Product Supply Optimization for Crowdfunding Campaigns

Guifeng Wang, Qi Liu, Hongke Zhao, Chuanren Liu, *Member, IEEE*,
Tong Xu *Member, IEEE*, Enhong Chen, *Senior Member, IEEE*

Abstract—Recent years have witnessed the rapid development of Finance Internet platforms, specifically, crowdfunding, which is for creators designing campaigns (projects) to collect funds from the public. Usually, the limited budget of a creator is manually divided into several perks (reward options), which should fit various market demand and further bring different monetary contributions for the campaign. Therefore, it is very challenging for each creator to design an effective allocation of perks when launching a campaign. Indeed, our aim is to enhance the funding performance of newly proposed campaigns, with a focus on optimizing the product supply of perks. In this paper, given the expected budget and the perks of a campaign, we propose a novel solution to automatically recommend the optimal product supply for balancing the expected return of this campaign against the risk. Along this line, we define it as a constrained portfolio selection problem, where the investment volume is measured by a multi-task learning method, and we adopt two kinds of methods for task splitting. Furthermore, we extend the investment volume prediction model with inner competition for capturing the competitive relationship between the perks in one campaign. Finally, extensive experiments on real-world crowdfunding data clearly prove the performance significantly.

Index Terms—multi-task learning (MTL), modern portfolio theory, competition mining, product supply, crowdfunding.

1 Introduction

With the rapid development of the Internet, crowdfunding, a relatively new practice of marketing which provides a revolutionary way for a large group of individuals (the 'crowd') to support ideas and campaigns across a wide range of domains (e.g. technology, film, art), has rapidly risen in popularity [1]. According to the official website data of two biggest American crowdfunding platforms (Kickstarter and Indiegogo), until April 2018, investors in Kickstarter has exceeded 14 million[1], and funds raised on Indiegogo in the past two years increased 1000%.[2]

When launching a campaign (project) on crowdfunding platforms, like Kickstarter and Indiegogo, the creators (individuals or startups) want to solicit as many funds as possible or expand their awareness from investors (i.e., backers, contributors, buyers) by carefully showing their stories, goals (funding amount), reward options (often vowing future products) and so on. Even though, statistics show that only around 40% of the campaigns succeed in reaching their pledged goals [2]. Therefore, improving the success rate of a campaign and inferring the impacts of specific factors (e.g., the campaign design, the campaign descriptions, the social networks of creators) on investor decision have become research hotspots.

- *G. Wang, Q. Liu, H. Zhao, T. Xu and E. Chen are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: wgf1109@mail.ustc.edu.cn, qiliuql@ustc.edu.cn, hkzh@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn.*
- *C. Liu is with the Decision Science and Management Information Systems Department, Drexel University, Philadelphia, PA 19104. E-mail: chuanren.liu@drexel.edu.*

Manuscript received Aug. 6, 2018; revised Oct. 31, 2018. (Corresponding author: Enhong Chen.)

1. www.kickstarter.com/
2. www.indiegogo.com/contact/press

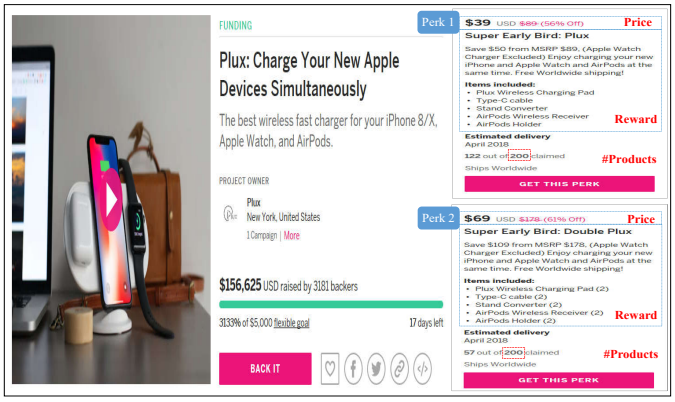


Fig. 1: An example of the campaign (Indiegogo.com).

Since the investors in crowdfunding are sufficiently heterogeneous in product valuations, a campaign usually offers a variety of rewards in the form of *perks* [3] for soliciting more funds. For instance, as shown in Figure 1, the creator of this campaign divided her budget and offered a line of perks with different levels of prices (e.g. $39), product rewards (e.g. one Plux) and the claimed product supply (e.g. 200 for Perk 1) to maximize the expected funding or awareness of products. The setting of perks is a kind of marketing strategy, for example, the more diversified perks are provided, the more backers will be covered. However, the number of different perks is limited, and all of them should be under the budget of a creator.

Therefore, the problem of how to automatically help creators optimally divide their budgets according to the market states, i.e. by optimizing the product supply of each perk, remains pretty much open. Indeed, it is very challenging to

recommend an appropriate product supply to each perk in a campaign. First, it is difficult to estimate the return and risk before the campaign is launched, since the investments are potentially affected by many static and temporal factors, such as the perk or campaign descriptions and the funding dynamics. Thus, how to reasonably estimate the return and risk remains pretty much open. Second, before launching a campaign, a creator will consider the budget, and the choice of the product supply for each perk is limited due to the entire budget of one campaign. How to design a flexible model that can take the budget into consideration becomes one challenge. Last but not least, in a competitive crowdfunding platform, especially, the fierce competition among different perks in one campaign is a significant factor to shake bakers' choices. How to mine the competitive relationships among perks in one campaign to improve the optimization of product supply of each perk? In summary, the special return and risk estimation, the limited budget and perks competition from one campaign compose three main challenges of the problem we study.

