Information Sciences 432 (2018) 376–391

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Finding potential lenders in P2P lending: A Hybrid Random Walk Approach

Hefu Zhang, Hongke Zhao, Qi Liu, Tong Xu, Enhong Chen*, Xunpeng Huang

Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China, China

ARTICLE INFO

Article history:
Received 11 March 2017
Revised 8 December 2017
Accepted 9 December 2017
Available online 11 December 2017

2010 MSC:
00-01
99-00

Keywords:
P2P lending
Random walk
Hybrid recommender system

ABSTRACT

P2P lending is a burgeoning online service that allows individuals to directly borrow money from each other. In these platforms, each loan has a specific duration for raising money from lenders. Following the "all-or-nothing" rule, many loans fail due to insufficient pledges/money in their funding durations. Thus, automatically accessing and finding potential lenders early is crucial for loans. However, this problem has some unique challenges (e.g., the temporality of loan) that are still being explored. To that end, in this paper, we present a holistic study on finding potential lenders in P2P lending. Specifically, we propose a hybrid random walk approach, i.e., $\mathcal{RWH}$, by combining both collaborative filtering and content-based filtering, which can be adapted to loans at any funding progress (e.g., the starting progress). In the content-based filtering of $\mathcal{RWH}$, the model extract dynamic features and adopt bagging to estimate the similarity between loans. Further more, to adapt to the loan temporality, $\mathcal{RWH}$ is dynamically established with temporal loans and lenders via a sliding window. Finally, we systematically evaluate our method on large-scale real-world datasets. The experimental results clearly demonstrate the effectiveness and robustness of our solutions.

1. Introduction

P2P lending is an online service that allows individuals to directly borrow money from each other without going through traditional financial intermediaries. In recent years, P2P lending has become a fast growing financial market attracting a massive number of users. For instance, in November 2015, Prosper announced that it had more than 2 million members and more than $5 billion in funded loans.[1]

There are two roles in P2P lending: *borrowers* who borrow money from others and *lenders* who lend their money to borrowers. A borrower seeking to borrow money creates a loan listing on the platform to raise pledges. As shown in Fig. 1, each loan has its declared funding amount and duration (often less than a month). All loans obey the "all-or-nothing" rule [13,36], which means that a loan will succeed and take all pledges if and only if this loan receives a sufficient amount of money in its funding duration. If a loan fails, the received pledges will also expire. Unfortunately, in P2P lending markets, only 20%–40% of loans succeed [15,23]. Further more, more than 60% of failed loans received less than 20% of their funding

* Corresponding author.
E-mail addresses: zhf2011@mail.ustc.edu.cn (H. Zhang), zhhk@mail.ustc.edu.cn (H. Zhao), qiliuql@ustc.edu.cn (Q. Liu), tongxu@ustc.edu.cn (T. Xu), cheneh@ustc.edu.cn (E. Chen), hxpsola@mail.ustc.edu.cn (X. Huang).

[1] https://www.prosper.com/abouts

https://doi.org/10.1016/j.ins.2017.12.017
0020-0255/

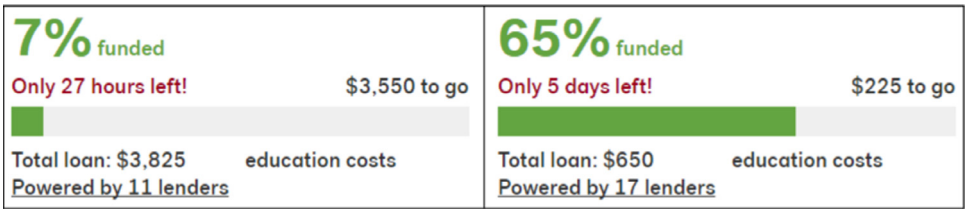


Fig. 1. Loans in P2P lending.

goals. For instance, the first loan in Fig. 1 ultimately failed because it did not reach its declared goal ($ 3,825). Thus, it is crucial for loans to raise sufficient money in time.

Although some studies of this issue have been performed, e.g., predicting loan success [15,35] and analysing funding dynamics [6,26,43,45], how to automatically find potential lenders for loans and further accelerate funding progress remain to be explored. Specifically, the potential lenders are those who are most likely to pledge a specific loan. According to a previous study, earlier pledges of a loan affect the decision-making of subsequent lenders and even the funding result [35]. *Thus, helping loans find potential lenders as early as possible is very important.* In addition, considering that lenders often miss loans of interest when browsing massive loans [42], lenders also require recommendation assistance.

However, this problem has some unique challenges. First, in P2P lending, the funding progress of a loan is *dynamic* and keeps changing during its funding progress (see Fig. 1). That is, as the received pledges of a loan increase, the potential lenders may also change. Second, every loan has *temporality* [8], i.e., a loan will eventually end whether it succeeds or expires. This characteristic greatly distinguishes our study from other recommendation problems. Only lenders who pledge during the loan's funding duration are useful for recommendation. Last but not least, many lenders periodically and continuously lend to multiple loans. How to explore these *time effects* and lending behaviors is also very much open.

In this paper, we propose to help raising-money loans dynamically find potential lenders. Specifically, we propose a **R**andom **W**alk based **H**ybrid approach, i.e., $\mathcal{RWH}$, for this task. Our contributions are as follows:

- First, we propose a hybrid method $\mathcal{RWH}$ that combines both collaborative filtering and content-based filtering. By adding loan-loan routes into a loan-lender collaborative filtering framework, $\mathcal{RWH}$ can be adapted to loans at any *dynamic* progress and works robustly.
- Second, in the content-based routes, instead of using loans' current similarity, a bagging learner is adopted with extracted *dynamic* features to estimate the final similarity of loan nodes, which is more effective and forward-looking.
- Third, $\mathcal{RWH}$ is dynamically established on small-scale networks with temporal loans and lenders to speed up the recommendation procedure and tackle *temporality* and *time effects* in our task.

In experiments, we systematically evaluate our method on large-scale real-world datasets, and the experimental results clearly demonstrate the effectiveness and robustness of $\mathcal{RWH}$. The rest of the paper is organized as follows. Section 2 reviews the relevant works in P2P lending and recommendations. Section 3 formalizes the studied problem and introduces the $\mathcal{RWH}$. Section 4 designs experiments to evaluate our approach. Finally, Section 5 concludes the paper and suggests future research directions.

2. Related works

In this section, we briefly introduce related works, which can be grouped into two categories.

2.1. P2P lending

Over the last few years, many research efforts have been devoted to the P2P lending field. Readers can refer to [3,41] for an overview of P2P lending. In this area, some studies have focused on analysing user behaviors, such as lending behaviors [6,27], and social interactions [9]. Other studies have aimed to evaluate loan quality or risk, such as [10,28,30]. In particular, the prediction of loan success or fully funded probability [15,20,35] and analyses of the dynamic of funding progression [6,26,38,43,45] have been well-studied. For example, in [15], the authors studied both borrower-related determinants (e.g., credit) and loan-related determinants (e.g., interest rate) of funding success on Prosper. In [26], the authors proposed to predict the funding of projects in Kickstarter by inferring the impacts of social media. Zhao et al. [43] studied the relationship between funding dynamics and the hidden market states using a Markov chain.

In addition, some researchers have studied loan or project recommendations in P2P lending or crowdfunding [32,44]. In [2], the authors proposed to recommend investors found on Twitter for Kickstarter projects. Lee et al. [19] proposes a fairness-aware loan recommendation system for Kiva that optimizes both accuracy and fairness. Zhang et al. [39] proposes a personalized recommendation model for P2P lending platforms applying surplus maximization. Zhao et al. [42] studied the loan portfolio selection problem in P2P lending and proposed a multi-objective selection strategy. However, all these recommendation techniques in P2P lending recommend loans to lenders. To the best of our knowledge, finding potential

Table 1
Notations

Notation	Description
U	the set of lenders $\{u_1, u_2, \ldots\}$
V	the set of loans $\{v_1, v_2, \ldots\}$
UV_i	the set of lenders who pledge loan v_i
VU_i	the set of loans that lender u_i pledges
$M_{i,j}$	the financial amount u_i pledges to v_j
$sim_{i,j}$	the content similarity of loans v_i and v_j
$P^{(t)}(u_i)$	the probability of locating lender u_i at time t
$P^{(t)}(v_i)$	the probability of locating loan v_i at time t

lenders for loans in P2P lending platforms has not been explored, and this study is the first to attempt to recommend potential lenders in P2P lending by exploring the dynamic of funding progression.

2.2. Recommender system

A recommender system [1,4] provides suggestions of items that may interest users. Generally, recommendations are based on collaborative filtering (CF) [29], content-based methods (CB) [25] and hybrid techniques [18]. CF methods make recommendations by exploring the user-item relationship in different ways, e.g., memory-based CF [37] and model-based CF [14,17,24]. Content-based methods build a preference profile for a user by analysing the items selected by her. Recommendations are given by comparing the content of possible items with the user's profile and the most similar items are recommended. Usually, CF methods typically have greater accuracy than CB methods, but CF methods can occasionally suffer from cold-start and sparsity problems [12]. To overcome the shortcomings of these two kinds of methods, many efforts have sought to combine CF and CB, i.e., hybrid approaches. A simple strategy is to combine the results of CF and CB methods after implementing them separately. A second strategy is to add CF characteristics to advance CB methods or to add CB characteristics to enhance CF. In this paper, we also integrate CB filtering into the CF recommendation framework.

