Risk-Hedged Venture Capital Investment Recommendation

Xiaoxue Zhao, Weinan Zhang,* Jun Wang
Department of Computer Science, University College London
{x.zhao, w.zhang, j.wang}@cs.ucl.ac.uk

ABSTRACT

With the increasing accessibility of transactional data in venture finance, venture capital firms (VCs) face great challenges in developing quantitative tools to identify new investment opportunities. Recommendation techniques have the possibility of helping VCs making data-driven investment decisions by providing an automatic screening process of a large number of startups across different domains on the basis of their past investment data. A previous study has shown the potential advantage of using collaborative filtering to catch and predict the VCs' investment behaviours [17]. However, two fundamental challenges in venture finance make conventional recommendation techniques difficult to apply. First, risk factors should be cautiously considered when making investments: for a potential startup, a VC needs to specifically estimate how well this new investment can fit into its *holding* investment portfolio in such a way that investment risk can be *hedged*. Second, The investment behaviours are much sparser than conventional recommendation applications and a VC's investments are usually limited to a few industry categories, making it impossible to use a topic-diversification method to hedge the risk. In this paper, we solve the startup recommendation problem from a risk management perspective. We propose 5 risk-aware startup selection and ranking algorithms to catch the VCs' investment behaviours and predict their new investments. Apart from the contribution on the new risk-aware recommendation model, our experiments on the collected CrunchBase dataset show significant performance improvements over strong baselines.

Keywords

Recommendation Diversification, Recommender Systems, Portfolio Theory, Venture Finance

1. INTRODUCTION

Early-stage investment is a key driving force of technological innovation and is vitally important to the wider economy, especially in high-growth and hi-tech industries, such as life sciences, clean-tech and information technology. Traditionally, investment opportunities are either referred or identified through manual technology scans [18]. The main stages of an investor's decision making process involve deal origination, screening, evaluation, structuring and post investment activities. These stages align with those identified by other research into venture capital (VC) investment [7]. In recent years, the traditional venture financing landscape has also shown signs of evolving. Some commentators [2] depict an industry "trifurcating" with (i) top-tier firms, e.g., Sequoia Capital, (ii) incubators and accelerators, e.g., Y Combinator, and, finally, (iii) firms that are taking a more quantitative approach to funding, e.g., Correlation Ventures. There is potentially a fourth factor in the emergence of entirely new funding sources such as "crowdfunding" which generally operate through online platforms, e.g., AngelList. Shifts towards more quantitative and data driven approaches along with new opportunities for online private investment provide additional impetus and scope for applying data mining and intelligent recommendations to this domain [17].

Whilst recommendation is useful in helping make investment decisions, this new domain, however, is quite distinct from existing applications of recommender systems (e.g., for movies and music) and represents unique cha-llenges. First, an essential question about venture finance is to estimate and control the risk, which is not well modelled for traditional recommender systems. Specifically, in a recent work where the collaborative filtering method is applied to the startup recommendations [17], the authors purely explored the similarity between new investment opportunities and VCs' holding investments. Promoting similar opportunities may be attractive to the VCs at the first sight, but such similarity-based methods fail to catch VCs' underlying main investment intention, which is to examine how well the new investment will fit into the current investment portfolio to hedge the risk and increase the return [9]. Figure 1 further illustrates our motivation by employing a latent-factor model on our collected dataset from CrunchBase (`https://www.crunchbase.com/`). This figure compares the mean and variance between the probabilistic matrix factorisation (PMF) [14] recommendations and VCs' groundtruths (detailed in Section 3). The PMF recommendations are made to contain the same number of items as the groundtruths for each VC in order to make comparison. We can see that for 76.3% VCs, PMF recommendations have higher mean values, but in 83.5% cases, the VCs' true investments have lower variances and thus lower risk.

Furthermore, investments usually involve a long term process, which is different from content consumption behaviours such as movie watching. It is because a VC's decision on more investments does not necessarily indicate terminations of previous holding investments, but adding them into the existing portfolio. This emphasises the need that instead of

*Joint first authorship.

RecSys'15, September 16–20, 2015, Vienna, Austria.
© 2015 ACM. ISBN 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2800181.

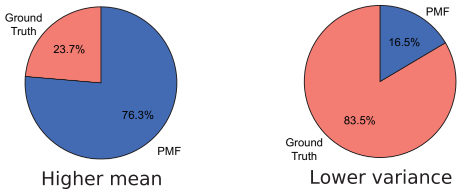


Figure 1: Mean and variance comparisons between PMF recommendations and groundtruth investments in CrunchBase. Left: percentage of higher mean for individual VCs. Right: percentage of lower variance for individual VCs.

diversifying the recommendation list alone, as many personalised recommendation systems have studied [16, 24, 22], it is more sensible to diversify the *joint* investment portfolio including both the VCs' holding investments and potential future ones. In other words, the recommended investment opportunities should be able to potentially hedge some risk of the portfolio the investor has already held. This point makes the VC investment recommendation significantly different from traditional recommendation tasks.

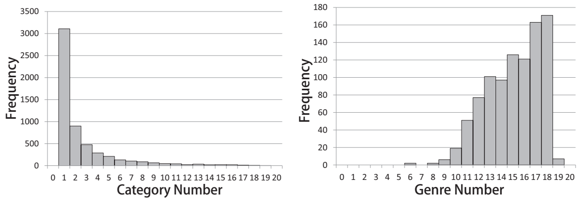
Finally, from the analysis of the CrunchBase data, we can find some unique characteristics in the VC investment behaviours (see Section 3). The dataset is much sparser as each VC usually only invests in a small number of investments. Meanwhile, a VC usually only focuses on a few industry categories, making the previous recommendation methods based on topic diversification, e.g., [27, 8], infeasible here.

In this paper, we address the above issues by integrating risk management into recommender systems and tailoring it for venture finance. Our contributions are fourfold. (i) We find that with advances in recommender systems, particularly collaborative filtering, it is possible to envision an automatic startups screening process according to the expectation and risk revealed by the VC-startup relationships. (ii) Based on the estimated risk, we propose 5 algorithms to systematically optimise the startup recommendations in a risk-aware manner. (iii) We conduct comprehensive experiments which show significant improvements on various measures, proving the effectiveness of our algorithms in recommending future investments and predicting VC investment behaviours. (iv) We publicise our collected CrunchBase venture finance dataset used in this paper, in an effort to facilitate scientific research on investment behaviours and recommendations in venture finance.