To address the first two challenges, limited budget and special estimation of return and risk, we provide a preliminary study [4] on the product supply optimization problem in the crowdfunding platform. Specifically, we propose a novel solution to automatically recommend the optimal product supply to each pre-defined perk (including the prices and the reward products are given) so as to maximize the expected return and minimize the risk of the campaign, simultaneously. Along this line, inspired by the modern portfolio theory [5], [6], we first define it as a constrained portfolio optimization problem, where the constraint is the budget of the campaign. Then, considering the relevance and heterogeneity among all perks in the platform when attracting investments, we propose a trace-norm multi-task learning method to estimate the future investment volume for each campaign. In this way, the risk on the settings of product supply to the perks/campaigns could also be estimated. Next, by solving the optimization problem with Alternating Optimization Method, the optimal product supply can be recommended to each perk in a new campaign.

Furthermore, we extend our previous work and study the product supply optimization problem with multiple competitive perks in one campaign. In our previous estimation model, we adopt the multi-task learning method with manual task partition (named PSO). Considering this partition method is not general enough, we firstly extend a task auto-splitting method in multi-task learning (MTL), and combine it into our proposed frameworks (named APSO). Secondly, although MTL method is able to capture the relevance and heterogeneity among all perks in the platform, and it expresses the competitive relationship between one perk and all the other perks, including other campaigns' perks, which is called the exterior competition. We think it cannot focus on the competitive relationship between perks in one campaign, which we called **inner competition**. In one campaign, perks turn to compete with each other to attract the attention of backers, as shown in Figure 1. Therefore, we argue, in order to predict the investment volume more accurately, it is essential to take the competition effect among perks in one campaign into consideration. Specifically, given a campaign with competitive perks, we study how to incorporate competition into APSO model, and jointly learn investment volume and perk competition in a unified framework.

In summary, the contributions of this paper are mainly shown in three aspects. Firstly, we propose an automatic task-splitting method for avoiding personal error. Secondly, we define inner competition to measure the competitive relationship between perks, and come up with a new product supply optimization method (named APSO-C) by incorporating the competition into the APSO model. Besides, we conduct extensive experiments on the real-world crowdfunding data that was crawled from Indiegogo.com. The results clearly prove that our multi-task learning method could more precisely estimate the investment volume of each perk, and meanwhile, the optimized product supply can improve the future performance of campaigns significantly. To the best of our knowledge, this is the first comprehensive attempt at assisting creators to enhance the performance of newly proposed campaigns, with a focus on optimizing the product supply from a data-driven way, and this idea can also be applied on optimizing other features of the campaign.

2 PRELIMINARIES

In this section, we first introduce the working mechanism of crowdfunding platform from creators' perspective. Then, we introduce the collected dataset from an American famous crowdfunding platform (Indiegogo.com) and how we construct features based on the data.

2.1 Creators' Job

In crowdfunding platforms, a creator is who launch a campaign or project in the platform for reaching her motivations, i.e., collect enough money or early advertise for their products. Before launching campaigns, creators need to do a lot of jobs for showing their creative ideas or fantastic products in different ways, for example, they reveal through photos view, video presentation, and textual description. Besides, creators should design proper perks (rewards) since they are an integral part of the crowdfunding process. Perks or rewards are offered by creators in exchange for contributions to their campaigns, and they are one of the best ways to spread the word and make contributors feel valued. Well-planned perks are the heart of any great campaign. Before launching campaigns, creators should design product supplies and prices of perks. As fresh creators are short of experience and they can not estimate the real demand, sometimes, their designs are not so desirable. So our aim is to optimize their designs on product supply for helping them reach their expectation.

2.2 Dataset and Constructed Features

Indiegogo data includes the campaign information, perk information, and some mutual records of creators and investors[3]. For instance, it contains 14,143 launched campaigns for more than 18 billion funds (including 98,923

3. The data used in this article could be reached from: https://drive.google.com/drive/folders/12bCdJm1IDcoGTJ8muRyAK8AcsbX2Kbi?usp=sharing

TABLE 1: The information of features.

Feature Level	Feature Type	Feature Source	Feature	Description
Perk Feature	Numerical	Perk Profile	Price	Unit price of the perk
			Featured	Whether the perk is recommended by the creator
			Shipping	Whether the perk is need to be shipping
			Delivery Term	How long will the investor get the reward
			Preset Num	Preset number of the product supply quantity
		Perk Summary	Perk Option Num	Number of perk options
			Avg Perk Price	Average of all perks' price of the campaign
			Var Perk Price	Variance of all perks' price of the campaign
	Textual	Perk Profile	Description	Detailed description of the perk
Campaign Feature	Numerical	Campaign Profile	Duration	Declared funding days of the campaign
			Goal	Declared funding amount of the campaign
			Currency	The currency for paying the perks, such as USD
			Created time	Created time of the campaign
			Funding type	The type of campaign, i.e. the funding amount is flexible or fixed
			Owner type	Purpose of the campaign, such as business, individual, non-profit
			Category	Category of the campaign, such as Technology, Art
			Team Members Num	Number of team members
			Location	Such as city, country of the creator
		Social Media	Display Form	Whether the campaign use some form (such as video, image except text) to display
			Social Exposure	Whether the campaign use some social form (such as Facebook, Twitter) to exposure
			Verification	Whether the creator was verified in Facebook
			Avg Verified Num	Average number of team members verified in Facebook
			Facebook Friends Num	The Facebook friends number of the creator
			Avg Facebook Friends Num	Average Facebooks friends number of the team members
		Mutual Record	Created Num	Number of the campaigns created by the creator before
			Avg Goal	Average claimed funding amount of the created campaigns by the creator
			Avg Funded Amount	Average funded amount of the created campaign by the creator
			Baked Campaign Num	Number of campaigns the creator/team members invested
			Avg Comment Num	Average number of campaigns the creator/team members commented
			Avg Reply Num	Average number of campaigns the creator replied
	Textual	Campaign Profile	Title	Title of the campaign
			Description	Detailed description of the campaign in text