Random walk is a specific technique applied in recommender systems [11,16,40]. When applied for recommendations, a random walk is performed on a network with user nodes and item nodes. The random walk jumps via user-item connections, and the probability of visiting a node is determined by ratings [11,16]. Random walk usually performs well with respect to accuracy, but the time cost is always very high since they are constructed on networks with all items and users.

In this study, we add item-item, i.e., loan-loan, connections to the walking network to perform a hybrid recommendation. By adding loan-loan routes into a loan-lender collaborative filtering framework, the approach can be applied to all loans at any stage of funding progress, e.g., even cold-start loans. In addition, random walk networks are dynamically established with temporal loans and lenders to solve temporality in our problem and accelerate efficiency.

3. Methodology

In this section, we first introduce the framework of $\mathcal{RWH}$ in detail. Then, we apply a bagging method to learn loan similarity in loan-loan routes of $\mathcal{RWH}$. Finally, we will introduce how to dynamically construct networks with temporal loans and lenders and perform $\mathcal{RWH}$ on these networks. Table 1 lists the mathematical notations used in this paper.

3.1. Problem statement

Formally, given all historical loans $V = \{v_1, v_2, \ldots\}$, current raising-money loans $V_S = \{v_{s1}, v_{s2}, \ldots\}$, and lenders $U = \{u_1, u_2, \ldots\}$ with their pledging records in the market, our goal is to automatically find potential lenders from U for each raising-money loan $v_s \in V_S$ as early as possible.

Challenges. Finding potential lenders in P2P lending has the following unique challenges:*dynamic* and *temporality*.

Dynamic. In P2P lending, the funding progress of a loan is dynamic and continues to change during the fundraising period. Specifically, when a loan has just begun (only received a few pledges from lenders), it is difficult to access potential lenders through collaborative filtering due to the cold start and sparsity [12,33]. As the fundraising of the loan increases(receives many pledges from incremental lenders), the effects of collaborative filtering may improve. Thus, the approach should adapt to loans at different stage of funding progress.

Temporality. Both loans and lenders have temporalities. Lenders should be recommended to a loan before its funding duration ends, which requires a fast recommending procedure. Lenders in P2P lending also have temporalities: i.e., in a certain period, only a small portion of temporal lenders are available due to periodic pledging. Thus, if the approach accesses candidate lenders without distinguishing temporal lenders, the quality of the user experience may decrease, especially for those lenders in inactive periods.

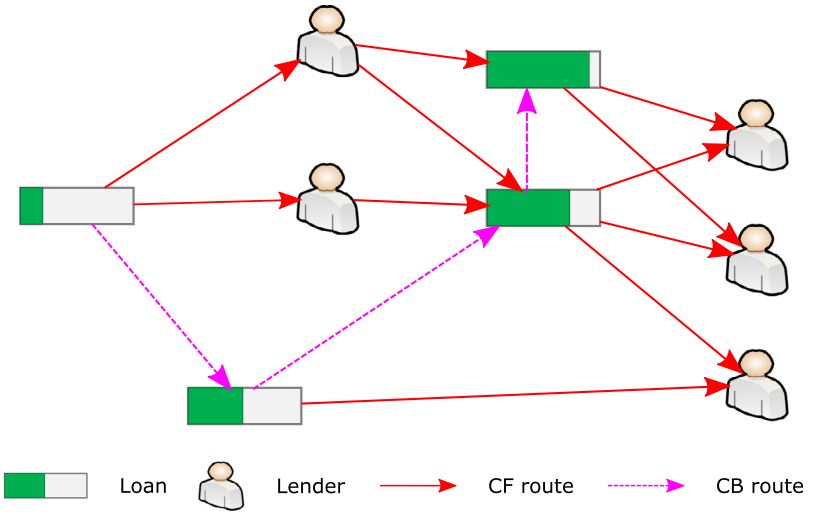


Fig. 2. Graph structure of $\mathcal{RWH}$.

3.2. Framework of $\mathcal{RWH}$

To address these challenges, $\mathcal{RWH}$ combines collaborative and content-based filterings.

Initially, $\mathcal{RWH}$ adjusts the weights of both collaborative filtering routes and content-based filtering routes for different loans, which can be adapted to different stage of funding progress for loans. In addition, for loan-loan routes, $\mathcal{RWH}$ adopts a bagging learner with dynamic features to estimate the final loan similarities, which is more effective and forward-looking.

To adapt to the temporality, $\mathcal{RWH}$ is dynamically established on small-scale networks with temporal loans and lenders using a sliding window. With small and dynamic networks, $\mathcal{RWH}$ will converge faster. When the funding progress of a loan changes, $\mathcal{RWH}$ can update the network and perform the walk again to obtain new recommendation results. The small scale and dynamics of $\mathcal{RWH}$ make updating recommendations feasible.

In this subsection, we will introduce the framework of $\mathcal{RWH}$ and its walking processes. Fig. 2 shows the graph structure of $\mathcal{RWH}$. The red lines in $\mathcal{RWH}$ are the seeking routes from loans to lenders or from lenders to loans which can be regarded as collaborative filtering, whereas the purple lines are the seeking routes from loans to loans which can be regarded as content-based filtering. The loan-lender routes are built according to the lenders' lending or pledging behaviors on loans, and the loan-loan routes are built according to the loan similarity which will be detailed in Section 3.3. There are no lender-lender routes in $\mathcal{RWH}$ because there is no external lender profile information to build them. There is also no need to build lender-lender connections with their common lending because this type of route can be indirectly implemented via the lender-loan-lender routes.

The combination of both loan-lender routes and loan-loan routes in $\mathcal{RWH}$ allows it to adapt to all loans at any funding progress. As shown in Fig. 2, the green part of a loan represents its funding progress. For example, the starting loan has only received a small amount of money/pledges, whereas the other loans have reached the middle or later periods of their funding durations. The funding progress bar of the loan node will influence the route selection of $\mathcal{RWH}$.

After the routes of the graph are built, we can consider how $\mathcal{RWH}$ walks. Since the recommendations of potential lenders are for each raising-money loan, i.e., $v_s \in V_S$, $\mathcal{RWH}$ starts from a target loan v_s at each time. For each raising-money loan v_s, $\mathcal{RWH}$ works as follows.

At any time, $\mathcal{RWH}$ **may be at either a lender node** u_i **or a loan node** v_j. For any node position, $\mathcal{RWH}$ has two options for the next move:

- with probability $\alpha\,(0 < \alpha < 1)$, $\mathcal{RWH}$ restarts from the starting loan node v_s\$;
- with probability $1-\alpha$, $\mathcal{RWH}$ continues walking.

In Fig. 2, loan nodes link both other loan nodes and lender nodes, whereas lender nodes only link loan nodes. Thus, $\mathcal{RWH}$ will make different selections at different types of nodes when it continues to seek walk.

If $\mathcal{RWH}$ **is at a loan node** v_j . First, $\mathcal{RWH}$ will choose to move to loans (purple routes) or lenders (red routes). In $\mathcal{RWH}$, the loan-loan routes (purple routes) represent the content-based connections and the loan-lender routes (red routes) represent the collaborative filtering connections. The route selection follows a random variable R with a *Bernoulli distribution*: $R = 1$ means $\mathcal{RWH}$ will select loan-loan routes, whereas $R = 0$ means $\mathcal{RWH}$ will select loan-lender routes. Different loan nodes should have different Bernoulli distributions which might be mainly influenced by the dynamic funding progress of this loan. Intuitively, if a loan has a low funded percentage, the probability of moving to a loan should be high since there are only less loan-lender route options; on the contrary, for an almost fully funded loan, it should have a higher probability

of moving to a lender. Without loss of generality, the route selection probability of $\mathcal{RWH}$ at a loan node v_j can be calculated as follows:

$$\begin{aligned} p(R=1|v_j) &= \beta(v_j,\theta) = \frac{2e^{-\theta FP_{v_j}}}{1+e^{-\theta FP_{v_j}}}, \\ p(R=0|v_j) &= 1-\beta(v_j,\theta) = \frac{1-e^{-\theta FP_{v_j}}}{1+e^{-\theta FP_{v_j}}}, \end{aligned} \tag{1}$$

where FP_{v_j} represents the funded percentage of v_j and $\theta \in \mathbb{R}^+$ is a parameter that can be used to adjust the value of $\beta(.)$. When FP_{v_j} =0, $\beta(v_j, \theta)$=1, because v_j does not receive any pledges and collaborative filtering cannot work. As the loan receives more pledges, $p(R=1|v_j)$ decreases, whereas $p(R=0|v_j)$ increases. The effects of θ will be discussed in Section 4. Thus, when $\mathcal{RWH}$ is at a loan node v_j, it will

- move to other loans along the loan-loan routes with probability $p(R=1)$;
- move to other lenders who have pledged on v_j along the loan-lender routes with probability $p(R=0)$.