The rest of this paper is organised as follows. Related work is discussed in Section 2. In Section 3, we describe and analyse the CrunchBase dataset. In Section 4 we present the proposed risk-aware solutions for startup recommendations. Our experimental evaluation is reported in Section 5 and Section 6 summarises and concludes this paper.

2. RELATED WORK

Venture Finance. Venture finance refers to the financing of private companies through the use of venture capital. Venture capital is a form of private equity, a medium to long-term form of finance provided in return for an equity stake in potentially high growth companies. Early-stage investment is typified by venture capital firms who deploy capital towards high-risk ventures. Venture capital has five main characteristics [13]: is a financial intermediary; invests only in private companies; takes an active role in monitoring and helping portfolio companies; primary goal is to maximise financial return by exiting investments through sale or an initial public offering (IPO); invests to fund the internal growth of companies. Whilst there have been some applications of recommender systems to the broader domain of finance, including microfinance [3], there has seemingly been little previous academic research in applying such techniques directly to venture finance. To our knowledge, [17] is the first and the only one that has studied collaborative filtering on venture finance recommendation. However, [17] only showed some empirical results of a direct application of recommendation algorithms to venture finance, lacking a more sophisticated consideration or adjustment of recommendation methods to the unique domain, where the risk is a major concern. It is worth mentioning that [25] also considered risk in recommendation optimisation for a P2P lending investment recommendation problem. However, the authors failed to address the correlations between investments or to analyse the investors' risk-averse levels, making it significantly different from our paper.



(a) CrunchBase (b) MovieLens

Figure 2: Behaviour comparison by category/genre distribution.

Diversification in Recommender Systems. Recommendation diversification has become a hot research topic recently, and there are mainly two types of solutions. The first type of work tries to diversify the recommendation result explicitly [27, 22, 11]. For example, the authors of [27] explicitly introduced an intra-list similarity metric based on content-based features of the items and increased topic diversification by reducing the intra-list similarity. The authors of [22] defined an item set diversity based on averaged item-pair distances in a learnt latent space, and formulated the diversification problem into a binary optimisation problem to balance the accuracy and diversity. In [11], the authors discussed the diversification of recommended items as a temporal process so that the system delivers novel items with respect to the recommendations made in the past. And [19] discussed several evaluation metrics for diversification and novelty of the recommendation result. Generally, this type of work first defines a diversity metric, e.g., item category coverage or item distance in a latent space, and then proposes solutions to increase the measured diversity.

The second type of work achieves diversification by introducing the risk management concept from finance [9], where the investors can combine negatively or weakly correlated investments to diversify the risk of deficit using quantitative methods such as the modern portfolio theory [5]. In the recommendation scenario, items with the same topics or attributes usually have positive correlations with respect to users' preferences on them, making it higher risk to include them in one recommendation list. Thus by controlling the risk in the recommendation list, diversification of topics and results can be naturally achieved [16, 24, 4]. The authors of [16] first introduced the idea of portfolio theory to the mean-variance analysis of user-item preference modelling and then proposed a re-ranking algorithm to balance between the uncertainty and expectation of user preferences. The authors found that by increasing the risk-averse level, the item category diversification in the recommended ranked list can be improved. The authors of [24] studied a bi-portfolio scenario, where the recommendation lists from personalised and non-personalised methods were modelled as two portfolios. By proposing a risk-aware switching algorithm, they answered the dilemma of whether the personalised or non-personalised recommendations should be used. In [4], a contextual bandit algorithm is proposed to control the risk in mobile recommendations which dynamically address the user's risk levels. However, there has been no academic research how to address risk control in venture finance recommendation, which is the main focus of this paper.

3. THE CRUNCHBASE DATASET

CrunchBase is a repository of startup companies, individual partners, and financial institutes focusing on the US high-tech sectors [1]. With its self-description as a "free database of technology companies, people and investors that anyone can edit", CrunchBase maintains the investment events between investors (including financial institutes and individual partners) and investment opportunities (usually startup companies) associated with the total amount of raised funding[1] and time. According to [17], financial organisations and individual partners are significantly different in their investment behaviours. Thus in search of consistent properties, in this work we focus on only the financial organisations. We crawled the CrunchBase data from its official API[2] in May 2014. In total, we collected 62,926 investment events between 7,706 VCs and 18,026 startups from 1987 to 2014. We publicise the dataset online for research use[3].

By comparing the statistics between CrunchBase and the MovieLens 1M dataset (a well-known dataset for collaborative filtering research), we have identified quite different characteristics. First, as shown in Figure 2, VCs in the CrunchBase dataset tend to invest in a small number of industry categories, whereas users in the MovieLens dataset tend to rate a variety of movies, which often span more than 15 different genres. The reasons could be that, on one hand, VC's investment numbers in CrunchBase are generally much lower than the user rating numbers in MovieLens, due to the severe sparsity of the CrunchBase dataset (detailed below); on the other hand, VCs may be cautious in investing in unfamiliar industry categories to avoid risk.

Second, the CrunchBase dataset is much sparser than conventional recommendation data. The rating ratio of the MovieLens 1M dataset is about 4.46%, and it is 1.17% for another well-known movie-rating dataset Netflix. These ratios are already very low, but the observed investment ratio of CrunchBase is even lower: only 0.045%, about 1/97 of MovieLens 1M's and 1/25 of Netflix's. Such sparsity is reasonable since private investment activity is not as commonplace as simply watching movies. Also, the final investment decision will require the consent of both the company and VCs and usually involves a lengthy due diligence process [6].

The distinct characteristics of the venture finance investment behaviours revealed by the CrunchBase data motivate us to build new risk-aware recommendation algorithms tailored for the unique investor-investment ecosystem:

- From Figure 1, VCs tend to invest in opportunities with risk concerns rather than pure recommendations based on similarity.
- VCs usually cannot make extremely large numbers of investments. For each new investment opportunity, the VC may consider how it can fit into its holding portfolio. This motivates us to optimise the portfolio including both the invested startups and those to be recommended together.
- VCs normally focus on a small number of industry categories, unlike the wide range of genres in users' movie watching behaviours. This suggests that we cannot simply use a topic-diversification method commonly used for recommendation list diversification [27].

