilities of transitioning from transient states to transient states, $\mathbf{R}$ is an $(n-r)^k \times r$ matrix containing the probabilities of the one-step transition from the transient states to absorbing states.

The sub-matrix $\mathbf{0}$ is a zero matrix of size $n^k - (n-r)^k \times (n-r)$, and $\mathbf{I_1}$ is matrix of size $n^k - (n-r)^k \times r$ containing the probabilities

$$p_{i_k i_{k-1},\ldots,i_1 i_0} = \begin{cases} 1 & \text{if } i_0 = i_{\hat{j}}, \\ 0 & \text{Otherwise,} \end{cases} \tag{3.17}$$

where $\hat{j} = \max\{1 \leq j \leq k \mid i_j \in \mathcal{A}\}$ denotes the greatest index $\hat{j}$ such that $i_{\hat{j}}$ is in an absorbing state, in other words, the earliest point in the observable history where the process is in an absorbing state.

### 3.3.2 High-Order Sequence-Based clustering

The machine learning technique used to reinforce the Markov chain models will be SBC. We begin by defining sequence matrices and outlining how these objects will be clustered.

**Definition 3.3.1.** Let $m \in \mathbb{N}$, $X_1^m, X_2^m, ..., X_{T_m}^m$ be a sequence of random variables, and $\mathcal{S}$ be the state space. Then the $m$th sequence vector of length $T_m \in \mathbb{N}$ is defined by the vector $\mathbf{x}_m(T_m) = (X_1^m, X_2^m, ..., X_{T_m}^m)$ with states $X_t^m \in \mathcal{S}$.

**Definition 3.3.2.** Let $N_{i_k,\ldots,i_0}$ be the number of transitions to state $i_0$ given that the

previous $k$ states are $i_k, i_{k-1}, ..., i_1$ for the $m$th sequence with sequence vector $\mathbf{x}_m(T)$. The corresponding sequence matrix $\mathbf{S}_m$ of order $k$ is then an $n^k \times n$ matrix whose entries are denoted by

$$S_m^k[i_k, ..., i_0] = \frac{N_{i_k,...,i_0}}{\sum_{i_j=1}^{n} N_{i_k,...,i_1,i_j}} \tag{3.18}$$

and so the entries $S_m^k[i_k, ..., i_0]$ are the relative transition frequency to state $i_0$ conditional on the previous states being $i_k, ..., i_1$.

As outlined in Park et al. (2008), for a given sequence we can construct a sequence matrix which can be clustered using the K-means algorithm. The K-means algorithm is used to partition a data set into $C$ clusters by minimizing the within cluster variance which is equivalent to minimizing the squared Euclidean distance between each data point in each cluster. Treating the elements of $\mathbf{S}_m^k$ as coordinates in Euclidean space, the Euclidean distance between sequence $\mathbf{x}_{m_1}$ and $\mathbf{x}_{m_2}$ is defined as

$$d_{\text{euc}} = \sqrt{\sum_{i_k,...,i_0} |S_{m_1}^k[i_k, ..., i_0] - S_{m_2}^k[i_k, ..., i_0]|^2}. \tag{3.19}$$

For each sequence vector $\mathbf{x}_m(T_m)$ we can generate the sequence matrix $\mathbf{S}_m$. The sequences matrices are then partitioned into $C$ different clusters based on their distance to a random initialization of $C$ centroids. How close a sequence matrix is to a centroid is based on the Euclidean distance. The assignment of a sequence matrix to a centroid is therefore defined as

$$c^* = \arg\min_c \|\mathbf{S}_m - \boldsymbol{\mu}_c\|, \tag{3.20}$$

where $\boldsymbol{\mu}_c$ is the centroid of the $c$th cluster and $c^*$ is the new cluster that the $m$th

sequence is assigned to. New centroids are calculated once all assignments are made and the assignment process is repeated considering these new centroids. As the within cluster sum of squares is reduced, the algorithm stops after convergence or when a maximum number of iterations has been reached. It should be noted that the algorithm solution may not be a global minimum as it can get stuck in a local minimum. After each sequence matrix has been assigned to a cluster we can then model the transition dynamics of each cluster using high-order absorbing Markov chains.

### 3.3.3 Estimating Transition Matrices: The MTD model

To describe the transitions of the high-order Markov chains, we use high-order transition matrices. As described in Berchtold and Raftery (2002) and Raftery (1985a), it is a highly laborious task to explicitly estimate every parameter of a $k$th-order Markov chain. This is due to the rapid increase in the number of parameters to be estimated. Since each row of the transition matrices represents a probability distribution, and so must sum to 1, there are $n-1$ parameters in each row that must be estimated. Therefore, the fully parameterized high-order Markov model results in $n^k(n-1)$ total parameters to be estimated. To manage the large number of parameters Raftery (1985a) introduced the following MTD model for modelling time-homogeneous high-order Markov chains. The transition probabilities are estimated in

the following fashion

$$Pr\{X_t = i_0 \mid X_{t-k} = i_k, ..., X_{t-1} = i_1\} \tag{3.21}$$

$$= \sum_{g=1}^{k} \lambda_g Pr\{X_t = i_0 \mid X_{t-g} = i_g\} \tag{3.22}$$

$$= \sum_{g=1}^{k} \lambda_g q_{i_g i_0}, \tag{3.23}$$

subject to the following constraints

$$\sum_{g=1}^{k} \lambda_g = 1, \tag{3.24}$$

$$\lambda_g \geq 0, \tag{3.25}$$

where $\lambda_g$ are the respective lag parameters of the past observations $X_{t-g}$, and $q_{i_g i_o}$ are the non-negative elements of an $n \times n$ transition matrix $\mathbf{Q}_{\text{MTD}}$ whose rows sum to 1. Note that this transition matrix is not necessarily the first-order transition matrix. Constraints (3.24) and (3.25) ensure that the model returns probabilities. Given the proposed MTD model, we can see that there are $k-1$ independent lag parameters by constraint (3.24), and $n(n-1)$ parameters of $\mathbf{Q}_{\text{MTD}}$ to estimate. This results in $(k-1)+n(n-1)$ total parameters to estimate. Hence the model proposed by Raftery (1985a) achieves a substantially more parsimonious model.

A benefit of the MTD model is that the lag parameters can be used to interpret the dependence of the model on varying degrees of lag. For example, if $\lambda_1$ is substantially greater than $\lambda_2$ and $\lambda_3$, then the probability of predicting the present value is more dependent on the first lag than the second or third lag, suggesting the observed behaviour follows more closely to that of a first-order Markov chain. On the other

hand, a greater value of $\lambda_g$ for $g > 1$ would suggest a greater dependence on a later lag and hence, the observed behaviour follows more closely to that of a higher-order Markov chain. A more general interpretation can be obtained by relaxing constraint (3.25) allowing for negative $\lambda_g$ (Raftery & Tavare, 1994). Allowing for negative lag parameters presents an opportunity to consider the inverse relationship between the lag and the present.

In this study, we will use a generalization whereby the transition matrix $\mathbf{Q}_{\text{MTD}}$ is no longer assumed to be constant and instead, allowed to vary for each lag. This generalization of the MTD model was first proposed by Raftery (1985b) and is called MTDg. Similar to equation (3.23), the model is defined as

$$Pr\{X_t = i_0 \mid X_{t-k} = i_k, ..., X_{t-1} = i_1\} \tag{3.26}$$

$$= \sum_{g=1}^{k} \lambda_g q_{i_g i_0}^{(g)}, \tag{3.27}$$

where $q_{i_g i_0}^{(g)}$ are the non-negative elements of $\mathbf{Q}_{\text{MTD},g}$, an $n \times n$ transition matrix describing the relationship between the $g$th lag and the present. The number of parameters to be estimated is then $(k-1) + kn(n-1)$. The number of parameters in the MTDg model is greater than the number of parameters in the MTD model. The trade-off for the increase in the number of parameters is the increase in generality. In practice, due to the existence of absorbing states we can further reduce the number of parameters to estimate by at least $|\mathcal{A}|$. A diagram of the high-order SBC model can be found in Figure 3.1.

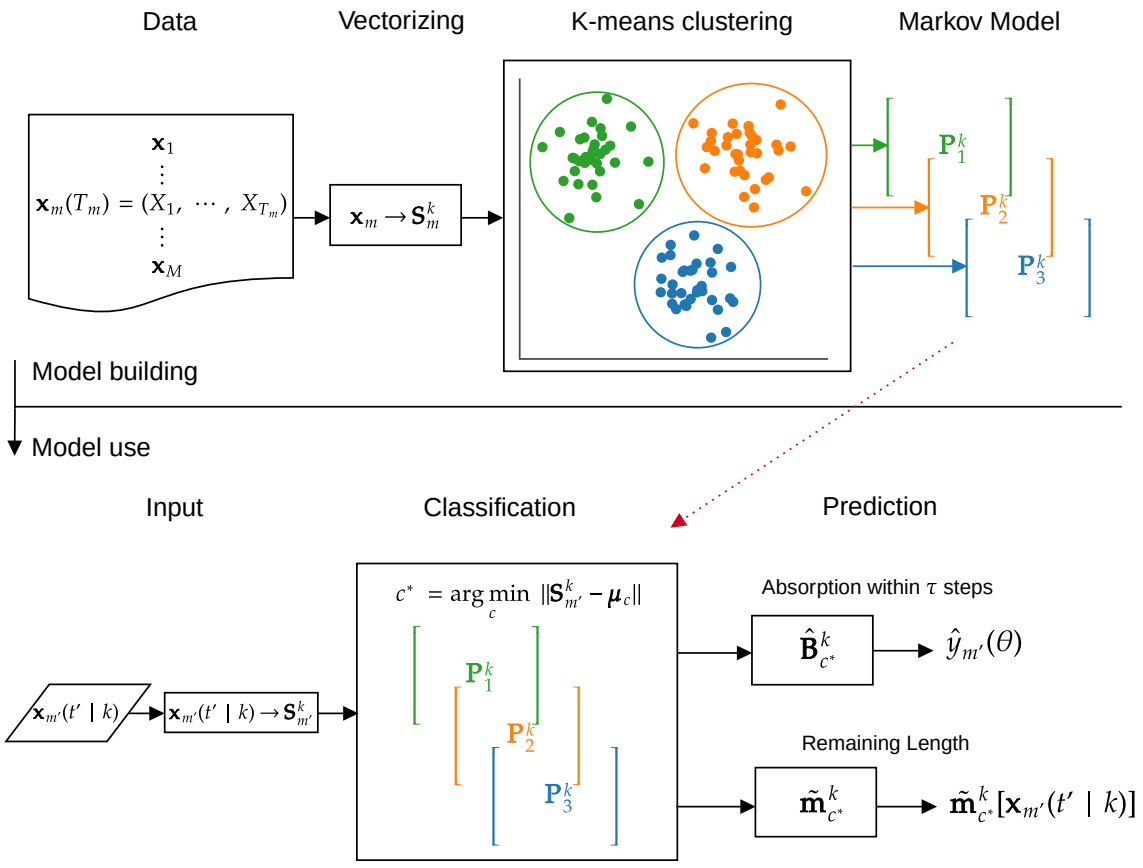



Figure 3.1: A diagram detailing the high-order Markov model combined with SBC.

## 3.4 Classification using Sequence-Based Clustering

We will be investigating how well high-order absorbing Markov chains can model credit rating transitions and Web-usage behaviour. In the credit rating context, we will be testing the effectiveness of high-order absorbing Markov chains in predicting the default of companies within some period of time. In the Web-usage context, we tested the predictive capabilities of the model in predicting the termination of a browsing session within some number of Web page changes, capturing a similar

notion of default in the credit rating setting. Finally, the estimated average length of each browsing session using high-order absorbing Markov chains will be compared to the empirical session duration. In other words, for a $k$th-order Markov chain, and number of transition steps $\tau$, we tested our model against three scenarios:

1. The classification of firms most likely to default within $\tau$ transitions based on the last $k$ credit ratings assigned to the firm.

2. The classification of Web sessions likely to terminate within $\tau$ transitions based on the last $k$ Web pages visited.

3. Determining the remaining length of a browsing session based on the last $k$ Web pages visited.

Each scenario will be evaluated using K-fold cross-validation. Therefore, we will be splitting the data sets into their respective training and testing sets. The model is trained using the training set and is evaluated based on its classification performance using the testing set.

Using the testing set, we assign a single firm to one of the $C$ clusters based on the Euclidean distance between $\mathbf{S}_m^k$ and the clusters' centroid $\boldsymbol{\mu}_c$. After assigning the sequence to a cluster we can estimate the future behaviour of the sequence by using the cluster's representative transition matrix $\mathbf{P}_c^k$.

### 3.4.1 Absorption Within $\tau$ Transitions

In both the credit risk and Web-usage setting, there is only one absorbing state. This absorbing state represents receiving the default credit rating in the credit risk setting, and the end of a Web browsing session in the Web-usage setting. We will evaluate

whether the estimated probabilities given by high-order Markov chains with SBC can be used to accurately predict whether a sequence will truly be absorbed within $\tau$ transitions.

To calculate the probability of absorption within the next $\tau$ transitions, we can use the representative transition matrix $\mathbf{P}_c^k$ of the $c$th cluster. We first define the following sub-matrices of $\mathbf{P}_c^k$ used in the calculations.

**Definition 3.4.1.** Given a state space $\mathcal{S}$ and absorbing set $\mathcal{A}$, let $\mathbf{P}_c^k$ be the $k$th-order representative transition matrix of cluster $c$ of size $n^k \times n$, we define sub-matrices $\mathbf{Q}_c^k$ and $\mathbf{R}_c^k$ as in equation (3.16), containing the probabilities of transitions between the transient states and the probabilities of transitioning to an absorbing state, respectively.

Using equation (3.12), the given sub-matrices $\mathbf{Q}_c^k$ and $\mathbf{R}_c^k$ as defined above, the probability of absorption within $\tau$ transitions for a sequence assigned to the $c$th cluster is given by the $\tau$-term sum

$$\mathbf{B}_c^{k,(\tau)} = \mathbf{R}_c^k + \mathbf{Q}_c^k\mathbf{R}_c^k + \mathbf{Q}_c^k(\mathbf{Q}_c^k\mathbf{R}_c^k) + ... + \overbrace{\mathbf{Q}_c^k(\mathbf{Q}_c^k...(\mathbf{Q}_c^k(\mathbf{Q}_c^k}^{\tau-1}\mathbf{R}_c^k))...). \tag{3.28}$$

**Definition 3.4.2.** Let $m \in \mathbb{N}$, $X_1^m, X_2^m, ..., X_T^m \in \mathcal{S}$ be a sequence of random variables for the $m$th sequence. Then for $k < t \leq T_m$, the current state vector of length $k$ is the vector $\mathbf{x}_m(t \mid k) = (X_{t-k+1}^m, ..., X_t^m)$.

During the training process, we cluster sequence matrices based off the entire history of the sequence. Therefore, the sequence matrices will be based off of $\mathbf{x}_m(T_m)$. The point in the sequence where we make a prediction after $\tau$ transitions will be denoted by $t'$. During the testing process, the assigned probability of default is

dependent on the current and past $k$ states of the sequence at time $t'$. That is, dependent on $\mathbf{x}_m(t' \mid k)$.

Given the $m$th sequence that was assigned to the $c$th cluster, the probability of default within $\tau$ is then given by the entry $B_c^{k,(\tau)}[X^m_{t'-k+1}, ..., X^m_{t'}, X^m_{t'+\tau} = a] = B_c^{k,(\tau)}[\mathbf{x}_m(t' \mid k), a]$ where $a \in \mathcal{A}$. We will refer to this probability of absorption as the risk score of the sequence. During the K-fold cross-validation process, we assign a risk score to each sequence within the current test fold. By the time we reach the Kth fold of the cross-validation process, all sequences in the data set will have been assigned a risk score.

What follows is to assign the classification labels for each sequence. Since we are only interested in whether a sequence has been absorbed or not, we treat the absorption classification problem as a binary classification problem. The class labels are defined $\mathcal{L} = \{1, 0\}$ where 1 represents a sequence absorbed within $\tau$ transitions and 0 represents a sequence not absorbed within $\tau$ transitions. The decision for labelling the sequences is dependent on the threshold $\theta$. For a chosen threshold value $\theta$, the $m$th sequence can be assigned a label $y_m$ based on its risk score given by $\mathbf{B}_c^{k,(\tau)}$

$$\hat{y}_m(\theta) = \begin{cases} 1 & \text{if } B_c^{k,(\tau)}[\mathbf{x}_m(t' \mid k), a] > \theta, \\ 0 & \text{if } B_c^{k,(\tau)}[\mathbf{x}_m(t' \mid k), a] \leq \theta. \end{cases} \tag{3.29}$$

In other words, if the risk score is greater than this threshold, the sequence is predicted to default within $\tau$ transitions. If the risk score is less than this threshold, the sequence is predicted not to default within $\tau$ transitions.

### 3.4.2 User-Session Duration Prediction

For the user session duration prediction problem, we plan to predict the remaining length of a Web user's browsing session. Length here does not refer to time but a number of transitions that the user visits until they end their browsing session. To estimate this length we use the SBC model to compute the average remaining length until absorption. This predicted length will be compared to the actual remaining length until absorption.

As with 3.4.1, we use the results from Elayat (1973) to calculate the remaining length until absorption by any given absorbing state $a \in \mathcal{A}$. We again consider the sub-matrices $\mathbf{Q}_c^k$ and $\mathbf{R}_c^k$ of the representative transition matrix $\mathbf{P}_c^k$. The average remaining length until absorption can then be calculated by first considering the average number of times the process will be in a transient state. So, given by the following series

$$\mathbf{M}_c^{k,(n)} = \mathbf{I} + \mathbf{Q}_c^k + \mathbf{Q}_c^{k,(2)} + ... + \mathbf{Q}_c^{k,(n)}, \tag{3.30}$$

multiplying on the left side by $\mathbf{Q}_c^k$ we have

$$\mathbf{Q}_c^k \mathbf{M}_c^{k,(n)} = \mathbf{Q}_c^k + \mathbf{Q}_c^{k,(2)} + \mathbf{Q}_c^{k,(3)} + ... + \mathbf{Q}_c^{k,(n+1)} \tag{3.31}$$

and subtracting equation (3.30) by (3.31) results in

$$\mathbf{M}_c^{k,(n)} - \mathbf{Q}_c^k \mathbf{M}_c^{k,(n)} = \mathbf{I} - \mathbf{Q}_c^{k,(n+1)}. \tag{3.32}$$

Then, passing to the limit in equation (3.32) by taking $n \to \infty$ we obtain

$$\lim_{n \to \infty} \mathbf{M}_c^{k,(n)} = \mathbf{M}_c^k \tag{3.33}$$

and

$$\lim_{n\to\infty} \mathbf{Q}_c^{k,(n+1)} = 0, \tag{3.34}$$

which gives

$$\mathbf{M}_c^k - \mathbf{Q}_c^k \mathbf{M}_c^k = \mathbf{I}. \tag{3.35}$$

Again, because the associative law does not hold here we solve for the solution matrix $\mathbf{M}_c^k$ by writing equation (3.35) in vector notation, resulting in

$$\hat{\mathbf{Q}}_c^k \mathbf{m}_c^k = \mathbf{i}_c^k, \tag{3.36}$$

where $\hat{\mathbf{Q}}_c^k$ is a $(n-r)^{k+1} \times (n-r)^{k+1}$ square matrix, $\mathbf{m}_c^k = \text{vec}(\mathbf{M}_c^k)$ is a $(n-r)^{k+1} \times 1$ column vector, and $\mathbf{i}_c^k = \text{vec}(\mathbf{I})$ is a $(n-r)^{k+1} \times 1$ column vector defined by

$$\hat{\mathbf{Q}}_c^k = \begin{bmatrix} \hat{\mathbf{Q}}_{11} & \hat{\mathbf{Q}}_{12} & \cdots & \hat{\mathbf{Q}}_{1(n-r)} \\ \hat{\mathbf{Q}}_{21} & \hat{\mathbf{Q}}_{22} & \cdots & \hat{\mathbf{Q}}_{2(n-r)} \\ \vdots & \vdots & & \vdots \\ \hat{\mathbf{Q}}_{(n-r)1} & \hat{\mathbf{Q}}_{(n-r)2} & \cdots & \hat{\mathbf{Q}}_{(n-r)(n-r)} \end{bmatrix}, \tag{3.37}$$

such that

$$\hat{\mathbf{Q}}_{ij} = \mathbf{0} \quad \text{for } i \neq j \tag{3.38}$$

and

$$\hat{\mathbf{Q}}_{11} = \hat{\mathbf{Q}}_{22} = \cdots = \hat{\mathbf{Q}}_{(n-r)(n-r)}, \tag{3.39}$$

where $\hat{\mathbf{Q}}_{ll}$ is a square matrix of size $(n-r)^k \times (n-r)^k$. To form the main diagonal

block matrices we consider first partitioning the $l$th row $\mathbf{q}_l$ of $\mathbf{Q}_c^k$ by

$$\mathbf{Q}_c^k = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_{(n-r)^k} \end{bmatrix}, \tag{3.40}$$

then we define the permutation on $\mathbf{Q}_c^k$ by the transformation defined by the block matrix $\mathbf{T}$ of size $(n-r)^{k+1} \times (n-r)^k$. This matrix is first formed by considering $n-r$ identity matrices of size $(n-r)^k \times (n-r)^k$. The transformation matrix is then defined by

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_{(n-r)} \end{bmatrix}. \tag{3.41}$$

Before we apply this transformation we will form a new partition on $\mathbf{T}$ by forming $(n-r)^k$ blocks of size $(n-r) \times (n-r)^k$. In other words, the identity matrices themselves are further partitioned into $(n-r)^{k-1}$ blocks per identity matrix resulting in the matrix of the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_{(n-r)^k} \end{bmatrix}. \tag{3.42}$$

The main diagonal blocks $\hat{\mathbf{Q}}_{ll}$ can now be computed using block-matrix multiplication

in the following fashion

$$\hat{\mathbf{Q}}_{ll} = \mathbf{I} - \mathbf{Q}_c^k \mathbf{T} \tag{3.43}$$

$$= \mathbf{I} - \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_{(n-r)^k} \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \vdots \\ \mathbf{T}_{(n-r)^k} \end{bmatrix} \tag{3.44}$$

$$= \mathbf{I} - \begin{bmatrix} \mathbf{q}_1 \cdot \mathbf{T}_1 \\ \mathbf{q}_2 \cdot \mathbf{T}_2 \\ \vdots \\ \mathbf{q}_{(n-r)^k} \cdot \mathbf{T}_{(n-r)^k} \end{bmatrix} \tag{3.45}$$

for $l = 1, 2, ..., r$. Using the main diagonal blocks, we can now form $\hat{\mathbf{Q}}_c^k$ and equation (3.36). Therefore, the average number of times the sequence visits a transient state is given by

$$\mathbf{m}_c^k = (\hat{\mathbf{Q}}_c^k)^{-1} \mathbf{i}_c^k. \tag{3.46}$$

Finally, the remaining length until absorption is given by the vector $\tilde{\mathbf{m}}_c^k$ defined by

$$\tilde{\mathbf{m}}_c^k = \mathbf{M}_c^k \mathbf{1}', \tag{3.47}$$

where $\mathbf{1} = (1, 1, ..., 1)$ is a vector of ones of length $n - r$. In other words, each entry of $\tilde{\mathbf{m}}_c^k$ contains the corresponding row sum of $\mathbf{M}_c^k$.

## 3.5 Data

In this section we will describe the data used in this chapter. The experiments will be conducted using two different data sets, the credit rating transition data and Web-usage transition data. The behaviour of credit rating data is different from Web-usage data both in terms of the length of the sequence and frequency of transitions between different states. By testing our model against both credit rating and Web-usage data, we can observe the performance of the model across a variety of different data sets.

### 3.5.1 Credit Rating Data

The credit rating transition data used in this chapter was collected and provided by the National Information & Credit Evaluation Inc., a major bond-rating company in Korea. The data set consists of monthly corporate credit ratings from 1986-09-01 to 2018-09-01 for 1899 firms in Korean indices such as the KOSDAQ and KOSPI. As in Chapter 2, we will be considering quarterly data, in other words, we only consider the rating the firm received every quarter. Firms in this data set can take any rating from the following set of 22 credit ratings

$$\{\text{AAA, AA+, AA, AA-, A+, A, A-, BBB+, BBB, BBB-,}$$
$$\text{BB+, BB, BB-, B+, B, B-, CCC+, CCC, CCC-, CC, C, D}\}.$$

The firms that take the "D" rating are considered to be in default. A number of firms in the data set were "closed" after some time. With this in mind, we treat these firms to be in default. Firms that were missing credit rating sequences, made for sale, or were merged with another firm were removed from the data set. Finally, any firm

that transitioned out of the default state was removed. After processing the data set, 1648 firms remained in the data set.

| New Ratings | Old Ratings |
|---|---|
| AAA | {AAA} |
| AA | {AA+, AA, AA-} |
| A | {A+, A, A-} |
| BBB | {BBB+, BBB, BBB-} |
| BB | {BB+, BB, BB-} |
| B | {B+, B, B-, CCC+, CCC, CCC-, CC, C} |
| D | {D} |

Table 3.2: The 7 aggregated classes.

We mitigate the negative effects of the imbalanced data set on transition matrix estimation by combining similar categories together. The aggregation reduces the number of credit rating classes from 22 to 7 classes with 6 ratings being transient states and the default state representing the absorbing state. The aggregated classes are shown in Table 3.2.

The number of sequences that can be used in the SBC model is dependent on the order and input date $t'$ used. This is because from the point $t'$ we need to condition on an additional $k-1$ states before time $t'$. Therefore, not all sequences might contain enough transitions to condition on. In this study, we take $t'$ to be 2002-01-01. With further processing, the actual number of firms used for training and testing can be found in Table 3.3.

The number of positive (absorbed) and negative (not absorbed) labels will be dependent on $\tau$. We take $\tau \in \{12, 20, 40\}$. The distribution of the class labels can be found in Figure B.1a in Appendix B.2. It is clear from these distributions that the minority class is the default class for all values of $\tau$.

| Order | Number of Valid Sequences |
|---|---|
| 1 | 677 |
| 2 | 653 |
| 3 | 630 |
| 4 | 569 |

Table 3.3: The number of valid sequences used in the credit rating experiments given that $t'$ is January 1, 2002.

### 3.5.2 Web-Usage Data

The Web-usage data set we will be using is the Wikispeedia data set. The Wikispeedia data was collected from the Stanford Network Analysis Project (Leskovec & Krevl, 2014). This data set contains human Web browsing navigation paths on the website Wikipedia. The users that participated in this game were asked to navigate from a given source to a given target article by clicking only on the hyperlinks embedded throughout the Wikipedia article the user is currently browsing. There are a total of 76102 sequences. From the raw data of finished browsing sessions, there are a total of 15 transient states. Each transient state represents the topic of the Wikipedia Web page that is visited. At the end of the browsing session, users were asked to rate the difficulty of their run. This resulted in 7 absorption states labelled 16 to 22, with 16 representing users finishing their run without assigning a difficulty, 17 to 21 representing a run rated from very easy to brutal, and 22 representing an unfinished run. In practice, we are only concerned with whether a sequence has been absorbed or not. Hence, after removing sequences ending in state 22, we relabelled the absorbing state of the remaining sequences to be 16, representing the termination of the sequence. Additionally, we exclude sequences with less than 6 transitions and longer than 100, ultimately resulting in 37486 sequences remaining.