If $\mathcal{RWH}$ selects the loan-loan routes when $R=1$, the probability of moving from loan node v_j to another specific loan node v_k is calculated based on their similarity:

$$p(v_k|v_j, R=1) = \frac{sim_{j,k}}{\sum_{v_{k'}\in V} sim_{j,k'}}, \tag{2}$$

where $sim_{j,k}$ is the similarity between loan v_j and v_k. The similarity computation is another important problem in our study and will be introduced in Section 3.3.

By contrast, if $\mathcal{RWH}$ selects a particular lender route when $R=0$, the probability of moving from loan node v_j to a specific lender node u_i can be calculated as follows:

$$p(u_i|v_j, R=0) = \frac{M_{i,j}}{\sum_{u_k\in UV_j} M_{k,j}}, \tag{3}$$

where UV_j is the collection of lenders who have pledged/lent to loan v_j before the current time and $M_{i,j}$ is the financial amount that u_i pledged to v_j. A loan receives all pledges in a predefined short time, e.g., one week or ten days. Thus, we do not consider the time effect in loan-lender probabilities in this paper.

If $\mathcal{RWH}$ is at a lender node u_i. $\mathcal{RWH}$ will move to the loans that u_i lent in the past. The probability of selecting a particular loan v_j from lender u_i is defined as similar to the move from loan node to lender node:

$$p(v_j|u_i) = \frac{M_{i,j}}{\sum_{v_k\in VU_i} M_{i,k}}, \tag{4}$$

where VU_i is the collection of loans that lender u_i has pledged/lent to before the current time. However, in contrast to the multiple pledges of a loan above, one lender's different behaviors may be long-term, and her time-varying lending behaviors may have great time effects on her subsequent pledges. For example, a lender's recent pledges on raising-money loans have greater effects on her next selections than her pledges on ended loans because recent pledges and next selections form a portfolio [44]. Thus, we use a time-aware factor $f(Time_{i,j})$ to adjust the selection probability. Let $M'_{i,j} = M_{i,j} f(Time_{i,j})$, where $Time_{i,j}$ is the number of days since the pledge behavior happened. The selection probability with the time-aware factor is as follows:

$$p(v_j|u_i) = \frac{M'_{i,j}}{\sum_{v_k\in VU_i} M'_{i,k}}. \tag{5}$$

The value of function $f(Time_{i,j})$ should decrease when the value of $Time_{i,j}$ increases. For instance, the exponential function $f(Time_{i,j}) = e^{-Time_{i,j}}$ is effective.

Then the probability of $\mathcal{RWH}$ visiting a specific lender node u_i or a loan node v_j can be deduced:

$$p^{(t+1)}(u_i) = \sum_{v_j\in VU_i} p(u_i|v_j, R=0)\, p^{(t)}(v_j)\, p(R=0|v_j), \tag{6}$$

$$\begin{aligned} p^{(t+1)}(v_j) = &\sum_{v_k\in V} p(v_j|v_k, R=1)\, p^{(t)}(v_k)\, p(R=1|v_k) \\ &+ \sum_{u_i\in UV_j} p(v_j|u_i)\, p^{(t)}(u_i). \end{aligned} \tag{7}$$

The above formulas can be transformed into vector-matrix forms:

$$\boldsymbol{P}_{\boldsymbol{u}}^{(t+1)} = \boldsymbol{M}_{\boldsymbol{vu}}^{T}(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{P}_{\boldsymbol{v}}^{(t)}, \tag{8}$$

$$\boldsymbol{P}_{\boldsymbol{v}}^{(t+1)} = \boldsymbol{SBP}_{\boldsymbol{v}}^{(t)} + \boldsymbol{M}_{\boldsymbol{uv}}^{T}\boldsymbol{P}_{\boldsymbol{u}}^{(t)}. \tag{9}$$

where $\boldsymbol{P}_{\boldsymbol{u}}^{(t)} = [p^{(t)}(u_1), p^{(t)}(u_2), \ldots, p^{(t)}(u_{|U|})]^T$ represents the probability vector of visiting lender nodes at time t, and $\boldsymbol{P}_{\boldsymbol{v}}^{(t)} = [p^{(t)}(v_1), p^{(t)}(v_2), \ldots, p^{(t)}(v_{|V|})]^T$ represents the probability vector of visiting loan nodes at time t. Eqs. (8), (9) provide the major concepts by which $\mathcal{RWH}$ approximates the result, i.e., the probability that each lender will lend money to the target loan. $\boldsymbol{M_{vu}}$ is a $|V| \times |U|$ matrix, $(\boldsymbol{M_{vu}})_{ij} = p(u_j|v_i, R = 0)$; $\boldsymbol{M_{uv}}$ is a $|U| \times |V|$ matrix, $(\boldsymbol{M_{uv}})_{ij} = p(v_j|u_i)$. In the same form, $\boldsymbol{S}$ is a $|V| \times |V|$ matrix with $\boldsymbol{S}_{ij} = p(v_j|v_i, R = 1)$. $\boldsymbol{B}$ is a diagonal matrix and retains $\beta(v_i, \theta)$ for each loan v_i. $\boldsymbol{B}_{ii} = \beta(v_i)$, so that $(\boldsymbol{I} - \boldsymbol{B})_{ii} = 1 - \beta(v_i, \theta)$.

Iterative calculations of Eqs. (8), (9) may not confirm a stationary result. Restarting of each step must be considered:

$$\boldsymbol{P}_{\boldsymbol{u}}^{(t+1)} = (1-\alpha)(\boldsymbol{M}_{\boldsymbol{vu}}^{T}(\boldsymbol{I} - \boldsymbol{B})\boldsymbol{P}_{\boldsymbol{v}}^{(t)}) + \alpha\boldsymbol{P}_{\boldsymbol{u}}^{(0)}, \tag{10}$$

$$\boldsymbol{P}_{\boldsymbol{v}}^{(t+1)} = (1-\alpha)(\boldsymbol{SBP}_{\boldsymbol{v}}^{(t)} + \boldsymbol{M}_{\boldsymbol{uv}}^{T}\boldsymbol{P}_{\boldsymbol{u}}^{(t)}) + \alpha\boldsymbol{P}_{\boldsymbol{v}}^{(0)}. \tag{11}$$

Then we can implement the random walk process by Eqs. (10), (11). When the iteration is over and $\boldsymbol{P_u}$ reaches a stationary distribution, i.e., we obtain the probabilities of visiting different lenders, the stationary results $\boldsymbol{P^*_u}$ and $\boldsymbol{P^*_v}$ can be calculated. Let matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ be defined as follows:

$$\boldsymbol{X} = \boldsymbol{SB}, \tag{12}$$

$$\boldsymbol{Y} = \boldsymbol{M}_{\boldsymbol{uv}}^{T}\boldsymbol{M}_{\boldsymbol{vu}}^{T}(\boldsymbol{I} - \boldsymbol{B}). \tag{13}$$

The form of $\boldsymbol{P}_{\boldsymbol{v}}^{(t+1)}$ is changed to the following:

$$\boldsymbol{P}_{\boldsymbol{v}}^{(t+1)} = (1-\alpha)(\boldsymbol{XP}_{\boldsymbol{v}}^{(t)} + \boldsymbol{YP}_{\boldsymbol{v}}^{(t-1)}) + \alpha\boldsymbol{P}_{\boldsymbol{v}}^{(0)}. \tag{14}$$

Let $\boldsymbol{P}_{\boldsymbol{v}}^{(t+1)} = \boldsymbol{P}_{\boldsymbol{v}}^{(t)} = \boldsymbol{P}_{\boldsymbol{v}}^{(t-1)} = \boldsymbol{P}_{\boldsymbol{v}}^{*}$, the stationary distribution $\boldsymbol{P}_{\boldsymbol{v}}^{*}$ is the following:

$$\boldsymbol{P}_{\boldsymbol{v}}^{*} = \alpha[\boldsymbol{I} - (1-\alpha)(\boldsymbol{X} + \boldsymbol{Y})]^{-1}\boldsymbol{P}_{\boldsymbol{v}}^{(0)}. \tag{15}$$

Substituting Eq. (15) into (10), we can obtain $\boldsymbol{P}_{\boldsymbol{u}}^{*}$. Defining $\boldsymbol{Z} = [\boldsymbol{I} - (1-\alpha)(\boldsymbol{X} + \boldsymbol{Y})]^{-1}$ for convenience, the expression of $\boldsymbol{P}_{\boldsymbol{u}}^{*}$ is the following:

$$\boldsymbol{P}_{\boldsymbol{u}}^{*} = \alpha(1-\alpha)\boldsymbol{ZM}_{\boldsymbol{vu}}^{T}(\boldsymbol{I} - \boldsymbol{B})\boldsymbol{P}_{\boldsymbol{v}}^{(0)} + \alpha\boldsymbol{P}_{\boldsymbol{u}}^{(0)}. \tag{16}$$

Thus, the approach can rank lenders by the vector $\boldsymbol{P}_{\boldsymbol{u}}^{*}$. Top-ranked lenders (remove those lenders who have pledged to loan v_s) have the highest probabilities of pledging and will be recommended. The random walk process is conducted separately for each raising-money loan. In this way, $\mathcal{RWH}$ finally reaches our goal: finding the lenders for each loan who are willing to pledge it at its current funding progress.