4. METHODOLOGY

4.1 Problem Formulation

Let us denote a VC (venture capital firm) as u and the available startup (investment opportunity) pool as $\mathcal{I}$. Suppose that VC u has already invested in m startups from the pool, and the recommender system is to seek another n startups from the pool for this VC to invest in. Without loss of generality, we denote the m holding investments (startups that the VC has already invested in) as $\boldsymbol{i} = (i_1, i_2, \ldots, i_m)$, and denote the startups to recommend as $\boldsymbol{j} = (j_1, j_2, \ldots, j_n)$ where $\boldsymbol{j} \subset \mathcal{I} \backslash \boldsymbol{i}$. We will also refer to the available startup set for VC u as $\mathcal{I}^u$ ($\mathcal{I}^u = \mathcal{I} \backslash \boldsymbol{i}$) in the sequel.

We define a joint portfolio $p(\boldsymbol{j})$ as a linear combination of the $m+n$ startups (m invested startups and n recommendations) with normalised weights:

$$p(\boldsymbol{j}) = \{(i_1, w_{i_1}), \ldots (i_m, w_{i_m}), (j_1, w_{j_1}), \ldots (j_n, w_{j_n})\} \quad (1)$$

with $\sum_{\alpha=1}^{m} w_{i_\alpha} + \sum_{\beta=1}^{n} w_{j_\beta} = 1$. Here the weights stand for estimated importance of each startup in the portfolio and will finally determine the ranking of the recommendation list [24]. Note that $\boldsymbol{i}$ is omitted in this notation for the sake of simplicity because the holding startups denoted by $\boldsymbol{i}$ are always contained in the joint portfolio.

We further denote VC u's preference on the joint portfolio $p(\boldsymbol{j})$ as $R_{u,p(\boldsymbol{j})}$, which is a weighted linear combination of the preferences on its component, as will be discussed later. According to PMF, $R_{u,p(\boldsymbol{j})}$ can be modelled as a random variable [14]. The utility function $U[R_{u,p(\boldsymbol{j})}]$ based on the random variable $R_{u,p(\boldsymbol{j})}$ is defined as a trade-off between the expected reward $\mathbb{E}[R_{u,p(\boldsymbol{j})}]$ and the associated risk. The risk is usually defined as the variance of the reward $\mathrm{Var}[R_{u,p(\boldsymbol{j})}]$ [12, 21, 24]. In the risk-averse case, it is subtracted from the expected reward $\mathbb{E}[R_{u,p(\boldsymbol{j})}]$ to form the utility function.

The objective function is thus to find n startups (with rankings) to recommend to the VC so that the VC's utility over the joint portfolio is optimised:

$$\boldsymbol{j}^* = \underbrace{\arg\max_{\boldsymbol{j}}}_{\text{startup selection}} \underbrace{\left[\max_{\boldsymbol{w_i},\boldsymbol{w_j}} \left(\mathbb{E}[R_{u,p(\boldsymbol{j})}] - b\,\mathrm{Var}[R_{u,p(\boldsymbol{j})}]\right)\right]}_{\text{portfolio optimisation}}. \quad (2)$$

Here we have already used vectors $\boldsymbol{w_i}$ and $\boldsymbol{w_j}$ to denote the weight vector of startups $\boldsymbol{i}$ and startups $\boldsymbol{j}$ respectively. The n recommended startups are then ranked according to their weights optimised in the process. Parameter b is the VC's risk-averse level. A higher b means that the VC is more risk-averse and is willing to sacrifice the expected reward to hedge the risk. It can be optimised globally (for all the VCs) or personally (adapted for each individual VC) from the data. We will show in the experiment section how b is determined and calibrated from the data.

We can see that there are two sub-problems in the objective function:

- **Portfolio optimisation**: given a candidate recommendation set of startups $\boldsymbol{j}$, to find the optimally allocated weights to maximise the utility of the joint portfolio $p(\boldsymbol{j})$. This part is discussed in Section 4.2.
- **Startup selection and ranking**: given a pool of available startups $\mathcal{I}^u$, to select a subset $\boldsymbol{j} \subseteq \mathcal{I}^u$ ($|\boldsymbol{j}| = n$) to form the joint portfolio $p(\boldsymbol{j})$. This part is discussed in Section 4.3.

[1] It is the total raised money in one round for a startup instead of indicated for each funding party. Therefore we choose not to use the funding amount information in this work.

[2] CrunchBase API: `http://developer.crunchbase.com`

[3] `http://www0.cs.ucl.ac.uk/staff/w.zhang/cb.html`

Algorithm 1 Sampling-based Startup Selection (Sampling)

```
Require: VC $u$, current invested startups $i_1, \dots, i_m$, candidate startup set $\mathcal{I}^c$, risk-averse parameter $b$, recommendation size $n$, utility function $U$, sampling time $T$.
  Initialise $\boldsymbol{j}^* \leftarrow \emptyset$
  for $t = 1 \dots T$ do
    Sample $\boldsymbol{j}_t = (j_1, j_2, \dots, j_n)$ from $\mathcal{I}^c$
    Build portfolio $p(\boldsymbol{j}_t)$ based on the joint set $(i_1, \dots, i_m, j_1, j_2, \dots, j_n)$ via Eq. (1)
    Calculate the maximum utility $U[R_{u,p(\boldsymbol{j}_t)}]$ via Eq. (3)
    if $U[R_{u,p(\boldsymbol{j}_t)}]$ is the largest utility so far then
      Update $\boldsymbol{j}^* \leftarrow \boldsymbol{j}_t$
    end if
  end for
  return Rank list $\boldsymbol{j}^*$
```

The two sub-problems are inter-connected. To optimise the portfolio we need to provide the selected startup subset, and to determine the optimal startups to recommend, we need to optimise each joint portfolio and compare between different startup selections.

4.2 Portfolio Optimisation

We first focus on the portfolio optimisation problem:

$$\max_{\boldsymbol{w_i}, \boldsymbol{w_j}} U[R_{u,p(\boldsymbol{j})}] = \max_{\boldsymbol{w_i}, \boldsymbol{w_j}} \mathbb{E}[R_{u,p(\boldsymbol{j})}] - b\,\mathrm{Var}[R_{u,p(\boldsymbol{j})}]. \tag{3}$$

To further simplify the notations, we will use $\boldsymbol{w}$ as the concatenation $(\boldsymbol{w_i}, \boldsymbol{w_j})$ in the sequel. We will also denote the startups contained in the joint portfolio as $\boldsymbol{\kappa}$ which is a concatenation $(\boldsymbol{i}, \boldsymbol{j})$. A startup in the joint portfolio is thus denoted by a single symbol κ ($\kappa \in \boldsymbol{\kappa}$). Now, the optimisation problem is simplified as $\max_{\boldsymbol{w}} \mathbb{E}[R_{u,p(\boldsymbol{j})}] - b\,\mathrm{Var}[R_{u,p(\boldsymbol{j})}]$. Here we allow flexible weights $\boldsymbol{w_i}$ of existing startups in the portfolio optimisation process as we assume the VC can adjust importance and priorities of them.