perks, on average 7 perks in one campaign) and their funding information from July 2011 to May 2016 with 217,156 investors, 1,862,097 investment records. Here, we remove the unfinished campaigns because their investment volume still changes.

From this dataset, we extract 23 features from the campaign level and 9 features from the perk level, the details of them are illustrated in Table 1.

2.2.1 Campaign Features

The campaign features mainly include campaign profile, social media, and mutual record. The features of campaign profile and social media are extracted from campaign entities, creator and team members. The features of mutual records, e.g., *Avg Funded Amount, Avg Comment Num*, are statistical result mainly extracted from historical campaign entities funded or created by the creator or team member.

We can see the features are very heterogeneous, including numerical ones (e.g., goal, facebook friends num), categorical ones (e.g., category, location) and text (e.g., campaign description, title). For data preprocessing, we construct the features as numerical vectors. Specifically, we transform categorical data with less than 10 dimensions into N binary-valued features (numerical ones) using **one-hot encoding** [7] (i.e. dummy feature). For example, we convert *Owner Type* into a 3-dimensional binary vector, in which only the value in the corresponding category is set to one and the other values are set to zeros. As for the categorical data with more than 10 dimensions, such as *Category, Location including city and country*, we adopt the **count encoding** method which replaces the variables by the respective count frequencies of the variables in the dataset. Meanwhile, the **doc2vec** method [8] is adopted to convert textual data into numerical vectors, e.g., campaign title and campaign description are turned into 5 and 100 dimension vectors, respectively.

2.2.2 Perk Features

The perk features contain perk profile and perk summary. To know the relationships of different perks under one campaign, we extract *Avg Perk Price, Var Perk Price* in perk summary. We preprocess all the perk features in the same way as introduced for campaign features, except the perk description, which is represented by a 10 dimension vector.

3 Problem and Method Overview

In this section, we first detail the problem of product supply optimization in competitive crowdfunding, then give an outline of the way for solving this problem. For better illustration, Table 2 lists some mathematical notations. The input and output variables are distinguished by the transverse line, where the upper variables are given (input variables), and the rest variables need to be learned (output variables).

3.1 Problem Formulation

In crowdfunding, what the creators are concerned most is the success of their campaigns. In this paper, we aim to assist these creators to enhance the performance of their newly proposed campaigns by optimizing the product supply of each perk. Specifically, the problem can be defined as:

Given the entire budget B_i from the creator to campaign i, the perk settings (including the class number of perks n_i in this campaign, the prices $P_i = \{p_{i1}, p_{i2}, ..., p_{in_i}\}$,

TABLE 2: Several important mathematical notations.

Notations	Type	Description
M	number	the number of campaigns in the market
n_i	number	the number of perks in campaign i
N	number	the number of all perks in the market
L	number	the number of learning tasks
P_i	vector	p_{ij} is the price of the j-th perk in campaign i
S_i	vector	s_{ij} is the given product supply of the perk
E_i	vector	e_{ij} is the reward of the j-th perk in campaign i
C_i	vector	c_{ij} is the number of investments under the setting s_{ij}
h_i	vector	h_{ij} is 1 or the price of the j-th perk in campaign i
C_t	vector	each entry is the label (e.g. c_{ij}) of one perk
X_t	matrix	each row stores the feature vector of one perk
C'_i	vector	c'_{ij} is the number of investments under the setting s'_{ij}
C'_t	vector	the estimated number of investments (e.g. c'_{ij}) of one perk
S'_i	vector	s'_{ij} is the optimized product supply
W_t	matrix	contains the importance of each feature to every task
μ_t	number	the mean of t-th Gaussian Distribution
σ_t^2	number	the variance of t-th Gaussian Distribution
Z_j	vector	z_{jt} is the cluster indicator variable of perk j belongs to task t

the claimed product numbers $S_i = \{s_{i1}, s_{i2}, ..., s_{in_i}\}$ and the rewards $E_i = \{e_{i1}, e_{i2}, ..., e_{in_i}\}$ of the perks), and some other features (like campaign description), our goal is to learn the optimal product supply $S'_i = \{s'_{i1}, s'_{i2}, ..., s'_{in_i}\}$ of the perks, which can bring the maximum expected return for the target campaign.

$$\max_{S'_i} Return_i, \quad s.t. \quad B'_i \leq B_i, \tag{1}$$

where B'_i is the cost of campaign i under the product supply S'_i, and it can be measured by $B'_i \approx \sum_{j=1}^{n_i} s'_{ij} e_{ij}$. Similarly, the claimed budget B_i is $B_i \approx \sum_{j=1}^{n_i} s_{ij} e_{ij}$. The constraint condition is that the cost B'_i after optimizing could not exceed the expect cost (i.e., B_i) of the creator. In this paper, p_{ij} in P_i is the price for the investors to contribute on one product in perk j (e.g. p_{i1} and p_{i2} for two perks in Figure 1 are \$39 and \$69, respectively), and e_{ij} in E_i is the reward that the creator should provide to the investors (e.g. e_{i1} and e_{i2} for the two perks in Figure 1 are 1x Plux and 2x Plux, respectively) after the campaign success. For simplicity, we assume that the creator has enough number of rewards e_{ij} for each product supply s'_{ij}, as long as the budget constraint is satisfied.