3.3. Similarity calculation

In this subsection, we focus on computing the loan similarity for loan-loan routes in $\mathcal{RWH}$. Li and Tang [22] and Li et al. [21] studies information propagation and similarity measure problems, which are relevant to the similarity calculation. We aim to help loans seek potential lenders. Thus, in this paper, loan similarity may lose its original meaning, i.e., loans with similar properties. Specifically, loans that have more common linked lenders are more similar. In addition, we only discuss the similarity between two loans whose funding durations overlap. Two loans that are not raising money at the same time have a similarity equal to 0 because they won't be selected by the same lender at the same time. For the loans with completed funding durations, we can obtain the similarities of any two loans by the Jaccard similarity:

$$sim_{j,k} = \frac{|UV_j \cap UV_k|}{|UV_j \cup UV_k|}. \tag{17}$$

However, Jaccard similarity cannot be directly applied in $\mathcal{RWH}$. As the description in the walking process, $\mathcal{RWH}$ selects the loan-loan routes with higher probabilities only when the loan node is at the start-up stage or has only received very few pledges. In this situation, Jaccard similarities between loans will not work well. Consequently, we must find a way to predict the loans' final similarity based on their currently captured features. The objective of the predictor is the final Jaccard similarity when both loans are completed. In the following, we develop a learner for similarity calculation combining both the *static features* and *temporal dynamic features* of the loans. The learner is trained with ended loan pairs whose dynamic features extracted at a random position of funding progress and their final Jaccard similarities are known. After training, the learner can predict the final Jaccard similarities of temporal loans in $\mathcal{RWH}$ with their currently captured features.

Features for calculating loan similarity. Loans in P2P lending have both *static features* and *dynamic features*. Specifically, static features are the properties of a loan that are given by the borrower when the loan is created, e.g., Category and BorrowerRate. The dynamic features are extracted from the dynamic funding progress and incremental lenders who

Table 2
Feature examples.

Name	Description	Type
BorrowerRate	the maximum interest rate the borrower is willing to pay	static
Category	borrowing purpose	
CreationDate	the creation date of the loan	
...	...	
AmountFunded	the amount funded	dynamic
AmountRemaining	the amount remaining to be funded	
TimeFromCreation	time interval from the CreationDate to current time	
...	...	

pledge this loan during its funding duration. A loan receives pledges one by one during its funding period, and thus the features extracted from the lenders change over time, e.g., AmountFunded and AmountRemaining. The effectiveness of dynamic features has been explored in prior studies [20,42].

We represent all features as numerics [9,42]. For temporal features, such as CreationDate, we convert raw features to a serial date number that represents the whole and fractional number of days from a fixed preset date [9,42]. For categorical features, such as Category, we convert a variable with n categories into a n-dimensional binary vector in which only the value in the corresponding category is set to one. All features are normalized for comparability. Table 2 summarizes some examples of feature. In summary, we use 14 static features and 6 dynamic features.

Model for calculating loan similarity. After preparing the features for similarity, we adopt bootstrap aggregating, i.e., bagging [5] to estimate the loan similarity in $\mathcal{RWH}$. Bagging is chosen for the following reasons. First, we should calculate the similarity of loans, which refers to the Jaccard similarity. Bagging can estimate the similarity based on the use of a regression model. More importantly, bagging is sensitive to differences in features between loans and reflects these differences in the results. Moreover, as proven in [5], the bagging predictor results in a lower mean-squared error and better performance than a single regression tree or other conventional machine learning models.

Calculating similarity can be formalized as a regression problem. The input two vectors $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ represent the features of loans v_i and v_j. The output y of the predictor represents the similarity $sim(\boldsymbol{x_i}, \boldsymbol{x_j})$. The objective of y is the final Jaccard similarity of the two loans. Suppose we have a training set $T = \{(\boldsymbol{x_1^*}, y_1), (\boldsymbol{x_2^*}, y_2), \ldots\}$, where $\boldsymbol{x_k^*} = \boldsymbol{x_{ki}} - \boldsymbol{x_{kj}} (\boldsymbol{i} < \boldsymbol{j})$. We train a bagging regressor $B(\boldsymbol{x}^*)$ with P sub-regressors; each is a regression tree $b(\boldsymbol{x}^*)$,

$$B(\boldsymbol{x}^*) = \sum_{p=1}^{P} \gamma_p b_p(\boldsymbol{x}^*), \tag{18}$$

where γ_p is the weight of each regression tree, which is set to $\frac{1}{P}$ in this study. Each regression tree is trained with sampled subsets. The learner randomly samples P subsets $\{T_1, \ldots, T_P\}$ from the training set T. For each subset T_i, a regression tree $b_i(\boldsymbol{x}^*)$ grows with the subset. The bagging regressor $B(\boldsymbol{x}^*)$ is a linear combination of $\{b_1(\boldsymbol{x}^*), \ldots, b_P(\boldsymbol{x}^*)\}$. Thus, the regression result of $B(\boldsymbol{x}^*)$ is the arithmetic mean of the results of P regression trees.

With this bagging model and captured features of loans in $\mathcal{RWH}$, the selecting probability in loan-loan routes can be further calculated as:

$$p(v_k|v_j, R = 1) = \frac{sim_{j,k}}{\sum_{v_{k'} \in V} sim_{j,k'}} = \frac{B(\boldsymbol{x_k} - \boldsymbol{x_j})}{\sum_{v_{k'} \in V} B(\boldsymbol{x'_k} - \boldsymbol{x_j})}. \tag{19}$$

3.4. Dynamic establishment of $\mathcal{RWH}$

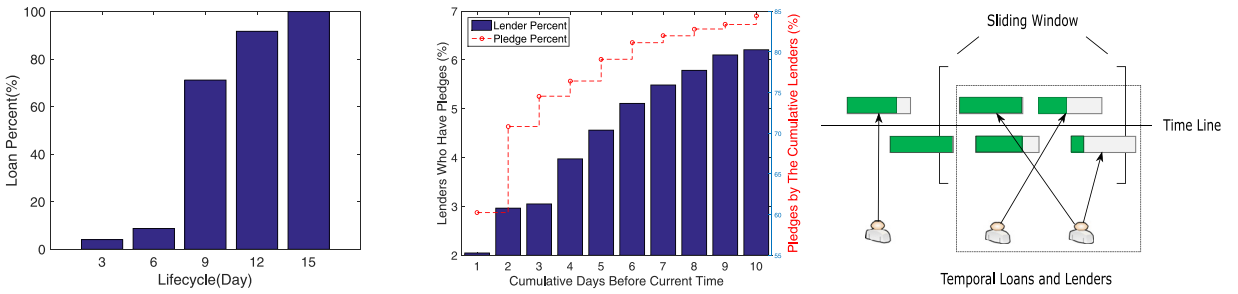
In the above, we introduced the framework of $\mathcal{RWH}$, which combines both loan-lender routes and loan-loan routes for adapting to all loans at any stage of funding progress. However, considering the temporality of loans, we should establish networks and perform $\mathcal{RWH}$ dynamically to provide recommendations as soon as possible. As the funding progresses, we should update the corresponding parts (links and weights) in the $\mathcal{RWH}$ network and perform $\mathcal{RWH}$ on it again to obtain new results. If we use the whole sets of lenders and loans to establish a full-scale network in each time, the rebuild-and-recommend process will access many inactive lenders and also cost much more time.

Thus, in contrast to traditional random walk techniques, we propose to dynamically establish $\mathcal{RWH}$ on small-scale networks with *temporal loans* and *temporal lenders*. Specifically, temporal loans are those that are raising money currently, and temporal lenders are those who are willing to lend money to these temporal loans.

Specifically, we have the following two observations about the temporalities of both loans and lenders in P2P lending.

Observation 1. Most loans have short durations for raising money. In the Prosper dataset, loans durations are usually no more than 15 days. Fig. 3(a) shows the distribution of loans with different funding durations.

Observation 2. Many lenders often periodically and continuously pledge a series of loans [44]. In the Prosper dataset, lenders who have pledges in recent days contribute the most pledges to the current temporal loans in these loans' re-



(a) Loan Funding Durations. (b) Temporal Lenders of Loans. (c) Sliding Window.

Fig. 3. Observations and sliding window.