Again, the number of positive and negative (absorbed and not absorbed) labels

will be dependent on $\tau$. We take $\tau \in \{3, 5, 10, 15\}$. The distribution of the class labels can be found in Figure B.1b in Appendix B.2. Unlike in the credit rating data set, absorption appears to belong to the majority class and becomes even more so as $\tau$ increases. This is not unexpected as by increasing $\tau$, one would expect more sequences to terminate.

## 3.6 Results and Discussion

In this chapter there are a number of questions we are interested in. Will using high-order absorbing Markov chains prove more effective than first-order absorbing Markov chains? Will there be an increase in performance, if any, when we cluster the data set and model each cluster using high-order absorbing Markov chains? How does the high-order SBC model fare when applied to fundamentally different data sets such as the credit rating and Web-usage data sets?

Before presenting the results, we first provide an overview of the classification metrics used to evaluate the model using the credit rating and Wikispeedia data sets. Then, we present the results after applying the model for the credit rating and Wikispeedia data sets. Note that the SBC model will be often referred to as the clustering model while the use of the Markov model with no clustering present will be referred to as the benchmark model.

While collecting the results, we vary several parameters to explore the performance of the model. The first parameter we explore is the performance for different values of $\tau$. As noted earlier, for the credit rating and Wikispeedia data set, we use $\tau \in \{12, 20, 40\}$ and $\tau \in \{3, 5, 10, 15\}$, respectively. For each choice of $\tau$, we vary the clustering choice for the K-means algorithm, denoted by $c$ ranging from 1 to 20. For each clustering choice $c \in \{1, 2, ..., C\}$, we use K-fold stratified cross validation

where $K = 5$, to partition the data set into a training and testing set. To estimate the transition matrices using the MTDg model we use the EM algorithm described in Lèbre and Bourguignon (2008). Due to the stochastic nature in estimating the transition matrices we run an $M$ number of experiments for each $c$ chosen.

Using the credit rating data, we will shuffle the data set before applying cross-validation for each of the $M$ experiments. Using the Wikispeedia data set, we will sample 3000 sequences without replacement from the set of valid sequences. The reported performance measures found in the tables are presented using the best clustering choice $c$ and are averaged across $M$ number of experiments with the respective standard deviation presented in parentheses. We choose $M = 10$. The algorithms used can be found in Appendix B.

### 3.6.1 Performance Measures

When evaluating whether absorption takes place within $\tau$ steps, we evaluate the model using the binary F1, accuracy, precision, and recall. We define the class label set by $\mathcal{L} = \{1, 0\}$ where 1 represents the absorption of a sequence within $\tau$ transitions and 0 represents the non-absorption of sequence within $\tau$ transitions. The state space and absorbing set using the credit rating data is $\mathcal{S} = \{1, ..., 7\}$ and $\mathcal{A} = \{7\}$ while the state space and absorbing set using the Wikispeedia data is $\mathcal{S} = \{1, ..., 15\}$ and $\mathcal{A} = \{16\}$. To begin the evaluation of our model, we need to count the number of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) predictions. These counts can be organized into a confusion matrix.

| | | Predicted | |
|---|---|---|---|
| | | Absorbed | Not Absorbed |
| Actual | Absorbed | TP | FN |
| | Not Absorbed | FP | TN |

Given the confusion matrix, we can calculate our desired classification metrics. The accuracy is calculated as

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{3.48}$$

In the case of binary classification, the accuracy may not be a good metric when the data set is imbalanced and this may result in bias during evaluation, for example, for a highly imbalanced data set, a high accuracy can be achieved by predicting only the majority class. Thus, we also report the binary F1 for absorption prediction within $\tau$ steps. The binary F1 can be calculated using the following equation for the positive class

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} = \frac{2\text{Pr} \cdot \text{Re}}{\text{Pr} + \text{Re}}, \tag{3.49}$$

where Pr and Re denote the precision and recall, respectively. By using the F1, high counts of the TNs will not be able to hide low counts of TP. The F1 also considers the balance between the precision and recall and not only the number of correct predictions that are made, further reducing the impact of the class imbalance. The

precision (Pr) and recall (Re) are calculated as

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{3.50}$$

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \mathrm{TPR}. \tag{3.51}$$

Therefore, the precision is the number of correctly predicted positive cases out of the total number of predicted positives. The recall is the number of correctly predicted positive cases out of the total number of actually positive cases. This is also known as the hit rate. Depending on the context, one might prioritize one measure over another.

To begin populating the confusion matrix, we need to compare the estimated probabilities against a threshold level $\theta$ for classification. To determine a reasonable threshold level, we make use of the receiver operating characteristic (ROC) curve. The ROC curve is a convenient representation of the model performance for all levels of threshold, constructed by plotting the true positive rate (TPR) against the false positive rate (FPR) while varying the threshold between 0 and 1. The TPR is equal to the recall while the FPR is defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{3.52}$$

The optimal $\theta$ can be chosen based on the Youden's index $J(\theta)$. The Youden's index is defined in terms of the TPR and FPR as

$$J(\theta) = \mathrm{TPR}(\theta) - \mathrm{FPR}(\theta) \tag{3.53}$$

and represents the trade off between the true positive rate and true negative rate.

The optimal threshold $\hat{\theta}$ is chosen such that $J$ is maximized.

For each of the classification scenarios, we will be aggregating confusion matrix counts across the K folds. From this aggregated count, we calculate the performance metrics. Forman and Scholz (2010) noted that this is the most unbiased method in computing the F1-measure as defined in equation (3.49) when there is a high class imbalance.

To evaluate the performance of the model for predicting the duration of the user-session, we compute the error between the expected remaining length and the true remaining length to absorption. That is, we evaluate the performance using the mean absolute error (MAE) and the mean absolute error per sequence (MAEPS). Let $M$ be the total number of sequences in the testing set. Then the MAE is calculated as

$$\text{MAE} = \frac{\sum_{m=1}^{M} \sum_{t>k}^{T_m-1} |\tilde{\mathbf{m}}_c^k[\mathbf{x}_m(t \mid k)] - \tilde{\mathbf{x}}_m(t)|}{\sum_{m=1}^{M} T_m - 1}, \tag{3.54}$$

where $\tilde{\mathbf{x}}_m(t) = T_m - t$, the length of the $m$th sequence at time $t$. The MAEPS can be calculated by

$$\text{MAEPS} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{T_m - 1} \sum_{t>k}^{T_m-1} |\tilde{\mathbf{m}}_c^k[\mathbf{x}_m(t \mid k)] - \tilde{\mathbf{x}}_m(t)|. \tag{3.55}$$

### 3.6.2 Model Performance in Default Prediction

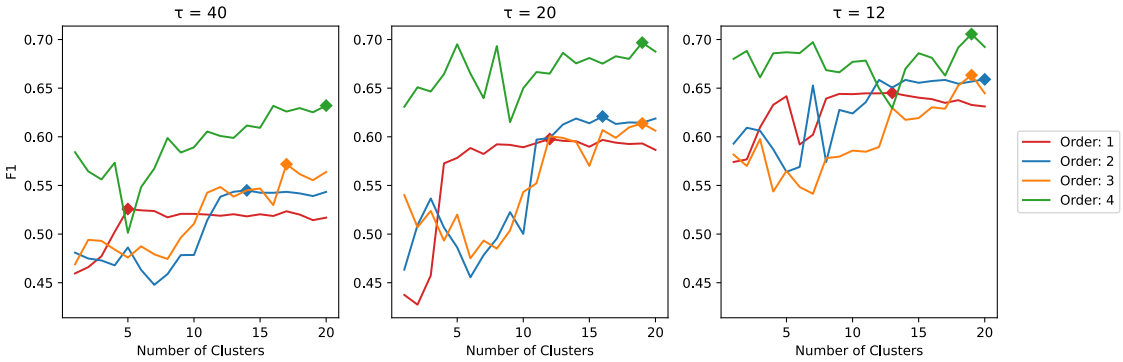
In the default classification scenario, we tested the model's ability to predict whether a sequence will transition into default within $\tau = 40, 20$, and 12 quarters or $10, 5$, and 3 years, respectively. The plots of the F1 score and accuracy against the number of clusters in Figure 3.2 show that as the number of clusters increases, the F1 score and accuracy also increase. Comparing the y-axis of the respective subplots, we can observe based on the initial position of the curves that as $\tau$ decreases, the overall

performance of the F1 score and accuracy tend to increase regardless of the number of clusters. This effect is especially more pronounced when looking at the accuracy. While a general trend of performance improvement is observed as the number of clusters increases, the degree of improvement also appears to depend on the Markov model's order. For example, we can observe a large increase in the performance when considering the first-order models as the number of clusters increase. On the other hand, this level of improvement is not as pronounced in the 4th order models. In fact, the performance of the 4th order model appears to remain high regardless of the number of clusters used in this case. That is, we find that the 4th order model shows the greatest performance while clustering provides marginal benefits.

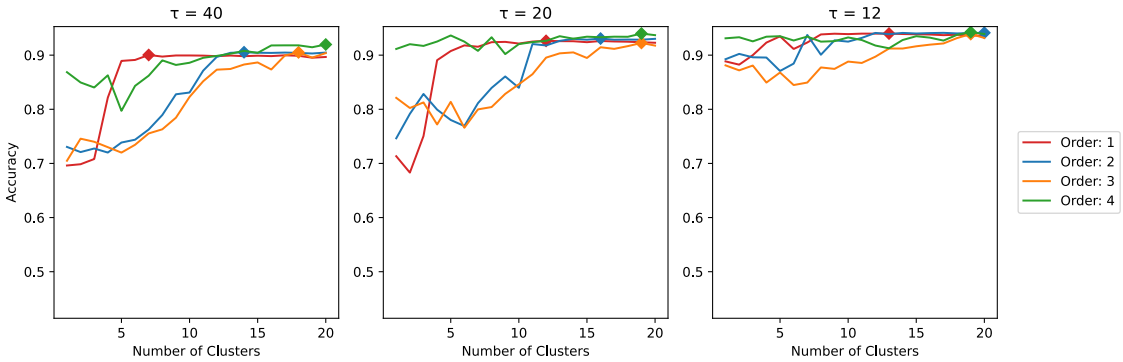
However, it should be noted that to achieve a good level of performance, it may not be necessary to use the highest order model. For example, when $\tau = 20$, similar levels of accuracy achieved using the 4th order model can also be achieved by opting to use the 1st order model with around 12 clusters instead. It appears that SBC is a method that permits this level of flexibility, offering a more parsimonious alternative in this classification problem.

Considering the best performance of the model across different orders, values of $\tau$, and clustering choice, we tabulate the results in Table 3.4 and Table 3.5. When making comparisons between the respective performance metrics, we utilize the t-test for hypothesis testing, where statistical significance is reported using three different alpha levels $\alpha = 0.1, 0.05$, and $0.001$. That is, statistical significance is reported when the p-value $p$ is less than or equal to $\alpha$, with increasing significance for more strict levels of $\alpha$.

Comparing the F1 score of the high-order benchmark models over the first-order benchmark model, we found the greatest improvement of $27\%, 44\%$, and $18\%$ when $\tau = 40, 20$, and 12 was found using the 4th-order model, respectively. When




(a) F1 score




(b) Accuracy

Figure 3.2: The F1 score and accuracy plotted against the number of clusters used. The diamond markers represent the greatest score achieved across the different number of clustering choices.

considering the accuracy, we found the greatest improvement of 25%, 28%, and 4.8% when $\tau = 40, 20$, and 12 using the 4th-order model, respectively. While in both cases, we found a statistically significant level of improvement for all high-order clustering models over the first-order benchmark model, the greatest improvement can be found when using the 4th-order clustering or benchmark models, as noted.

While the high-order benchmark models appear to show promise, the addition of clustering can further enhance the predictive accuracy. Comparing the F1 score

of the high-order clustering models over the first-order benchmark model, we found the greatest improvement using the 4th-order clustering model with an improvement of $38\%, 60\%$, and $23\%$ when $\tau = 40, 20$, and 12, respectively. When considering the accuracy, we found the greatest improvement of $32\%, 32\%$, and $6.1\%$ in the accuracy when $\tau = 40, 20$, and 12, respectively. Again, the improvements found here are statistically significant for all orders over the first-order benchmark model. Ultimately, the 4th order model dominates in terms of performance, but SBC also provides an appreciable boost to the performance of the model in terms of F1 and accuracy over the non-clustering model.

| $\tau$ | Order | (C) F1 | (B) F1 | (C) Accuracy | (B) Accuracy |
|---|---|---|---|---|---|
| 40 | 1 | 0.5258 (0.0185)*** | 0.4596 (0.0093) | 0.9003 (0.0022)*** | 0.6962 (0.0137) |
| | 2 | 0.5449 (0.0030)*** | 0.4809 (0.0130)*** | 0.9054 (0.0011)*** | 0.7305 (0.0228)** |
| | 3 | 0.5718 (0.0183)*** | 0.4689 (0.0162) | 0.9051 (0.0045)*** | 0.7049 (0.0315) |
| | 4 | **0.6321 (0.0287)***** | 0.5842 (0.0651)*** | **0.9197 (0.0125)***** | 0.8685 (0.0545)*** |
| 20 | 1 | 0.5978 (0.0029)*** | 0.4375 (0.0246) | 0.9264 (0.0009)*** | 0.7133 (0.0414) |
| | 2 | 0.6209 (0.0040)*** | 0.4635 (0.0458)* | 0.9306 (0.0010)*** | 0.7464 (0.0582) |
| | 3 | 0.6138 (0.0205)*** | 0.5400 (0.0914)** | 0.9225 (0.0071)*** | 0.8211 (0.0867)** |
| | 4 | **0.6968 (0.0358)***** | 0.6309 (0.0442)*** | **0.9399 (0.0106)***** | 0.9114 (0.0227)*** |
| 12 | 1 | 0.6452 (0.0044)*** | 0.5741 (0.0414) | 0.9399 (0.0012)** | 0.8885 (0.0375) |
| | 2 | 0.6591 (0.0086)*** | 0.5930 (0.0415) | 0.9413 (0.0023)** | 0.8925 (0.0268) |
| | 3 | 0.6634 (0.0141)*** | 0.5817 (0.0425) | 0.9389 (0.0042)** | 0.8811 (0.0332) |
| | 4 | **0.7056 (0.0453)***** | 0.6801 (0.0725)** | **0.9425 (0.0137)**** | 0.9311 (0.0270)** |

Table 3.4: The F1 score and accuracy from the credit rating default prediction classification scenario. Values under the column label (C) and (B) represent the clustering and the benchmark model respectively. Bold faced values correspond to the highest level of performance for each metric. Values appended with *, **, and *** represent a statistically significant improvement over the first-order benchmark model at $p \leq 0.1$, $p \leq 0.05$, and $p \leq 0.001$, respectively.

Comparing the precision of the high-order benchmark models over the first-order benchmark model, we found the greatest improvement of $63\%, 87\%$, and $27\%$

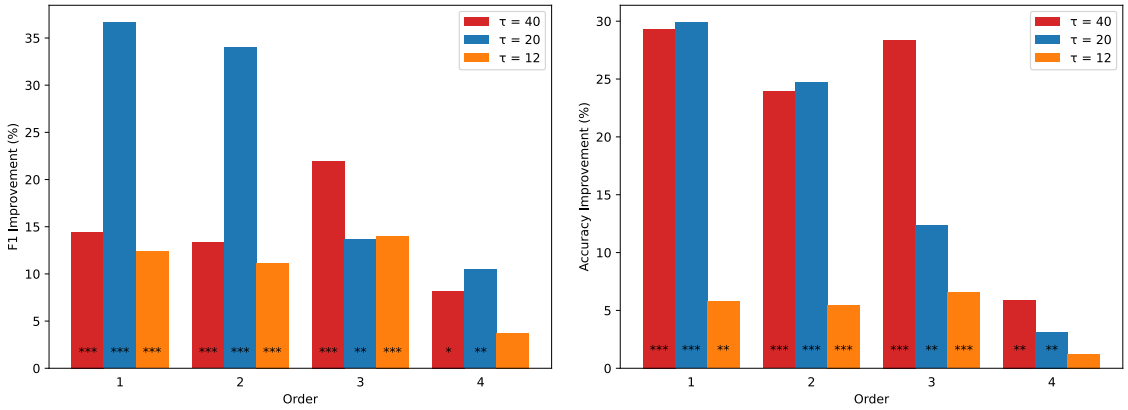
when $\tau = 40, 20$, and 12, respectively. It should be noted that while high-order benchmark models are superior to the first-order benchmark model, not all 4th-order models performed the best. In some cases, the first-order clustering models, offered significantly superior performance. That is, there are an improvement of $211\%, 232\%$, and $84\%$ when $\tau = 40, 20$, and 12, for $k = 1, 2$, and 1 respectively. With respect to the recall, we found fewer instances where clustering or high-order models offered statistically significant levels of improvement. Only the 3rd-order benchmark models outperformed the first-order benchmark when $\tau = 40$ and 12. We found the greatest improvement of $3.8\%, 6.7\%$, and $23\%$ in the accuracy when $\tau = 40, 20$, and 12, for $k = 3, 1$, and 3 respectively. Therefore, SBC may be able to yield significant levels of improvement if precision is a concern.

| $\tau$ | Order | (C) Precision | (B) Precision | (C) Recall | (B) Recall |
|---|---|---|---|---|---|
| 40 | 1 | **0.9864 (0.0409)**$^{***}$ | 0.3165 (0.0097) | 0.8654 (0.0469)$^{*}$ | 0.8404 (0.0259) |
| | 2 | 0.9792 (0.0193)$^{***}$ | 0.3393 (0.0168)$^{**}$ | 0.8388 (0.0277) | 0.8296 (0.0335) |
| | 3 | 0.8530 (0.0753)$^{***}$ | 0.3187 (0.0221) | 0.8725 (0.0506)$^{*}$ | **0.8978 (0.0543)**$^{**}$ |
| | 4 | 0.7352 (0.1163)$^{***}$ | 0.5186 (0.1238)$^{***}$ | 0.8058 (0.0674) | 0.7159 (0.0818) |
| 20 | 1 | 0.9791 (0.0155)$^{***}$ | 0.2943 (0.0299) | **0.9267 (0.0459)**$^{**}$ | 0.8686 (0.0541) |
| | 2 | **0.9897 (0.0171)**$^{***}$ | 0.3246 (0.0581) | 0.8024 (0.0685) | 0.8476 (0.0632) |
| | 3 | 0.7819 (0.0819)$^{***}$ | 0.4617 (0.1765)$^{**}$ | 0.8421 (0.0807) | 0.7750 (0.1306) |
| | 4 | 0.7087 (0.0977)$^{***}$ | 0.5512 (0.0745)$^{***}$ | 0.7750 (0.0376) | 0.7518 (0.0361) |
| 12 | 1 | **0.9818 (0.0201)**$^{***}$ | 0.5349 (0.0764) | 0.6701 (0.0721) | 0.6403 (0.0680) |
| | 2 | 0.9674 (0.0356)$^{***}$ | 0.5397 (0.0834) | 0.6946 (0.0623) | 0.6757 (0.0472) |
| | 3 | 0.8416 (0.0546)$^{***}$ | 0.4895 (0.0675) | **0.7870 (0.0667)**$^{**}$ | 0.7348 (0.0472)$^{**}$ |
| | 4 | 0.7304 (0.1332)$^{**}$ | 0.6831 (0.1878)$^{**}$ | 0.7655 (0.0426)$^{***}$ | 0.7164 (0.0489)$^{**}$ |

Table 3.5: The precision and recall from the credit rating default prediction classification scenario.

While SBC does provide an appreciable boost to the performance of the Markov models for the F1 score, accuracy, and precision, the degree of improvement varies depending on the order of the model and $\tau$. From the plots in Figure 3.3, we observed

that lower order Markov models experienced the greatest benefit from the introduction of SBC. In fact, the level of improvement between the clustering model compared to benchmark model when $k = 4$, was measured to be lower in magnitude and even not statistically significant for $\tau = 12$. The value of $\tau$ also appeared to play a role in the degree of improvement, with the most significant boost being observed when $\tau = 20$ for the F1 score, after SBC was introduced. In general, when $\tau = 12$, the degree of performance increase is reduced. This suggests that SBC may be more beneficial for longer-term predictions even with the introduction of higher-order Markov models. This suggests that if long-term credit risk assessment is a concern, then a combination of high-order Markov chains and SBC could prove beneficial. Whereas, for shorter-term credit risk assessment, first-order Markov chains with SBC may be sufficient. This observation appears to be consistent with the results in Le et al. (2021).




(a) F1 Improvement

(b) Accuracy Improvement

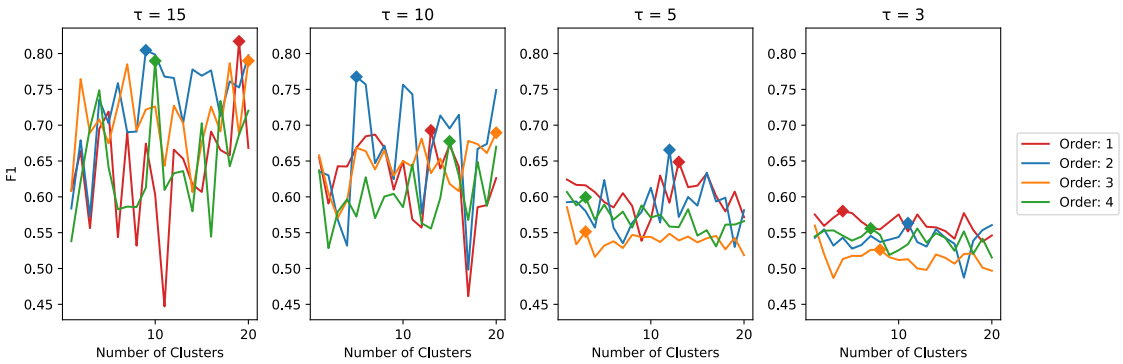
Figure 3.3: The improvement in performance of the clustering model over the benchmark model with respect to the F1 score and accuracy for each order and $\tau$ using the credit rating data set. Bars are annotated with *, **, and *** and represent a statistically significant improvement over the respective benchmark model at $p \leq 0.1$, $p \leq 0.05$, and $p \leq 0.001$, respectively.

Therefore, both high-order Markov chains and clustering can be used to maximize default prediction performance. If computing resources are a concern or if the problem state space is too large, using the first-order Markov chains with SBC may be a viable alternative. By opting to use the alternative, clustering combined with lower-order Markov chains, we can avoid the process of estimating high-order transition matrices, which is computationally intensive and requires significant storage capacity. For example, for $\tau = 20$, a second-order clustering model provides nearly the same level of performance as a 4th-order benchmark model.

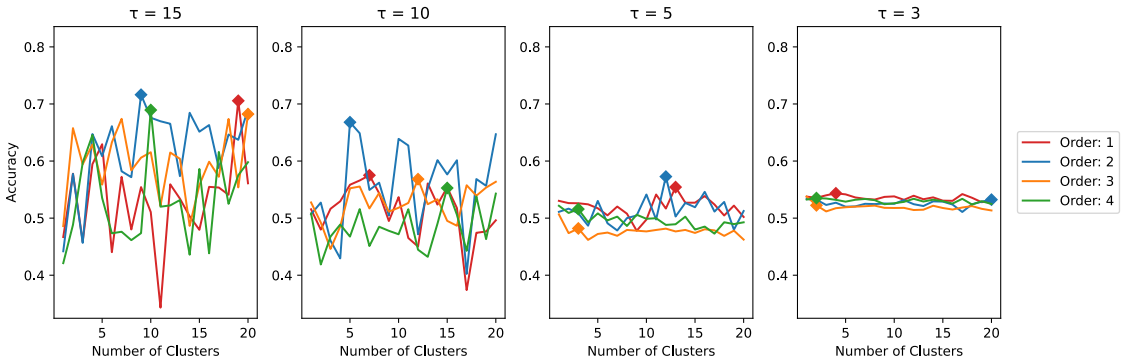
### 3.6.3 Model Performance in Web-Usage Mining

Using the Wikispeedia data, we first test the model's ability to predict whether a sequence will terminate within $\tau = 15, 10, 5$, and 3 steps. The plots of the F1 score and accuracy against the number of clusters in Figure 3.4 show that for $\tau > 5$, as the number of clusters increases, the F1 score and accuracy also increase. For shorter $\tau$ lengths, we observed a marginal increase in the utility when using SBC. This is a similar observation in the credit rating case. In addition, the higher-order Markov models did not appear to dominate as greatly as it was observed in Figure 3.2 for short $\tau$. Unlike in the credit rating case however, for $\tau > 5$, the 2nd-order clustering model appears to provide a slightly greater level of performance for the majority of clustering choices.

For the duration prediction scenario, we can see in Figure 3.5 that as the number of clusters increases, the MAE and MAEPS are reduced for all choices of $k$. That is, as the order of the model increases, we find that the performance is dominated by higher-order models in both the MAE and MAEPS regardless of the number of clusters used.




(a) F1 Score

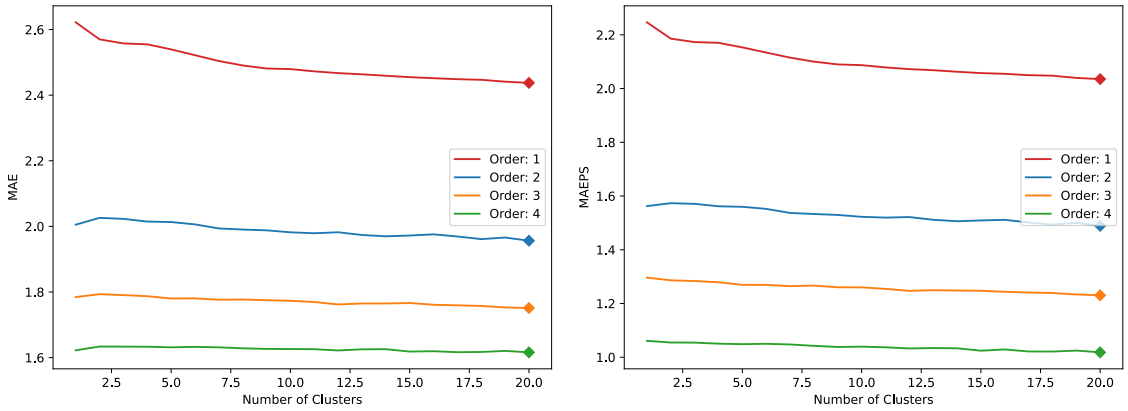



(b) The accuracy

Figure 3.4: The F1 score and accuracy as the number of clusters increase when predicting the termination of a Web session within $\tau$ steps.

Considering the best performance of the model across different orders, values of $\tau$, and clustering choices, we tabulate the results in Table 3.6 and Table 3.7. As with the credit rating data, when making comparisons between the respective performance metrics, we utilize the t-test for hypothesis testing with the same levels of $\alpha$ used.

Comparing the F1 score and accuracy of the high-order benchmark models over the first-order benchmark model, we measured no improvement that was statistically significant. However, the F1 score and accuracy of the high-order clustering models do show some promise over the first-order benchmark model when $\tau > 5$. We observed




(a) The MAE (b) The MAEPS

Figure 3.5: The MAE and MAEPS as the number of clusters increase when estimating the remaining length of a Web session.

an improvement of 34% and 17% in the F1 score for the clustering model with $\tau = 15$ when $k = 1$ and 2, respectively. With respect to the accuracy, we observed an improvement of $53\%, 30\%$, and $1\%$ using the clustering model for $\tau = 15, 10$, and 3, when $k = 2, 2$, and 1, respectively.