4.2.1 Portfolio-Level Preference

As mentioned before, we associate weights as the importance of startups in the portfolio. We also define the ranking order of startups by the importance (weight) order among all recommended items. Now that the problem is translated into a ranking problem, and thus we adopt a generalised definition of weight which can be either positive or negative [24]. An advantage of this treatment also lies in its analytical solution for weight optimisation.

As mentioned before, a VC u's preference on a portfolio is a random variable $R_{u,p(\boldsymbol{j})}$, which is a linear combination of the preference random variables of individual startups denoted by $r_{u,\kappa}$:

$$R_{u,p(\boldsymbol{j})} = \sum_{\kappa} w_\kappa r_{u,\kappa} = \boldsymbol{w}^T \boldsymbol{r}, \tag{4}$$

where $\boldsymbol{r}$ is the vector representation of the VC's preferences of startups in the portfolio. By denoting the mean and variance of the preference $r_{u,\kappa}$ as $\mu_{u,\kappa}$ and $\sigma^2_{u,\kappa}$, the expectation and variance of $R_{u,p(\boldsymbol{j})}$ are calculated as:

$$\mathbb{E}[R_{u,p(\boldsymbol{j})}] = \sum_{\kappa} w_\kappa \mathbb{E}[r_{u,\kappa}] = \boldsymbol{w}^T \boldsymbol{\mu}, \tag{5}$$

$$\mathrm{Var}[R_{u,p(\boldsymbol{j})}] = \sum_{\kappa} \sum_{\kappa'} w_\kappa w_{\kappa'} \,\mathrm{Cov}(\kappa, \kappa') = \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}. \tag{6}$$

Here we have used $\boldsymbol{\mu}$ to denote the vector of the preference expectations, and $\boldsymbol{\Sigma}$ to denote the covariance matrix whose (κ, κ')-th element is given by the covariance $\mathrm{Cov}(\kappa, \kappa') = \rho_{\kappa,\kappa'} \sigma_{u,\kappa} \sigma_{u,\kappa'}$, where $\rho_{\kappa,\kappa'}$ is the correlation between startup κ and κ' [12, 21] and can be estimated via industry category overlap [17] or latent factor vector cosine [24].

In this paper, we follow [26, 24] and use the PMF model [14] to obtain the probabilistic representations of the VC-startup preferences. Assuming that uncertainty in the preference originates from the uncertainty of the user latent factor estimation [26], we can estimate the expectation and variance of $r_{u,\kappa}$ as follows:

$$\mu_{u,\kappa} = \mathbb{E}[\boldsymbol{p}_u]^T \boldsymbol{q}_\kappa, \tag{7}$$

$$\sigma^2_{u,\kappa} = \boldsymbol{q}_\kappa^T \,\mathrm{Cov}[\boldsymbol{p}_u] \boldsymbol{q}_\kappa. \tag{8}$$

Here $\boldsymbol{p}_u$ and $\boldsymbol{q}_\kappa$ can be estimated by the MAP solution of $p(r_{u,i} | \boldsymbol{p}_u, \boldsymbol{q}_i, \sigma^2) = \mathcal{N}(r_{u,i} | \boldsymbol{p}_u^T \boldsymbol{q}_i, \sigma^2)$. Due to space limit, the technical details are omitted here and we refer to the PMF setting in [26, 24].

4.2.2 Portfolio Weight Optimisation

Integrating Eqs. (5) and (6) into Eq. (3), we translate the portfolio optimisation problem into the portfolio weight optimisation problem:

$$\max_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{\mu} - b \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}, \tag{9}$$

which is a standard quadratic optimisation problem. In the case when $\boldsymbol{w}$ can take any value in $\mathbb{R}^{m+n}$, there is an analytic solution explicitly given in [24]. Without loss of generality we assume an optimised portfolio is ranked according to the item weight such that

$$w_{i_1} > w_{i_2} > \cdots > w_{i_m}, \text{ and } w_{j_1} > w_{j_2} > \cdots > w_{j_n},$$

i.e., elements in $\boldsymbol{i}$ and $\boldsymbol{j}$ are ranked by their importance.

4.3 Startup Selection and Ranking

In Section 4.2, we discussed the model to estimate the maximum investment utility $U[R_{u,p(\boldsymbol{j})}]$ given the recommended startups $\boldsymbol{j}$. In this section, we discuss the algorithms to efficiently find the optimal recommendation set $\boldsymbol{j}$ from a large candidate corpus $\mathcal{I}^u$.

Considering the fact that the possible startup combination space is extremely large ($C^n_{|\mathcal{I}^u|}$), we need to first reduce the candidate set by pre-selecting a size-N candidate startup set $\mathcal{I}^c \subseteq \mathcal{I}^u$ ($|\mathcal{I}^c| = N$) with the highest expected preferences $\mu_{u,\kappa}$ estimated from PMF (Eq. (7)). Then within the candidate set $\mathcal{I}^c$ we determine the final ranked list of startups $\boldsymbol{j}$. All of our proposed algorithms share the procedure of first choosing the size-N candidate set and then determining the final size-n ranked recommendations.

With the candidate set $\mathcal{I}^c$ we propose the following 5 different algorithms to find the optimal selections and their ranking.

4.3.1 Startup Selection by Sampling

A straightforward solution is to use a sampling method to approximate the optimal solution, which greatly reduces the computational cost. The details are presented in Algorithm 1. By sampling n-sized startup combinations among the N candidates for T times and picking the combination with the highest utility, we can get a globally $1/T$ best combination in expectation. As $T \to C^n_{|\mathcal{I}^c|}$, the performance of the sampling-based method will converge to the globally optimal solution, i.e., the portfolio $p(\boldsymbol{j}^*)$ leading to the highest utility $U[R_{u,p(\boldsymbol{j}^*)}]$.