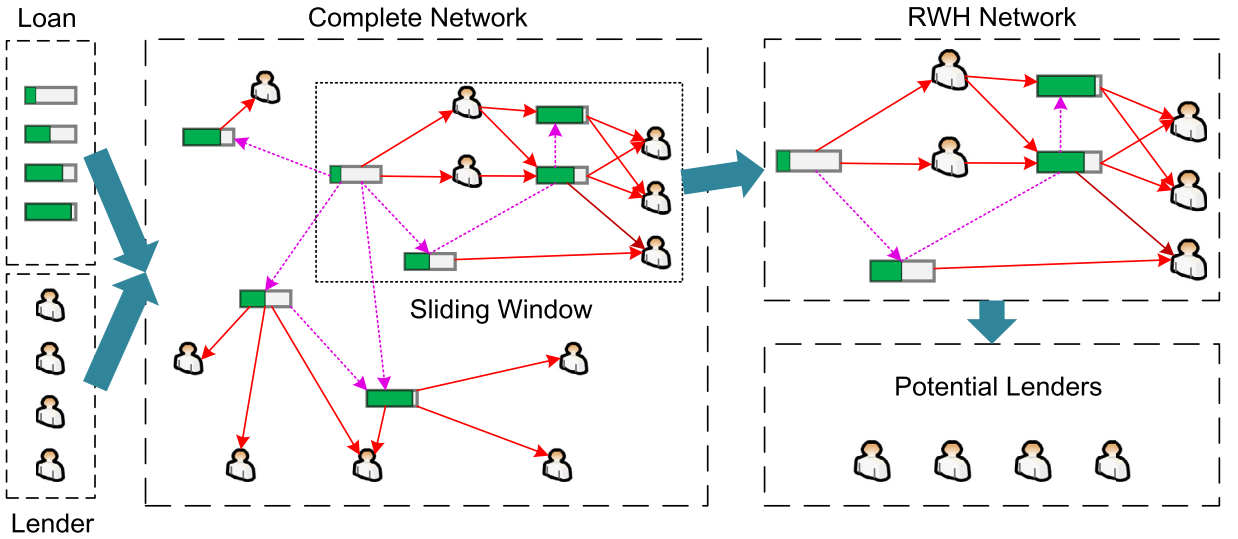


Fig. 4. Flow chart of $\mathcal{RWH}$.

maining durations. For example, as shown in Fig. 3(b), 6% of lenders who have pledges in the previous 10 days contribute 85% of pledges to current temporal loans.

These observations inspired us to establish $\mathcal{RWH}$ on networks with temporal lenders and loans rather than all loans and lenders. Because many lenders pledge periodically and continuously, lenders who have pledged in recent days contribute the most pledges to the temporal loans. In other words, these temporal lenders are the most important potential candidates in the current market. As for loans, only two loans whose durations overlap are tied more closely and can be selected by the same lender at the same time to form a portfolio. Thus, these observations guarantee the performance of establishing $\mathcal{RWH}$ on small-scale networks with temporal loans and lenders.

Specifically, we dynamically establish $\mathcal{RWH}$ by a *sliding window*, as shown in Fig. 3(c). Suppose a target loan v_s is raising money at time $T_b^{(t)}$, then the start time of the window is $T_b^{(w)} = T_b^{(t)} - 30(Days)$. We let the window starts 30 days earlier before the current time because 30 days is twice the maximum loan duration. A period of 30 days thus guarantees that this window contains all temporal loans whose durations may overlap with that of v_s. The window ends at the current time ($T_{current}$), i.e., $T_e^{(w)} = T_{current}$. Thus, the window contains the current temporal lenders and loans. A temporal lender's pledge time $T^{(tU)}$ and a temporal loan's start time $T^{(tV)}$ should satisfy the following conditions:

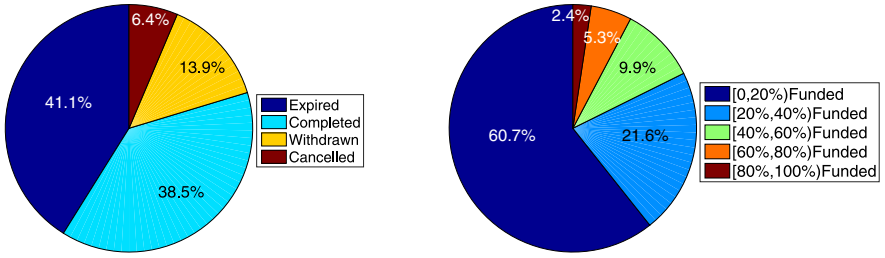
$$T^{(tU)}, T^{(tV)} \in [T_b^{(w)}, T_e^{(w)}]. \tag{20}$$

Thus, an $\mathcal{RWH}$ network is only established with these temporal lenders tU and temporal loans tV, where $tU \subset U$, and $tV \subset V$. In addition, to calculate probability in the small network, tU and tV are used rather than the whole sets.

Fig. 4 presents a flow chart for the use of $\mathcal{RWH}$ in the real world. First, loans and lenders constitute the complete network, in which the loan-loan route represents the loan-loan similarity, whereas the loan-lender route represents the pledge record. Second, an $\mathcal{RWH}$ network is separated from the complete network by the sliding window. Finally, random walking on the $\mathcal{RWH}$ is performed to obtain the final results, i.e., potential lenders for the target loan. In general, $\mathcal{RWH}$ has an obviously small scale. It can work faster and find potential lenders as early as possible for raising-money loans.

Table 3
Data statistics of the Prosper dataset.

#loans	#lenders	#pledges	#test_loans	#test_lenders
100,327	60,333	8,054,183	21,846	50,847



(a) Funding Results. (b) Distribution of Expired Loans.

Fig. 5. Statistical analysis of the Prosper dataset.

Another benefit of this small scale is that we can easily rebuild the network as the funding progresses rather than changing the structure of the full-scale network. With the time of funding, we only need to update the small network and perform the recommendation again. In addition, the mechanism of dynamically establishing a network avoids accessing inactive lenders in the current market, thus protecting the user experience.

In summary, we propose a random walk-based hybrid approach, $\mathcal{RWH}$, that combines both loan-lender routes and loan-loan routes to find potential lenders for raising-money loans. We also adopt a bagging method with extracted dynamic features to calculate loan similarity in loan-loan routes. Furthermore, we design a dynamic establishment for $\mathcal{RWH}$ by a sliding window to adapt to the temporality and dynamic nature of this problem. In next section, we will evaluate our approach experimentally.

4. Experiments

In this section, we design experiments to evaluate the effectiveness of our approach. First, we introduce the experimental dataset, pre-process and setup. Then, we respectively report the experimental results for effectiveness (Section 4.4.1), robustness (Section 4.4.2), efficiency (Section 4.4.3), and parameter optimization (Section 4.4.4) and the results for a crowdfunding dataset(Section 4.4.5).

4.1. Dataset

The first experimental dataset was collected from Prosper.[2] This dataset contains all the records for loans, lenders and pledges (i.e., bids in this dataset) for nearly 6 years on this platform. We mainly use three tables of this data for our experiments. The *Listing* table provides the features of listed loans and how these loans ended (expired or succeeded). The *Member* table contains member information, such as addresses and credit levels. Notice that both borrowers and lenders are all recorded in the *Member* table. The *Bid* table is used to link the *Listing* table and the *Member* table to obtain information about who pledges to a certain loan. In our experiments, we only use loans with at least five pledges. Table 3 shows the statistics of this experimental dataset.

Fig. 5 presents some statistical analysis results for funding. As shown in Fig. 5(a), only 38.5% of loans (*Completed*) receive sufficient pledges in time. *Withdrawn* loans were withdrawn by the borrowers. *Cancelled* loans were cancelled by Prosper. At least 41.1% loans (*Expired*) failed because they did not received enough money. We analyse the specific funding distribution for the failed loans in Fig. 5(b). Sparsity is obvious in the data set, i.e., more than 60% of failed loans received less than 20% of their funding goals. Thus, finding lenders is crucial for loans, and this finding and recommending should be performed as early as possible (e.g., in the start-up stage of the funding process) in the raising-money duration.

4.2. Pre-processing

There is no environment for conducting an online test either in practice or in the literature. Thus, in this paper, we simulated the real-world lending and online recommending procedure.

[2] http://www.prosper.com/tools/DataExport.aspx.

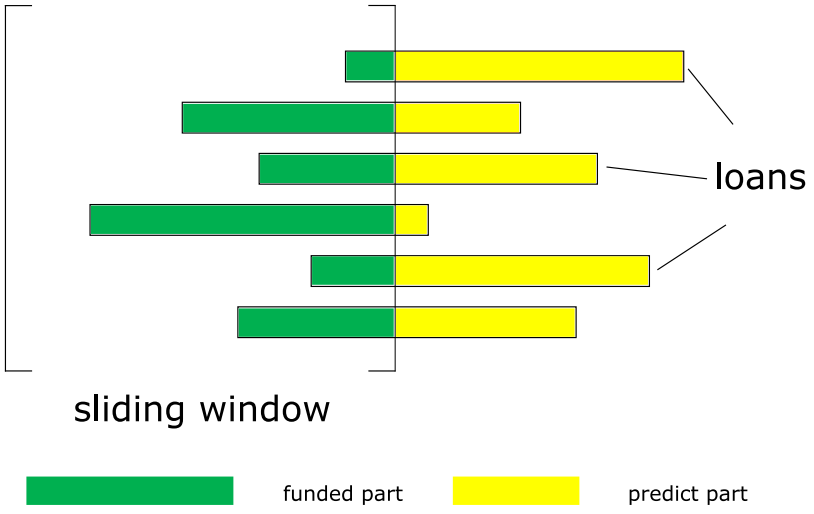


Fig. 6. Dividing with a cut time.

Before the online recommending experiments, we trained the bagging regressor offline in advance to calculate the loan similarity. In particular, we randomly sampled 50,000 pairs of loans from the training loans and randomly select 40,000 of them to build the training set and 10,000 to build the validation set. we then obtain the bagging regression model.