Now let us use an example for depicting the research problem intuitively. We suppose the campaign in Figure 1 contains only 2 (n_i) perks whose price $P_i = \{\$39, \$69\}$ and the product numbers preseted by the creator are $S_i = \{200, 200\}$, respectively. A few days before the end of funding, we observed the investment numbers were $\{197, 57\}$. By tracing the campaign investments, we think the investment number in perk 1 will exceed the preseted number 200 in the future, and the preseted product number in perk 2 is too much which causes the wasting of budgets. According to our proposed model, it is believed that changing S_i to a more suitable number setting S'_i (e.g. $\{250,100\}$) is helpful, which helps to achieve a more potential return (shown in Eq. (1)) under the limited budget.

3.2 Method Overview

To maximize the expected return for fresh campaigns in test set by Eq. (1), we first need to use training data to learn connection between the expected return and the features including supply number S', which also means to learn connection between supply number S' and investment volume C' (since return is determined by C') which shown in Eq. (2). After above training, we can get optimal S' for fresh campaigns in test set based on this connection (e.g. Eq. (1)). Meanwhile, because the real investment volume C in train set is known, inspired by the Modern Portfolio Theory, we adopt the difference between C' and C as a penalty factor for balancing the *Risk* and *Return* of creators in training step, which for campaign i is shown as:

$$\max_{\pi, S'_i} \rho_i Return_i - Risk_i, \quad s.t. \quad B'_i \leq B_i, \tag{2}$$

where ρ_i is a heuristic parameter, π is the learned connection between C' and S', and we will use other specific notations to replace π later. Here $Return_i$ and $Risk_i$ functions will be introduced detailedly in Section 4.

Based on above main idea, we propose a two-stage solution including offline (testing) stage and online (training) stage. In the offline stage, we learn the deep connection between C' and S' based on historical campaigns, which shown in Eq. (2). Considering the balance between return and risk for each creator, we adopt modern portfolio theory and use an alternative optimization algorithm to learn the parameters.

In the online stage, we optimize the product supply for fresh campaigns under a limited budget. Specifically, we extract their features and maximize the expected return of each campaign for recommending the optimized product supply for each perk in the campaign, which has been shown in Eq. (1).

Depending on the two-stage solution, we firstly propose two models based on two kinds of splitting task methods. We call the product supply optimization based on the manual task partition method as **PSO** model, and the method based on automatic task partition as **APSO** model, which will be introduced detailedly in Section 4.

What's more, we study the investment prediction problem with multiple competitive perks in a campaign, and the model is termed as APSO with *Competition* (**APSO-C**) which will be introduced detailedly in Section 5.

4 (A)PSO: (Automatic) Product Supply Optimization Model

In this section, we first illuminate our entire set of framework about modern portfolio theory and multi-task learning method with two ways of task splitting in Section 4.1 and 4.2, respectively, and then give a description of the two stages which including the offline stage of investment volume prediction (in Section 4.3) and online stage of product supply optimization (in Section 4.4).

4.1 General Framework

In the offline stage, we build the general modern portfolio theory framework which includes return and risk functions to satisfy creators' motivations, and these two functions are all based on the predicted investment volume. Specifically, because of the relationships between perks, we adopt multi-task learning method to estimate the unknown investment volume of each perk. If we have already known the task

each perk belonging to, then the estimated investment volume of j-th perk from campaign i is formulated as c'_{ij}:

$$c'_{ij} = x_{ij} W_{t_{ij}}^{\top}, \tag{3}$$

here x_{ij} is the features including campaign i's features and perk j's, the weight parameter $W_{t_{ij}}$ is the learned connection like π in Eq. (2) from task t which perk j of campaign i belongs to. How to construct x_{ij} and learn $W_{t_{ij}}$ will be introduced later.

Based on this, we will introduce the return and risk functions detailedly, and explain how to predict the investment volume by multi-task learning method in this subsection.

Return Function. According to the findings in previous studies, the motivations/goals for creators launching campaigns are mainly classified into two categories: raising enough money and expanding awareness of the products (boosting their brands) [1], [9]. Thus, the expected return of campaign i can be measured by:

$$Return_i = \sum_{j=1}^{n_i} c'_{ij} h_{ij}, \quad h_{ij} \in \{p_{ij}, 1\}, \tag{4}$$

where c'_{ij} in C'_i ($C'_i = \{c'_{i1}, c'_{i2}, ..., c'_{in_i}\}$) is the number of investments of j-th perk in campaign i under the product supply setting S'_i, which is shown in Eq. (3). h_{ij} measures the motivation of the campaign, i.e., $h_{ij} = 1$ if the creator mainly try to influence more people and $h_{ij} = p_{ij}$ if she aims to collect more money. We should also note that, there may be more sophisticated choices of h_{ij}, e.g. by balancing between p_{ij} and 1. However, this is not the major focus of this paper, and we will leave it for future study.