As with the F1 score and accuracy, in Table 3.7, the greater statistically significant performance was generally observed in the clustering model over the first-order benchmark, with respect to the precision and recall. However, we note that the magnitude of the performance increase is low overall. The most notable increase in performance is observed when considering the recall. We observed an improvement of 59% and 35% in the recall when $\tau = 15$ and 10, using the 2nd-order clustering model over the first-order benchmark model. It appears that while for the credit rating data set, one can find improvement in both the clustering and benchmark models, in other cases such as the Wikispeedia data set, a combination of clustering and high-order models may be necessary to see any improvement in performance.

In Figure 3.6, we compare the performance of the clustering model to the perfor-

| Tau | Order | (C) F1 | (B) F1 | (C) Accuracy | (B) Accuracy |
|---|---|---|---|---|---|
| 15 | 1 | **0.8170 (0.0856)**** | 0.6087 (0.1479) | 0.7056 (0.1132)*** | 0.4669 (0.1244) |
| | 2 | 0.8046 (0.2011)* | 0.5838 (0.1394) | **0.7162 (0.2011)**** | 0.4419 (0.1169) |
| | 3 | 0.7898 (0.1455)** | 0.6088 (0.2059) | 0.6823 (0.1805)** | 0.4860 (0.2106) |
| | 4 | 0.7897 (0.1756)* | 0.5381 (0.2159) | 0.6894 (0.2018)** | 0.4212 (0.2089) |
| 10 | 1 | 0.6927 (0.0899) | 0.6549 (0.0370) | 0.5751 (0.1694) | 0.5155 (0.0371) |
| | 2 | **0.7675 (0.1649)*** | 0.6353 (0.1178) | **0.6680 (0.1822)**** | 0.5068 (0.1024) |
| | 3 | 0.6895 (0.1194) | 0.6579 (0.0993) | 0.5683 (0.1797) | 0.5275 (0.1175) |
| | 4 | 0.6775 (0.1298) | 0.6370 (0.1119) | 0.5528 (0.1298) | 0.5082 (0.1188) |
| 5 | 1 | 0.6487 (0.1049) | 0.6242 (0.0277) | 0.5544 (0.0763) | 0.5303 (0.0181) |
| | 2 | **0.6655 (0.1223)** | 0.5925 (0.0609) | **0.5728 (0.0921)** | 0.5110 (0.0335) |
| | 3 | 0.5513 (0.0373) | 0.5857 (0.0580) | 0.4819 (0.0227) | 0.5066 (0.0325) |
| | 4 | 0.5994 (0.0552) | 0.6068 (0.0629) | 0.5158 (0.0358) | 0.5217 (0.0393) |
| 3 | 1 | **0.5801 (0.0235)** | 0.5754 (0.0290) | **0.5435 (0.0071)*** | 0.5377 (0.0093) |
| | 2 | 0.5633 (0.0853) | 0.5429 (0.0378) | 0.5324 (0.0161) | 0.5335 (0.0109) |
| | 3 | 0.5262 (0.0476) | 0.5600 (0.0251) | 0.5224 (0.0228) | 0.5378 (0.0066) |
| | 4 | 0.5560 (0.0363) | 0.5448 (0.0313) | 0.5352 (0.0094) | 0.5326 (0.0114) |

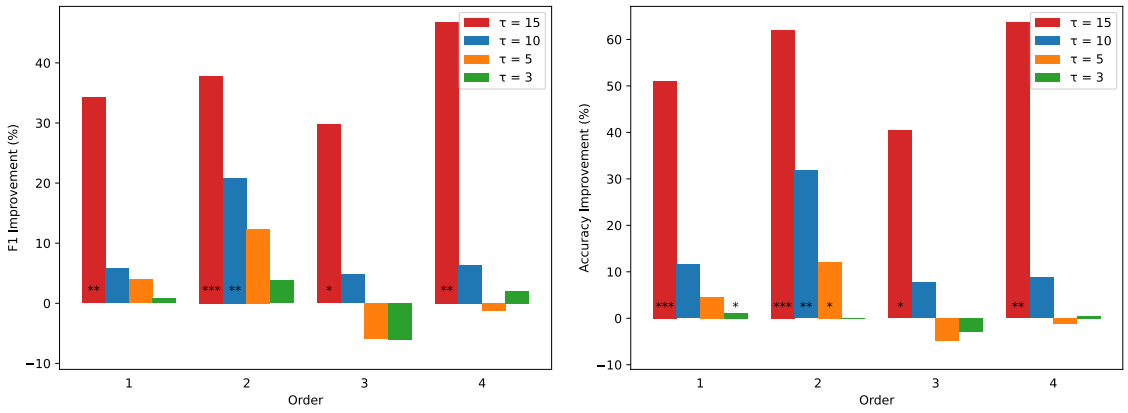
Table 3.6: The F1 score and accuracy when predicting when a Web session will terminate within $\tau$ steps.

mance of the benchmark model for each $k$ and $\tau$. Unlike in the credit rating data, we observe that the 2nd-order Markov models experienced the greatest benefit from the introduction of SBC, providing a slightly greater performance boost than the 1st or 3rd-order clustering. As with the credit rating data, it appears that the performance boost gained using SBC in terms of the F1 score and accuracy, is dependent on whether we are forecasting long-term or shorter-term changes. In this case, the majority of the performance enhancement provided by SBC appears only when $\tau > 5$. In fact, for $\tau \leq 5$ we find that there is actually some decrease in performance, as in the case when $k = 3$.

Considering the remaining length prediction scenario, the results found in Table 3.8 suggest that using high-order transition matrices can lead to a significant reduction in the error between the actual and predicted duration over using first-order transition matrices. Without clustering, the 4th-order model achieves the greatest reduction,

| $\tau$ | Order | (C) Precision | (B) Precision | (C) Recall | (B) Recall |
|---|---|---|---|---|---|
| 15 | 1 | **0.9681 (0.0130)**$^{**}$ | 0.9588 (0.0085) | 0.7202 (0.1220)$^{***}$ | 0.4614 (0.1360) |
| | 2 | 0.9632 (0.0077)$^{**}$ | 0.9610 (0.0091) | **0.7321 (0.2233)**$^{**}$ | 0.4332 (0.1301) |
| | 3 | 0.9649 (0.0070)$^{**}$ | 0.9617 (0.0098)$^{**}$ | 0.6955 (0.1958)$^{**}$ | 0.4824 (0.2340) |
| | 4 | 0.9669 (0.0130)$^{**}$ | 0.9639 (0.0090)$^{**}$ | 0.7052 (0.2192)$^{**}$ | 0.4093 (0.2308) |
| 10 | 1 | 0.9227 (0.0122)$^{**}$ | 0.9131 (0.0119) | 0.5835 (0.2105) | 0.5120 (0.0459) |
| | 2 | 0.9200 (0.0151)$^{**}$ | 0.9151 (0.0110) | **0.6961 (0.2246)**$^{**}$ | 0.4994 (0.1311) |
| | 3 | 0.9219 (0.0155)$^{**}$ | 0.9150 (0.0132) | 0.5742 (0.2205) | 0.5276 (0.1519) |
| | 4 | **0.9230 (0.0186)**$^{**}$ | 0.9147 (0.0137) | 0.5561 (0.1621) | 0.5039 (0.1521) |
| 5 | 1 | 0.7802 (0.0147) | 0.7776 (0.0171) | 0.5808 (0.1582) | 0.5223 (0.0359) |
| | 2 | 0.7805 (0.0144) | 0.7822 (0.0167)$^{*}$ | **0.6121 (0.1847)** | 0.4816 (0.0717) |
| | 3 | 0.7878 (0.0192)$^{**}$ | 0.7838 (0.0177)$^{**}$ | 0.4276 (0.1248) | 0.4719 (0.0701) |
| | 4 | **0.7918 (0.0218)**$^{**}$ | 0.7822 (0.0180)$^{*}$ | 0.4902 (0.0742) | 0.5017 (0.0851) |
| 3 | 1 | 0.6336 (0.0210)$^{**}$ | 0.6248 (0.0170) | **0.5423 (0.0965)** | 0.5361 (0.0515) |
| | 2 | 0.6316 (0.0238)$^{*}$ | 0.6386 (0.0159)$^{***}$ | 0.5401 (0.1562) | 0.4749 (0.0535) |
| | 3 | 0.6363 (0.0190)$^{**}$ | 0.6356 (0.0191)$^{**}$ | 0.4649 (0.1163) | 0.5024 (0.0425) |
| | 4 | **0.6410 (0.0172)**$^{**}$ | 0.6364 (0.0200)$^{***}$ | 0.5005 (0.0567) | 0.4782 (0.0454) |

Table 3.7: The precision and recall when predicting when a Web session will terminate within $\tau$ steps.




(a) F1 Improvement

(b) Accuracy Improvement

Figure 3.6: The improvement in performance of the clustering model over the benchmark model with respect to the F1 score and accuracy for each order and $\tau$ using the Wikispeedia data set.

reducing the MAE and MAEPS by 31% and 46%, respectively. Comparing the effectiveness of high-order Markov models over first-order Markov models, with

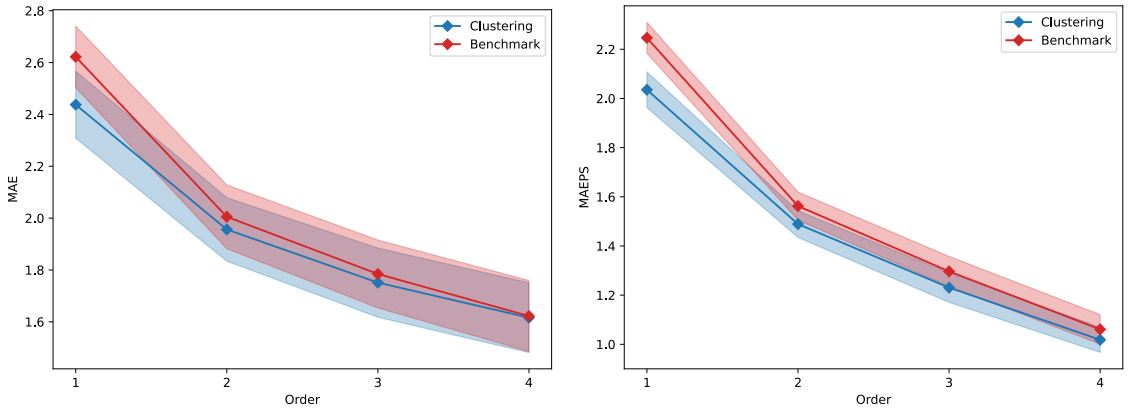
| Order | (C) MAE | (B) MAE | (C) MAEPS | (B) MAEPS |
|---|---|---|---|---|
| 1 | 2.4375 (0.1292)*** | 2.6221 (0.1187) | 2.0353 (0.0726) | 2.2465 (0.0631) |
| 2 | 1.9563 (0.1224)*** | 2.0052 (0.1229)*** | 1.4889 (0.0540)*** | 1.5624 (0.0569)*** |
| 3 | 1.7512 (0.1334)*** | 1.7845 (0.1308)*** | 1.2306 (0.0598)*** | 1.2965 (0.0611)*** |
| 4 | **1.6161 (0.1345)***** | 1.6224 (0.1371)*** | **1.0183 (0.0509)***** | 1.0610 (0.0599)*** |

Table 3.8: The MAE and MAEPS for the remaining length prediction.

clustering, the greatest reduction is achieved using the 4th-order model, with a reduction of 31% and 45% in the MAE and MAEPS respectively. From Table 3.8, a clear reduction in the MAE and MAEPS is observed for both the clustering and benchmark case as the order increases.

In Figure 3.7, the degree of improvement of the clustering model over the benhmark model is presented for the MAE and MAEPS. While the level of reduction when using the SBC model is superior to the benchmark regardless of the order of the Markov model, the level of improvement appears to decrease as the order increases for the MAE. The level reduction in error for both the MAE and MAEPS appears to decrease for orders greater than 1.

As in the credit rating context for the F1 score and accuracy, the overall performance is greater when clustering is used regardless of the order of the transition matrices used compared to the benchmark model. While the performance of the clustering case is overall greater than the benchmark case, the first-order clustering model still shows quite a substantial increase in performance. This suggests that it may be beneficial to seriously consider solely using the SBC model with first-order transition matrices instead of using solely using first-order or higher-order transition matrices. As in the credit rating context, the benefit of this alternative is that one can avoid the more computationally demanding and data-hungry process of estimating high-order transition matrices. This gain from using high-order transition matrices




(a) The MAE

(b) The MAEPS

Figure 3.7: The improvement in MAE and MAEPS of the clustering model over the benchmark model. The shaded region represents the standard deviation of the respective model.

in the remaining length prediction scenario however is large enough to warrant the consideration of using this as the gain in performance is quite significant.

### 3.6.4 Summary of Results

In the credit rating context, we find that clustering results in improved performance for the F1 score and accuracy regardless of the order. For the precision and recall, more consideration may be necessary in the choice of cluster and model order. In general, using high-order Markov chains with clustering outperforms the first-order benchmark case. We observe an overall increase in performance as $\tau$ decreases; however, at the same time, we also observe diminishing returns on the benefits of using high-order Markov models.

In the Web-usage context, for absorption within $\tau$ steps prediction, we find that generally, clustering results in improved performance over the benchmark models for $\tau > 5$. As the order of the transition matrices used increases, we generally observed

the performance decreasing, except for some choices of model order for $\tau > 5$. For the remaining length prediction, while clustering results in some improvement with respect to the MAE and MAEPS, we find that the greatest impact in reducing the MAE and MAEPS was observed when we introduced high-order Markov models, implying an improvement in accuracy when estimating the average duration of user Web sessions. This observation can be seen with similar degrees of improvement in both the clustering and benchmark models as the order increases. Overall, the results show that depending on the data set and context, extending the first-order SBC model can further enhance the quality of transition matrix estimation in the high-order case. That is, using SBC in conjunction with higher-order Markov models can be beneficial in providing improved prediction accuracy of future behaviour both in credit rating and Web-usage sequences.

## 3.7 Conclusion

Historical credit rating sequences and Web-usage sequences embody a wealth of information that can provide valuable insights into the future behaviour of the process. In some cases, it is sufficient to only consider the present state of a process to infer future behaviour. However, in other cases, the future behaviour of a process may also depend on events prior to the present. Using high-order Markov models, one can capture not only the immediate state dependencies but also previous state dependencies as well. The purpose of the work done in this chapter was to determine if high-order Markov models augmented with SBC can offer superior predictive capabilities compared to traditional first-order Markov models. This new model was validated against benchmark models where no clustering was present. Two real world data sets were used, a Korean historical credit rating sequence data set, and a

Wikispeedia browsing data set. The credit rating data was used to assess our model's performance in predicting defaults. The Wikispeedia data set was used to evaluate our model's performance in making accurate predictions of absorption within $\tau$ steps and the remaining length of user sessions. This work extends the first-order SBC model presented in Le et al. (2021) and Park et al. (2023).

We contribute to the literature on high-order Markov chains and SBC in the context of credit rating and Web-usage mining in a number of ways. We find that high-order Markov chains, with and without clustering, can accurately estimate the default probability of a firm given its credit rating history. To the best of our knowledge, we are the first to validate the use of high-order Markov models with SBC using real world data sets. The usefulness of our model in the context of credit risk and Web-usage mining is evident from its ability to improve upon more traditional first-order models. The clustering algorithm used was the K-means algorithm, a non-parametric unsupervised clustering algorithm. As a result, we are not required to make restrictive assumptions about the data itself. For the credit rating data, SBC allows us to partition sequences with similar dynamics together, capturing non-Markovian behaviour such as rating momentum.

We assume that transitions are dependent on more than the current state by employing high-order Markov chains. In doing so, we further capture rating momentum within the clusters themselves. For practitioners concerned about capturing these credit rating characteristics in default prediction, first-order SBC model may be a viable alternative over strictly using high-order Markov chains as SBC allows us to capture some of these unique characteristics. Using SBC in this fashion allows practitioners to forgo the high data cost of estimating high-order transition matrices while maintaining respectable levels of performance over just using a first-order Markov model. If resources permit, practitioners may opt to use a combination of SBC and

high-order Markov chains of their desired order in order to maximize performance gains.

With respect to Web-usage mining, web developers may find SBC useful in improving website design, ad placement during a browsing session, and recommendation systems as SBC can improve the prediction of when a user might end their browsing session. Furthermore, we find that high-order Markov chains, with and without clustering, can model the remaining length of Web browsing behaviour well. This feature is valuable when developers are concerned about the remaining length of a session. For example, in virtual queue time estimation, redesigning website structure to minimize the length of time a user might take to make a purchase, as well as ad placements.

# Chapter 4

# Reducing Systemic Risk using Reinforcement Learning

In this chapter, we introduce a novel framework to assess and manage systemic risk in a multi-layer financial network by taking advantage of reinforcement learning (RL). The reduction of systemic risk in the financial network is achieved by applying the deep deterministic policy gradient algorithm (DDPG) to reorganize the interbank lending structure of the network into an orientation that better mitigates the spread of contagion. In addition, we propose a new multi-layer DebtRank (DR) algorithm taking into account how contagion spreads from one layer to another. The work presented in this chapter has been published in Le and Ku (2022).

## 4.1 Introduction

A key property of a financial network is its interconnectedness. This interconnectedness, however, is a mechanism for amplifying shocks and distress, leading to contagion

and potentially resulting in catastrophic failure of the network. The risk of financial collapse due to the failure of some portion of the financial network leading to economic decline is called systemic risk. The importance of financial stability and systemic risk in the financial sector has been underlined after the global financial crisis, and the monitoring and regulation of systemic risk have become a major concern for regulators, governments, and financial institutions. The insights gained from the crisis include the importance of interconnectedness among financial institutions and markets and the necessity of adopting a system-wide view of stability and risk. One can get useful insights from analogous problems related to the large-scale (in)stability of systems with many interconnected components and feedback loops in other disciplines. It is important to understand the mechanisms underlying systemic risk and financial instability, metrics for identifying sources of systemic risk, and tools for monitoring these sources in practice.

A high level of systemic risk can have several effects on members of the financial system. These effects include impacts on sovereign credit ratings (Huang & Shen, 2015; Pagano & Sedunov, 2016; Sehgal et al., 2018), impacts on hedge fund returns (Bianchi et al., 2010), and reducing the benefits of diversification in portfolios (Busse et al., 2014; Strobl, 2016). In this chapter, we explore how RL can be used to reorganize the connections of a multi-layer financial network in order to reduce the overall systemic risk in the network.

Complex networks provide a convenient representation of the financial system. Typically, the nodes in the network represent the banks or financial institutions, while the edges connecting the nodes represent the relationship between the financial actors. Such a natural representation of the financial system has spurred the development of models adopting this framework to study systemic risk in financial systems (Bardoscia et al., 2021; Eisenberg & Noe, 2001; Gai et al., 2011; Glasserman & Young, 2015;

Jackson & Pernoud, 2021; Li et al., 2019; Macchiati et al., 2021). The structure of the network has been intimately tied to the levels of systemic risk present in the network (Allen & Gale, 2000; Boss et al., 2004; Gai & Kapadia, 2010; Nier et al., 2007). This idea has introduced a line of study investigating how we can capitalize on the connection between systemic risk and the network structure to reduce systemic risk. Poledna and Thurner (2016) and Poledna et al. (2017) implement a systemic risk tax incentive using agent-based modelling and observe a reduction in systemic risk in the self-organized network. Diem et al. (2020) reduce systemic risk in a direct-exposure network using mixed-integer linear programming to reorganize the network. They also observe that the reorganization of the direct exposure network can yield lower levels of systemic risk when compared to Basel III-like equity increases. They highlight some network characteristics expressing low levels of systemic risk, suggesting that these characteristics should be taken into account when designing policies for tackling financial market stability. Another study by Pichler et al. (2021) reorganizes an overlapping bond portfolio network, represented by a bipartite network, in a similar manner by framing the reorganization problem as an optimization problem.

In order to manage systemic risk, we need to be able to measure systemic risk. In a network setting, systemic risk can be measured by using network-based measures (Eisenberg & Noe, 2001; Furfine, 2003; Neveu, 2018). The seminal work by Eisenberg and Noe (2001) introduces a clearing algorithm while also providing an estimate of the systemic risk based on the number of "waves" of defaults required for a firm to fail in the algorithm. The approach taken by Furfine (2003) uses interbank payment data to simulate the knock-on effects of the failure of a single firm. These measures do not consider how distress prior to default can lead to contagion. The popular DR measure by Battiston et al. (2012) tackles this problem by taking into account the build-up and propagation of distress and its effect on the equity of banks in the

network. The DR algorithm measures the systemic risk of financial institutions by propagating their initial distress through the network and calculating the induced loss as a result. This algorithm was extended by Bardoscia et al. (2015) to allow banks to propagate distress more than once and by Silva et al. (2017) to consider a feedback mechanism between the real economy and financial systems.

In reality, there are many different types of financial products and contracts. The failure of an institution to honour one type of contract is not always felt in isolation. In fact, Cuba et al. (2021), Montagna and Kok (2013), and Poledna et al. (2015) find that only considering the systemic risk in single-layer networks severely underestimates the total systemic risk of a financial system. Poledna et al. (2015) extend the DR measure to the multi-layer case, allowing for the comparison of systemic risk between layers as well as the systemic risk of the combined network of projected layers. Poledna et al. (2021) modify the DR algorithm to account for overlapping portfolios in bipartite networks, and hence account for indirect exposures. Cao et al. (2021) extend the DR algorithm to the multi-layer case, accounting for investments of debt and equity between financial institutions.

Recently, there have been efforts to apply machine learning techniques to improve systemic risk assessment. Li et al. (2013) use support vector machines to predict systemic risk in the Chinese banking system. Cerchiello et al. (2016) use financial twitter and market data in predicting when shocks to the financial system might occur. Using algorithmic text analysis, Nyman et al. (2021) use financial reports and news articles to measure relative sentiment shifts based off excitement and anxiety summary statistics, finding potential in predicting increases in distress in the financial system. Most recently, So et al. (2022) proposed the use of Latent Dirichlet Allocation on financial news article data, allowing real-time prediction of systemic risk. For a more detailed review of machine learning applications in systemic risk, see the survey

by Kou et al. (2019).

Although there are a number of studies on machine learning applications in systemic risk, currently, only a handful of studies have adapted RL techniques to a financial network setting. A. Liu et al. (2018) make use of a multi-agent model based on temporal difference RL to replicate lending and borrowing dynamics. In particular, RL here is used to help decide each banks' counterparties. Their work demonstrates how the risk preferences of individual banks can assemble networks that are less at risk for contagion. Although not modifying the interbank relationships themselves, Petrone et al. (2021) proposes a framework in which an RL agent provides capital investments for different banks in the network, replicating the capital injections given by the government to increase the resilience of banks and minimize losses in the network.

In our study, we take a different approach to reorganizing the financial network by using RL. The main goal of this chapter is to construct a RL framework to minimize systemic risk in a multi-layer financial network. In pursuing this goal, we made the following contributions. First of all, we develop the constraint DDPG algorithm to reorganize the interbank lending structure of a multi-layer network by modifying the classical DDPG algorithm, proposed by Lillicrap et al. (2015). To minimize the effects of network reorganization on the balance sheets of each bank in the network, we incorporate a safety layer inspired by Dalal et al. (2018). The flexibility that is offered by RL allows us to easily extend the optimization procedure from the single layer case to the multi-layer case circumventing the technical optimization challenges noted by Diem et al. (2020). Second, we propose a new multi-layer DR to measure systemic risk in our networks. Both optimization procedures in Diem et al. (2020) and Pichler et al. (2021) were done by minimizing the total direct impact, an approximation of the DR. Using RL we directly incorporate the DR measure into the model's objective

via the reward function. The types of assets used in this study will have different maturities. Therefore, to account for how contagion might spread in a multi-layer network of loans with differing maturities, we propose a modified DR measure. We further highlight the versatility of using RL by considering preferential reduction in DR through the modification of the reward function to account for highly leveraged banks.

The remainder of this chapter is organized into 5 sections. In Section 4.2 we outline how we model our multi-layer complex network and present both the conventional DR and our proposed multi-layer DR measures. In Section 4.3 we detail the implementation of our RL agent in the context of reducing systemic risk in a multi-layer complex network environment. In Section 4.4 we outline how to constrain the action of the RL agent to preserve specific properties of the complex network and also present the experimental details for the single-layer and multi-layer case along with the parameters and hyperparameters used in our model. In Section 4.5 we present the results and discussion. Finally, in Section 4.6 we conclude the chapter.

## 4.2 Network Model

### 4.2.1 Multi-Layer Complex Networks

We modelled the interbank liability network as a multi-layer weighted directed graph with $\mathcal{M} = \{G, Y\}$ where $G = \{(V, E_\alpha) \mid \alpha \in \{1, 2, ..., M\}$ is the set of graphs in the multi-layer network and $Y = \{\alpha \mid \alpha \in \{1, 2, ..., M\}\}$ is the index set for the different layers of the multi-layer network. The set of nodes in the multi-layer network is denoted by $V = \{i \mid i \in \{1, 2, ..., N\}\}$. $E_\alpha = \{(i, j) \mid i, j \in V, i \neq j\}$ denotes the set of edges connecting nodes $V$ in layer $\alpha$. Note that each layer contains the same set of

| Notation | Definition |
|---|---|
| $M$ | Total number of layers |
| $\mathbf{Q}$ | Adjacency matrix |
| $\mathbf{L}^{\alpha}$ | Liability interbank network, weighted adjacency matrix for layer $\alpha$ |
| $\Delta\mathbf{L}^{\alpha}$ | Change in the interbank liability network weights for layer $\alpha$ |
| $N$ | Total number of banks |
| $A$ | Total assets in the network |
| $\theta^{\alpha}$ | Interbank loan ratio for layer $\alpha$ |
| $r$ | Degree heterogeneity of network |
| $\beta$ | Cash deposit ratio |
| $\gamma$ | Equity capital ratio |
| $W_{ij}^{\alpha}$ | Impact of bank $i$ on bank $j$ in layer $\alpha$ |
| $e_j$ | Equity of bank $j$ |
| $\mathbf{e}$ | Vector containing equities of each bank in the network |
| $s_i(t)^{\alpha}$ | State of bank $i$ as distress propagates in layer $\alpha$ at time $t$ |
| $h_i(t)^{\alpha}$ | Level of distress of bank $i$ as distress propagates in layer $\alpha$ at time $t$ |
| $R_i(\mathbf{L}^{\alpha}, \mathbf{e})$ | The DR of bank $i$ for a given liability network $\mathbf{L}^{\alpha}$, vector $\mathbf{e}$ |
| $v^{\alpha}$ | Relative economic value of layer $\alpha$ for layer $\alpha$ |
| $k_i$ | Leverage ratio of bank $i$ |
| $w(k_i)$ | Credit weight function dependent on the leverage ratio $k_i$ |
| $s_t$ | State of environment at time $t$ |
| $a$ | Action of the RL agent |
| $\mu(s)$ | Policy of the RL agent |
| $r(s, a)$ | Reward function for state $s$ and action $a$ |
| $\mathbf{D}$ | Constraint matrix |
| $\mathbf{K}$ | Basis matrix of the null($\mathbf{D}$) |
| $d$ | Density of the multi-layer network |
| $J$ | Jaccard similarity index |
| $c$ | Clustering coefficient |
| $k_{\text{total},i}^{\alpha}$ | Total degree node $i$ in layer $\alpha$ |

Table 4.1: A table to reference for some notation used throughout this chapter.

nodes and the only difference between the layers is the topology of the edges.