Since the experimental data contains the detailed timestamp of each pledge, we can restore the funding market with a specific *recommendation time*. First, we randomly pick a time point as a cut, i.e., the *recommendation time*. Then we restore every loan that is divided by the cut to its previous status. The pledges after the cut are removed from the training sets and are used to test the recommendation. At the same time, a sliding window is established, and only the loans and lenders inside the sliding window are retained for the next steps (refer to Fig. 3)(c). When a starting loan node (a loan that is raising money at the *recommendation time*) is determined, the graph can be established in a recursive procedure. Once a loan node is in the graph, lenders who have pledges on this loan and other loans whose funding durations overlap that of this loan will be added. If a lender is added to the graph, the loans she has pledged are added to the graph as well. The random walk probabilities are calculated when nodes are added. Probabilities of loan-lender routes are calculated with the pledge amount, whereas probabilities of loan-loan routes are determined by their similarities. We can easily update the network for other raising-money loans via minor modifications of the existing loan and lender node relationships. With the window sliding over time, we can also update the network for new loans. Fig. 6 illustrates a market with loans with varying funded percentages at a selected time. Overall, we periodically establish and update networks for sequential cutting timestamps over time. All loans and lenders that are included in the established networks constitute the test set of the experiments.

4.3. Experimental setup

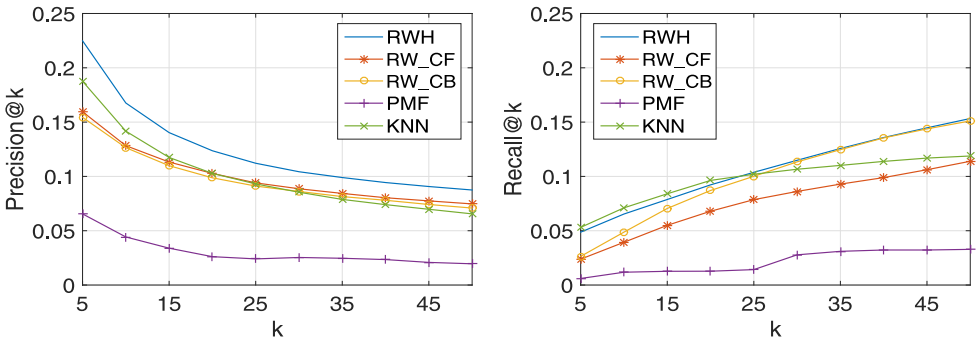
The whole process of $\mathcal{RWH}$ is denoted as RWH in the experiments. The parameter α of $\mathcal{RWH}$ is set to 0.8 [34], and θ is set to 10 without special instructions. The sliding window size is set to 30 days, i.e., double the maximum loan duration on Prosper.

4.3.1. Baselines

In the experiments, we construct several baseline methods for comparison with $\mathcal{RWH}$. These baseline methods fall into three types.

Variants of $\mathcal{RWH}$. We consider other random walk approaches, i.e., variants of $\mathcal{RWH}$, that only adapt collaborative filtering or content-based filtering, denoted as RW_CF and RW_CB , respectively. In RW_CF, there are only loan-lender routes and no loan-loan routes. Thus, the walking network of RW_CF is a strictly bipartite graph. The transition probabilities between nodes are calculated by the pledge amount. The stable iteration result is also a vector showing the probability of visiting each lender. The RW_CB approach only consists of loan-loan routes. Like RWH, the transition probabilities between loans are calculated by the bagging regression model. The stable iteration result of RW_CB is a vector showing the probabilities of visiting other loans. analysing the iteration results indicates that the greater the probability of visiting another loan, the better the recommendation results will be for the lenders who pledge the visited loan.

Traditional Recommendation Methods. We also implement k-Nearest Neighbour (KNN) [7] for lender recommendation. We choose k Nearest Neighbour loans for a target loan. The lenders who have pledged these neighbour loans will be recommended. In these experiments, the distance between neighbour loans is computed by their Jaccard similarity. We also rank the recommended lenders by the number of times they pledge the neighbour loans.



(a) Precision@k (b) Recall@k

Fig. 7. Precision@k and Recall@k for the Prosper dataset.

Model-based Collaborative Filtering. For the popular recommendation models, we implement the widely-used Probabilistic Matrix Factorization (PMF) [17]. In these experiments, the probability matrix is filled by calculating the percentage of the pledge amount. We factorize the probability matrix and obtain 30 factors for both loans and lenders. Recommendation results are given by the approximation of the pledge matrix.

4.3.2. Metrics

We evaluate these methods on precision, recall [31] and also the running time cost. Precision and recall are two widely used metrics of effectiveness, whereas running time cost records the efficiency results of all methods. The definitions of precision and recall are as follows:

$$Precision@k = \frac{|L(k) \cap T(s)|}{k}, \qquad Recall@k = \frac{|L(k) \cap T(s)|}{|T(s)|}, \tag{21}$$

where k is the candidate list size, $L(k)$ is the recommended candidate lender list for loan v_s, and $T(s)$ is the true set of lenders of loan v_s after the predicting time. These two metrics measure how the recommendations match a lender's selections. In the following experiments, all reported results are the average of all test loans.

4.4. Experimental results

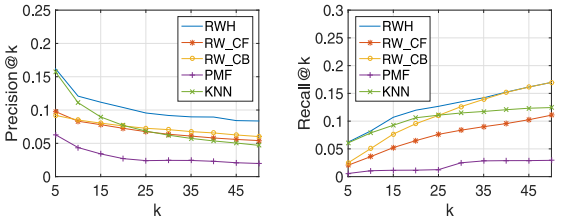
In this subsection, we mainly report the experimental results from the aspects of effectiveness, robustness, efficiency and parameter optimization. Specifically, we also report the experimental results for a crowdfunding dataset.

4.4.1. Effectiveness

Fig. 7 shows the results for precision and recall with different lender list lengths k on the Prosper dataset. The precision results of the three random walk methods and KNN all decrease as top k increases. RWH is better than both RW_CF and RW_CB because RWH exploits the advantages of both. KNN has higher precision than RW_CF and RW_CB only for the top 5 and top 10 but performs worse than these two methods when the top k becomes larger. Thus, Nearest Neighbours provide better help on recommendation when the candidate list is short. PMF performs worse than the other methods, possibly because of the sparsity of the dataset. We also note that RWH, RW_CB and RW_CF perform better on recall, especially when the candidate list size is larger. Based on the overall evaluation, RWH outperforms the other approaches by 2% to 4% on precision. Among all approaches, RWH shows the best performance.

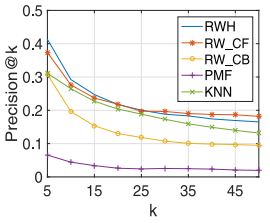
4.4.2. Robustness

We further divide the starting or target loans into groups according to their completed funding percentages, e.g., 0%–20%, at the recommending time. Fig. 8 show the results for each group, from which we can analyse the performance of the methods with different stage of funding progress of the loan. As for the total dataset, RWH performs best in most cases. As shown in Fig. 8(a) and (b), the funded percentage of loans is 0% to 20%, indicating that these loans are in the start-up funding stage and facing an extreme cold-start problem. All approaches perform worse than they perform on other groups. Moreover, RWH provides obviously better results than the other approaches in this case. As the funded percentage of the loans increases, the performance of all five approaches improves. In most cases, RWH always performs better than all of other approaches, e.g., with 5% to 10% improvements in precision and recall. We can conclude that RWH performs the best of these five approaches by summarizing the results of the five groups of experiments.

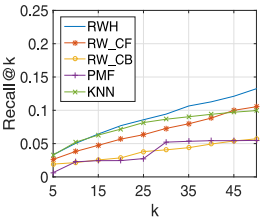


(a) Precision of 0%–20%

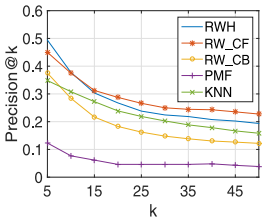
(b) Recall of 0%–20%



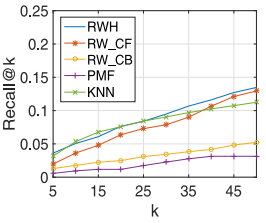
(c) Precision of 20%–40%



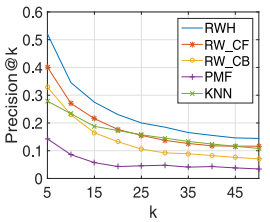
(d) Recall of 20%–40%



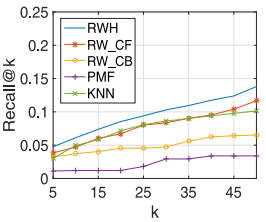
(e) Precision of 40%–60%



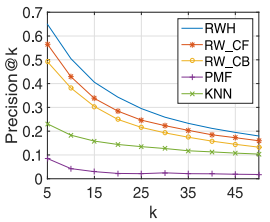
(f) Recall of 40%–60%



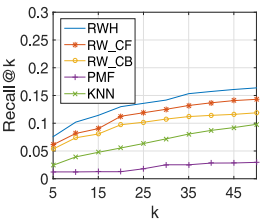
(g) Precision of 60%–80%



(h) Recall of 60%–80%



(i) Precision of 80%–100%



(j) Recall of 80%–100%

Fig. 8. Precision@k of each group.