Risk Function. Besides the budgets from creators, the product supply of each perk is also constrained by the number of potential investors in the market. For instance, if the number of investments c_{ij} is much smaller than the estimation c'_{ij}, the product rewards are overstocked by the creator, and vice versa. Indeed, since the number of investors cannot be perfectly estimated, the future return of a new campaign is actually unknown and Eq. (4) is computed with uncertainty/risk. In finance, risk is usually measured by *(co)variance* of the investments (e.g. stocks) [5] [6]. Correspondingly, in crowdfunding, the risk of setting product supply as S_i for campaign i can be estimated by the variance of the returns:

$$Risk_i \approx \sum_{j=1}^{n_i} (c'_{ij} h_{ij} - c_{ij} h_{ij})^2. \tag{5}$$

Modern Portfolio Theory. Therefore, summarizing Eq. (2) for all the campaigns in modern portfolio theory:

$$\begin{aligned} &\max_{W, S'_i} \sum_{i=1}^{M} \rho_i Return_i - \sum_{i=1}^{M} Risk_i, \\ &s.t. \quad B'_i \le B_i, \quad \forall i \in [1, M], \end{aligned} \tag{6}$$

where M is the number of campaigns, the learned parameter W is like π in Eq. (2). Obviously, as a portfolio of investment choices in the platform, every campaign (perk) is related to others. Thus, instead of computing independently, the risk in Eq. (6) should be measured more precisely by exploiting this kind of relevance.

Multi-task Learning (MTL). In this paper, considering the number of investments of a perk is related to other perks even other campaigns, we propose such a measurement by the trace-norm multi-task learning method. Specifically, instead of a formal definition on the correlation/relevance among different campaigns, we automatically learn the risk of the portfolio from a data-driven way to replace Eq. (5):

$$\sum_{i=1}^{M} Risk_i = \sum_{t=1}^{L} ||(X_t W_t^{\top} - C_t) \times h_t||_F^2 + \lambda ||W||_*, \tag{7}$$

where $\times$ denotes the cross product of two vectors. All perks in the campaigns are now clustered into L learning tasks based on their characteristics. For the t-th task, the input comprises (X_t, C_t, h_t), where $X_t \in \mathbb{R}^{m_t \times d}$ is the input matrix for the t-th task with m_t perks and d features, i.e. X_{ij} is the feature vector of the j-th perk in campaign i. While label $C_t \in \mathbb{R}^{m_t \times 1}$ is the corresponding target vector (i.e. entries from C, under the product supply setting S), and similarly, vector h_t contains the motivations of these perks. $W = [W_1, W_2, ..., W_L]$ is a $d \times L$ weight matrix, containing the importance/coefficient of each feature to one task. Obviously, the estimated value in t-th task is $C'_t = X_t W_t^{\top}$, and the estimated number of investments in j-th perk of campaign i is shown in Eq. (3). Different choices of regularization terms may reflect different task relationships [10]. Without loss of generality, in this study, we formulate our model by trace norm, which is given by the sum of the singular values: $||W||_* = \sum_t (\sigma_t(W))$, for capturing the task relationship by constraining the parameter vectors of different tasks to share a low dimensional subspace.

4.2 Task Partition

Before using the MTL method, we should first put the perks into L different learning tasks to get X_t and C_t. In this subsection, we introduce two task splitting methods, one is a manual way, and the other is an automatic way.

4.2.1 Manual Task Partition

Indeed, the price is the direct reflection of two different products. Considering that, products with near prices are more related than distant ones [11], we propose to split perks into different tasks according to their prices, i.e. the perks with the similar prices will be put into the same task. Without loss of generality, on the basis of perks' distribution on different prices, we generate 7 tasks ($L = 7$) whose price ranges (in \$) (from task $T1$ to task $T7$) are $(0, 10], (10, 20], (20, 30], (30, 40], (40, 50], (50, 200]$, and $(200, +\infty)$, respectively. In order to better illustrate, we call the MTL with manual task partition method as **PSO**.

4.2.2 Automatic Task Partition

Up to now, we split the task by perk prices, which is a manual way and maybe not so reasonable. Therefore, we propose a strategy for splitting task automatically [12]. The main idea of splitting task is equal to cluster perks whose feature space are similar into the same task. To distinguish this method and MTL with manual task partition method (PSO), we named automatic one as **APSO**.

Inspired by [13], [14], we propose to use EM-process to split tasks. The main idea of EM-processes is dividing objects with similar feature space into the same clusters. Now, we introduce how we use it. Besides the observable variables X, we also assume there are some unobservable cluster indicator variables, such as $Z \in \mathbb{R}^{N\times L}$, where N is the number of all perks (here we put all perks from campaigns together for splitting tasks), and each row of Z represents a perk, for j-th perk, it can be abbreviated as $Z_j = (z_{j1}, z_{j2}, ..., z_{jL}), j = 1, 2, ..., N, t = 1, 2, ..., L$. Specifically, $z_{jt} \in \{0, 1\}$, where $z_{jt} = 1$ means the j-th perk belongs to the t-th task, and vice versa. Due to the complexity of splitting task, true Z value is unknown, we assume that the error of estimated investment volumes follows Gaussian Distribution in each task, which means $c_j \sim \mathcal{N}(\hat{c}_j + \mu_t, \sigma_t^2)$, the formulation is given as:

$$\Phi(c_j|\theta_t) = \frac{1}{\sqrt{2\pi}\sigma_t} exp[-\frac{(c_j - x_j W_t^\top - \mu_t)^2}{2\sigma_t^2}], \tag{8}$$

here $\theta_t = (W_t, \mu_t, \sigma_t)$, $\hat{c}_j = x_j W_t^\top$ and c_j are the estimated investment volume and true investment volume respectively. Then, the probability density is the form of Gaussian mixture model shown as:

$$P(c_j|\theta) = \sum_{t=1}^{L} \alpha_t \Phi(c_j|\theta_t), \tag{9}$$

where $\alpha_t \in (0, 1)$ is the *mixing weight* ($\sum_{t=1}^{L} \alpha_t = 1, t = 1, 2, ..., L$), θ represents the collection of parameters $(\alpha_1, ..., \alpha_L, \theta_1, ..., \theta_L)$.

By combining the cluster indicator parameters, we will get the likelihood function:

$$\begin{aligned} P(c, Z|\theta) &= \prod_{j=1}^{N} P(c_j, Z|\theta) \\ &= \prod_{t=1}^{L}\prod_{j=1}^{N} [\alpha_t \Phi(c_j|\theta_t)]^{z_{jt}} \\ &= \prod_{t=1}^{L} \alpha_t^{\sum_{j=1}^{N} z_{jt}} \prod_{j=1}^{N} \Phi(c_j|\theta_t)^{z_{jt}}. \end{aligned} \tag{10}$$

In our case, we use EM algorithm for task splitting (perk clustering), so we add another step (C-step). Specifically, Z is unobservable, so we estimate Z as a decimal between 0 and 1 in E-step. Then, in C-step, we cluster each perk depends on their Z in different tasks. Finally, we minimize the loss function to learn θ in M-step.

The first step in applying the EM algorithm to the problem at hand is to initialize mean vectors μ, covariance vectors σ and weight matrixes W to represent each of the L tasks. Without loss of generality, we initialize each element in the start mean as 0, variance and weight matrix as 1, besides, we denote $\mathbf{A} = (A_1, A_2, ..., A_L)$ as split tasks, and A_i represents i-th task containing perks belong to it. Starting from an initial partition $\mathbf{A}^{(0)}$, the m-th iteration ($m > 0$) is defined as follows:

E-step. The E-step computes the expected value of the complete log-posterior which is usually called the Q-function, we have:

$$\begin{aligned} Q(\theta, \theta^{(m)}) &= E[\log P(c, Z|\theta)|c, \theta^{(m)}] \\ &= \sum_{t=1}^{L}\sum_{j=1}^{N} \hat{z}_{jt}^{(m)} \log \alpha_t + \sum_{j=1}^{N} \hat{z}_{jt}^{(m)} [\log(\frac{1}{\sqrt{2\pi}}) \\ &\quad - \log \sigma_t - \frac{1}{2\sigma_t^2}(c_j - \hat{c}_j - \mu_t)^2]. \end{aligned} \tag{11}$$

To complement this function, we need to compute the current posterior probabilities $\hat{z}_{jt}^{(m)}$ for $j = 1, 2, .., N$ and $t = 1, 2, ..., L$ given as follows, and the detailed derivation is given in Appendix A.

$$\hat{z}_{jt}^{(m)} = \mathbb{E}(z_{jt}^{(m)}|c, \theta) = \frac{\alpha_t^{(m)} \Phi(c_j|\theta_t^{(m)})}{\sum_{t=1}^{L} \alpha_t^{(m)} \Phi(c_j|\theta_t^{(m)})}. \tag{12}$$

Something should be noticed that the computation of $\hat{z}_{jt}$ is relevant to c_j, so as for testing dataset, we treat the mean of all perks' c_j which from task t as the value of c_j here.

C-step. This step is for classifying, we assign each perk to the task which provides the maximum posterior probability $\hat{z}_{jt}^{(m)}, t = 1, 2, ..., L$, (if the maximum posterior probability is not unique, we choose the task with the smallest index). Let $\mathbf{A}^{(m)}$ denote the resulting partition. For j-th perk:

$$t' = \arg\max_{t=1}^{L} \hat{z}_{jt}^{(m)}, A_{t'}^{(m)} = A_{t'}^{(m)} \cup \{j\}. \tag{13}$$

M-step. After splitting the tasks, the estimates of the relevant parameters $\theta^{(m+1)}$ are updated by maximizing the Q-function. For $t = 1, 2, ..., L$, we update them as:

$$\begin{aligned} \alpha_t^{(m+1)} &= \frac{\sum_{j=1}^{N} \hat{z}_{jt}}{N}, \\ \mu_t^{(m+1)} &= \frac{\sum_{j=1}^{N} \hat{z}_{jt}(c_j - \hat{c}_j)}{\sum_{j=1}^{N} \hat{z}_{jt}}, \\ \sigma_t^{2^{(m+1)}} &= \frac{\sum_{j=1}^{N} \hat{z}_{jt}(c_j - \hat{c}_j - \mu_t)^2}{\sum_{j=1}^{N} \hat{z}_{jt}}, \\ w_t^{(m+1)} &= \frac{\sum_{j=1}^{N} \hat{z}_{jt}(c_j - \mu_t^{(m)}) * x_j}{\sum_{j=1}^{N} \hat{z}_{jt} * x_j x_j^\top}. \end{aligned} \tag{14}$$

The above steps are repeated until a suitable stopping criterion is reached. In our experiments, we stop if the evaluation metrics (i.e., RMSE or nMSE) increases by less than some small value (i.e., $1.0e^{-2}$ or $1.0e^{-6}$) from one iteration to the next. We repeat the EM-process ten times, then take the average of ten results as our final results, the split task $\mathbf{A}$ is returned for subsequent computation.