4.3.2 Startup Selection by Individual Score Ranking

This is a simple ranking algorithm that ranks the startup utility by considering individual startups joining the current portfolio. We denote the joint portfolio including one candidate startup j as $p(j)$, and the maximum utility $U[R_{u,p(j)}]$ with j will act as the ranking score of j. Based on the score of each candidate startup, we can rank them and choose the

Algorithm 2 Individual Startup Selection (Idv)

```
Require: VC $u$, current invested startups $i_1, \ldots, i_m$, candidate startup set $\mathcal{I}^c$, risk-averse parameter $b$, recommendation size $n$, utility function $U$.
  for each candidate startup $j$ in $\mathcal{I}^c$ do
    Build portfolio $p(j)$ based on the joint set $(i_1, \ldots, i_m, j)$ via Eq. (1)
    Calculate the maximum utility $U[R_{u,p(j)}]$ via Eq. (3)
  end for
  return Rank list $\boldsymbol{j}^*$ of $n$ startups with highest $U[R_{u,p(j)}]$
```

Algorithm 3 Sequential Startup Selection (Seq)

```
Require: VC $u$, current invested startups $i_1, \ldots, i_m$, candidate startup set $\mathcal{I}^c$, risk-averse parameter $b$, recommendation size $n$, utility function $U$.
  Initialise startup set $\boldsymbol{j}^* \leftarrow \emptyset$
  for $l = 1 \ldots n$ do
    Select the optimal startup $j_l^*$ in $\mathcal{I}^c$ such that
        $j_l^* = \arg\max_{j_l \in \mathcal{I}^c} U[R_{u,p(\boldsymbol{j}^*, j_l)}]$
    where the portfolio $p(\boldsymbol{j}^*, j_l)$ is built based on startups $\{i_1, \ldots, i_m, j_1^*, j_2^*, \ldots, j_{l-1}^*, j_l\}$ via Eq. (1)
    $\boldsymbol{j}^* \leftarrow \boldsymbol{j}^* \cup \{j_l^*\}$
    $\mathcal{I}^c \leftarrow \mathcal{I}^c \backslash j_l^*$
  end for
  return Rank list $\boldsymbol{j}^*$ with the selection order
```

top-n startups with the highest scores. This procedure is given in Algorithm 2.

As we can see, Algorithm 2 is quite straightforward: selecting each startup based on the utility it brings. However, this algorithm fails to consider the correlation among the n recommended startups.

4.3.3 Sequential Startup Selection

Inspired by [21, 16], we select the startups incrementally to approximate the optimal solution with a large computational cost reduced. For each iteration, in a greedy fashion, we select one startup which can bring the highest increase in the utility function when being added into the current portfolio. This procedure is described in Algorithm 3.

Sequential methods have been adopted in previous work on top-N recommendation [16] and webpage ranking [21]. Though it is a greedy method, it has shown high efficiency and good empirical performances.

4.3.4 Startup Selection by Weight Ranking

With the candidate startup set $\mathcal{I}^c$, we can build a portfolio p_g with all the candidate startups and the invested startups $\boldsymbol{i}$. Then we can apply the portfolio optimisation according to Eq. (3) to obtain the optimal weights for all the candidates. We rank their weights and select the top n. This algorithm is illustrated in Algorithm 4.

This algorithm takes into account the inter-relationship between each pair of candidate startups in $\mathcal{I}^c$. However, by selecting the top n candidates with the highest portfolio weights, the resulting portfolio is already different from the global portfolio p_g. In other words, the top n candidates are selected based on a globally learnt weight ranking rather than a direct optimisation on the joint portfolio with only these n candidates added, which is a discrepancy.

4.3.5 Startup Selection by Weight Filtering

Here we implement a backward sequential method shown in Algorithm 5. In each iteration, we build the global portfolio p_g based on the invested startups and the startups in the candidate set, optimise the portfolio to obtain the optimal weights according to Eq. (3), and remove the candidate startup with the lowest weight from the candidate startup set. This process iterates until the resulting candidate startup set shrinks to the size of n. Similar to the weight ranking algorithm, the weight filtering algorithm is also based on the weights obtained by optimising the portfolio constructed by the overall startup set rather than the selected subset, and thus suffers from the same discrepancy.

Algorithm 4 Weight-based Startup Selection (Weight)

```
Require: VC $u$, current invested startups $i_1, \ldots, i_m$, candidate startup set $\mathcal{I}^c$, risk-averse parameter $b$, recommendation size $n$, utility function $U$.
  Build the portfolio $p_g$ based on the joint set $\boldsymbol{g} = \{i_1, \ldots, i_m\} \cup \mathcal{I}^c$ via Eq. (1)
  Calculate the optimal weights $\boldsymbol{w_g}$ via Eq. (3)
  Sort the candidate startups by their weight in $\boldsymbol{w_g}$
  return Rank list $\boldsymbol{j}^*$ of $n$ startups with highest weights
```

Algorithm 5 Filtering-based Startup Selection (Filtering)

```
Require: VC $u$, current invested startups $i_1, \ldots, i_m$, candidate startup set $\mathcal{I}^c$, risk-averse parameter $b$, recommendation size $n$, utility function $U$.
  while $|\mathcal{I}^c| > n$ do
    Build portfolio $p_g$ based on the joint startup set $\boldsymbol{g} = \{i_1, \ldots, i_m\} \cup \mathcal{I}^c$ via Eq. (1)
    Obtain the optimal weights $\boldsymbol{w_g}$ via Eq. (3)
    Obtain $j_f$ with the lowest weight in $\boldsymbol{w_g}$
    Update $\mathcal{I}^c \leftarrow \mathcal{I}^c \backslash j_f$
  end while
  return Rank list of startups in $\mathcal{I}^c$ by the optimal weights
```

4.4 Adaptive Risk-Averse Level

With different industry category focuses and investment strategies, different VCs may have different risk-averse levels, represented as the parameter b in our model Eq. (3). All the above discussed algorithms take b as a model parameter, yet b can also be learnt for each VC u and thus the portfolio can be optimised in a personalised manner.

In order to adaptively learn this parameter for each VC, we conduct a cross validation on the training data, tune the parameter b_u for each VC u and pick its optimal value for each VC which maximises the startup ranking evaluation measure (e.g., NDCG) on the validation data. Then the learnt b_u for VC u will be used in the test phase.