4.4.3. Efficiency

We also report the efficiency results of the different methods. The results (three separate times) are shown in Fig. 9. Since iteration steps are not required for KNN, it runs fastest. RWH requires slightly more time than RW_CF and RW_CB because its random walk network is more complicated than those of the others. All three random walk approaches and PMF require iteration steps to obtain results. However, all of the three random walk approaches use only temporal loans and lenders to build the network, and thus they have acceptable efficiency performances and run much faster than PMF does.

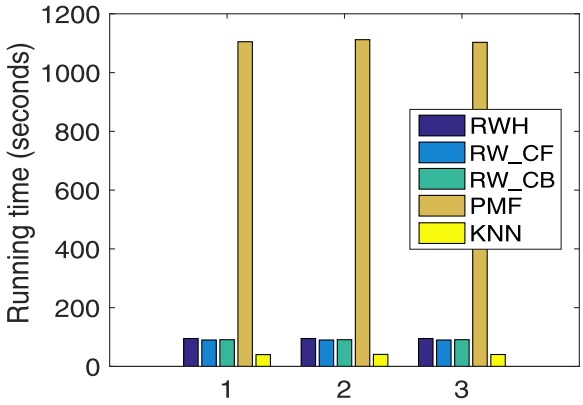


Fig. 9. Running time results.

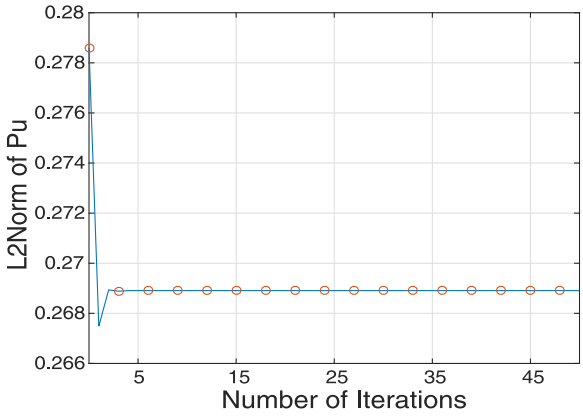


Fig. 10. Rates of convergence for $\mathcal{RWH}$.

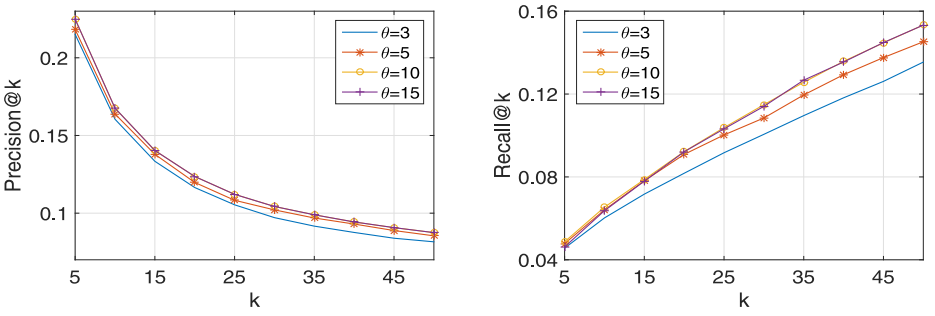
Since PMF uses the whole rating matrix to compute, it requires a much longer running time. Fig. 10 represents how the L2Norm of $\boldsymbol{P_u}$ changes with iterations. When applied to the Prosper dataset, $\mathcal{RWH}$ exhibits good convergence properties. The L2Norm of vector $\boldsymbol{P_u}$ shocks 2–3 steps and then converges gently. Compared with the running time results, $\mathcal{RWH}$ is qualified for real-world recommendations.

4.4.4. Optimization of parameters

Parameter θ**.** In our approach, the variable $R \sim Bernoulli(\beta(v_i, \theta))$ is introduced to determine whether $\mathcal{RWH}$ should move to another loan node or a lender node from the current loan node. Thus, the value of function $\beta(v_i, \theta)$ can be seen as the weights of collaborative filtering and content-based filtering. A larger θ causes the function $\beta(v_i, \theta)$ to become smaller. Thus, the approach will depend more on collaborative filtering routes than on loan nodes with a higher funded percentage. By contrast, if the dataset has high sparsity, θ should be smaller to enhance the performance. In this paper, we tune the value of parameter θ to optimize the experimental performance.

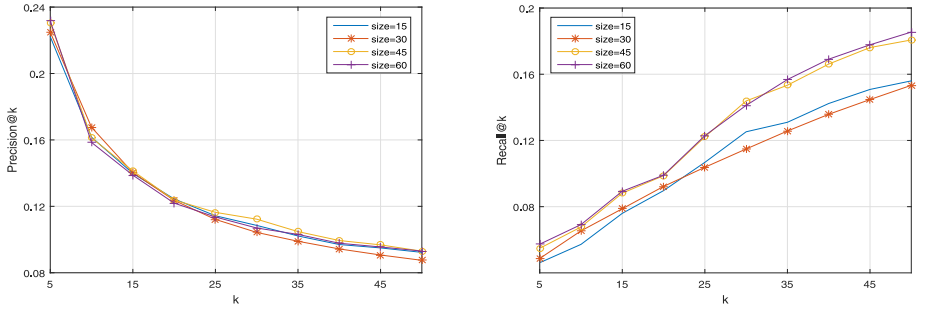
Fig. 11 shows the results for tuning the parameter θ. When θ increases from 3 to 5 and from 5 to 10, $\mathcal{RWH}$ exhibits obvious improvements in both precision and recall. When θ continues to increase to 15, the performance of $\mathcal{RWH}$ begins to decay slowly. For the best performance, we set $\theta = 10$ in our experiments.

Parameter sliding window size. The size of the sliding window controls the scale of $\mathcal{RWH}$ and can both accelerate the random walk procedures and influence the accuracy of the results. To study the relationship between window size and results, we tune the value of this parameter. Fig. 12 shows the tuning results for sliding window size. As the size is tuned from 15 to 60, the precision and recall slowly improve. Although the performances of accuracy improve as the size increases, the time cost in the experiments also increases obviously. To also increases the *temporality* of $\mathcal{RWH}$, we chose an acceptable sliding window size of 60 days.



(a) Precision@k (b) Recall@k

Fig. 11. Tuning parameter θ.



(a) Precision@k (b) Recall@k

Fig. 12. Tuning the sliding window size.

Table 4
Data statistics of the Indiegogo dataset.

#campaigns	#contributions	#contributors	#test_campaigns	#test_contributors
1,294	437,076	248,155	622	33,545

4.4.5. Test on the crowdfunding dataset

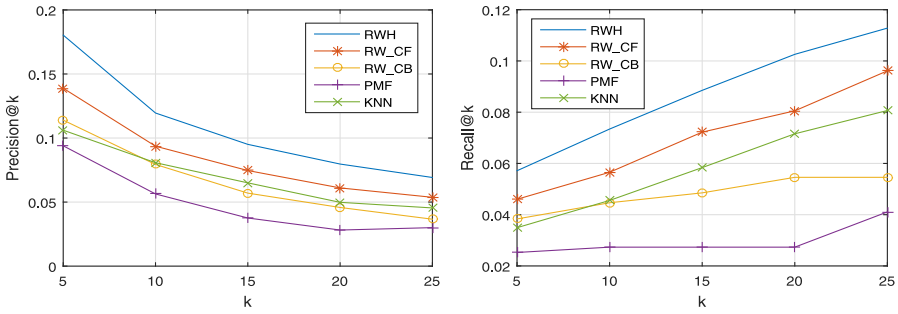
In this subsection, we evaluate the effectiveness of $\mathcal{RWH}$ on a real-world crowdfunding dataset. Other testing aspects are omitted due to data sparsity and small size. The dataset was collected from Indiegogo,[3, 4] an online crowdfunding platform that can be treated as a special case of P2P lending from the view of *borrowing-money or funding motives* [41]. Indiegogo adopts a different reward type compare to Prosper. The former is a reward-based platform, whereas the latter is a lending-based platform. Despite these differences, Indiegogo and Prosper have some similar concepts. For example, *campaign* is similar to *loan, contributor* is similar to *lender*, and *pledge* is similar to *contribution*. These common characteristics enable similar implementation of $\mathcal{RWH}$ on the Indiegogo dataset. Table 4 shows the statistics of the Indiegogo dataset.

The experiments with the Indiegogo dataset followed the same pre-processing and experimental setup as the Prosper dataset. Fig. 13 shows the effectiveness of each approach on the Indiegogo dataset. Note that the Indiegogo dataset has higher sparsity than Prosper, and the performance of each approach is not better than the results in Fig. 7. KNN performs better than RW_CB when top k is larger than 10, possibly because the collaborative filtering-based metrics fit the Indiegogo dataset better. In the experiments on both datasets, RWH performs best among all five approaches.

From the above experimental results, we can see that our approach, i.e., $\mathcal{RWH}$, can provide the best results with acceptable time cost in most cases. In particular, $\mathcal{RWH}$ is robust and adapted to loans at any stage of funding progress. This is because $\mathcal{RWH}$ is a hybrid method which combines both collaborative filtering and content-based filtering. The hybrid method overcome the shortcomings each of these two types of methods. In addition, the small scale of the $\mathcal{RWH}$ network ensures the efficiency of the method. This characteristic is crucial in P2P lending because of the dynamic and temporality of funding progression.