4.3 The Investment Volume Prediction

So far, we already know the whole framework and task partition way, so we can rewrite the $Return_i$ in Eq. (6) and combine it with $Risk_i$ in Eq. (7). Thus, the optimization problem becomes:

$$\begin{aligned} \max_{W,S'} \quad & \sum_{i=1}^{M}\sum_{j=1}^{n_i} \rho_{ij}[s'_{ij} w_{t_{ij}}^{s'} + x_{ij}^{-s'}(W_{t_{ij}}^{-s'})^\top] h_{ij} \\ & - (\sum_{t=1}^{L} ||(X_t W_t^\top - C_t) \times h_t||_F^2 + \lambda ||W||_*), \\ s.t. \quad & s'_{ij} \geq 0 \quad and \quad B'_i \leq B_i, \quad \forall i \in [1, M]. \end{aligned} \tag{15}$$

Here, x_{ij} is represented by $x_{ij} =< s'_{ij}, x_{ij}^{-s'} >$, where s'_{ij} is the optimized product supply and $x_{ij}^{-s'}$ stores other features of this perk except s'_{ij}. Similarly, matrix W_t is also split into two parts, $W_t^{s'}$ and $W_t^{-s'}$. Note that index t_{ij} indicates the j-th perk in campaign i belongs to the t-th task.

In APSO, we adopt EM-process to split task, it is important to note that we already got a matrix W in M-step, for achieving a more accurate result quickly, we use the W come from EM-process (Section 4.2) to initialize the W in Eq. (15).

4.4 Product Supply Optimization

In summary, given the campaigns/perks and their features X, for solving the product supply optimization problem on testing campaigns, we first put these perks into different learning tasks, and then learn the W and S' based on the training ones by solving Eq. (15). Thus, the optimal S'_i for each testing campaign can be computed by only maximizing the expected return:

$$\max_{S'_i} \sum_{j=1}^{n_i} \rho_{ij}[s'_{ij} w_{t_{ij}}^{s'} + x_{ij}^{-s'}(W_{t_{ij}}^{-s'})^\top] h_{ij}, \quad s.t. \quad s'_{ij} \geq 0, \quad B'_i \leq B_i. \tag{16}$$

Please note the difference between Eq. (15) and (16), i.e., for the new campaign in testing, the risk is unknown without any investment records. Finally, S'_i (i.e. the output of Eq. (16)) is recommended to the creator when she is publishing this campaign. Now, we show the way of solving Eq. (15).

Algorithm 1 Alternating Optimization Method

Require: $X, C, S, B, N, tol_W, tol_{S'}$
Ensure: W, S'
1: set $S' = S, W_0 = \mathbf{0}$
2: **for** $k = 1$ to N **do**
3: Update W based on Eq. (17),
4: Update S' based on Eq. (18),
5: Update X by replacing S'.
6: **if** stopping criteria is satisfied **then**
7: Break
8: **end if**
9: **end for**
10: **return** W, S'

4.5 Optimization Algorithm

Here, we propose to solve Eq. (15) by the Alternating Optimization Method, which is similar to the Block Coordinate Descent method [15], where the variable is optimized alternatively with the other variables fixed. Because Eq. (15) is continuous and separately convex, the alternating optimization algorithm is convergent. Besides, the main reason to choose this algorithm is that the investment volumes and product supply number change in real time. Please refer to Algorithm 1 for the holistic method, and the detailed gradients of W and S' are shown as (Here, we simply fix $h_{ij} = 1$ for better illustration):

$$\nabla f(W_t) = (W_t X_t^\top X_t - 2C_t^\top X_t) - \sum_{i=1}^{M}\sum_{j=1}^{n_i} \rho_{ij} x_{ij}, \tag{17}$$

$$\nabla f(s'_{ij}) = 2W_{t_{ij}}^{s'} s'_{ij} W_{t_{ij}}^{s'} + 2(x_{ij}^{-s'} W_{t_{ij}}^{-s'} - C_{ij}) W_{t_{ij}}^{s'} - \rho_{ij} W_{t_{ij}}^{s'}. \tag{18}$$

In Algorithm 1, B is each campaign's budget and S is the product supply claimed by creators. Since each s'_{ij} is a part (an entry) of the feature matrix X, X should be updated whenever s'_{ij} changes. We should also note that we update W for each task, and update S' for each perk. In each iteration step, we adopt Accelerated Gradient (AG) method to update W and S' for achieving the optimal rate of convergence by specifying the step-size policy, and we use projection method to satisfy the constraints.

During the implement, we first update W for each task. For task t which contains n_t perks, each perk has d dimensional features, so the computational complexity is $O(n_t^2 d^2)$. Then we update S' and X, their computational complexity can be regarded as $O(1)$ compared to the complexity in updating W, so the overall computational complexity is $O(n_k^2 d^2 N)$, where k is the index of the task that has the biggest n_k^2. That is, the algorithm should be stopped when the changing of W or S' is less than a threshold (i.e. tol_W or $tol_{S'}$) or the iteration reaches a maximum number N. In practice, we set tol_W ($tol_{S'}$) as $1.0e^{-5}$, and set N as $1.0e^5$, which we think is of high-quality enough.