In the context of financial networks, each node in the graph will represent a bank. The directed edge from bank $i$ to $j$ in layer $\alpha$ represents the loan of bank $i$ to bank $j$ in layer $\alpha$. This lending amount is denoted by $L_{ij}^{\alpha}$. In the case that bank $j$ defaults, $L_{ij}^{\alpha}$ also represents the impact of bank $j$ on bank $i$ as this is the amount lost by bank $j$. We define the adjacency matrix of the graph as $\mathbf{Q}$ and denote its elements using

$$Q_{ij}^{\alpha} = \begin{cases} 1 & \text{if bank } i \text{ lends to bank } j \text{ in layer } \alpha, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Using the methods described in Li et al. (2019) and Maeno et al. (2013), we can simulate a liability interbank network $\mathbf{L}^{\alpha}$ with parameters $N, A, \theta^{\alpha}, r, \beta$, and $\gamma$ representing the total number of banks, the total assets in the network, the interbank loan ratio, the networks' degree heterogeneity, the cash deposit ratio, and the equity capital ratio, respectively. Then the lending amounts that appear on the balance sheet are calculated as

$$L_{ij}^{\alpha} = \frac{Q_{ij}^{\alpha}(k_{\text{out},i}^{\alpha} k_{\text{in},i}^{\alpha})^{r}}{\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}^{\alpha}(k_{\text{out},i}^{\alpha} k_{\text{in},i}^{\alpha})^{r}} \theta^{\alpha} A, \tag{4.2}$$

where $k_{\text{out},i}^{\alpha}$ and $k_{\text{in},i}^{\alpha}$ are the outgoing degree and incoming degree of the $i$th bank in the $\alpha$ layer, respectively. Once the lending amounts are defined for every bank, we can then calculate the rest of the balance sheet. To calculate the balance sheets of the banks, we define the following, let $\theta = \sum_{\alpha=1}^{M} \theta^{\alpha}$, $l_i = \sum_{\alpha=1}^{M}\sum_{j=1}^{N} L_{ij}^{\alpha}$, $b_i = \sum_{\alpha=1}^{M}\sum_{i=1}^{N} L_{ij}^{\alpha}$, and $\text{TL} = \sum_{i=1}^{N} l_i$ where $\theta$ is the total proportion of the assets used for lending, $l_i$ is the total lending amount of bank $i$, $b_i$ is the total borrowing of bank $i$, and TL is the total amount used for lending in the network respectively. The balance sheet can

| Assets | Liabilities & Equity |
|---|---|
| Interbank loans, $\sum_{\alpha=1}^{M} l_i^{\alpha}$ | Interbank borrowings, $\sum_{\alpha=1}^{M} b_i^{\alpha}$ |
| Cash, $c_i$ | Deposits, $d_i$ |
| Other Assets, $o_i$ | Equity, $e_i$ |

Figure 4.1: Balance sheet of the $i$-th bank for all layers of the multi-layer network.

then be calculated using the following set of equations

$$o_i = \max(b_i - l_i, 0) + \Big[[(1-\theta) - \beta(1-\gamma) + \beta\theta]A - \sum_{i=1}^{N} \max(b_i - l_i, 0)\Big] l_i/\mathrm{TL}, \tag{4.3}$$

$$e_i = \frac{\gamma(l_i + o_i) - \beta\gamma b_i}{1 + \beta\gamma - \beta}, \tag{4.4}$$

$$d_i = \frac{(1-\gamma)(l_i + o_i) - \beta b_i}{1 + \beta\gamma - \beta}, \tag{4.5}$$

$$c_i = \frac{\beta(1-\gamma)(l_i + o_i) - \beta b_i}{1 + \beta\gamma - \beta}, \tag{4.6}$$

where $o_i$, $e_i$, $d_i$, and $c_i$ are the other assets, equity, deposits, and cash of bank $i$. Then the total assets on bank $i$'s balance sheet is $a_i = c_i + l_i + o_i$ and by basic accounting principles $a_i = d_i + b_i + e_i$. The simulated balance sheet for each bank in the network can be found in Figure 4.1.

### 4.2.2 DebtRank

We will be measuring the systemic risk contribution of banks in the complex network in terms of their DR. The algorithm used to calculate the DR was first introduced by Battiston et al. (2012) and was extended to multi-layer networks by Poledna

et al. (2015). It should be noted however that Poledna et al. (2015) do not take into account how distress might propagate between the different layers. In our study, we introduce our own mechanism for contagion to spread between the different layers of the multi-layer network. For the completeness, we first provide a brief introduction of the conventional DR for a single-layer network. The impact of bank $i$ on bank $j$ can be defined by

$$W_{ij} = \min\left[1, \frac{L_{ji}}{e_j}\right], \tag{4.7}$$

where $L_{ji}$ is the lending amount from bank $j$ to bank $i$ and $e_j$ is the equity of bank $j$. If $L_{ji} < e_j$ then the impact of bank $i$ on bank $j$ is $L_{ji}/e_j < 1$. Therefore, given an adequate level of $e_j$, the impact of bank $i$ on bank $j$ can be mitigated by the buffer $e_j$. Given a sufficiently low $e_j$, the impact of bank $i$ on bank $j$ could lead to the default of bank $j$. For each bank, we define two state variables. Let $h_i \in [0, 1]$ represent the level of distress of bank $i$ and $s_i \in \{U, D, I\}$ be a discrete variable taking three possible states $U, D$, and $I$, representing the undistressed, distressed, and inactive states, respectively. The dynamics of $h_i$ follows

$$h_i(t) = \min\left\{1, h_i(t-1) + \sum_{j|s_j(t-1)=D} W_{ji} h_j(t-1)\right\}, \tag{4.8}$$

$$s_i(t) = \begin{cases} D & \text{if } h_i(t) > 0; s_i(t-1) \neq I, \\ I & \text{if } s_i(t-1) = D, \\ s_i(t-1) & \text{otherwise,} \end{cases}$$

where $h_i$ is calculated for all $i$ at each time step. The DR of a bank $i$ is calculated after some finite time $T$ has passed or once all the banks are in state $U$ or $I$. The

DR of bank $i$, denoted by $R_i$, can be calculated as

$$R_i = \sum_j h_j(T)v_j - h_i(1)v_i, \tag{4.9}$$

where $v_i$ is the relative economic value of each bank defined as

$$v_i = \frac{l_i}{\sum_{k=1}^{N} l_k} \quad \forall i \in V. \tag{4.10}$$

The relative economic value of each bank is the contribution of bank $i$'s lending relative to the entire interbank network. When a bank is in distress, some or all of its value is lost (the bank is considered in default if all of its value is lost). Therefore, the DR can be interpreted as the relative economic value of the network that is potentially lost due to the distress caused by bank $i$ propagating through the network. Given that the DR is dependent only on $L_{ij}$ and $e_j$, we define $R_i(\mathbf{L}, \mathbf{e}) = R_i$ for a given liability network $\mathbf{L}$ and vector $\mathbf{e}$ whose entries are the equities of the respective banks.

### 4.2.3 DebtRank Accounting for Differing Maturity of Loans

In order to take into account how distress in one layer propagates to other layers, we use the index set $Y = \{\alpha \mid \alpha \in \{1, 2, 3, ..., M\}\}$ and let the first layer, $\alpha = 1$, represent the interbank liability network of loans with the shortest maturities and the last layer, $\alpha = M$, represent the interbank liability network of loans with the longest maturities. Therefore $L_{ij}^1$ represents the interbank liability matrix of the loans with the shortest maturities and $L_{ij}^M$ represents the interbank liability matrix of the loans with the longest maturities. It is assumed that any distress experienced impacts the short-term liability before the long-term liability. Then the impact matrix for layer

$\alpha > 1$ is given by

$$W_{ij}^{\alpha} = \begin{cases} \dfrac{L_{ji}^{\alpha}}{\max\left(L_{ji}^{\alpha},\ e_j - \sum_{\kappa=1}^{\alpha-1}\sum_{p=1}^{N} L_{jp}^{\kappa} h_p^{\kappa}(T)\right)} & \text{if } L_{ji}^{\alpha} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{4.11}$$

and for $\alpha = 1$, the impact matrix is given by

$$W_{ij}^{1} = \min\left(1, \frac{L_{ji}^{1}}{e_j}\right), \tag{4.12}$$

where $h_i^{\kappa}(T)$ is the distress that bank $i$ experiences in layer $\kappa$ at time $T$. For $\alpha > 1$ the equity is reduced by the lending amount affected by the distress in the previous layers. The DR of the first layer is calculated using the usual initial conditions of the conventional DR as shown in Equation (4.12), and the dynamics for layers $\alpha > 1$ follows similarly to the conventional DR algorithm. In other words, we let $h_i^1 \in [0, 1]$ represent the level of distress of bank $i$ resulting from the initial distress of bank $i$ in the first layer and $s_i^1 \in \{U, D, I\}$ be the discrete state variable where $U, D$, and $I$ represent the undistressed, distressed, and inactive state of a node, respectively. Then, for each layer, the dynamics of $h_i^{\alpha}$ follows

$$h_i^{\alpha}(t) = \min\left\{1, h_i^{\alpha}(t-1) + \sum_{j | s_j^{\alpha}(t-1) = D} W_{ji}^{\alpha} h_j^{\alpha}(t-1)\right\}, \tag{4.13}$$

$$s_i^{\alpha}(t) = \begin{cases} D & \text{if } h_i^{\alpha}(t) > 0; s_i^{\alpha}(t-1) \neq I, \\ I & \text{if } s_i^{\alpha}(t-1) = D, \\ s_i^{\alpha}(t-1) & \text{otherwise,} \end{cases}$$

where $h_i^\alpha$ is calculated for all $i$ at every time step. The calculation for layer $\alpha$ is stopped as soon as all the banks in layer $\alpha$ are in the state $U$ or $I$ after some finite time $T$ has passed. The initial distress and state for all nodes $i$ in layers $\alpha > 1$ are set according to the following equations

$$h_i^\alpha(0) = h_i^{\alpha-1}(T), \tag{4.14}$$

$$s_i^\alpha(0) = \begin{cases} D & \text{if } h_i^{\alpha-1}(T) > 0, \\ U & \text{otherwise.} \end{cases} \tag{4.15}$$

Therefore, any node that was distressed in the previous layer will maintain the same levels of distress at time $T$ in the next layer.[2] Furthermore, any nodes that become inactive after becoming distressed will have their state set to distressed or undistressed and will be able to propagate distress again in subsequent layers.

The DR of each layer $\alpha$ is calculated after some finite time $T$ has passed or once all banks are in state $U$ or $I$. Again, we define the total amount loaned by a bank $i$ in layer $\alpha$ as $l_i^\alpha = \sum_{j=1}^N L_{ij}^\alpha$. Then DR of a bank $i$ is calculated as

$$R_i^\alpha = \begin{cases} \sum_j h_j^\alpha(T) v_j^\alpha - h_i^\alpha(1) v_i^\alpha & \text{if } \alpha = 1, \\ \sum_j h_j^\alpha(T) v_j^\alpha & \text{if } \alpha > 1, \end{cases} \tag{4.16}$$

where $v_i^\alpha$ is the relative economic value of each node $i$ in layer $\alpha$ defined as

$$v_i^\alpha = \frac{l_i^\alpha}{\sum_{k=1}^N l_k^\alpha} \quad \forall i \in V. \tag{4.17}$$

[2]Equation (4.14) can be modified to include a recovery rate to allow banks to reduce the level of distress that is transferred to the next layer.

Note that the DR of each layer takes into account the distress from the previous layers, and therefore the DR of the multi-layered complex network will be greater than the conventional DR measure. Given that the DR is dependent only on $L_{ij}^{\alpha}$ and $e_j$, we define $R_i(\mathbf{L}^{\alpha}, e) = R_i^{\alpha}$ for a given liability network $\mathbf{L}^{\alpha}$ in layer $\alpha$ and vector $\mathbf{e}$ whose entries are the equities of the respective banks. The total DR of the multi-layer network is then calculated by

$$R(\mathbf{L}, \mathbf{e}) = \sum_{\alpha=1}^{M} \sum_{i=1}^{N} v^{\alpha} R_i(\mathbf{L}^{\alpha}, \mathbf{e}), \tag{4.18}$$

where

$$v^{\alpha} = \frac{\sum_{i=1}^{N} l_i^{\alpha}}{\sum_{\alpha=1}^{M} \sum_{i=1}^{N} l_i^{\alpha}}. \tag{4.19}$$

### 4.2.4 DebtRank Weighted by Leverage and Credit Risk

The DR measures the economic value lost due to the spread of the distress from a single bank. This measure does not provide insight into how likely a bank is to default. Credit risk is the measure of the likelihood that a bank will default on a debt obligation. One might be more concerned about a bank with high DR and high credit risk than a bank with a low DR but high credit risk, as the impact on the network due to the failure of the first bank is far greater than impact on the network due to the failure of the second bank. After generating the balance sheets of the banks in the complex network, we can use the leverage ratio as a proxy of the bank's credit risk. There are several different leverage ratios that are commonly used in finance. We will use debt-to-assets ratio $k_i$ for a bank $i$ as defined below

$$k_i = \frac{d_i + b_i}{c_i + l_i + o_i}. \tag{4.20}$$

This measure will be used to modify the objective of the RL agent to preferentially reduce the systemic risk of higher leveraged banks. To implement this desired behaviour we weight the individual DR of the banks by their respective level of credit risk using a credit weight function $w(k_i)$ dependent on the leverage ratio of the bank. The credit weighted DR is then

$$R^w(\mathbf{L}, \mathbf{e}, k) = \sum_{\alpha=1}^{M} \sum_{i=1}^{N} w(k_i) v^\alpha R_i(\mathbf{L}^\alpha, \mathbf{e}), \tag{4.21}$$

where k is the vector of the leverage ratios of each bank. The degree of importance placed on the level of credit risk when reducing systemic risk can be changed by modifying the form of $w(k_i)$. An alternative estimate of the credit risk can also be used instead of the defined value of $k_i$ in Equation (4.20).

## 4.3 Reinforcement Learning

In our study, we take a RL approach to reorganizing the financial network. Since the DR reduction process can be constantly changed, we require an off-policy agent that maps a high dimensional state space to a high dimensional continuous action space. So we will adopt the DDPG algorithm. DDPG, proposed in Lillicrap et al. (2015), is an actor–critic based deep RL algorithm that has made remarkable achievements in the financial perspective. It uses a neural network as a Q-function approximator. To address the relatively unstable learned action function, they propose the use of a replay buffer and soft target updates to improve convergence to the optimal policy.

The classical DDPG algorithm has been developed by considering a Markov decision process with a state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $p(s_{t+1} \mid s_t, a_t)$, and reward function $r$. The return from a state is defined as the sum of

discounted future reward

$$\mathrm{R}_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i), \tag{4.22}$$

where $\gamma \in [0, 1]$ is the discount factor, $s_i \in \mathcal{S}$ and $a_i \in \mathcal{A}$ are the observation and the agent's action, respectively. The state-action value is defined by

$$Q^{\mu}(s, a) = \mathbb{E}_{r_{i\geq t}, s_{i>t} \sim E, a_{i>t} \sim \mu}[\mathrm{R}_t | s_t, a_t], \tag{4.23}$$

and we use the recursive Bellman equation

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))], \tag{4.24}$$

where $r_t, s_{t+1} \sim E$ indicates that the current reward and the future state are sampled from the environment. The parametrized actor function $\mu(s|\theta^{\mu})$ maps the states $\mathbf{S}$ to action $\mathcal{A}$. The $Q$-function will be approximated by the critic by minimizing the loss

$$L(\theta^{Q}) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}}[(Q(s_t, a_t \mid \theta^{Q}) - y_t)^2], \tag{4.25}$$

$$y_t = r(s_t, a_t) + \gamma Q'(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'}), \tag{4.26}$$

where $\mathcal{D}$ is the replay buffer that stores the transitions of the DDPG agent and $\mu'(s \mid \theta^{\mu'})$ and $Q'(s, a \mid \theta^{Q'})$ are the target actor and critic networks, respectively. The weights of the target networks are slowly updated using the learned networks' weights. The purpose of the target networks is to improve the stability of learning. The policy $\mu : \mathcal{S} \rightarrow \mathcal{A}$ of the agent is learned using the actor network. We train the actor network by maximizing the expected return $J$ with respect to $\theta^{\mu}$

$$J = \mathbb{E}_{s_t \sim \mathcal{D}}[Q(s, a \mid \theta^{Q}) \mid_{s=s_t, a=\mu(s_t|\theta^{\mu})}]. \tag{4.27}$$

The actor network is updated with the policy gradient using the results from Silver et al. (2014)

$$\nabla_{\theta^\mu} J = \mathbb{E}_{s_t \sim \mathcal{D}}[\nabla_a Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s \mid \theta^\mu) \mid_{s=s_t}]. \tag{4.28}$$

In the context of a complex network of banks, we have a single agent that assigns different amounts of lending to each bank in the network. In our problem setting we wish to reward the agent every time a network configuration results in a lower overall DR. In the following section, we will express our problem setting in an RL framework.

### 4.3.1 The Observation Space

The environment consists of $N$ financial institutions or banks. We set $M$ different types of lending relationships the banks can establish with one another. In our environment, each layer represents different maturity lengths of loans. Therefore, there exist $M(N^2 - N)$ lending relationships that the agent can assign to form the complex network. Some examples of different relationships include deposits and loans, security cross-holdings, derivatives, and foreign exchange, and loans with differing maturities (Li et al., 2019; Poledna et al., 2015). Every bank in the environment is given a balance sheet as described in Section 4.2. The objective of the agent is to find the network configuration with the least amount of systemic risk measured using Equation (4.16). In our implementation, we let the observation of the agent consist of the vectorization of the interbank liability network. The vectorization of the liability matrix $\mathbf{L}$ is defined by

$$\text{vec}(\mathbf{L}) = (L_{11}, ..., L_{1N}, L_{21}, ..., L_{2N}, ..., L_{N1}, ..., L_{NN}). \tag{4.29}$$

Therefore the observation, $s_t$, of the DDPG agent is given as

$$s_t = \{\text{vec}(\mathbf{L}^\alpha(t))|\alpha = 1, 2, ..., M\}. \tag{4.30}$$

### 4.3.2 The Action Space

Our DDPG agent will interact with the environment by modifying lending amounts of each bank in the network based on the observation $s_t$. The interbank liability network at time $t$ is denoted $\mathbf{L}(t)$. The action provided by the RL agent will be denoted by $\Delta\mathbf{L}(t)$. The quantity $\Delta\mathbf{L}^\alpha(t)$ is a modification to the current lending network. Through this action, a new interbank lending network $\mathbf{L}^\alpha(t+1)$ will be constructed. Therefore the new interbank lending network is given by

$$\mathbf{L}^\alpha(t+1) = \mathbf{L}^\alpha(t) + \Delta\mathbf{L}^\alpha(t). \tag{4.31}$$

After the agent acts on the environment and modifies the complex network, we wish to conserve the total borrowing and total lending amounts of each bank in the network. Additionally, we require the new lending amounts of each bank to be non-negative. Let the lending relationships at time $t$ be given by $\mathbf{L}^\alpha(t)$. Then $l_i^\alpha(t) = \sum_{i=1}^N L_{ij}^\alpha(t)$ and $b_i^\alpha(t) = \sum_{j=1}^N L_{ji}^\alpha(t)$ is the total $\alpha$-type lending and borrowing amount of the $i$th bank, respectively, at time $t$. Our objective is then to find an $\mathbf{L}^\alpha(t)$ where the

following constraints are satisfied

$$\sum_{j=1}^{N} L_{ij}^{\alpha}(t) = l_i^{\alpha}(0) \quad \forall i \in V, \alpha \in Y, t \geq 0, \tag{4.32}$$

$$\sum_{j=1}^{N} L_{ji}^{\alpha}(t) = b_i^{\alpha}(0) \quad \forall i \in V, \alpha \in Y, t \geq 0, \tag{4.33}$$

$$L_{ij}^{\alpha}(t) \geq 0 \quad \forall i \in V, \alpha \in Y, t \geq 0. \tag{4.34}$$

In order to construct the action of the RL agent, we will be using the framework outlined in Section 4.4.1 where we outline how to satisfy the lending and borrowing constraints by using a linear transformation, and in Section 4.4.2 where we outline how to satisfy the non-negativity constraints by introducing a safety layer using quadratic programming (QP). An overview of our constraint DDPG structure and its interaction with the safety layer can be found in Figure 4.2.

### 4.3.3 Reward and Episode Termination

The objective of our problem is to minimize the systemic risk with respect to the multi-layer DR. To do so, we intend to reward the agent every episode when the DR is reduced. Here we define the DR component of the reward function

$$r(s,a) = \max\left(1 - \lambda \frac{R(\mathbf{L}(t+1), \mathbf{e})}{R(\mathbf{L}(t), \mathbf{e})}, 0\right), \tag{4.35}$$

where $\lambda \in \mathbb{R}$. In this way, the agent is given a reward if the DR in the next step is lower than the previous step's DR. The factor $\lambda$ can be used to set a threshold on how low the DR must be before the agent is given a positive reward. In our experiments, we set $\lambda = 1$. The environment is also designed such that the episode ends if the DR

achieved at time $t+1$ is higher than the DR at time $t$. Comparing the DR at time $t+1$ to the DR at time $t$ instead of at time $t=0$ has the added benefit that the DR measured at the end of an episode is the lowest DR achieved in that episode.

Incorporating the credit risk weighted DR into the reward function results in

$$r^w(s,a,k) = \max\Big(1 - \lambda \frac{R^w(\mathbf{L}(t+1), \mathbf{e}, k)}{R^w(\mathbf{L}(t), \mathbf{e}, k)}, 0\Big). \tag{4.36}$$

Therefore, the RL agent will be rewarded more when banks with a large leverage ratio (i.e., more risky in terms of credit risk) have their DR reduced. In this way, we incentivize reducing the DR of a bank with a high leverage ratio over reducing the DR of a bank with a lower leverage ratio.

## 4.4 Proposed Approaches

The classical DDPG agent cannot be directly applied to our problem of reorganizing the multi-layer complex network as there are a number of properties that we wish to preserve. To preserve the operational well-being of the banks in the network, we require that the total lending and borrowing amounts on the stylized balance sheets to be conserved after reorganization. This idea is expressed through constraints (4.32) and (4.33). Additionally, after reorganization our model does not allow for negative lending. This idea is expressed through constraint (4.34). These constraints are achieved by using the transformation outlined in Section 4.4.1 and the safety layer presented in Section 4.4.2. The parameters and hyperparameters used in the construction of the complex multi-layer network and our DDPG agent, respectively, are outlined in Table 4.2. The constrained DDPG algorithm is outlined in Appendix C.

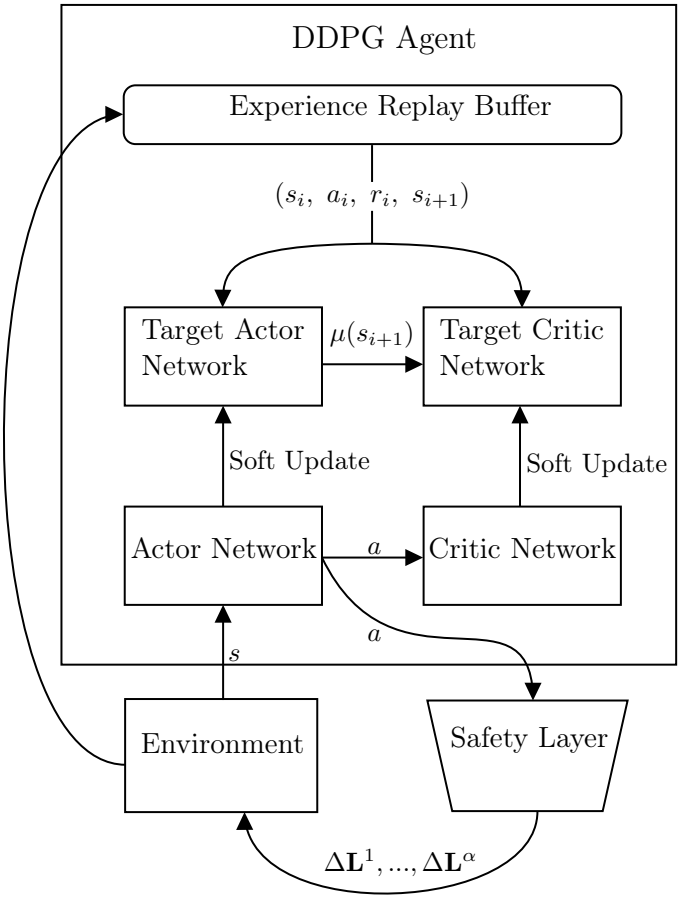



Figure 4.2: A diagram of the constraint DDPG architecture interacting with the safety layer and environment.

### 4.4.1 Lending and Borrowing Constraints

This section will describe how we can modify the network without violating the lending and borrowing constraint. Note that the $\alpha$ and $t$ in the notation are dropped in this section. This is because the framework outlined in this section is independent of the layer in the multi-complex network and the time when the DDPG agent acts in the environment. We define $\Delta\mathbf{L}$ as the change in the liability network and the new network configuration as $\mathbf{L}' = \mathbf{L} + \Delta\mathbf{L}$. We wish to find a mapping for the actions generated by the policy to $\Delta\mathbf{L}$. For a given liability matrix $\mathbf{L}$ we note that $L_{ij} = 0$ for $i = j$, therefore, we are only concerned with finding values of $\Delta L_{ij}$ where $i \neq j$, the off-diagonal elements of $\Delta\mathbf{L}$. For a matrix $\mathbf{A}$ of size $N \times N$ we define offdiag($\mathbf{A}$) to be the vector of size $N(N-1)$ containing the off-diagonal elements of $\mathbf{A}$.

In order to modify the interbank liablity network while satisfying constraints (4.32) and (4.33), we require that $\Delta\mathbf{L}$ satisfy the following constraints

$$\sum_{j=1}^{N} \Delta L_{ij} = 0 \quad \forall i \in V, \tag{4.37}$$

$$\sum_{j=1}^{N} \Delta L_{ji} = 0 \quad \forall i \in V. \tag{4.38}$$

To accomplish this, we solve the following homogeneous system of linear equations

$$\mathbf{D}\mathbf{x} = \mathbf{0}, \tag{4.39}$$

where we define the solution vector by

$$\begin{aligned}\mathbf{x} =& \text{offdiag}(\Delta\mathbf{L}) && (4.40)\\ =&(\Delta L_{12}, \Delta L_{13}, ..., \Delta L_{1N}, && (4.41)\\ &\Delta L_{21}, \Delta L_{23}, ..., \Delta L_{2N}, ...,\\ &\Delta L_{N1}, \Delta L_{N2}, ..., \Delta L_{N,N-1}).\end{aligned}$$

We then define a constraint matrix $\mathbf{D}$ of size $2N \times N(N-1)$ by

$$\mathbf{D} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & ... & \mathbf{C}_N \\ \mathbf{J}_1 & \mathbf{J}_2 & ... & \mathbf{J}_{N-1} \end{bmatrix}, \tag{4.42}$$

where $\mathbf{C}_n$ are the sub-matrices of size $N \times (N-1)$ for $1 \leq n \leq N$ whose entries are equal to 1 in the $n$th row and 0 in all other rows. $\mathbf{J}_n$ are the sub-matrices of size $N \times N$ defined by the following recursion

$$\mathbf{J}_1 = \mathbf{I}, \tag{4.43}$$

$$\mathbf{J}_{n+1} = \mathbf{E}_{(N-1),N}...\mathbf{E}_{23}\mathbf{E}_{12}\mathbf{J}_n, \tag{4.44}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{E}_{ij}$ is the elementary matrix corresponding to the column swapping transformation between columns $i$ and $j$. The sub-matrices $\mathbf{C_n}$ in this system constrain each row of $\Delta\mathbf{L}$ to sum to zero while the sub-matrices $\mathbf{J}_n$ constrain each column of $\Delta\mathbf{L}$. An example of the structure of the system for $N = 4$

is presented below

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta L_{12} \\ \Delta L_{13} \\ \Delta L_{14} \\ \Delta L_{21} \\ \Delta L_{23} \\ \Delta L_{24} \\ \Delta L_{31} \\ \Delta L_{32} \\ \Delta L_{34} \\ \Delta L_{41} \\ \Delta L_{42} \\ \Delta L_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.45}$$

For $N \geq 3$, we have $2N \leq N(N-1)$ so the homogeneous system (4.39) has infinitely many solutions and $\mathbf{x} \in \text{null}(\mathbf{D})$. This null space is useful because the solutions here satisfy Equations (4.37) and (4.38).