[3] https://www.indiegogo.com/.

[4] The dataset is available in http://home.ustc.edu.cn/%7Ezhhk/DataSets.html.



(a) Precision@k (b) Recall@k

Fig. 13. Precision@k and Recall@k on the Indiegogo dataset.

5. Conclusion

In this paper, we presented a holistic study on finding potential lenders for raising-money loans in P2P lending. Specifically, we proposed a random walk-based hybrid approach, i.e., $\mathcal{RWH}$, which combined both collaborative filtering and content-based recommendation. In the loan-loan routes of $\mathcal{RWH}$, we extracted dynamic features from loans and adapted a bagging regression to estimate the final similarity between two loans. Furthermore, for efficiency and to adapt to the temporality in P2P lending, we proposed to dynamically establish small-scale networks with temporal loans and lenders. To evaluate our approach, we constructed extensive experiments on Prosper and Indiegogo data. The analysis and experimental results demonstrated the significance of our study and the effectiveness of our solutions.

In the future, we will adapt our approach to other similar applications and evaluate it with other datasets. We also plan to design a mechanism to detect the optimal time to perform lender recommendations for each loan in their funding duration.

Acknowledgement

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2016YFB1000904), and the National Natural Science Foundation of China (Grants No. 61727809, U1605251 and 61703386), and the Youth Innovation Promotion Association of CAS (No. 2014299), and the Anhui Provincial Natural Science Foundation (Grant No. 1708085QF140), and the Fundamental Research Funds for the Central Universities (Grant No. WK2150110006).

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, Knowl. Data Eng. IEEE Trans. 17 (6) (2005) 734–749.
[2] J. An, D. Quercia, J. Crowcroft, Recommending investors for crowdfunding projects, in: 23rd WWW, ACM, 2014, pp. 261–270.
[3] S.C. Berger, F. Gleisner, Emergence of financial intermediaries in electronic markets: the case of online p2p lending, BuR-Bus. Res. 2 (1) (2009) 39–65.
[4] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl. Based Syst. 46 (2013) 109–132.
[5] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[6] S. Ceyhan, X. Shi, J. Leskovec, Dynamics of bidding in a p2p lending service: effects of herding and predicting loan success, in: 20th WWW, ACM, 2011, pp. 547–556.
[7] K. Choi, Y. Suh, A new similarity function for selecting neighbors for each target item in collaborative filtering, Knowl. Based Syst. 37 (2013) 146–153.
[8] J. Choo, C. Lee, D. Lee, H. Zha, H. Park, Understanding and promoting micro-finance activities in kiva. org, in: 7th WWW, ACM, 2014, pp. 583–592.
[9] J. Choo, D. Lee, B. Dilkina, H. Zha, H. Park, To gather together for a better world: Understanding and leveraging communities in micro-lending recommendation, in: 23rd WWW, ACM, 2014, pp. 249–260.
[10] R. Emekter, Y. Tu, B. Jirasakuldech, M. Lu, Evaluating credit risk and loan performance in online peer-to-peer (p2p) lending, Appl. Econ. 47 (1) (2015) 54–70.
[11] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, Knowl.Data Eng. IEEE Trans. 19 (3) (2007) 355–369.
[12] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, L. Schmidt-Thieme, Learning attribute-to-feature mappings for cold-start recommendations, in: 10th ICDM, IEEE, 2010, pp. 176–185.
[13] E.M. Gerber, J.S. Hui, P.-Y. Kuo, Crowdfunding: why people are motivated to post and fund projects on crowdfunding platforms, in: Proceedings of the International Workshop on Design, Influence, and Social Technologies: Techniques, Impacts and Ethics, 2012.
[14] A. Hernando, J. Bobadilla, F. Ortega, A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model, Knowl. Based Syst. (2016).
[15] M. Herzenstein, R.L. Andrews, U.M. Dholakia, E. Lyandres, The democratization of personal consumer loans? determinants of success in online peer–to-peer lending communities, Research Paper, Boston University School of Management, 2008. 6
[16] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: 15th SIGKDD, ACM, 2009, pp. 397–406.
[17] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (2009) 30–37.
[18] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, L. Getoor, Hyper: a flexible and extensible probabilistic framework for hybrid recommender systems, in: Proceedings of the 9th ACM Conference on Recommender Systems, ACM, 2015, pp. 99–106.

[19] E.L. Lee, J.K. Lou, W.M. Chen, Y.C. Chen, S.D. Lin, Y.S. Chiang, K.T. Chen, Fairness-aware loan recommendation for microfinance services, in: Proceedings of the 2014 International Conference on Social Computing, ACM, 2014, p. 3.
[20] Y. Li, V. Rakesh, C.K. Reddy, Project success prediction in crowdfunding environments, 9th WSDM, ACM, 2016.
[21] Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015) 2085–2098.
[22] Z. Li, J. Tang, Weakly supervised deep matrix factorization for social image understanding, IEEE Trans. Image Process. 26 (1) (2017) 276–288.
[23] D. Liu, D. Brass, Y. Lu, D. Chen, Friendships in online peer-to-peer lending: pipes, prisms, and relational herding, Mis Q. 39 (3) (2015) 729–742.
[24] Q. Liu, Y. Ge, Z. Li, E. Chen, H. Xiong, Personalized travel package recommendation, in: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, 2011, pp. 407–416.
[25] P. Lops, M. De Gemmis, G. Semeraro, Content-based Recommender Systems: State of the Art and Trends, in: Recommender systems handbook, Springer, 2011, pp. 73–105.
[26] C.-T. Lu, S. Xie, X. Kong, P.S. Yu, Inferring the impacts of social media on crowdfunding, in: 7th WSDM, ACM, 2014, pp. 573–582.
[27] B. Luo, Z. Lin, A decision tree model for herd behavior and empirical evidence from the online p2p lending market, Inf. Syst. e-Bus. Manage. 11 (1) (2013) 141–160.
[28] C. Luo, H. Xiong, W. Zhou, Y. Guo, G. Deng, Enhancing investment decisions in p2p lending: an investor composition perspective, in: 17th SIGKDD, ACM, 2011, pp. 292–300.
[29] X. Luo, M. Zhou, Y. Xia, Q. Zhu, An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems, Ind. Inf. IEEE Trans. 10 (2) (2014) 1273–1284.
[30] M. Malekipirbazari, V. Aksakalli, Risk assessment in social lending via random forests, Expert Syst. Appl. 42 (10) (2015) 4621–4631.
[31] D.M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011).
[32] V. Rakesh, W.-C. Lee, C.K. Reddy, Probabilistic group recommendation model for crowdfunding domains, 9th WSDM, ACM, 2016.
[33] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: 25th SIGIR, ACM, 2002, pp. 253–260.
[34] Y. Shi, M. Larson, A. Hanjalic, Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges, ACM Comput. Surv. (CSUR) 47 (1) (2014) 3.
[35] J. Solomon, W. Ma, R. Wash, Don't wait!: how timing affects coordination of crowdfunding donations, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, ACM, 2015, pp. 547–556.
[36] Z. Wei, M. Lin, Market mechanisms in online peer-to-peer lending, Manage. Sci. (2016).
[37] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, H.-P. Kriegel, Probabilistic memory-based collaborative filtering, Knowl. Data Eng. IEEE Trans. 16 (1) (2004) 56–69.
[38] Y. Zhang, Y. Xiong, X. Kong, Y. Zhu, Netcycle: collective evolution inference in heterogeneous information networks, in: SIGKDD, ACM, 2016, pp. 1365–1374.
[39] Y. Zhang, Q. Zhao, Y. Zhang, D. Friedman, M. Zhang, Y. Liu, S. Ma, Economic recommendation with surplus maximization, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 73–83.
[40] Z. Zhang, D.D. Zeng, A. Abbasi, J. Peng, X. Zheng, A random walk model for item recommendation in social tagging systems, ACM Trans. Manage. Inf. Syst.(TMIS) 4 (2) (2013) 8.
[41] H. Zhao, Y. Ge, Q. Liu, G. Wang, E. Chen, H. Zhang, P2p lending survey: platforms, recent advances and prospects, ACM Trans. Intell. Syst.Technol. (TIST) 8 (6) (2017) 72.
[42] H. Zhao, Q. Liu, G. Wang, Y. Ge, E. Chen, Portfolio selections in p2p lending: amulti-objective perspective, in: SIGKDD, ACM, 2016, pp. 2075–2084.
[43] H. Zhao, Q. Liu, H. Zhu, Y. Ge, E. Chen, Y. Zhu, J. Du, A sequential approach to market state modeling and analysis in online p2p lending, IEEE Trans. Syst. Man, Cybern.Syst. (2017).
[44] H. Zhao, L. Wu, Q. Liu, Y. Ge, E. Chen, Investment recommendation in p2p lending: a portfolio perspective with risk management, in: ICDM, IEEE, 2014, pp. 1109–1114.
[45] H. Zhao, H. Zhang, Y. Ge, Q. Liu, E. Chen, H. Li, L. Wu, Tracking the dynamics in crowdfunding, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 625–634.