5. EXPERIMENTS

After describing the experiment setup in Section 5.1, we present the experimental results in three parts. (i) In Section 5.2, we focus on the case of recommending the next startup, i.e., $n = 1$. With only one startup to be recommended, the correlation between the recommended startup and the existing portfolio plays the key role in the decision process. (ii) In Section 5.3, we study the cases where multiple recommendations are made, i.e., $n = 3, 5, 10$. In these cases, not only the correlation between each of the new items and the existing investments, but also correlations among the recommended ones are important. (iii) In Section 5.4, we further perform a statistical data analysis on the optimal risk-averse level b among the VCs.

5.1 Experimental Setup

5.1.1 Data Processing

As described in Section 3, we base our experiments on the CrunchBase dataset that we collected. We first divide the CrunchBase dataset into training set and test set with 2:1 ratio for each VC according to investment time. Splitting this way, the total investment number is 69,422 in the training set and 24,138 in the test set.

We label a recorded investment from a VC to a startup as 1, i.e., a positive observation. Since it is a one-class training data [15], we follow [15] to perform a user-oriented negative item sampling process, i.e., for each VC, we sample the same number of negative data points as its observed positive points and label them with 0. We train the PMF model to obtain the latent factors for the VCs and startups as well as the probabilistic representation of the VC latent factors (as discussed in Section 4.2). Note that our focus in this paper is not on the performance comparison against the state-of-the-art recommendation methods, but on investigating how the proposed portfolio-based algorithms can *improve* the recommendation results. The choice of PMF enables a coherent view of the effectiveness of the proposed method as it enables pure model-based mean/variance/covariance estimation for building portfolios.

5.1.2 *Compared Algorithms*

Three types of state-of-the-art algorithms are compared: the conventional recommendation algorithms, portfolio-based algorithms and adaptive-b portfolio-based algorithms. As described in Section 4.3, we always first determine a candidate item set $\mathcal{I}^c$ obtained as the top-N items from PMF, before applying any item selection and ranking algorithm.

Random sampling (Random). As a baseline, we compare our results with randomly-chosen n startups from the candidate set.

PMF. PMF method directly gives the top-n startup determined by the maximum a posteriori (MAP) estimation of the VC's preference regarding each startup.

Portfolio-based methods. These methods include Sampling, Sequential Selection (Seq), Individual Score Ranking (Idv), Weight Ranking (Weight), and Weight Filtering (Filtering). Details of each algorithm are described in Section 4.3.

Adaptive-b portfolio-based methods. These methods adopt a personalised risk-averse level b for each VC, as described in Section 4.4. We denote them with '-A' following the algorithm's name.

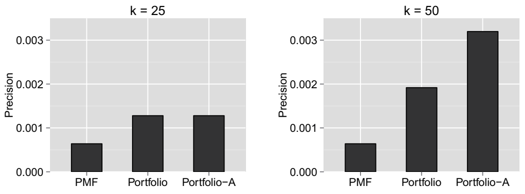
5.1.3 *Evaluation Measures*

As the task falls into the category of top-N recommendation based on implicit data, we follow previous work [26, 23] to evaluate the recommendation performances with the following ranking evaluation measures: Precision (P@n), Normalised Discounted Cumulative Gain (NDCG@n) [10], and Mean Reciprocal Rank (MRR@n) [20]. For each algorithm, we calculate the recommendation performances (with respect to these three measures) in regard to each test VC, then average for all test VCs to get the average performances.

5.2 Next Startup Recommendation

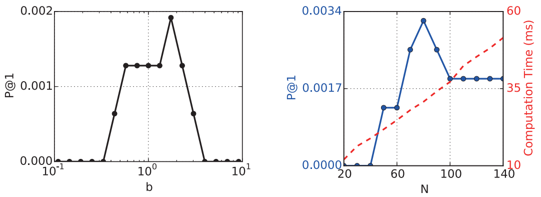
In this subsection, we focus on the case of $n = 1$, i.e., only one startup is recommended for each test VC. In this case, Sampling, Idv, and Seq are essentially the same, denoted as Portfolio. We compare Portfolio and its adaptive-b version Portfolio-A with the baseline algorithm PMF. As P@1, NDCG@1 and MRR@1 provide exactly the same result in the case $n = 1$, we only use P@1 as the measure here.

Figure 3 shows the result comparison between PMF, Portfolio and the adaptive-b version Portfolio-A, for different latent space dimensions ($k = 25$ and $k = 50$). The candidate size N and the risk-averse level b are both tuned to optimal to obtain the Portfolio performance.



(a) k = 25 (b) k = 50

Figure 3: Precision comparison when $n = 1$.



(a) Risk-averse level b (b) Candidate set size N

Figure 4: Performance against parameters when $n = 1$. The computation time is calculated as per test VC.

From Figure 3, we have the following observations. (i) For both cases ($k = 25$ and $k = 50$), Portfolio and Portfolio-A perform significantly better than PMF. (ii) Comparing between $k = 25$ and $k = 50$, the performance of PMF keeps unchanged, whilst the performance improvements by Portfolio and Portfolio-A are even higher when $k = 50$. (iii) In the case of $k = 50$, Portfolio is outperformed by Portfolio-A. These facts show the effectiveness of our proposed algorithms over PMF, indicating that recommendations with risk concerns are superior in catching the VCs' investment behaviour, and different VCs have different risk-averse levels. We will extend these discussions in Section 5.4.

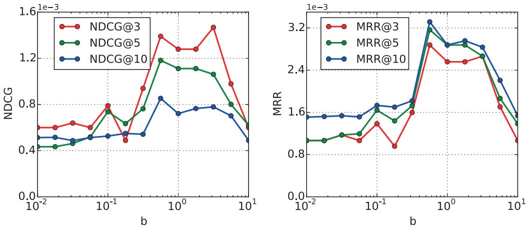
In Figure 4, we show the effect of parameters b and candidate size N. From Figure 4(a), we can see that the performance peaks when the global risk-averse level $b = 1$. The global b reflects the overall risk-averse level for all test VCs, and the peak value indicates that a certain risk-averse level optimally catches the VCs' overall investment behaviour and leads to the best recommendation result. In Figure 4(b), we show the effect of tuning the candidate size N with the corresponding computational time shown as a reference. We can see that when the candidate set gets larger, the performance first increases and then drops to a lower level. It indicates that though an increasing candidate size N adds more options for the algorithm to choose, an oversized candidate set may also mislead the algorithm due to overfitted estimation of latent factors from PMF. The computation time increases linearly as the candidate set enlarges, so one may find a trade-off between the candidate size N and the computation speed as desired.