In order for the DDPG agent to make a choice of $\Delta\mathbf{L}$, we will have it solve system (4.39). To accomplish this, we express the vector (4.40) as a linear combination of the basis vectors of the null space of $\mathbf{D}$. That is, let $\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_d$ be the basis vectors of null($\mathbf{D}$), then all solutions to system (4.39) are given by

$$\mathbf{K}\mathbf{u} = \mathbf{x}, \tag{4.46}$$

where $\mathbf{K}$ is the basis matrix and $\mathbf{u} = (u_1, u_2, ..., u_d)$ is a vector of size $d$ and $u_i \in \mathbb{R}$ for $i = 1, 2, ..., d$.

For each step in the environment the actor network will need to provide a set of vectors $\mathbf{u}$ for each layer. The action space is therefore $\mathcal{A} = \mathbb{R}^{Md}$ and the output of the actor network is then

$$\mu(s) = (u_1, u_2, ..., u_{(Md)}). \tag{4.47}$$

We can then partition the elements of vector (4.47) into $M$ vectors of length $d$ to be applied to the respective layers of the complex network. We will use

$$\mathbf{u}^\alpha = (u_{(\alpha-1)d+1}, u_{(\alpha-1)d+2}, ..., u_{\alpha d}) \tag{4.48}$$

to calculate the non-diagonal values of $\Delta\mathbf{L}^\alpha$ using Equation (4.46).

Although a more intuitive approach might be to have the DDPG agent directly calculate $\Delta\mathbf{L}$, if we were to do that, the action space would then be $\mathbb{R}^{M(N^2)}$. However, by using the basis matrix $\mathbf{K}$ as described above we can reduce the dimension of the action space.

**Theorem 4.4.1.** *Let* $\mathbf{D}$ *be the constraint matrix as described by Equation* (4.42) *for a liability network of size* $N \times N$ *where* $N \in \mathbb{N}$ *such that* $N \geq 3$. *Then the dimension of the action space for a single layer network is reduced by* $2N - 1$, *from* $N(N-1)$ *to* $N^2 - 3N + 1$.

*Proof.* Given a constraint matrix $\mathbf{D}$ as described by Equation (4.42), let $D_i$ be the $i$th row of matrix $\mathbf{D}$. We note that the first row can be written as the following linear combination

$$D_1 = \sum_{i=N+1}^{2N} D_i - \sum_{i=2}^{N} D_i, \tag{4.49}$$

and so, $D_1$ is linearly dependent and can be made into a zero vector by subtracting

the first row by Equation (4.49), resulting in

$$\begin{bmatrix} \mathbf{0} & \mathbf{C}_2 & ... & \mathbf{C}_N \\ \mathbf{J}_1 & \mathbf{J}_2 & ... & \mathbf{J}_{N-1} \end{bmatrix}, \tag{4.50}$$

where $\mathbf{0}$ is the $N \times (N-1)$ zero matrix. Second, swapping the first $N$ rows with the last $N+1$ to $2N$ rows gives

$$\begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & ... & \mathbf{J}_{N-1} \\ \mathbf{0} & \mathbf{C}_2 & ... & \mathbf{C}_N \end{bmatrix}. \tag{4.51}$$

Third, we shift row $D_{N+1}$ down until we get

$$\begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & ... & \mathbf{J}_{N-1} \\ \mathbf{0} & \mathbf{C}_1 & ... & \mathbf{C}_{N-1} \end{bmatrix}. \tag{4.52}$$

Finally adding $-D_N$ to $D_{N+1}$ gives

$$\begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & ... & \mathbf{J}_{N-1} \\ \mathbf{0} & \mathbf{C}_1 + \mathbf{Z} & ... & \mathbf{C}_{N-1} + \mathbf{Z} \end{bmatrix}, \tag{4.53}$$

where $\mathbf{Z}$ is a sub-matrix of size $N \times (N-1)$ with elements

$$Z_{ij} = \begin{cases} -1 & \text{if } i = 1, j = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.54}$$

The matrix $\mathbf{D}$ is now in reduced row echelon form and by inspection we find that the

$\text{Rank}(\mathbf{D}) = 2N - 1$. Now by the rank-nullity theorem, we have

$$\text{Rank}(\mathbf{D}) + \text{Nullity}(\mathbf{D}) = N(N-1), \tag{4.55}$$

and the $\text{Nullity}(\mathbf{D})$ is then given by

$$\begin{aligned} \text{Nullity}(\mathbf{D}) &= N(N-1) - \text{Rank}(\mathbf{D}), && (4.56)\\ &= N(N-1) - (2N-1), && (4.57)\\ &= N^2 - 3N + 1. && (4.58) \end{aligned}$$

Therefore, the dimension of the action space for a single layer network is $N^2 - 3N + 1$. □

By theorem 4.4.1 the exact size of the action space is reduced by $2N - 1$ and, in fact, the dimension of the action space is $d = N^2 - 3N + 1$. Again we note that the matrix $\mathbf{K}$ is layer-independent and only needs to be calculated once. Therefore the action space of concern is denoted by $\mathcal{A} = \mathbb{R}^{M(N^2-3N+1)}$.

### 4.4.2 Safety Layer: Non-negativity Constraints

The framework described above conserves the total lending and borrowing amounts of each bank but still allows for the possibility of negative lending amounts after a single step through the environment. In order to maintain the non-negativity conditions on $\mathbf{L}^{\alpha}(t+1)$, we pass the action through a QP problem inspired by Dalal et al. (2018).

This amounts to solving the following QP problem

$$\begin{aligned} \underset{\mathbf{x}^\alpha}{\arg\min} \quad & \frac{1}{2}\left\|\mathbf{K}\mathbf{x}^\alpha - \mathbf{K}\mathbf{u}^\alpha\right\|^2 \\ \text{subject to} \quad & \text{offdiag}(\mathbf{L}^\alpha(t)) + \mathbf{K}\mathbf{x}^\alpha \succeq \mathbf{0}, \end{aligned} \tag{4.59}$$

where the inequality $\succeq$ represents an element-wise inequality. The constraints of problem (4.59) ensure that after a step $\Delta\mathbf{L}$ the elements of $\mathbf{L}(t+1)$ will be non-negative. The QP problem itself aims to perturb the off-diagonal elements of $\Delta\mathbf{L}$ in the Euclidean norm in order to satisfy constraint (4.34). Practically, we utilize the CPLEX solver to solve problem (4.59).

By imposing these constraints, the actions from the agent will result in a liability network where the total lending and borrowing amounts appearing on their balance sheet are preserved. And after modifying the network structure, the non-negativity constraint will also be preserved. This will reduce the impact of the change in lending relationships on the banks' operations. We also note that by using the methods described in Section 4.4.1, we are able to avoid including constraints (4.32) and (4.33) to problem (4.59). A diagram detailing the flow of the policy through the safety layer can be found in Figure 4.3.

### 4.4.3 Initializing the Complex Network and DDPG Agent

To generate our complex network we will start by using the R package *systemicrisk* to build the interbank liability matrix $\mathbf{L}^\alpha$. In doing so, we can forgo the use of Equation (4.2). Following Diem (2020), we begin by randomly sampling the row and column

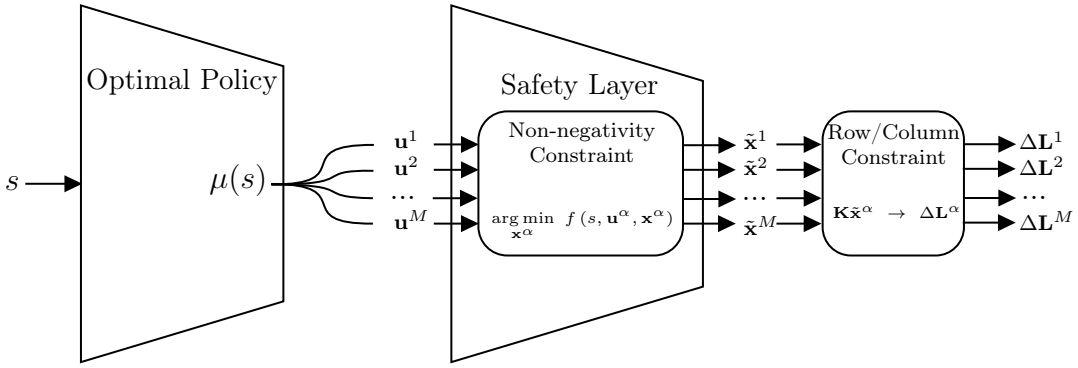



Figure 4.3: A diagram describing how the optimal policy is modified using the safety layer. The optimal policy $\mu(s)$ is first partitioned into $M$ policies for the respective layers of the multi-layer network. Each of these policies are then fed into the safety layer to operate on the policy to constrain the rows/columns and non-negativity condition of $\mathbf{L}(t+1)$ respectively. The constrained action $\mathbf{K}\tilde{\mathbf{x}}^\alpha$ contains the off-diagonal elements of a feasible $\Delta\mathbf{L}$.

sum vectors $\hat{\mathbf{l}}^\alpha$ and $\hat{\mathbf{a}}^\alpha$ of $\mathbf{L}^\alpha$, respectively, where

$$\hat{\mathbf{l}}^\alpha = (\hat{l}_1^\alpha, \hat{l}_2^\alpha, ..., \hat{l}_N^\alpha),$$
$$\hat{\mathbf{a}}^\alpha = (\hat{a}_1^\alpha, \hat{a}_2^\alpha, ..., \hat{a}_N^\alpha).$$

In our experiments, we consider three different network sizes where $N \in \{10, 20, 30\}$. Let $b \in \hat{\mathcal{B}}, m \in \hat{\mathcal{M}}, s \in \hat{\mathcal{S}}$ be the set of indices denoting the big, medium, and small banks respectively. The elements of the row and column sum vectors are sampled from the following uniform distributions

$$\hat{l}_b^\alpha \sim U(6000, 10000), \hat{l}_m^\alpha \sim U(2000, 6000), \hat{l}_s^\alpha \sim U(500, 2000),$$

and

$$\hat{a}_b^\alpha \sim U(0, 2000), \hat{a}_m^\alpha \sim U(0, 700), \hat{a}_s^\alpha \sim U(0, 150),$$

where $\hat{\mathcal{B}} = \{1\}, \hat{\mathcal{M}} = \{2, 3\}, \hat{\mathcal{S}} = \{4, ..., N\}$ for $N = 10$ and $\hat{\mathcal{B}} = \{1, 2\}, \hat{\mathcal{M}} = \{3, 4, 5\}, \hat{\mathcal{S}} = \{6, ..., N\}$ for $N = 20, 30$. The *systemicrisk* package estimates interbank liability matrices satisfying $\hat{\mathbf{l}}^\alpha$ and $\hat{\mathbf{a}}^\alpha$ based on Bayesian methodology developed by Gandy and Veraart (2017).

With the liability matrices given, we can set the total asset value of the entire network using $A = s \sum_\alpha \sum_{i,j} L_{ij}^\alpha$ where $s > 1$. The relative proportion of the network value that is used for lending can then be calculated as

$$\theta^\alpha = \frac{\sum_{i,j} L_{ij}^\alpha}{A}. \tag{4.60}$$

Finally, we can generate the balance sheet of each bank in the network using Equations (4.3) to (4.6). We set the cash deposit ratio to be $\beta = 0.18$ and the equity capital ratio for each bank $i$ to be sampled from the following interval, $\gamma_i \in (0.07, 0.2)$. When modifying the reward function to incorporate credit risk, we set $\gamma_i = 0.2$ with $k_i \in (0.07, 0.12) \cup (0.85, 1.0)$. The important quantity to consider on the balance sheet is the equity given by Equation (4.4) as this value is used in the calculation of the DR. Other values of the balance sheet may be used to calculate any other relevant financial variables, as required.

To test the effectiveness of applying RL in reducing systemic risk, we train and evaluate the RL agent on a number of different network structures. We also tested the flexibility of using RL by incorporating the notion of credit risk using the modified reward function, Equation (4.36).

Given the simulated multi-layer complex network we can begin reducing the systemic risk using DDPG. We use the Tensorflow TF-Agents framework to accomplish this task. For both actor and critic networks, we use three-layer neural networks with node sizes $(256, 256, 256)$. The parameter settings for the DDPG agent can be found in Table 4.2. Training was done for 8000 total iteration steps.

Table 4.2: Parameter settings for the DDPG agent.

| Hyperparameter | Value | Description |
|---|---|---|
| actor lr | $3 \times 10^{-5}$ | Actor learning rate |
| critic lr | $3 \times 10^{-4}$ | Critic learning rate |
| $\gamma$ | 0.80 | Discount factor |
| $\tau$ | 0.001 | Factor for the soft update of target networks |
| $N_{\text{mini}}$ | 256 | Batch size |
| $\mathcal{D}$ | 750 | Replay buffer size |
| $T$ | 50 | Maximum number of steps per episode |

### 4.4.4 Single Layer Case

In the single layer case, we will be investigating the effectiveness of the RL agent in reducing the systemic risk of the network and the effect of modifying the reward function for preferential reduction in DR for particular banks across the single layer network. For the single-layer case experiments, we let $N \in \{10, 20, 30\}$ and $M = 1$. The reward function used in reducing the systemic risk is Equation (4.35). To preferentially reduce systemic risk for highly leveraged banks, we consider the reward function (4.36) using four different weighting schemes. The DR distribution for the complex networks when exploring the effectiveness of the different weight functions is approximately uniform. This allows us to more easily judge the differences between the weight functions. In this way we can compare the different weight functions in a

fair manner.

$$w_{\text{uniform}}(k_i) = 1.0, \tag{4.61}$$

$$w_{\text{linear}}(k_i) = k_i, \tag{4.62}$$

$$w_{\text{exp},v}(k_i) = e^{vk_i}. \tag{4.63}$$

The first and second weighting schemes use a constant weight of 1.0 and linear weights comprising of the leverage ratio defined by Equations (4.61), and (4.62) respectively. The third and fourth weighting schemes use an exponential weight dependent on the leverage ratio defined by Equation (4.63) with parameter $v = 1.0$ and $v = 10.0$ respectively.

### 4.4.5 Multi-layer Case

In the mutli-layer case, we will be investigating the effectiveness of the RL agent in reducing the systemic risk of the network and presenting some observations on the network characteristics of multi-layer networks. For the multi-layer experiments we let $N \in \{10, 20, 30\}$ and $M \in \{1, 2, 3\}$. Therefore, the RL agent will be tasked with reducing systemic risk under nine different network sizes. Similar to the single-layer case, the reward function used in reducing the systemic risk is Equation (4.35). The DR however, is calculated using the multi-layer DR algorithm outlined in Section 4.2.3.

Given the optimized complex networks, we can observe the different characteristics of a multi-layer complex network after it has been modified by our RL model. We present the density, Jaccard distance, clustering coefficient, and average-weighted neighbour degree. The density for layer $\alpha$ of the multi-layer complex network is given

by

$$d = \frac{m^{\alpha}}{N(N-1)}, \tag{4.64}$$

where $m^{\alpha}$ is the number of edges in layer $\alpha$ and $N$ is the number of banks. This is the number of edges divided by the total number of possible edges. Therefore, the density can serve as a measure of sparsity.

To compare the differences between each layer of the network before and after optimization, we use the Jaccard distance measure. Given the two sets of edges $E_{\alpha}$ and $E_{\kappa}$ we can calculate the Jaccard distance by

$$d_J(E_{\alpha}, E_{\kappa}) = 1 - J(E_{\alpha}, E_{\kappa}), \tag{4.65}$$

where $J(E_{\alpha}, E_{\kappa})$ is the Jaccard similarity index between layers $\alpha$ and $\kappa$. The Jaccard similarity index is calculated as

$$J(E_{\alpha}, E_{\kappa}) = \frac{|E_{\alpha} \cap E_{\kappa}|}{|E_{\alpha} \cup E_{\kappa}|}. \tag{4.66}$$

The Jaccard distance measures the dissimilarity between the different layers by comparing the proportion of similar edge connections between the layers. The directed networks are converted to undirected networks before calculating the Jaccard distance.

Next we compute the clustering coefficient, treating each layer of the network as a directed unweighted graph. The clustering coefficient $c_i$ for node $i$ in layer $\alpha$ of the multi-layer complex network is given by

$$c_i = \frac{T_i}{2(k^{\alpha}_{\text{total},i}(k^{\alpha}_{\text{total},i} - 1) - 2k^{\alpha}_{\leftrightarrow,i})}, \tag{4.67}$$

where $T_i$ is the number of all directed triangles formed by node $i$ and

$$k^{\alpha}_{\text{total},i} = k^{\alpha}_{\text{out},i} + k^{\alpha}_{\text{in},i} \tag{4.68}$$

is the total degree of node $i$ in layer $\alpha$ and $k^{\alpha}_{\leftrightarrow,i}$ is the number of bilateral edges between node $i$ and its neighbours. Equation (4.67) measures the clustering coefficient for a directed unweighted graph. The clustering coefficient is the ratio between all directed triangles and the number of possible triangles which measures the tendency of the network to form tightly connected neighbourhoods (Fagiolo, 2007).

Finally, treating each layer of the network as a directed weighted graph, we compute the average-weighted neighbour degree (Diem et al., 2020). The average-weighted neighbour degree $k^{\alpha}_{\text{nn},i}$ for $i$ in layer $\alpha$ of the multi-layer complex network is given by

$$k^{\alpha}_{\text{nn},i} = \frac{1}{s^{\alpha}_i} \sum_{j=1}^{N} (L^{\alpha}_{ij} + L^{\alpha}_{ji}) k^{\alpha}_{\text{total},j}, \tag{4.69}$$

where

$$s^{\alpha}_i = \sum_{j=1}^{N} L^{\alpha}_{ij} + L^{\alpha}_{ji} \tag{4.70}$$

is the weighted node degree of bank $i$ in layer $\alpha$.

## 4.5 Results and Discussion

We will be evaluating the effectiveness of our constraint DDPG model in reducing systemic risk for two cases: (1) the single-layer case and (2) the multi-layer cases. It should be noted that given the nature of RL, it cannot be guaranteed that the reduced DR is a global optimum. However, with this trade-off, we are granted the flexibility of RL allowing the DDPG agent to directly consider the recursive DR

measure in its reward. Additionally, we introduce the idea of preferential systemic risk reduction to the DDPG agent by modifying the reward function whose results can be found in Section 4.5.1. Furthermore, to adapt the RL model to a multi-layer case we simply extend the DDPG agent's action to different layers and modify the reward function to include the DR of different layers whose results can be found in Section 4.5.2. The DDPG agent was tested on two types of networks. The first type consisting of a few number of large banks and the second type consisting of banks of similar sizes. In all cases we use the same set of hyperparameters described in Table 4.2. With more sophisticated hyperparameter tuning methods, it is suspected that the DR can be further reduced.

### 4.5.1 Single-layer Complex Network

The results in Figures 4.4 and 4.5 were generated using Equation (4.35) as the reward function with $N = 30$ and $M = 1$. The DR calculated in this section uses the conventional DR algorithm. For this particular network, the DR was reduced from 10.21 to 2.58. The DDPG agent achieved a reduction of 74.73%. From Figure 4.4, it can be observed that every bank has had their DR significantly reduced after optimization. Although the degree of reduction in DR between each bank varies, we did not observe an increase in DR for any bank.

From Figure 4.5 we can see how the network changes before and after optimization by the DDPG agent. An obvious increase in the sparsity can be observed. Using the policy that provides the greatest minimization of the DR we calculate average DR reduction across 100 episodes. From Table 4.3 we find that the DR was reduced significantly for all network types of size $N \in \{10, 20, 30\}$ and $M = 1$, the level of reduction achieved ranged from 70% to 75%.

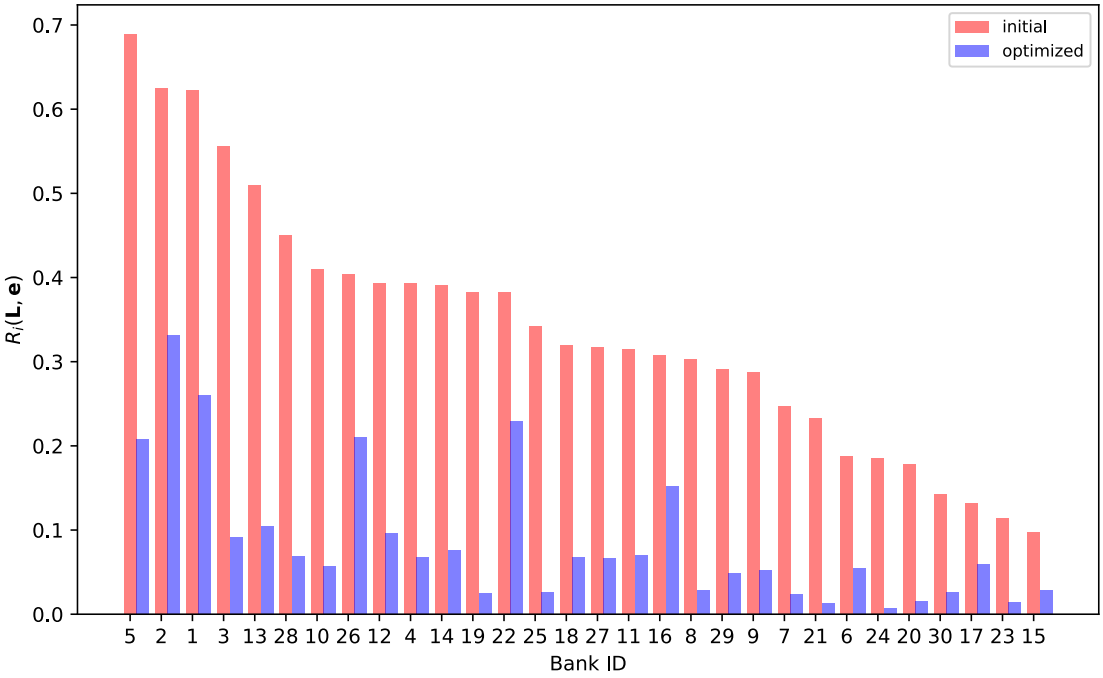



Figure 4.4: The DR of single-layer network of size $N = 30$. The banks are ordered from largest to smallest based on their DR. The red bars represent the initial levels of DR while the blue bars represent the optimized levels of DR.

The reward functions in this chapter are designed to incentivize the DDPG agent to reduce the overall systemic risk of the network. By introducing the weight factors (4.61)-(4.63) and using reward function (4.36), we aim to incentivize the agent to reduce the overall systemic risk while also preferentially reducing the DR of highly leveraged banks. The level of reduction in DR using the uniform weight factor (4.61) will be treated as the benchmark (where there is effectively no weight factor on the DR of each bank). In Figure 4.6, we compare the total DR for each leverage group. For the low leverage banks, we find that the total DR when using the linear and exponential weight functions is similar or greater compared to the benchmark total DR. The opposite observation is made for the high-leverage group. That is, when

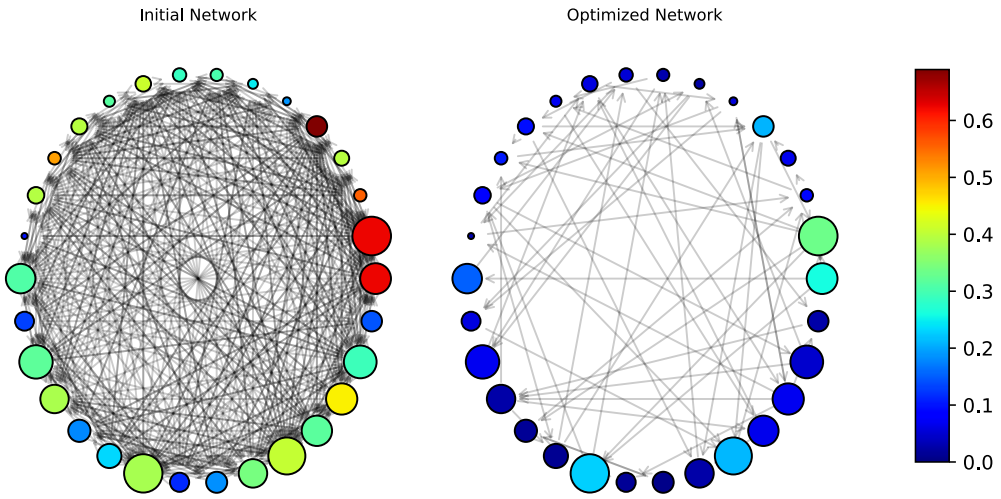



Figure 4.5: The structure of the single-layer complex network of size $N = 30$, where red represents high DR and blue represents low DR. The size of the circle represents the initial equity of the banks. The directed edges represent the lending relationship from bank $i$ to bank $j$

using linear and exponential weight functions, the total DR is lower compared to the benchmark total DR.

Note that despite higher leveraged banks receiving a greater weight in magnitude when using the exponential weight functions (4.63) compared to linear weight functions (4.62), we find that this does not necessarily mean there is a greater prioritization for the reduction of DR with respect to credit risk. From Figure 4.6, we observe that the linear weight provides a much more significant bias to reduce the DR of high leverage banks when compared to using the exponential weights with $v = 10.0$. This variability may be due to the stochastic nature of RL in training and the initial network structure to be optimized. Despite this observation, the modification to the reward function appears to encode the desired preferential reduction in systemic risk for highly leveraged banks when compared to not applying any weighting.

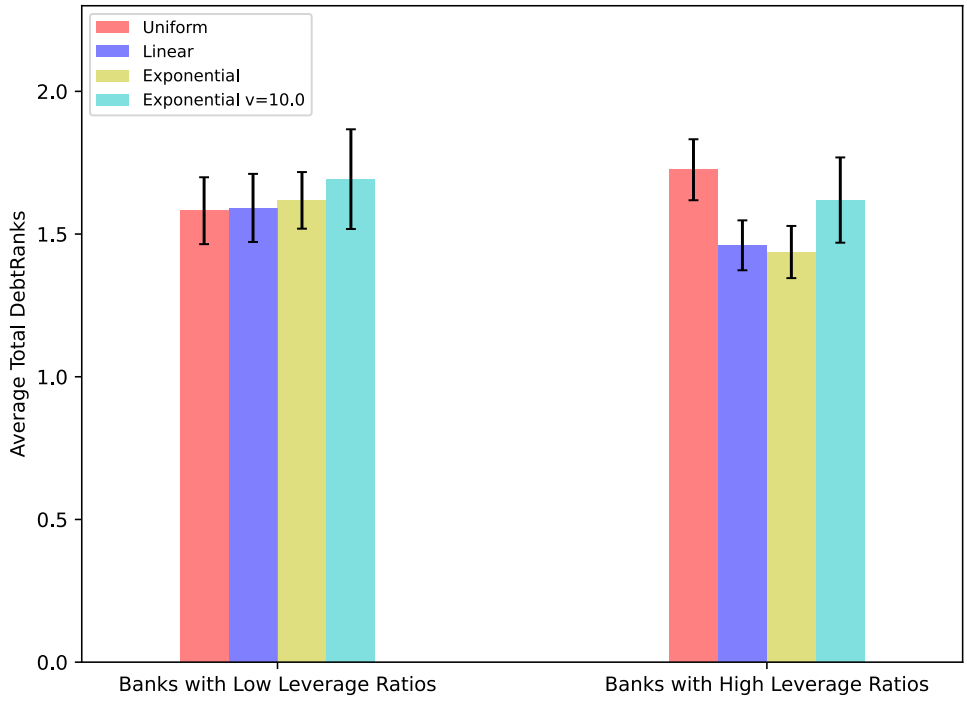



Figure 4.6: The low and high leverage banks were separated into two groups and the average total optimized DR was plotted for a network of size $N = 30, M = 1$. The average and standard deviation (presented as error bars) was calculated using 100 episodes.

Additionally, regardless of the weight function used, the main goal of the DDPG agent was achieved where the DR of the networks is reduced overall. The change in the DRs can be found in Table 4.3. The DDPG agent achieved a reduction as low as 67% to as high as 77% in DR depending on the weight function used. Preferentially reducing systemic risk of highly leveraged banks is beneficial because although some banks might have high levels of systemic risk, their credit risk might also be lower. In this case, reducing the systemic risk of banks with higher credit risk can be prioritized

by simply modifying the reward function.

Table 4.3: The average initial DR, optimized DR, and % reduction under each weighting scheme (Uniform, Linear, and Exponential with $v = 1.0$ and $v = 10.0$). The results are generated using single layer networks of size $N \in \{10, 20, 30\}$ over 100 episodes. The standard deviations are presented in the brackets.

| $N$ | Uniform | | | Linear | | |
|---|---|---|---|---|---|---|
| | Initial DR | Optimized DR | % Reduction | Initial DR | Optimized DR | % Reduction |
| 10 | 7.85 (0.41) | 2.31 (0.26) | 70.52 (3.94) | 7.84 (0.38) | 2.61 (0.40) | 66.62 (5.53) |
| 20 | 11.37 (0.48) | 2.69 (0.16) | 76.27 (1.52) | 11.36 (0.50) | 3.14 (0.19) | 72.34 (2.09) |
| 30 | 13.28 (0.49) | 3.31 (0.18) | 75.02 (1.55) | 13.22 (0.57) | 3.05 (0.19) | 76.87 (1.65) |
| | Exponential ($v = 1.0$) | | | Exponential ($v = 10.0$) | | |
| 10 | 7.86 (0.32) | 2.36 (0.26) | 69.98 (3.50) | 7.83 (0.31) | 2.60 (0.35) | 66.75 (4.76) |
| 20 | 11.44 (0.49) | 3.13 (0.27) | 72.62 (2.41) | 11.44 (0.52) | 3.07 (0.32) | 73.07 (3.11) |
| 30 | 13.28 (0.53) | 3.08 (0.16) | 76.82 (1.36) | 13.23 (0.50) | 3.34 (0.36) | 74.76 (2.73) |

### 4.5.2 Multi-layer Complex Network

All DRs in this section were reduced using Equation (4.35) as the reward function for the DDPG agent. Figures 4.7 and 4.8 were generated using a complex network of size $N = 30, M = 3$. The initial DR that was calculated before applying the DDPG agent was 11.10. After training and evaluation, we found that the DR was reduced to 6.53, a reduction of 41%. In Figure 4.7, a notable increase in DR can be observed across layers. This is expected as the multi-layer DR algorithm accounts for the inter-layer contagion spreading through successive layers. As distress accumulates from one layer to another, the equity of the banks may not be sufficient to cover the default of loans in higher layers. The equity of the banks in distress are reduced by the lending amount proportional to the distress experienced in the previous layer described by Equation (4.11). Despite the additional level of systemic risk accumulated in a multi-layer

network, we can see that by modifying the previous layers' structure we can reduce the overall DR by some amount in the subsequent layers. This is further evident when we note that the initial DR of the first, second and third layer are 1.14, 3.21, and 6.74 respectively. After reduction, the DR in the first, second and third layers was 0.62, 2.05, and 3.85. Therefore, the reduction achieved across the layers was 46%, 36%, and 43%, respectively. We find the greatest reduction in systemic risk in the first layer, despite having a relatively low level of initial DR compared to the other layers. Considering that the distress from the first layer propagates to the following layers, targeting the layer where the distress originates for optimization may prove to be the most effective. In this example, the DDPG agent targets the first layer, but other network configurations and different shock propagation dynamics may present different nodes or layers to prioritize.

The average optimized DRs can be found in Table 4.4. For all network types in the multi-layer case, we found that the DDPG agent was able to achieve an average reduction ranging from 8% to 57%. The lowest reduction achieved was in the case of similar sized banks with $N = 30, M = 3$ with a reduction of 8%. This may be due to the already low average initial DR. The network structure consisting of similarly sized banks in a multi-layer setting may also present some difficulty in DR reduction for the DDPG agent. Bear in mind Glasserman and Young (2015) report that contagion effects are more pronounced when node sizes are heterogeneous, suggesting that the performance of the DDPG agent under the "Few Large Banks" scenario should take higher precedence. By allowing contagion to accumulate across successive layers, we find that the increase in overall distress results in an overall lower level of systemic risk reduction achieved after optimization.

From Figure 4.8, the reorganization of the multi-layer complex network results in an increase in sparsity of the network based on the density of the edges. The average

Table 4.4: The average DR of the initial and optimized networks along with the % reduction after optimization for multi-layer networks of size $N \in \{10, 20, 30\}$ and $M \in \{1, 2, 3\}$ over 100 episodes. The values in the brackets are the standard deviations. The DR values are calculated using Equation (4.35). The ratio of large to medium to small banks under the "Few Large Banks" scenario is 1:2:8 for $N = 10$ and 2:3:($N$-5) for $N = 20, 30$, while the distribution of bank sizes is uniform under the "Similar Sized Banks" scenario.

| $N$ | $M$ | Few Large Banks | | | Similar Sized Banks | | |
|---|---|---|---|---|---|---|---|
| | | Init. DR | Opt. DR | % Red. | Init. DR | Opt. DR | % Red. |
| 10 | 1 | 7.81 (0.30) | 3.40 (0.19) | 56.38 (2.73) | 7.86 (0.33) | 2.25 (0.24) | 71.32 (3.23) |
| | 2 | 7.00 (0.28) | 4.41 (0.22) | 36.91 (4.05) | 5.53 (0.21) | 3.32 (0.12) | 39.89 (2.83) |
| | 3 | 7.00 (0.24) | 4.70 (0.12) | 32.88 (2.79) | 4.87 (0.10) | 3.13 (0.11) | 35.72 (2.32) |
| 20 | 1 | 13.88 (0.52) | 4.11 (0.33) | 70.37 (2.71) | 11.62 (0.48) | 3.02 (0.12) | 73.99 (1.52) |
| | 2 | 11.98 (0.29) | 5.54 (0.09) | 53.75 (1.18) | 6.19 (0.17) | 4.24 (0.08) | 31.52 (2.21) |
| | 3 | 10.03 (0.20) | 5.95 (0.30) | 40.68 (3.33) | 4.90 (0.08) | 4.21 (0.05) | 14.05 (1.62) |
| 30 | 1 | 18.79 (0.64) | 4.04 (0.38) | 78.46 (2.10) | 13.13 (0.49) | 3.20 (0.24) | 75.63 (1.87) |
| | 2 | 13.38 (0.38) | 5.72 (0.51) | 57.21 (4.28) | 6.14 (0.11) | 4.15 (0.08) | 32.38 (1.51) |
| | 3 | 11.29 (0.21) | 6.66 (0.13) | 41.05 (1.48) | 4.81 (0.05) | 4.41 (0.03) | 8.24 (0.71) |

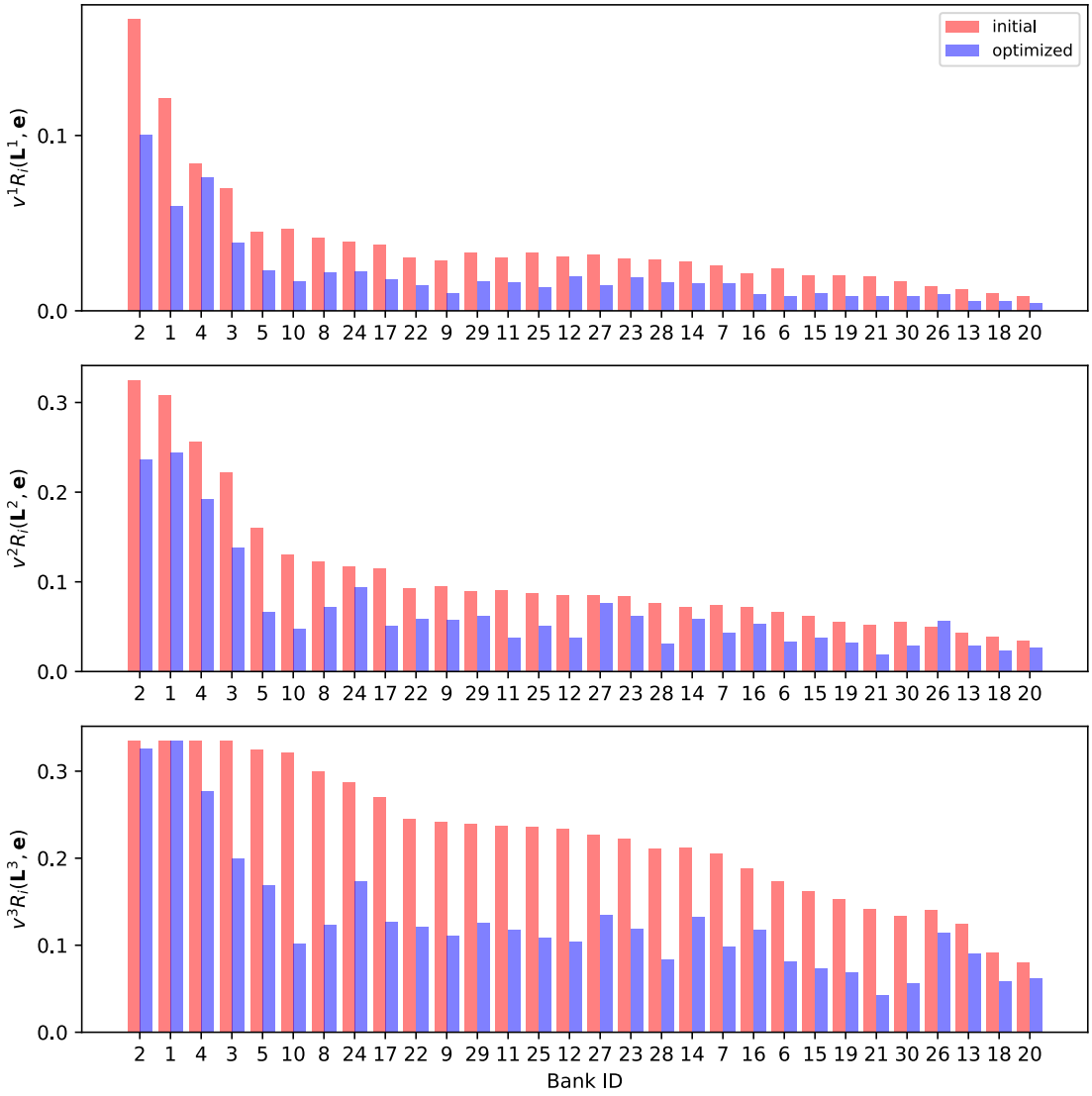



Figure 4.7: The DR of multi-layer network of size $N = 30$. The DRs in this plot are weighted by each layer's total relative economic value, $v^\alpha$. The banks are ordered from largest to smallest based on their DR. The red bars represent the initial levels of DR while the blue bars represent the reduced levels of DR.

density values for each layer of the corresponding networks can be found in Table 4.5. We observe that in every case of optimization, the average density was lowered. It

appears that given the larger network size, a larger reduction in density is observed. Although the increase in sparsity is consistent with what has been observed in other studies, this observation is not necessarily unique to an optimized network with low DR, as networks with high DR after optimization have been observed as well (Diem et al., 2020).

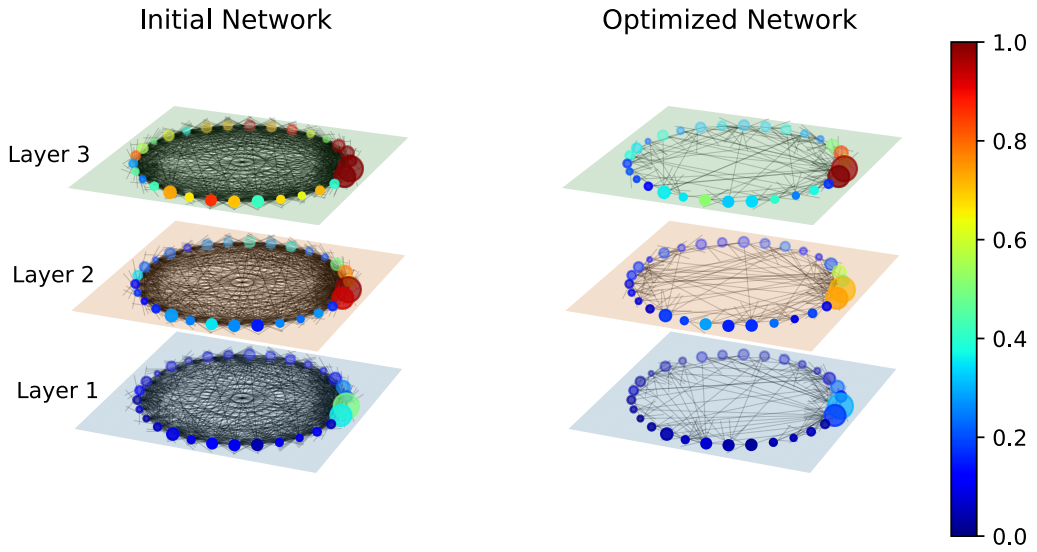



Figure 4.8: The structure of the multi-layer complex network of size $N = 30$, where red represents high DR and blue represents low DR. The size of the circle represents the initial equity of the banks. The directed edges represent the lending relationship from bank $i$ to bank $j$

Comparing the Jaccard distances in Table 4.6 across the different layers for the initial networks, we find that there is some similarity between all layers. After optimization any similarities between the layers are significantly reduced. That is, the topology of each layer becomes more dissimilar, suggesting that holding more dissimilar lending patterns across loans with differing maturities may produce complex networks more resilient against systemic risk. However, the benefits of diversification for the reduction of systemic risk are heavily debated and have been shown to be

Table 4.5: The average initial and optimized density for multi-layer complex networks of size $N \in \{10, 20, 30\}$, and $M \in \{1, 2, 3\}$ for each respective layer over 100 episodes. The values in the brackets are the standard deviations. The initial and optimized density are the density before and after optimizing the network configuration with respect to the DR, respectively.

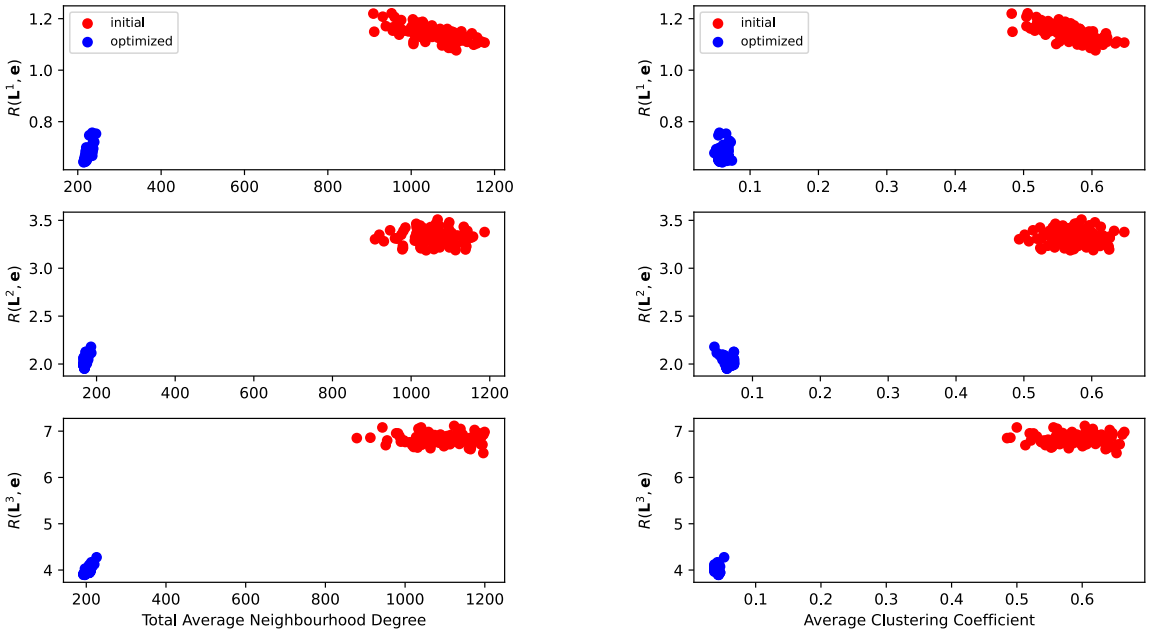
| $N$ | | 30 | | 20 | | 10 | |
|---|---|---|---|---|---|---|---|
| $M$ | $\alpha$ | Init. Density | Opt. Density | Init. Density | Opt. Density | Init. Density | Opt. Density |
| 1 | 1 | 0.55 (0.05) | 0.07 (0.00) | 0.57 (0.05) | 0.11 (0.00) | 0.47 (0.06) | 0.21 (0.00) |
| 2 | 1 | 0.54 (0.04) | 0.08 (0.01) | 0.55 (0.06) | 0.12 (0.01) | 0.56 (0.07) | 0.25 (0.01) |
| | 2 | 0.55 (0.04) | 0.09 (0.01) | 0.57 (0.05) | 0.10 (0.00) | 0.46 (0.05) | 0.22 (0.01) |
| 3 | 1 | 0.52 (0.04) | 0.07 (0.00) | 0.57 (0.05) | 0.11 (0.00) | 0.59 (0.07) | 0.22 (0.01) |
| | 2 | 0.54 (0.04) | 0.07 (0.00) | 0.54 (0.06) | 0.11 (0.01) | 0.56 (0.08) | 0.22 (0.01) |
| | 3 | 0.55 (0.04) | 0.07 (0.00) | 0.56 (0.04) | 0.11 (0.00) | 0.46 (0.05) | 0.21 (0.00) |

intimately related to systemic risk (Gai & Kapadia, 2019; Yang et al., 2020). Acemoglu et al. (2015) has shown that diversification protects well against small shocks but poorly against large shocks. Through empirical evidence, Yang et al. (2020) argues that large and medium sized banks contribute to systemic risk through diversification.

Figure 4.9a depicts the total average neighbour degree of the banks in the multi-layer complex network. We find that there is a significant change in the network characteristics after optimization. After optimization, a large number of banks will have a reduced total average neighbourhood degree. In all layers, it can be observed that before optimization, networks with high DR will have a large total average neighbourhood degree. After optimization, the networks have a reduced total DR and lower total average neighbourhood degree. This is consistent with Teteryatnikova (2014) who has shown through a tiered banking system that a negative correlation between neighboring banks' degree can increase the resilience of the network. Figure 4.9b shows the average clustering coefficient of the multi-layer network, we find that

Table 4.6: The matrices containing the average Jaccard dissimilarity between the layers of multi-layer complex networks of size $N \in \{10, 20, 30\}$, and $M = 3$ over 100 episodes. The values in the brackets are the standard deviations. The initial and optimized Jaccard dissimilarity are the Jaccard dissimilarity before and after optimizing the network configuration with respect to the DR, respectively.

| $N$ | $\alpha$ | Initial | | | Optimized | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 |
| 10 | 1 | 0 | 0.33 (0.09) | 0.41 (0.08) | 0 | 0.63 (0.02) | 0.64 (0.03) |
| | 2 | 0.33 (0.09) | 0 | 0.42 (0.08) | 0.63 (0.02) | 0 | 0.59 (0.03) |
| | 3 | 0.41 (0.08) | 0.42 (0.08) | 0 | 0.64 (0.03) | 0.59 (0.03) | 0 |
| 20 | 1 | 0 | 0.35 (0.05) | 0.34 (0.04) | 0 | 0.80 (0.01) | 0.83 (0.01) |
| | 2 | 0.35 (0.05) | 0 | 0.35 (0.05) | 0.80 (0.01) | 0 | 0.83 (0.01) |
| | 3 | 0.34 (0.04) | 0.35 (0.05) | 0 | 0.83 (0.01) | 0.83 (0.01) | 0 |
| 30 | 1 | 0 | 0.37 (0.03) | 0.37 (0.04) | 0 | 0.90 (0.00) | 0.90 (0.01) |
| | 2 | 0.37 (0.03) | 0 | 0.36 (0.04) | 0.90 (0.00) | 0 | 0.93 (0.00) |
| | 3 | 0.37 (0.04) | 0.36 (0.04) | 0 | 0.90 (0.01) | 0.93 (0.00) | 0 |




(a) Total average neighbourhood degree

(b) Average clustering coefficient

Figure 4.9: The total DR was plotted against (a) the total average neighbour degree $\sum_{i=1}^{N_{\text{avg}}} k_{nn,i}^{\alpha}$ in layer $\alpha$. (b) The average clustering coefficient $\frac{1}{N_{avg}} \sum_{i=1}^{N_{\text{avg}}} c_i$. Both plots were generated using multi-layer complex networks of size $N = 30$, and $M = 3$ over $N_{avg} = 100$ episodes.

after optimization, the average clustering coefficient is reduced for the majority of the networks. This suggests that the banks in the network begin to form less complete subgraphs with their neighbours after optimization. This characteristic may help reduce systemic risk, as the subgraphs have fewer channels for the spread of contagion.

### 4.5.3 Feasibility and Regulatory Guidance

In our model we delegate the task of discovering lower systemic risk networks to a single RL agent. To accomplish this task, the RL agent is incentivized through the use of reward functions. At the same time, we attempt to mitigate the disruption

to the operation of the banks by constraining the change to the total lending and borrowing amounts of a bank. In reality, there are many factors such as interest rates, credit worthiness, and liquidity that influence their decision making process. Explicitly modifying an actual interbank network presents a practicability issue as banks have several objectives and constraints to achieve and satisfy.

To clarify, we do not suggest imposing the network configuration designed by the RL agent on the participants of a real interbank network. Instead, the optimized networks from our model can serve as a benchmark to aid in designing regulatory policies when considering the multi-layer aspects of interbank networks, which is a use case that has been similarly suggested by Diem et al. (2020), Li et al. (2019), and Pichler et al. (2021). To encourage reorganization of the real interbank network, Poledna and Thurner (2016) and Poledna et al. (2017) propose to implement a systemic risk tax. This is a tax on transactions between any two counterparties to incentivize the formation of lower systemic risk networks. Their systemic risk tax is dependent on the change in expected systemic loss, a function of the DR of every bank in the network. In our model, the RL agent is guided by a relative change in DR of the network after every optimization pass. While the functional forms of the incentives are different, the marginal change to DR is a similar concept in both models.

An approach that may admit a more interpretable incentive mechanism would be to consider a multi-agent RL model such as the Multi-agent DDPG algorithm (Lowe et al., 2017). In this case, each bank can represent an agent and the environment can be designed to be competitive with respect to their own objectives or cooperative with respect to reducing overall systemic risk. The reward functions would then represent a direct incentive influencing the behaviour of each bank and hence, the evolution of the interbank network.

## 4.6 Conclusion

RL is an incredibly powerful tool that proves to be effective in the context of systemic risk management. In the final portion of this dissertation, we introduce a systemic risk reduction framework that takes advantage of RL by modifying the classical DDPG algorithm. The model reorganizes the interbank lending relationships of banks into a configuration that better mitigates the effects of contagion. The asset composition of the multi-layer networks consisted of short-term and long-term debts. In our model, the repayment of long-term debts is dependent on the solvency of short-term debts.

To calculate the systemic risk of such a network, we propose a new measure of DR accounting for the contagion that may spread from one layer to another as well as accounting for the impact of previous defaults on the individual banks' ability to repay future debts. The behaviour of the RL agent is guided by the reward function, and as a result, our RL agent is capable of solving problems in assessing and managing systemic risk.

To the best of our knowledge, this currently cannot be solved by traditional optimization techniques, since a recursive algorithm can be challenging to incorporate into the objective function of the optimization problem. We propose the DR reduction learning algorithm, called constraint DDPG, to find a network structure with reduced systemic risk. In order to satisfy the borrowing and lending constraints and maintain non-negativity with respect to the individual banks' lending after applying the DDPG agent's action, we modify the actor output in two ways. The first is by proposing a homogeneous system of linear equations whose solutions satisfy the lending and borrowing constraint. The second is the safety layer, which satisfies the non-negativity constraint by solving a QP problem. The effectiveness of our model was tested on different single-layer and multi-layer networks with varying sizes, layers,

and distribution of assets.

The performance of the RL agent was evaluated based on the level of DR reduction achieved. In all cases, a reduction in DR was observed, suggesting that RL is indeed an efficient tool in producing network structures that have reduced systemic risk in terms of DR. In the single-layer case, a reduction as high as 75% was observed while in the multi-layer case a reduction as high as 57% was observed. We find that the optimization process results in considerably different network topologies. The density, average neighbourhood degree, and clustering coefficient were observed to decrease after optimization. The Jaccard distance increased between the layers after optimization.

# Chapter 5

# Conclusion

In this dissertation, two areas of risk in finance were addressed. With respect to credit risk, we augmented the use of traditional Markov chain modelling with SBC to improve credit rating transition matrix estimation, fundamental in credit risk assessment. We then made further improvements to the predictive capabilities of the model by extending the first-order SBC model to higher-order Markov models. Additionally, the use of higher-order Markov chains allows for the consideration of credit rating momentum in the estimation process. To reduce systemic risk, we develop a novel and flexible RL framework capable of generating multi-layer network configurations possessing lower levels of systemic risk. At the same time, we extend the traditional DebtRank measure to consider transmission of distress between the different layers of the financial system. To conclude this dissertation, we summarize the work done in each chapter, discuss some limitations, and provide some future research directions.

## 5.1 Summary of Works

This dissertation can be effectively split into two parts. We opened the dissertation with studying the advantages of employing SBC for estimating credit rating transition matrices in Chapter 2. The transition matrices were estimated strictly from historical credit rating sequences. The credit rating data set consisted of credit rating sequences of Korean companies dating from 1986 to 2018. To cluster the sequences, the sequences must first be converted into sequence matrices. Given the sequence matrix representation, clusters of sequence matrices were generated using PCA-guided K-means. Next, first-order transition matrices, representative of each cluster, were estimated based on the sequences within the respective clusters. Finally, the estimated transition matrices were used to make predictions about the future behaviour of the crediting rating sequences given the current rating of the sequence. The proposed first-order SBC model was evaluated under 3 different long-term classification scenarios; 7 class credit rating prediction, credit rating transition direction (upgrade, stay, or downgrade) prediction, and default behaviour prediction. All three classification scenarios produced promising results suggesting that an ensemble of representative transition matrices based on each respective cluster better describes future credit rating behaviour than a single transition matrix.