5.3 Top-N Startup Recommendation

Here we present the results for multiple item recommendations. In this task, ranking measures NDCG@n and MRR@n are also used in addition to P@n. In Table 1, we compare the results between baseline algorithms Random and PMF, portfolio-based algorithms and adaptive-b portfolio-based algorithms. All the (hyper)parameters are optimised with cross-validation. From Table 1, we can make the following observations. (i) All the proposed algorithms have great improvements over the results of PMF for all three measures (with a few exceptions for Weight and Filtering), showing the effectiveness of our algorithms in a multiple item recommendation task generally. (ii) Among the (non-adaptive) portfolio-based methods, Sampling, Seq and Idv

Table 1: Performance comparison by different algorithms. The improvement(-A) is calculated from the best Portfolio(-A) algorithm over PMF for each measure. (All the numbers except the percentages are in the unit of 0.001.)

n	@3			@5			@10		
Measure / Rec	Pre	NDCG	MRR	Pre	NDCG	MRR	Pre	NDCG	MRR
Random	0.746	0.849	1.662	0.665	0.646	1.452	0.659	0.647	1.859
PMF	0.853	0.829	1.492	0.895	0.87	1.939	0.703	0.748	2.288
Sampling	1.492	1.729	3.41	1.279	1.235	3.218	0.959	0.955	3.177
Seq	1.279	1.429	2.771	1.151	1.126	2.931	0.831	0.871	2.931
Idv	1.279	1.239	2.451	1.151	1.105	2.771	0.959	0.987	2.937
Weight	1.066	1.279	2.558	0.767	0.946	2.558	0.703	0.741	2.835
Filtering	1.066	1.35	2.771	0.895	0.927	2.398	0.767	0.781	2.394
Improvement	74.9%	108.6%	128.6%	42.9%	42.0%	66.0%	36.4%	32.0%	38.9%
Sampling-A	5.968	6.126	11.509	5.243	5.395	12.479	3.964	4.112	12.293
Seq-A	1.705	1.879	3.73	1.662	1.74	3.986	1.087	1.299	4.276
Idv-A	1.705	1.768	3.41	1.662	1.616	3.89	1.087	1.304	4.452
Weight-A	1.066	1.279	2.558	0.767	1.008	2.685	0.703	0.786	2.92
Filtering-A	1.705	1.807	3.41	1.407	1.391	3.325	1.023	1.163	3.773
Improvement-A	599.6%	639.0%	671.4%	485.8%	520.1%	543.6%	463.9%	449.7%	437.3%



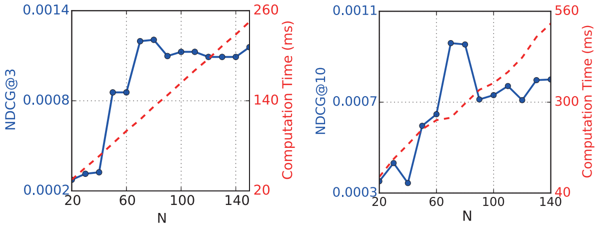
(a) NDCG@n (b) MRR@n

Figure 5: Performance against risk-averse level b.

perform better than Filtering or Weight. This fact indicates that top-down algorithms like Filtering and Weight, which filter out items according to the *direct* portfolio optimisation weight for the *overall* joint portfolio (invested startups plus *all* the candidate startups), do not work as well as the group-selection-based Sampling, or the bottom-up Seq and Simple. This is due to the discrepancy between the weights learnt by a global optimisation and the weights learnt directly for the chosen group, as mentioned in Section 4.3. (iii) Adaptive-b portfolio-based algorithms perform better than non-adaptive ones, showing that each VC's risk-averse level is indeed different, so by adaptively fitting the VC's own risk-averse level, the performance can be further improved. (iv) Sampling(-A) outperforms all other algorithms. Again we ascribe its superior performances to its group-selection nature, as according to Eq. (3), a group selection method can achieve the best results. The other methods Seq(-A), Idv(-A), Weight(-A) and Filtering(-A) are further approximations than Sampling to approach the exact solution. (iv) Among the two baselines, PMF performs better than Random, indicating the effectiveness of the PMF model to catch the latent factors of VCs and startups.

5.3.1 Parameter Tuning

In Figure 5 we present the influence of b evaluated by NDCG and MRR, with different $n = 3, 5, 10$. From Figure 5, we can see that for each n and each measure, the performance has a peak around $b = 1$, which is consistent with the case of P@1 in Section 5.2. Furthermore, comparing different top-n tasks, as n increases, NDCG@n decreases whilst MRR@n increases. This can be explained by the sparsity of the dataset. When only a small number of recommendations are made (e.g., 3), only a smaller number of VCs are provided with the correct recommendations within the recommendation list. Whereas when the number of recommendations is enlarged (e.g., 10), more users are provided with correct recommendation within the longer recommendation list. According to the definition of MRR [20], only the first correct recommendation counts. Thus, the result of MRR always increases with n in this case. On the other hand, NDCG considers the whole ranking list in a discounted manner, and, due to the sparseness of the dataset, it naturally decreases as n increases.



(a) Top 3 (b) Top 10

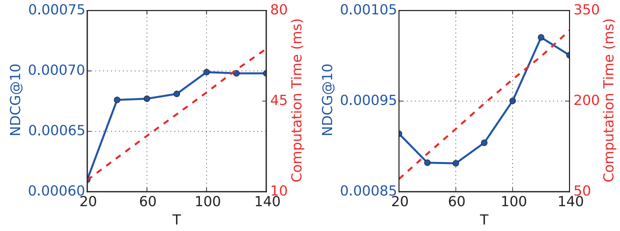
Figure 6: Impact of candidate size N on performance and computational time for Seq. The computation time is calculated as per test VC.

In Figure 6, we plot the influence of the candidate size N for the algorithm Seq, when $n = 3$ and $n = 10$. We can see that the performance first increases as the candidate size gets larger, then slightly drops after peaking around $N = 70$. This result may be due to the overfitting of PMF as mentioned before. Meanwhile, we plot the computation time for each N accordingly. We can see the computation time increases linearly with the candidate size N. Similar to the case when $n = 1$, we may trade off some performance for the computation speed by choosing a smaller candidate set than optimal.