Chapter 3 extended the work done in Chapter 2 by using higher-order Markov chains to represent each cluster. A higher focus was placed on the absorbing behaviour of these sequences where absorption represents the realization of default in credit risk. Just like in the first-order case, we test the higher-order model's ability to predict defaults by using strictly historical credit rating sequences from Korean companies from 1986 to 2018. In addition to credit risk assessment, we explored the higher-order SBC model's ability to generalize in the context of Web-usage mining. Web browsing

data from the Wikispeedia game was used test the model under two classification scenarios. First, we assessed its ability to predict the likelihood of absorption within a number of transitions. Second, we evaluated its ability to estimate the remaining length of a Web browsing session. Using the credit risk data, we found that high-order Markov models alone exhibit superior predictive performance over first-order models, where further improvement was observed with the addition of SBC. Using the Web-usage mining data, SBC provided significant improvement over both first-order and high-order models in predicting absorption within a specified number of transitions while high-order Markov models showed considerable promise in estimating the remaining duration of the browsing session.

Finally, we shifted our focus to the broader context of systemic risk in Chapter 4. We introduced a novel RL framework to assess and manage systemic risk in a multi-layer financial network by taking advantage of RL. New network configurations possessing reduced levels of systemic risk were discovered by applying the DDPG algorithm to reorganize the multi-layer interbank lending networks. The new network configurations were better able to mitigate the spread of contagion. The reorganization procedure itself was constrained in order to preserve the balance sheet of every bank. To achieve this, we developed a constraint DDPG model inspired by Dalal et al. (2018), consisting of a safety layer coupled with a linear mapping to satisfy the total borrowing and lending amounts of each bank. Moreover, we extended the traditional DR measure by taking into account how contagion spreads from one layer to another, resulting in a new multi-layer DR algorithm. Testing against networks of varying size and depth, our DDPG agent was able to significantly reduce systemic risk levels, suggesting the feasibility and utility of employing RL in managing systemic risk through aiding regulatory policy design. Additionally, we observed an increase in sparsity and an increase in network dissimilarity between the different layers of the

network after optimization.

## 5.2 Limitations and Future Extensions

While the ability of SBC appears to be quite promising there are some limitations. As demonstrated in Chapter 3, the performance of the model to predict absorbing behaviour is highly dependent on the data set used and so, one must take care to ensure the quality of the data is high, sufficiently processed, and can be appropriately modelled using Markov chains. With respect to the models themselves, an inherent limitation of high-order Markov chains is the curse of dimensionality. Indeed, the addition of SBC offers some relief by acting as a viable alternative to high-order models in the first-order case. While in some cases, this may be sufficient, in other cases, the best performance in predictive ability is desired. Hence, we provide evidence that a combination of SBC and higher-order Markov chains can potentially offer the greatest performance. Meaning, to achieve the greatest performance possible, higher-order Markov chains are necessary and the curse of dimensionality must be addressed in some way. By exploring and building upon parsimonous transition estimation methods such as the MTDg method by Raftery (1985b), our models can also serve to benefit greatly.

Regarding our credit risk assessment models, there are a number of possible extensions that can be made. The clustering algorithm used the Euclidean distance measure to distinguish similarity between different sequence matrices of firms. One can develop and incorporate a distance measure that considers transition risks. In the context of credit risk, the Euclidean distance measure is not able to distinguish clusters by their "riskiness" to start, one can refer to Trueck and Rachev (2005) for ideas on comparing transition matrices. Another extension could include using

more sophisticated models making use of other features of financial companies after clustering their respective sequence matrices.

A number of extensions can also be made with respect to the high-order portion of our model. In our model we use a single order choice for our high-order transition matrices for each cluster. However, depending on the resulting clusters generated, a single order choice may not be enough. That is, the order choice may be dependent on the characteristics of the cluster one wishes to represent. The question then is how to decide the degree to which the Markovian characteristics are expressed or not and which order of the Markov model to use for a particular cluster. SBC appears to be viable alternative to high-order Markov chains and given its highly tractable nature, one should be able to seamlessly integrate other models to represent each cluster generated. Therefore, it would be interesting to explore more sophisticated models and the effect SBC has on them.

Just like our credit risk models, the systemic risk RL framework proposed in Chapter 4 is not without limitations. The study was in part made possible thanks to the simulation methods outlined in Li et al. (2019) and Maeno et al. (2013). Meaning the results presented were based on a purely simulated multi-layer interbank lending network. Hence, the network structure and constraints used may not be representative of what is present in real world interbank lending networks. Another issue common with some RL problems is the idea of explainability or interpretability of the actions of the RL agent. Unlike applications of RL in other fields such as robotics or self-driving vehicles, by choosing the actions of the RL agent as parameters of a linear combination of bases, this task becomes extremely difficult. Even after transforming the actions into lending changes, the number of changes scales with the number of participants of the network which adds to the explainability difficulty.

Finally, we present some potential extensions of our work in systemic risk reduction

using RL. While we only consider a multi-layer interbank lending network, there are many different transmission channels for systemic risk. Our model can be extended to consider lending, security cross-holdings, derivatives, and foreign exchange transactions using the multi-layer exposure network model presented in Poledna et al. (2015) to start. Another extension could include a multi-action RL framework to consider equity levels or redistribution of wealth across different layers. At the moment, we assign a single DDPG agent with the task to reduce the systemic risk of an entire multi-layer complex network. The alternative approach to this problem is to design a multi-agent RL framework and let every bank be its own RL agent and work cooperatively to reduce the systemic risk. As a final remark, the versatile nature of the models presented in this dissertation allows for applications beyond the scope of credit risk and systemic risk. We hope this level of tractability can inspire innovation and exploration in other fields of research.

# Bibliography

Acemoglu, D., Ozdaglar, A., & Tahbaz-Salehi, A. (2015). Systemic risk and stability in financial networks. *American Economic Review*, *105*. https://doi.org/10.1257/aer.20130456 (cit. on p. 138).

Allen, F., & Gale, D. (2000). Financial contagion. *Journal of Political Economy*, *108*. https://doi.org/10.1086/262109 (cit. on pp. 4, 97).

Almeida, H., Cunha, I., Ferreira, M. A., & Restrepo, F. (2017). The real effects of credit ratings: The sovereign ceiling channel. *The Journal of Finance*, *72*(1), 249–290 (cit. on p. 47).

Altman, E. I., & Kao, D. L. (1992). The implications of corporate bond ratings drift. *Financial Analysts Journal*, *48*(3), 64–75 (cit. on pp. 3, 50).

Ansari, Z., Ahmed, W., Azeem, M., & Babu, A. V. (2015). Discovery of web usage profiles using various clustering techniques. *arXiv preprint arXiv:1509.00692* (cit. on p. 53).

Awad, M. A., & Khalil, I. (2012). Prediction of user's web-browsing behavior: Application of markov model. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*, 1131–1142. https://doi.org/10.1109/TSMCB.2012.2187441 (cit. on p. 52).

Baena-Mirabete, S., & Puig, P. (2018). Parsimonious higher order markov models for rating transitions. *Journal of the Royal Statistical Society. Series A: Statistics in Society*, *181*, 107–131. https://doi.org/10.1111/rssa.12267 (cit. on pp. 3, 51).

Bank for International Settlements. (2023). *The basel framework*. BIS. Retrieved June 16, 2023, from https://www.bis.org/basel_framework/index.htm?export=pdf (cit. on p. 2).

Bank for International Settlements. (2018). *Basel iii: Are we done now?* Retrieved June 25, 2023, from https://www.bis.org/speeches/sp180129.htm (cit. on p. 4).

Bardoscia, M., Barucca, P., Battiston, S., Caccioli, F., Cimini, G., Garlaschelli, D., Saracco, F., Squartini, T., & Caldarelli, G. (2021). The physics of financial networks. https://doi.org/10.1038/s42254-021-00322-5 (cit. on p. 96).

Bardoscia, M., Battiston, S., Caccioli, F., & Caldarelli, G. (2015). Debtrank: A microscopic foundation for shock propagation. *PLoS ONE*, *10*. https://doi.org/10.1371/journal.pone.0130406 (cit. on p. 98).

Battiston, S., Puliga, M., Kaushik, R., Tasca, P., & Caldarelli, G. (2012). Debtrank: Too central to fail? financial networks, the fed and systemic risk. *Scientific reports*, *2*, 541 (cit. on pp. 4, 97, 103).

Berchtold, A., & Raftery, A. (2002). *The mixture transition distribution model for high-order markov chains and non-gaussian time series* (3). www.stat.washington.edu/raftery (cit. on p. 61).

Bianchi, R., Drew, M., & Wijeratne, T. R. (2010). *Systemic risk, the ted spread and hedge fund returns* (Discussion Papers in Finance). Griffith University, Department of Accounting, Finance and Economics. https://EconPapers.repec.org/RePEc:gri:fpaper:finance:201004 (cit. on pp. 3, 96).

Borges, J., & Levene, M. (2007). Evaluating variable-length markov chain models for analysis of user web navigation sessions. *IEEE Transactions on Knowledge and Data Engineering*, *19*, 441–452. https://doi.org/10.1109/TKDE.2007.1012 (cit. on p. 52).

Borges, J., & Levene, M. (2005). Generating dynamic higher-order markov models in web usage mining. https://doi.org/https://doi.org/10.1007/11564126_9 (cit. on pp. 52 sq.).

Boss, M., Elsinger, H., Summer, M., & Thurner, S. (2004). Network topology of the interbank market. *Quantitative Finance*, *4*, 677–684. https://doi.org/10.1080/14697680400020325 (cit. on pp. 4, 97).

Busse, M., Dacorogna, M., & Kratz, M. (2014). The impact of systemic risk on the diversification benefits of a risk portfolio. *Risks*, *2*(3), 260–276 (cit. on pp. 4, 96).

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data mining and knowledge discovery*, *7*, 399–424 (cit. on p. 53).

Cao, J., Wen, F., Stanley, H. E., & Wang, X. (2021). Multilayer financial networks and systemic importance: Evidence from China. *International Review of Financial Analysis*, *78*(100). https://doi.org/10.1016/j.irfa.2021.10188 (cit. on p. 98).

Carpinone, A., Giorgio, M., Langella, R., & Testa, A. (2015). Markov chain modeling for very-short-term wind power forecasting. *Electric Power Systems Research*, *122*, 152–158. https://doi.org/10.1016/j.epsr.2014.12.025 (cit. on p. 46).

Cerchiello, P., Giudici, P., & Nicola, G. (2016). *Big data models of bank risk contagion* (DEM Working Papers Series No. 117). University of Pavia, Department of Economics and Management. https://EconPapers.repec.org/RePEc:pav:demwpp:demwp0117 (cit. on p. 98).

Chen, N., Ribeiro, B., Vieira, A., & Chen, A. (2013). Clustering and visualization of bankruptcy trajectory using self-organizing map. *Expert Systems with Applications*, *40*(1), 385–393. https://doi.org/https://doi.org/10.1016/j.eswa.2012.07.047 (cit. on pp. 8 sq.).

Ching, W. K., Fung, E. S., & Ng, M. K. (2004). Higher-order markov chain models for categorical data sequences. *Naval Research Logistics*, *51*, 557–574. https://doi.org/10.1002/nav.20017 (cit. on p. 48).

Christensen, J. H., Hansen, E., & Lando, D. (2004). Confidence sets for continuous-time rating transition probabilities. *Journal of Banking & Finance*, *28*(11), 2575–2602. https://doi.org/https://doi.org/10.1016/j.jbankfin.2004.06.003 (cit. on p. 19).

Cuba, W., Rodriguez-Martinez, A., Chavez, D. A., Caccioli, F., & Martinez-Jaramillo, S. (2021). A network characterization of the interbank exposures in peru. *Latin American Journal of Central Banking*, *2*, 100035. https://doi.org/10.1016/j.latcb.2021.100035 (cit. on p. 98).

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., & Tassa, Y. (2018). Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757* (cit. on pp. 99, 121, 146).

D'Amico, G., Dharmaraja, S., Manca, R., & Pasricha, P. (2019). A review of non-markovian models for the dynamics of credit ratings. *Reports on Economics and Finance*, *5*(1), 15–33 (cit. on pp. 3, 17, 50).

Deshpande, M., & Karypis, G. (2004). Selective markov models for predicting web page accesses. *ACM transactions on internet technology (TOIT)*, *4*(2), 163–184 (cit. on p. 52).

Dharmaraja, S., Pasricha, P., & Tardelli, P. (2017). Markov chain model with catastrophe to determine mean time to default of credit risky assets. *Journal of*

*Statistical Physics*, *169*(4), 876–888. https://doi.org/10.1007/s10955-017-1890-z (cit. on pp. 8, 50).

Diem, C. (2020, January). *Financial exposure network optimization via mixed integer linear programming*. https://csh.ac.at/vis/code/network_optimization/ (cit. on p. 122).

Diem, C., Pichler, A., & Thurner, S. (2020). What is the minimal systemic risk in financial exposure networks? *Journal of Economic Dynamics and Control*, *116*, 103900. https://doi.org/https://doi.org/10.1016/j.jedc.2020.103900 (cit. on pp. 97, 99, 128, 137, 141).

Dos Reis, G., Pfeuffer, M., & Smith, G. (2020). Capturing model risk and rating momentum in the estimation of probabilities of default and credit rating migrations. *Quantitative Finance*, *20*(7), 1069–1083 (cit. on p. 50).

Eisenberg, L., & Noe, T. H. (2001). Systemic risk in financial systems. *Management Science*, *47*(2), 236–249 (cit. on pp. 96 sq.).

Elayat, H. A. (1973, July). *Analysis of higher order markov and semi-markov chains*. https://hdl.handle.net/11244/31276 (cit. on pp. 56, 68).

Fagiolo, G. (2007). Clustering in complex directed networks. *Phys. Rev. E*, *76*, 026107. https://doi.org/10.1103/PhysRevE.76.026107 (cit. on p. 128).

Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explorations Newsletter*, *12*(1), 49–57. https://doi.org/10.1145/1882471.1882479 (cit. on pp. 28, 79).

Freixas, X., Laeven, L., & Peydró, J.-L. (2015). *Systemic risk, crises, and macroprudential regulation*. Mit Press. (Cit. on p. 5).

Frydman, H., & Schuermann, T. (2008). Credit rating dynamics and markov mixture models. *Journal of Banking & Finance*, *32*(6), 1062–1075. https://doi.org/https://doi.org/10.1016/j.jbankfin.2007.09.013 (cit. on pp. 3, 50).

Fuertes, A.-M., & Kalotychou, E. (2007). On sovereign credit migration: A study of alternative estimators and rating dynamics. *Computational Statistics & Data Analysis*, *51*(7), 3448–3469 (cit. on p. 51).

Furfine, C. (2003). Interbank exposures: Quantifying the risk of contagion. *Journal of Money, Credit, and Banking*, *35*, 111–128. https://doi.org/10.1353/mcb.2003.0004 (cit. on p. 97).

Gai, P., Haldane, A., & Kapadia, S. (2011). Complexity, concentration and contagion. *Journal of Monetary Economics*, *58*, 453–470. https://doi.org/10.1016/j.jmoneco.2011.05.005 (cit. on p. 96).

Gai, P., & Kapadia, S. (2010). Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *466*, 2401–2423. https://doi.org/10.1098/rspa.2009.0410 (cit. on pp. 4, 97).

Gai, P., & Kapadia, S. (2019). Networks and systemic risk in the financial system. *Oxford Review of Economic Policy*, *35*. https://doi.org/10.1093/oxrep/grz023 (cit. on p. 138).

Gandy, A., & Veraart, L. A. (2017). A bayesian methodology for systemic risk assessment in financial networks. *Management Science*, *63*, 4428–4446. https://doi.org/10.1287/mnsc.2016.2546 (cit. on p. 124).

García, V., Marqués, A. I., & Sánchez, J. S. (2015). An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *J. Intell. Inf. Syst.*, *44*(1), 159–189. https://doi.org/10.1007/s10844-014-0333-4 (cit. on p. 23).

Glasserman, P., & Young, H. P. (2015). How likely is contagion in financial networks? *Journal of Banking and Finance*, *50*, 383–399. https://doi.org/10.1016/j.jbankfin.2014.02.006 (cit. on pp. 96, 134).

Gunnvald, R. (2014). *Estimating probability of default using rating migrations in discrete and continuous time* (Publication No. 2014:49) [Master's thesis, KTH, Mathematical Statistics]. http://www.diva-portal.org/smash/record.jsf?pid=diva2%5C%3A747996&dswid=7601 (cit. on pp. 19 sq.).

Guo, X., Zhu, Z., & Shi, J. (2012). A corporate credit rating model using support vector domain combined with fuzzy clustering algorithm. *Mathematical Problems in Engineering*, *2012* (cit. on p. 9).

Güttler, A., & Raupach, P. (2008). The impact of downward rating momentum on credit portfolio risk. *Available at SSRN 2794019* (cit. on p. 3).

Huang, Y.-L., & Shen, C.-H. (2015). The Sovereign Effect on Bank Credit Ratings. *Journal of Financial Services Research*, *47*(3), 341–379. https://doi.org/10.1007/s10693-014-0193-7 (cit. on pp. 5, 96).

Irmatova, E. (2016). Relarm: A rating model based on relative pca attributes and k-means clustering. *arXiv preprint arXiv:1608.06416* (cit. on p. 9).

Jackson, M. O., & Pernoud, A. (2021). Systemic risk in financial networks: A survey. https://doi.org/10.1146/annurev-economics (cit. on p. 96).

Jafry, Y., & Schuermann, T. (2004). Measurement, estimation and comparison of credit migration matrices. *Journal of Banking & Finance*, *28*(11), 2603–2639. https://doi.org/https://doi.org/10.1016/j.jbankfin.2004.06.004 (cit. on pp. 20, 39).

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/https://doi.org/10.1016/j.patrec.2009.09.011 (cit. on p. 20).

Jarrow, R. A., Lando, D., & Turnbull, S. M. (1997). A markov model for the term structure of credit risk spreads. *The Review of Financial Studies*, *10*(2), 481–523. http://www.jstor.org/stable/2962353 (cit. on pp. 8, 49).

Khieu, H. D., & Pyles, M. K. (2012). The influence of a credit rating change on corporate cash holdings and their marginal value. *Financial Review*, *47*(2), 351–373 (cit. on p. 2).

Kiefer, N. M., & Larson, C. E. (2004). *Testing simple markov structures for credit rating transitions*. Comptroller of the Currency. (Cit. on p. 8).

Kou, G., Chao, X., Peng, Y., Alsaadi, F. E., & Herrera-Viedma, E. (2019). Machine learning methods for systemic risk analysis in financial sectors. *Technological and Economic Development of Economy*, *25*, 716–742. https://doi.org/10.3846/tede.2019.8740 (cit. on p. 99).

Kuncheva, L. I., & Sánchez, J. S. (2008). Nearest neighbour classifiers for streaming data with delayed labelling. *2008 Eighth IEEE International Conference on Data Mining*, 869–874 (cit. on p. 9).

Kuzilek, J., Vaclavek, J., Fuglik, V., & Zdrahal, Z. (2018). Student drop-out modelling using virtual learning environment behaviour data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11082 LNCS*, 166–171. https://doi.org/10.1007/978-3-319-98572-5_13 (cit. on p. 46).

Lando, D., & Skodeberg, T. M. (2002). Analyzing rating transitions and rating drift with continuous observations. *Journal of Banking & Finance*, *26*(2-3), 423–444. https://EconPapers.repec.org/RePEc:eee:jbfina:v:26:y:2002:i:2-3:p:423-444 (cit. on pp. 3, 20, 50 sq.).

Le, R., & Ku, H. (2022). Reducing systemic risk in a multi-layer network using reinforcement learning. *Physica A: Statistical Mechanics and its Applications*, *605*, 128029 (cit. on p. 95).

Le, R., Ku, H., & Jun, D. (2021). Sequence-based clustering applied to long-term credit risk assessment. *Expert Systems with Applications*, *165*. https://doi.org/10.1016/j.eswa.2020.113940 (cit. on pp. 7, 84, 93).

Lèbre, S., & Bourguignon, P.-Y. (2008). An em algorithm for estimation in the mixture transition distribution model. *Journal of Statistical Computation and Simulation*, *78*(8), 713–729 (cit. on p. 76).

Leskovec, J., & Krevl, A. (2014, June). SNAP Datasets: Stanford large network dataset collection. (Cit. on p. 74).

Li, S., Liu, M., Wang, L., & Yang, K. (2019). Bank multiplex networks and systemic risk. *Physica A: Statistical Mechanics and its Applications*, *533*, 122039. https://doi.org/https://doi.org/10.1016/j.physa.2019.122039 (cit. on pp. 97, 102, 111, 141, 148).

Li, S., Wang, M., & He, J. (2013). Prediction of banking systemic risk based on support vector machine. *Mathematical Problems in Engineering*, *2013*. https://doi.org/10.1155/2013/136030 (cit. on p. 98).

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (cit. on pp. 99, 109).

Liu, A., Mo, C. Y. J., Paddrik, M. E., & Yang, S. Y. (2018). An agent-based approach to interbank market lending decisions and risk implications. *Information*, *9*(6). https://doi.org/10.3390/info9060132 (cit. on p. 99).

Liu, Y. (2002). The evaluation of classification models for credit scoring. https://pdfs.semanticscholar.org/a62c/581a334e155cd6867a54d329b3db81ef2034.pdf (cit. on p. 32).

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 6379–6390 (cit. on p. 141).

Macchiati, V., Brandi, G., Matteo, T. D., Paolotti, D., Caldarelli, G., & Cimini, G. (2021). Systemic liquidity contagion in the european interbank market. *Journal of Economic Interaction and Coordination*. https://doi.org/10.1007/s11403-021-00338-1 (cit. on p. 97).

Maeno, Y., Morinaga, S., Nishiguchi, K., & Matsushima, H. (2013). Optimal portfolio for a robust financial system. *2013 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. https://doi.org/10.1109/cifer.2013.6611695 (cit. on pp. 102, 148).

Massa, S., & Puliafito, P. P. (2002). Web usage mining: Knowledge discovery using markov chains. *Management Information Systems*, *6* (cit. on p. 52).

Mlika, F., & Karoui, W. (2020). Proposed model to intelligent recommendation system based on markov chains and grouping of genres. *Procedia Computer Science*, *176*, 868–877 (cit. on p. 53).

Montagna, M., & Kok, C. (2013). *Multi-layered interbank model for assessing systemic risk* (Kiel Working Papers No. 1873). Kiel Institute for the World Economy (IfW Kiel). https://EconPapers.repec.org/RePEc:zbw:ifwkwp:1873 (cit. on p. 98).

Montiel, J., Bifet, A., & Abdessalem, T. (2017). Predicting over-indebtedness on batch and streaming data. *2017 IEEE International Conference on Big Data (Big Data)*, 1504–1513 (cit. on p. 10).

Morales, M. H., Rodríguez, J. T., & Montero, J. (2015). Credit rating using fuzzy algorithms. *Actas de la XVI Conferencia CAEPIA*, 539–548 (cit. on p. 9).

Neveu, A. R. (2018). A survey of network-based analysis and systemic risk measurement. *Journal of Economic Interaction and Coordination*, *13*, 241–281. https://doi.org/10.1007/s11403-016-0182-z (cit. on p. 97).

Newson, R. (2006). Confidence intervals for rank statistics: Somers' d and extensions. *The Stata Journal*, *6*(3), 309–334. https://doi.org/10.1177/1536867X0600600302 (cit. on p. 31).

Nicholls, M. G. (2007). Assessing the progress and the underlying nature of the flows of doctoral and master degree candidates using absorbing markov chains. *Higher Education*, *53*, 769–790. https://doi.org/10.1007/s10734-005-5275-x (cit. on p. 46).

Nicolau, J. (2014). A new model for multivariate markov chains. *Scandinavian Journal of Statistics*, *41*, 1124–1135. https://doi.org/10.1111/sjos.12087 (cit. on p. 48).

Nier, E., Yang, J., Yorulmazer, T., & Alentorn, A. (2007). Network models and financial stability. *Journal of Economic Dynamics and Control*, *31*, 2033–2060. https://doi.org/10.1016/j.jedc.2007.01.014 (cit. on pp. 4, 97).

Nyman, R., Kapadia, S., & Tuckett, D. (2021). News and narratives in financial systems: Exploiting big data for systemic risk assessment. *Journal of Economic Dynamics and Control*, *127*. https://doi.org/10.1016/j.jedc.2021.104119 (cit. on p. 98).

Pagano, M. S., & Sedunov, J. (2016). A comprehensive approach to measuring the relation between systemic risk exposure and sovereign debt. *Journal of Financial Stability*, *23*, 62–78. https://doi.org/https://doi.org/10.1016/j.jfs.2016.02.001 (cit. on p. 96).

Pal, G., Atkinson, K., & Li, G. (2021). Real-time user clickstream behavior analysis based on apache storm streaming. *Electronic Commerce Research*. https://doi.org/10.1007/s10660-021-09518-4 (cit. on p. 52).

Park, S., Ku, H., & Le, R. (2023). Predictive analytics for navigation data using sequence-based clustering and absorbing markov chains [Submitted for publication to *INFORMS Journal on Data Science*] (cit. on p. 93).

Park, S., Suresh, N. C., & Jeong, B.-K. (2008). Sequence-based clustering for web usage mining: A new experimental framework and ann-enhanced k-means algorithm. *Data & Knowledge Engineering*, *65*(3), 512–543. https://doi.org/https://doi.org/10.1016/j.datak.2008.01.002 (cit. on pp. 8, 11, 46, 53, 60).

Park, S., & Vasudev, V. (2017). Predicting web user's behavior: An absorbing markov chain approach. *Lecture Notes in Business Information Processing*, *296*. https://doi.org/10.1007/978-3-319-69644-7_17 (cit. on p. 53).

Parnes, D. (2010). Time to default and other sensitivities of credit ratings. *Quantitative Finance*, *10*, 947–952. https://doi.org/10.1080/14697680902946506 (cit. on p. 50).

Pasricha, P., Selvamuthu, D., & Arunachalam, V. (2017). Markov regenerative credit rating model. *The Journal of Risk Finance*, *18*(3), 311–325. https://doi.org/10.1108/JRF-09-2016-0123 (cit. on p. 50).

Petrone, D., Rodosthenous, N., & Latora, V. (2021). Artificial intelligence applied to bailout decisions in financial systemic risk management. *arXiv preprint arXiv:2102.02121* (cit. on p. 99).

Piccardi, T., Gerlach, M., Arora, A., & West, R. (2021). A large-scale characterization of how readers browse wikipedia. *ACM Transactions on the Web* (cit. on p. 52).

Pichler, A., Poledna, S., & Thurner, S. (2021). Systemic risk-efficient asset allocations: Minimization of systemic risk as a network optimization problem. *Journal of Financial Stability*, *52*. https://doi.org/10.1016/j.jfs.2020.100809 (cit. on pp. 97, 99, 141).

Plasse, J., & Adams, N. (2016). Handling delayed labels in temporally evolving data streams. *2016 IEEE International Conference on Big Data (Big Data)*, 2416–2424 (cit. on p. 10).

Poledna, S., Bochmann, O., & Thurner, S. (2017). Basel iii capital surcharges for g-sibs are far less effective in managing systemic risk in comparison to network-based, systemic risk-dependent financial transaction taxes. *Journal of Economic Dynamics and Control*, *77*, 230–246. https://doi.org/https://doi.org/10.1016/j.jedc.2017.02.004 (cit. on pp. 4, 97, 141).

Poledna, S., Martínez-Jaramillo, S., Caccioli, F., & Thurner, S. (2021). Quantification of systemic risk from overlapping portfolios in the financial system [Network models and stress testing for financial stability: the conference]. *Journal of Financial Stability*, *52*, 100808. https://doi.org/https://doi.org/10.1016/j.jfs.2020.100808 (cit. on p. 98).

Poledna, S., Molina-Borboa, J. L., Martínez-Jaramillo, S., van der Leij, M., & Thurner, S. (2015). The multi-layer network nature of systemic risk and its implications for the costs of financial crises. *Journal of Financial Stability*, *20*, 70–81. https://doi.org/https://doi.org/10.1016/j.jfs.2015.08.001 (cit. on pp. 4, 98, 103 sq., 111, 149).

Poledna, S., & Thurner, S. (2016). Elimination of systemic risk in financial networks by means of a systemic risk transaction tax. *Quantitative Finance*, *16*(10), 1599–1613. https://doi.org/10.1080/14697688.2016.1156146 (cit. on pp. 97, 141).

Pugh, N., & Park, H. (2021). High-order markov model for prediction of secondary crash likelihood considering incident duration. *Cogent Engineering*, *8*. https://doi.org/10.1080/23311916.2021.1978171 (cit. on p. 46).

Raftery, A. (1985a). A model for high-order markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, *47*, 528–539. https://doi.org/https://doi.org/10.1111/j.2517-6161.1985.tb01383.x (cit. on pp. 48, 61 sq.).

Raftery, A., & Tavare, S. (1994). Estimation and modelling repeated patterns in high order markov chains with the mixture transition distribution model. *Source: Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *43*, 179–199. https://www.jstor.org/stable/2986120 (cit. on p. 63).

Raftery, A. (1985b). A new model for discrete-valued time series: Autocorrelations and extensions. *Rassegna di Metodi Statistici ed Applicazioni*, *3*(4), 149–162 (cit. on pp. 63, 147).

Şahin, A., Sayimlar, A. D., Teksan, Z. M., & Albey, E. (2019). A markovian approach for time series prediction for quality control. *IFAC-PapersOnLine*, *52*, 1902–1907. https://doi.org/10.1016/j.ifacol.2019.11.480 (cit. on p. 46).

Sanchez, I. E. (2016). Optimal threshold estimation for binary classifiers using game theory. *F1000Research*, *5* (cit. on p. 33).

Sehgal, S., Mathur, S., Arora, M., & Gupta, L. (2018). Sovereign ratings: Determinants and policy implications for india. *IIMB Management Review*, *30*(2), 140–159. https://doi.org/https://doi.org/10.1016/j.iimb.2018.01.006 (cit. on pp. 5, 96).

Sharma, A., Jadi, D. M., & Ward, D. (2018). Evaluating financial performance of insurance companies using rating transition matrices. *The Journal of Economic Asymmetries*, *18*, e00102 (cit. on pp. 9, 49).

Silva, T. C., da Silva, M. A., & Tabak, B. M. (2017). Systemic risk in financial systems: A feedback approach. *Journal of Economic Behavior and Organization*, *144*, 97–120. https://doi.org/10.1016/j.jebo.2017.09.013 (cit. on p. 98).

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014, June). Deterministic policy gradient algorithms. In E. P. Xing & T. Jebara (Eds.),

*Proceedings of the 31st international conference on machine learning* (pp. 387–395, Vol. 32). PMLR. https://proceedings.mlr.press/v32/silver14.html (cit. on p. 111).

Singer, P., Helic, D., Taraghi, B., & Strohmaier, M. (2014). Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PloS one*, *9*(7), e102070 (cit. on p. 53).

So, M. K., Mak, A. S., & Chu, A. M. (2022). Assessing systemic risk in financial markets using dynamic topic networks. *Scientific Reports*, *12*. https://doi.org/10.1038/s41598-022-06399-x (cit. on p. 98).

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002 (cit. on pp. 27 sq.).

Somers, R. H. (1962). A new asymmetric measure of association for ordinal variables. *American Sociological Review*, *27*(6), 799–811. http://www.jstor.org/stable/2090408 (cit. on p. 31).

Strobl, S. (2016). Stand-alone vs systemic risk-taking of financial institutions. *The Journal of Risk Finance* (cit. on pp. 4, 96).

Taylor, H. M., & Karlin, S. (1998). *An introduction to stochastic modeling* (Third edition). Academic Press. (Cit. on p. 18).

Teteryatnikova, M. (2014). Systemic risk in banking networks: Advantages of "tiered" banking systems. *Journal of Economic Dynamics and Control*, *47*, 186–210. https://doi.org/10.1016/j.jedc.2014.08.007 (cit. on p. 138).

Thomas, L. C., Allen, D. E., & Morkel-Kingsbury, N. (2002). A hidden markov chain model for the term structure of bond credit risk spreads. *International Review of Financial Analysis*, *11*(3), 311–329. https://doi.org/https://doi.org/10.1016/S1057-5219(02)00078-9 (cit. on pp. 8, 49).

To, T. Y., Wu, E., & Zhang, L. (2022). Mind the sovereign ceiling on corporate performance. *Journal of Corporate Finance*, *75*, 102253. https://doi.org/https://doi.org/10.1016/j.jcorpfin.2022.102253 (cit. on p. 47).

Trueck, S., & Rachev, S. (2009). *Rating based modeling of credit risk: Theory and application of migration matrices*. Elsevier Science. https://books.google.ca/books?id=C8mxdgm%5C_K8EC (cit. on pp. 2, 31 sqq.).

Trueck, S., & Rachev, S. (2005). Credit portfolio risk and pd confidence sets through the business cycle. *Available at SSRN 675622* (cit. on pp. 2, 147).

Vermeer, S., & Trilling, D. (2020). Toward a better understanding of news user journeys: A markov chain approach. *Journalism Studies*, *21*, 879–894. https://doi.org/10.1080/1461670x.2020.1722958 (cit. on p. 52).

Wong, K. I., Wong, S. C., Bell, M. G. H., & Yang, H. (2005). Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing markov chain approach. *Journal of Advanced Transportation*, *39*, 81–104. https://doi.org/https://doi.org/10.1002/atr.5670390107 (cit. on p. 46).

Xu, Q., Ding, C., Liu, J., & Luo, B. (2015). Pca-guided search for k-means. *Pattern Recognition Letters*, *54*, 50–55. https://doi.org/https://doi.org/10.1016/j.patrec.2014.11.017 (cit. on p. 22).

Yakovyna, V., & Symets, I. (2021). Reliability assessment of cubesat nanosatellites flight software by high-order markov chains. *Procedia Computer Science*, *192*, 447–456. https://doi.org/10.1016/j.procs.2021.08.046 (cit. on p. 46).

Yang, H. F., Liu, C. L., & Chou, R. Y. (2020). Bank diversification and systemic risk. *Quarterly Review of Economics and Finance*, *77*. https://doi.org/10.1016/j.qref.2019.11.003 (cit. on p. 138).

# Appendix A

# First-Order Study Auxiliary Details

## A.1 Algorithm

**Algorithm 1** The classification algorithm for the rating prediction and transition direction scenarios.

Initialize the number of clusters $C$, the number of folds $K$, state space $\mathcal{S}$, input time $t$, predicted time $t'$, time step $\tau$, order $k$, and classification labels $\mathcal{L}$
Sample $S_{\text{size}}$ sequences from the data set
**for** $i = 1, M$ **do**
    For $1 \leq m \leq S_{\text{size}}$, generate the sequence matrix based on $\mathbf{x}_m(t)$
    Begin $K$-fold cross-validation process and split $K$ folds
    **for** $j = 1, K$ **do**
        Let the test set be the $K_j$ fold and form the training set by combining folds $K_l$ for $j \neq l$
        Cluster the training set using PCA-guided K-means.
        Generate the representative transition matrices $\mathbf{P}_c^{\tau}$ for each cluster $c$
        For all sequences in the test set, assign to a cluster based on respective $\mathbf{S}_m$
        Predict $\hat{X}_{t'}^{m}(\Delta t)$ and $\hat{d}_m(t, t')$ for each sequence $m$
    **end for**
    Calculate the $\text{F1}_{\mu,i}$, $\text{AA}_i$
**end for**
Average the performance measures across $M$ experiments

**Algorithm 2** The classification algorithm for default behaviour prediction.

Initialize the number of clusters $C$, the number of folds $K$, state space $\mathcal{S}$, input time $t$, predicted time $t'$, time step $\tau$, order $k$, and classification labels $\mathcal{L}$
Sample $S_{\text{size}}$ sequences from the data set
**for** $i = 1, M$ **do**
    For $1 \leq m \leq S_{\text{size}}$, generate the sequence matrix based on $\mathbf{x}_m(t)$
    Begin $K$-fold cross-validation process and split $K$ folds
    **for** $j = 1, K$ **do**
        Let the test set be the $K_j$ fold and form the training set by combining folds $K_l$ for $j \neq l$
        Cluster the training set using PCA-guided K-means.
        Generate the representative transition matrices $\mathbf{P}_c^{\tau}$ for each cluster $c$
        For all sequences in the test set, assign to a cluster based on respective $\mathbf{S}_m$
        Calculate $\mathbf{r}_c^{\tau}(X_t^m)$ for each sequence $m$
        Assign $\hat{y}_m(\theta)$ for each sequence $m$
    **end for**
    Calculate the $\text{F1}_i$, $\text{AA}_i$, $\text{Pr}_i$, $\text{Re}_i$, $d_{yx}$
**end for**
Average the performance measures across $M$ experiments

## A.2 Confusion Matrices

In Appendix A.2, we present the confusion matrices for each classification scenario to highlight the sample size and balance of the classes.

| | | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | AAA | AA | A | BBB | BB | B | C |
| Actual | AAA | 19.3 | 6.7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | AA | 4.1 | 44.0 | 8.9 | 23.7 | 3.2 | 0.0 | 2.0 |
| | A | 0.2 | 8.9 | 55.8 | 35.5 | 14.9 | 0.8 | 3.8 |
| | BBB | 0.0 | 0.1 | 13.9 | 100.5 | 13.8 | 1.0 | 9.6 |
| | BB | 0.0 | 0.1 | 3.3 | 18.3 | 69.5 | 6.9 | 1.8 |
| | B | 0.0 | 0.0 | 1.0 | 3.5 | 4.3 | 14.3 | 9.0 |
| | C | 0.6 | 0.4 | 0.6 | 1.7 | 0.9 | 0.4 | 82.4 |

(a) Clustering model using $\tau = 15$.

| | | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | AAA | AA | A | BBB | BB | B | C |
| Actual | AAA | 15.0 | 7.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | AA | 1.0 | 26.7 | 26.2 | 23.1 | 7.0 | 0.0 | 2.0 |
| | A | 0.0 | 5.0 | 45.8 | 47.2 | 18.0 | 0.0 | 4.0 |
| | BBB | 0.0 | 1.0 | 6.0 | 99.0 | 24.0 | 0.0 | 9.0 |
| | BB | 0.0 | 0.0 | 2.0 | 12.0 | 83.0 | 0.0 | 3.0 |
| | B | 0.0 | 0.0 | 0.0 | 7.0 | 11.0 | 0.0 | 14.0 |
| | C | 0.0 | 1.0 | 2.0 | 8.0 | 23.0 | 0.0 | 53.0 |

(b) Benchmark model using $\tau = 15$.

| | | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | AAA | AA | A | BBB | BB | B | C |
| Actual | AAA | 16.5 | 4.3 | 0.2 | 0.0 | 0.0 | 0.0 | 0.0 |
| | AA | 1.3 | 35.3 | 8.3 | 1.7 | 0.0 | 0.0 | 0.3 |
| | A | 0.0 | 7.7 | 64.2 | 40.0 | 5.7 | 1.0 | 3.3 |
| | BBB | 0.0 | 1.0 | 7.7 | 124.0 | 32.4 | 0.4 | 7.5 |
| | BB | 0.0 | 0.9 | 1.4 | 10.4 | 96.8 | 1.4 | 7.1 |
| | B | 0.0 | 0.0 | 0.1 | 2.4 | 8.6 | 11.9 | 6.9 |
| | C | 0.0 | 0.0 | 0.0 | 2.4 | 0.8 | 3.3 | 72.4 |

(c) Clustering model using $\tau = 5$.

| | | Predicted | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | AAA | AA | A | BBB | BB | B | C |
| Actual | AAA | 16.0 | 4.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | AA | 0.0 | 31.0 | 14.0 | 2.0 | 0.0 | 0.0 | 0.0 |
| | A | 0.0 | 4.0 | 65.0 | 43.0 | 6.0 | 1.0 | 3.0 |
| | BBB | 0.0 | 1.0 | 4.0 | 132.0 | 31.0 | 3.0 | 2.0 |
| | BB | 0.0 | 1.0 | 2.0 | 11.0 | 103.0 | 0.0 | 1.0 |
| | B | 0.0 | 0.0 | 0.0 | 4.0 | 11.0 | 15.0 | 0.0 |
| | C | 0.0 | 0.0 | 0.0 | 4.0 | 15.0 | 10.0 | 50.0 |

(d) Benchmark model using $\tau = 5$.

Table A.1: The confusion matrices using input date 2001-01-01 under the credit rating prediction scenario.

| | | Predicted | | |
|---|---|---|---|---|
| | | -1 | 0 | 1 |
| Actual | -1 | 65.1 | 21.9 | 4.0 |
| | 0 | 16.0 | 261.7 | 43.3 |
| | 1 | 11.4 | 76.8 | 89.9 |

(a) Clustering model with $\tau = 15$.

| | | Predicted | | |
|---|---|---|---|---|
| | | -1 | 0 | 1 |
| Actual | -1 | 39.6 | 38.0 | 13.4 |
| | 0 | 127.7 | 160.7 | 32.6 |
| | 1 | 61.8 | 100.8 | 15.4 |

(b) Benchmark model with $\tau = 15$.

| | | Predicted | | |
|---|---|---|---|---|
| | | -1 | 0 | 1 |
| Actual | -1 | 34.4 | 32.3 | 0.3 |
| | 0 | 17.6 | 380.5 | 13.9 |
| | 1 | 3.6 | 92.4 | 15.1 |

(c) Clustering model with $\tau = 5$.

| | | Predicted | | |
|---|---|---|---|---|
| | | -1 | 0 | 1 |
| Actual | -1 | 0.0 | 67.0 | 0.0 |
| | 0 | 0.0 | 412.0 | 0.0 |
| | 1 | 0.0 | 111.0 | 0.0 |

(d) Benchmark model with $\tau = 5$.

Table A.2: The confusion matrices using input date 2001-01-01 under the transition direction prediction scenario.

| | | Predicted | |
|---|---|---|---|
| | | 1 | 0 |
| Actual | 1 | 82.7 | 7.3 |
| | 0 | 40.3 | 459.7 |

(a) Clustering model with $\tau = 15$.

| | | Predicted | |
|---|---|---|---|
| | | 1 | 0 |
| Actual | 1 | 68.0 | 22.0 |
| | 0 | 128.6 | 371.4 |

(b) Benchmark model with $\tau = 15$.

| | | Predicted | |
|---|---|---|---|
| | | 1 | 0 |
| Actual | 1 | 65.0 | 3.0 |
| | 0 | 22.6 | 499.4 |

(c) Clustering model with $\tau = 5$.

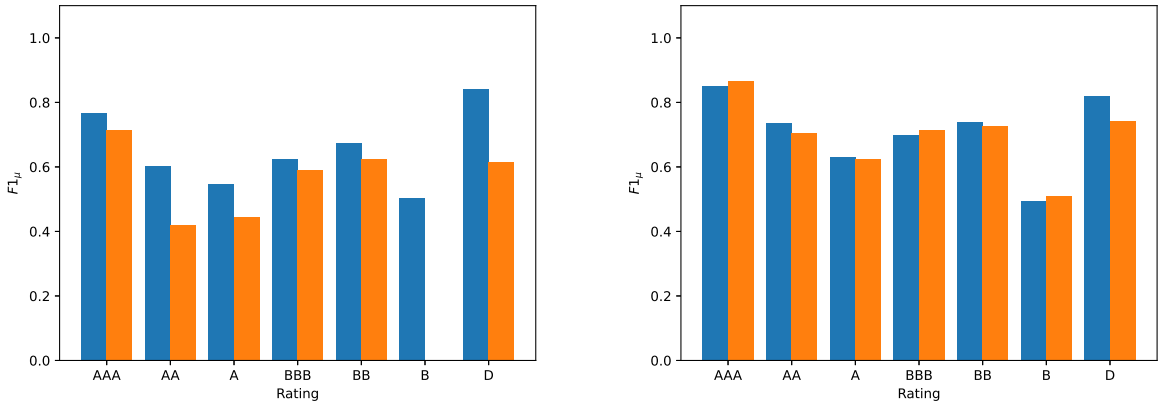
| | | Predicted | |
|---|---|---|---|
| | | 1 | 0 |
| Actual | 1 | 54.4 | 13.6 |
| | 0 | 90.8 | 431.2 |

(d) Benchmark model with $\tau = 5$.

Table A.3: The confusion matrices using input date 2001-01-01 under the default behaviour prediction scenario.

## A.3 Performance Measures by Credit Rating

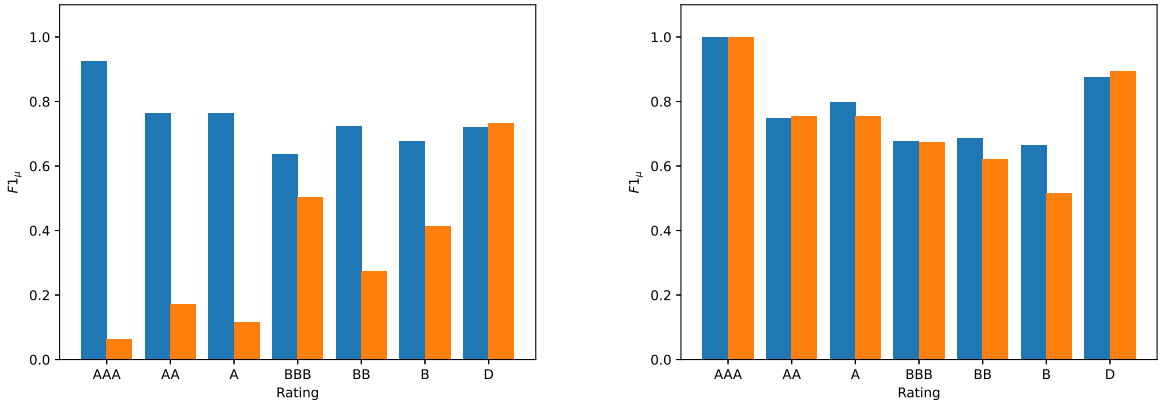
In Appendix A.3, we present the performance of the clustering model against the benchmark model across each credit rating using the *F1*-score for each classification scenario. The input date used for the results in the following section was set to 2001-01-01. The following results are generated by averaging over 1000 5-Fold cross-validation runs.




(a) Credit rating prediction with $\tau = 15$.

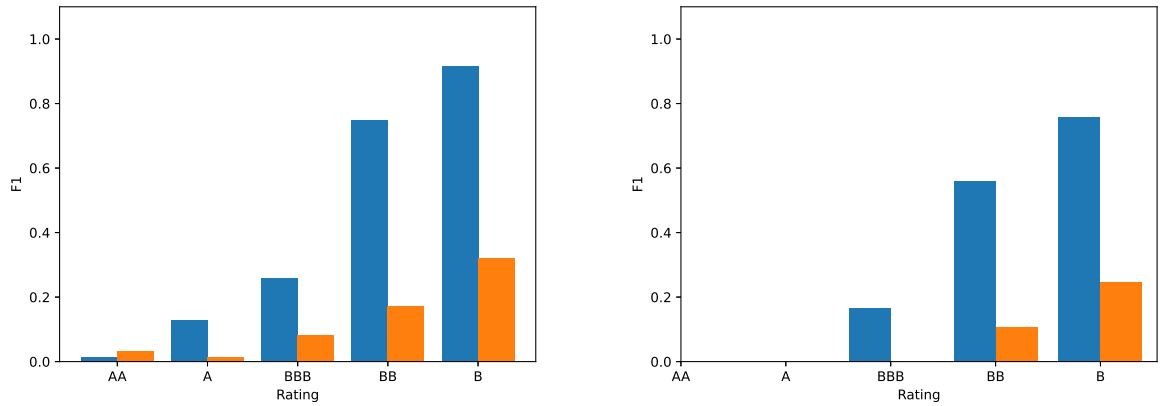
(b) Credit rating prediction with $\tau = 5$.

Figure A.1: The credit rating prediction performance of the model in terms of the F1 score for $\tau = 15$ and 5. The left (blue) and right (orange) bar graphs are the results using the clustering and benchmark models respectively.




(a) Transition direction prediction with $\tau = 15$. (b) Transition direction prediction with $\tau = 5$.

Figure A.2: The transition direction performance of the model in terms of the F1 score for $\tau = 15$ and 5. The left (blue) and right (orange) bar graphs are the results using the clustering and benchmark models respectively.




(a) Default behaviour prediction with $\tau = 15$. (b) Default behaviour prediction with $\tau = 5$.

Figure A.3: The default behaviour performance of the model in terms of the F1 score for $\tau = 15$ and 5. The left (blue) and right (orange) bar graphs are the results using the clustering and benchmark models respectively.

# Appendix B

# High-Order Study Auxiliary Details

## B.1 Algorithms

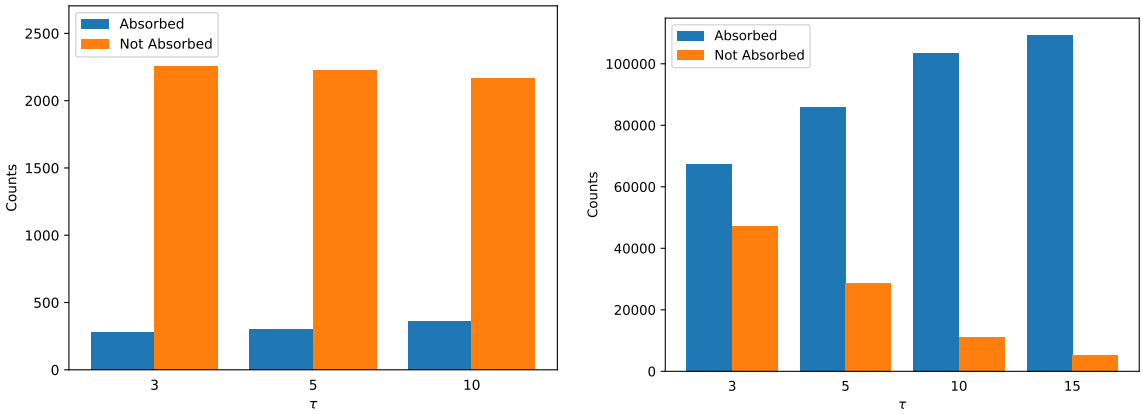
**Algorithm 3** The classification procedure using the high-order SBC model using the credit rating data.

Initialize the number of clusters $C$, the number of folds $K$, state space $\mathcal{S}$, input time $t$, time step $\tau$, order $k$, and classification labels $\mathcal{L}$
Sample $S_{\text{size}}$ sequences from the data set
**for** $i = 1, M$ **do**
    For sequences $1 \leq m \leq S_{\text{size}}$, generate the sequence matrix based on $\mathbf{x}_m(t')$
    Begin shuffled $K$-fold cross-validation process and split $K$ folds
    **for** $j = 1, K$ **do**
        Let the test set be the $K_j$ fold and form the training set by combining folds $K_l$ for $j \neq l$
        Cluster the training set using K-means.
        Generate the representative transition matrices $\mathbf{P}_c^k$ for each cluster $c$
        Calculate $\mathbf{B}_c^{k,(\tau)}$
        Assign $\hat{y}_m(\theta)$ for each sequence $m$ based on $\mathbf{x}_m(t' \mid k)$.
    **end for**
    Calculate the $\text{F1}_i$, $\text{Acc}_i$, $\text{Pr}_i$, and $\text{Re}_i$
**end for**
Average the performance measures across $M$ experiments

**Algorithm 4** The classification procedure using the high-order SBC model using the web-usage data.

Initialize the number of clusters $C$, the number of folds $K$, state space $\mathcal{S}$, input time $t$, time step $\tau$, order $k$, and classification labels $\mathcal{L}$
Sample $S_{\text{size},i}$ sequences from the data set, $M$ times without replacement
**for** $i = 1, M$ **do**
    For sequences $1 \leq m \leq S_{\text{size},i}$, of each sample, generate the sequence matrix based on $\mathbf{x}_m(t')$
    Begin $K$-fold cross-validation process and split $K$ folds
    **for** $j = 1, K$ **do**
        Let the test set be the $K_j$ fold and form the training set by combining folds $K_l$ for $j \neq l$
        Cluster the training set using K-means.
        Generate the representative transition matrices $\mathbf{P}_c^k$ for each cluster $c$
        Calculate $\mathbf{m}_c^k$ and $\mathbf{B}_c^{k,(\tau)}$
        Assign $\hat{y}_m(\theta)$ and $\tilde{\mathbf{m}}[\mathbf{x}_m(t' \mid k)]$ for each sequence $m$ for $t' > k$.
    **end for**
    Calculate the $\text{F1}_i$, $\text{Acc}_i$, $\text{Pr}_i$, and $\text{Re}_i$
    Calculate the $\text{MAE}_i$ and $\text{MAEPS}_i$
**end for**
Average the performance measures across $M$ experiments

## B.2 Binary Classification Label Distribution




(a) Credit rating data set

(b) Wikispeedia data set

Figure B.1: The classification label distribution for a sequence being absorbed or not: The distribution in a was formed from the 677, 653, 630, and 569 valid sequences. The distribution in b was aggregated from $M = 10$ samples.

# Appendix C

# Constrained DDPG Algorithm

**Algorithm 5** Constraint DDPG

Initialize the multi-layer network
Randomly initialize critic network $Q(s, a \mid \theta^Q)$ and actor $\mu(s \mid \theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \longleftarrow \theta^Q, \theta^{\mu'} \longleftarrow \theta^\mu$
Initialize replay buffer $\mathcal{D}$
**for** episode = 1, $N_{\text{episode}}$ **do**
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive an initial random observation state $s_1$
    **for** $t = 1, T$ **do**
        Calculate $a_t = \mu(s_t \mid \theta^\mu) + \mathcal{N}$ according to the current policy and exploration noise
        **for** $\alpha = 1, M$ **do**
            Take partition $\mathbf{u}^\alpha$ from $a_t$ and pass to the safety layer to find $\tilde{\mathbf{x}}^\alpha$
            Use $\tilde{\mathbf{x}}^\alpha$ to calculate $\Delta\mathbf{L}^\alpha(t)$ by equation (4.46)
            Calculate the new network by $\mathbf{L}^\alpha(t+1) = \mathbf{L}^\alpha(t) + \Delta\mathbf{L}^\alpha(t)$
        **end for**
        Calculate the reward $r_t$ based on equation (4.35) or (4.36) and observe the new state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{D}$
        Sample a random minibatch of $N_{\text{mini}}$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $\mathcal{D}$
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'})$
        Update critic by minimizing the loss:

$$L_{\text{loss}} = \frac{1}{N_{\text{mini}}} \sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2$$

        Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N_{\text{mini}}} \sum_i \nabla_a Q(s, a \mid \theta^Q) \mid_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s \mid \theta^\mu)|_{s_i}$$

        Update the target networks:

$$\theta^{Q'} \longleftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \longleftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

        **if** $R(\mathbf{L}(t+1), \mathbf{e}) \geq R(\mathbf{L}(t), \mathbf{e})$ **or** $t = T$ **then**
            End the episode
        **end if**
    **end for**
**end for**