In Figure 7, we plot the influence of the sampling time T in Algorithm 1 on the performances for both the non-adaptive and the adaptive-b cases. We can see that the performance peaks around $T = 120$ for both cases. The decrease of performance after the peak in Figure 7(b) may also be caused by the overfitting of the PMF model. Again, for the sampling method, we may also seek a trade-off between the ranking performance and efficiency by tuning the sampling time T.

5.4 Risk-Averse Level Analysis

In Figure 8(a), we plot the distribution of b optimised for individual VCs. We can see that VCs generally form two clusters: a risk-sensitive group whose risk-averse levels b are larger than 0.1 and a risk-neutral group whose risk-averse levels are much smaller. We also have interesting findings on the relationship between the number of investments made by a VC and its optimal risk-averse level b, shown in Figure 8(b). Here we applied a log-scale on the investment number, because the VCs' investment activity distribution is power-



(a) Sampling (b) Sampling-A

Figure 7: Impact of sampling time T on the performance of Sampling(-A) and computational time. The computation time is calculated as per test VC.

law [17]. We can find that on the log-log plot, the correlation of the two is negative: companies holding a large number of investments tend to be more risk-neutral, whilist companies with smaller investment scales tend to have higher risk-averse levels (more risk-sensitive). By inspecting company names, we can find some of the largest VCs in the world, such as Start-Up Chile, Sequoia Capital and Accel Partners, fall in the category of the risk-neutral group, whereas smaller VCs, such as Allegro Venture Partners, are more risk-averse. These companies are tagged on Figure 8(b) for reference. These observations coincide with the intuition that the fewer investments held by a VC, the more careful it should be in making new investments, whereas, for a VC with a great number of investments, the risk may have already been diversified in its holding portfolio, and thus there is less risk concerns in making new investments compared to smaller VCs.

6. CONCLUSIONS

In this paper, we proposed a portfolio optimisation framework to solve the information filtering problem in venture finance, specifically by optimising the joint portfolio of VC's holding investments and potential investment opportunities. We exploited the variance defined on latent factors using a probabilistic matrix factorisation model, and optimised the joint portfolio towards a trade-off between expected preference and uncertainty. We divided the problem into two connected sub-problems including an item selection problem and a portfolio optimisation problem, and proposed five different algorithms to solve it. Through the experiments, we demonstrated significant improvement by using our portfolio-based algorithms and adaptive-b portfolio-based algorithms, compared with a direct PMF approach. In addition, we discussed the influence of the risk-averse level b, and conducted a data analysis over the distribution of risk-averse levels among the VCs.

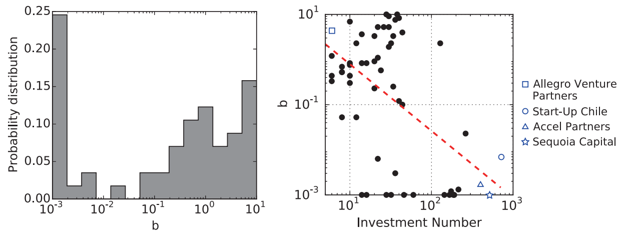
For future work, we are interested in investigating the relation between the users' past investment scales and their levels of risk appetite, and how this can affect the portfolio optimisation process. Also, we will investigate the dynamic portfolio optimisation problem where the portfolio is optimised over time. Finally, we are also interested in including more features to our model, such as the amount of investment, investment actions (e.g., buying back or reinvesting) and how these can facilitate the model.

7. REFERENCES

[1] O. T. Alexy, J. H. Block, P. Sandner, and A. L. Ter Wal. Social capital of venture capitalists and start-up funding. *Small Business Economics*, 39(4):835–851, 2012.

[2] S. Anthony. Is venture capital broken? *Harvard Business Review Blog*, 2012.

[3] T. Bhaskar and G. Subramanian. Loan recommender system for microfinance loans: Increasing efficiency to assist growth. *Journal of Financial Services Marketing*, 15(4):334–345, 2011.



(a) b distribution (b) Correlation

Figure 8: Data analysis on the personalised b. (a) Distribution of personalised b. (b) Correlation between VC's investment number and b for majority VCs.

[4] D. Bouneffouf. R-ucb: a contextual bandit algorithm for risk-aware recommender systems. *arXiv:1408.2195*, 2014.

[5] E. J. Elton and M. J. Gruber. *Modern portfolio theory and investment analysis*. J. Wiley and Sons, 2006.

[6] B. Flyvbjerg. Quality control and due diligence in project management: Getting decisions right by taking the outside view. *International Journal of Project Management*, 31(5):760–774, 2013.

[7] V. H. Fried and R. D. Hisrich. Toward a Model of Venture Capital Investment Decision Making. *Financial Management*, 23(3):28–37, 1994.

[8] R. Hu and P. Pu. Enhancing recommendation diversity with organization interfaces. In *IUI*, 2011.

[9] J. Hull. *Risk Management and Financial Institutions,+ Web Site*, volume 733. John Wiley & Sons, 2012.

[10] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *TOIS*, 20(4):422–446, 2002.

[11] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *SIGIR*, 2010.

[12] H. Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[13] A. Metrick and A. Yasuda. *Venture capital and the finance of innovation,2nd Edition*. John Wiley and Sons, Inc, 2010.

[14] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

[15] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, 2008.

[16] Y. Shi, X. Zhao, J. Wang, M. Larson, and A. Hanjalic. Adaptive diversification of recommendation results via latent factor portfolio. In *SIGIR*, 2012.

[17] T. Stone, W. Zhang, and X. Zhao. An empirical study of top-n recommendation for venture finance. In *CIKM*, 2013.

[18] T. T. Tyebjee and A. V. Bruno. A Model of Venture Capitalist Investment Activity. *Management Science*, 30(9):1051–1066, 1984.

[19] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*, 2011.

[20] E. M. Voorhees et al. The trec-8 question answering track report. In *TREC*, 1999.

[21] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, 2009.

[22] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*, 2008.

[23] W. Zhang, T. Chen, J. Wang, and Y. Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*, 2013.

[24] W. Zhang, J. Wang, B. Chen, and X. Zhao. To personalize or not: A risk management perspective. In *RecSys*, 2013.

[25] H. Zhao, L. Wu, Q. Liu, Y. Ge, and E. Chen. Investment recommendation in p2p lending: A portfolio perspective with risk management. In *ICDM*, 2014.

[26] X. Zhao, W. Zhang, and J. Wang. Interactive collaborative filtering. In *CIKM*, 2013.

[27] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.