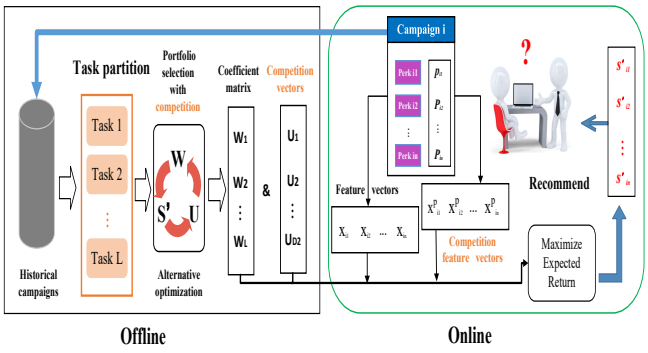
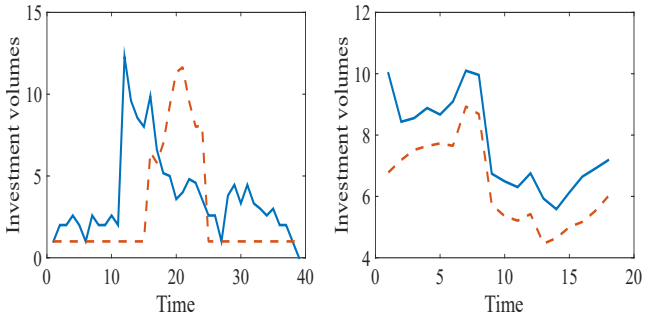


Fig. 2: The flowchart of automatic product supply optimization with competition (APSO-C).

5 APSO-C: Automatic Product Supply Optimization with Competition

Actually, like e-commerce market, crowdfunding platform is also a special competitive market, and the competition exists among not only different campaigns but also different perks in one campaign. It is natural for a backer vacillates in different perks from one campaign. As we can see from Figure 1, there exists funding competition between Perk 1 with price $39 to get one Plux and Perk 2 with price $69 to get double Plux. Because of the discount price in Perk 2, many backers would waver between them.

In a competitive market with multiple products, no matter is the PSO or APSO method, they both tackle the product supply optimization problem by dividing it into a set of single product supply optimization problem. However, it fails to consider the competition among different perks which from the same campaign. In this section, we



(a) Short-term competition (b) Long-term competition

Fig. 3: Scaled normalized investment volume time-series depicting competition between two similar perks, one is the leading perk (solid blue), the other one is the competitor (dashed orange).

address the problem of how to incorporate inner (perk) competition process into the proposed APSO model, and jointly learn coefficient matrix W, competition parameter U and optimized product supply S'. The entire flowchart of the APSO-C framework is illustrated in Figure 2.

5.1 Inner Competition in Crowdfunding

It is evident that competition is one of the most interesting elements of online marketing, where are various products competing to seize the market. Some researchers found that similar products (same product type but different brands) compete with each other in E-commerce marketing [16]. This phenomenon is consistent with our common sense.

Similarly, in the special market, competition is also common in crowdfunding platform. According to the survey of crowdfunding platform, there are some competitive perks with typical perk prices attract more contributors[4]. In our opinion, the perk competition not just depends on their prices, their rewards and other factors would also be considered from backers' perspective.

Based on this assumption, we profile some similar perk pairs where we see potential competition, and observe their funding amount in the same absolute time (even though one of them could have its funding started earlier than the other). We normalize the investment number of two competitive perks and show two representative types of competition in Figure 3. The first type of competition is referred as *short-term competition* (Figure 3(a)) where one perk (*competitor*) took over the other perk (*leader*) in investment volumes during a certain period, and then the leader regain the market share. The second type of competition is named *long-term competition* (Figure 3(b)) where the leader leading the market consistently, the competitor has no ability to seize the market. In this paper, we focus on the competition representation and do not distinguish their types, this figure is just to prove there exists competition.

5.2 Inner Competition Expression

As mentioned before, we predict the number of investment of j-th perk in campaign i shown in Eq. (3). Because of the existing of inner competition, the competitive perk may seize the market (investment volume) from the other one. Intuitively, we adopt a transition matrix $V \in \mathbb{R}^{n_i \times n_i}$ to model the competition for campaign i which contains n_i perks, with each element v_{jq} denotes the transferred investment volume from perk j to perk q. As we discovered, the more similar of two perks in the same campaign, the more competition between them. Based on this, we formulate their transferred investment volume under competition as:

$$v_{jq} = U * Sim(x_j^P, x_q^P). \tag{19}$$

Specifically, $U \in \mathbb{R}^{1 \times D2}$ is the weight parameter, here we focus on the inner competition, so x_j^P is feature vector of perk j only contains perk level features not include campaign level features, and $D2$ is the dimension of perk level features. Then, we define the similarity from perk j to perk q with direction as:

$$\begin{aligned} Sim(x_j^P, x_q^P) &= K(x_j^P - x_q^P) * I(x_j^P - x_q^P), \\ I(r) &= \begin{cases} 1, & r \geq 0; \\ -1, & r < 0. \end{cases} \end{aligned} \tag{20}$$

Here, $x_j^P - x_q^P$ is a directed distance from perk j to perk q, for computing the similarity between them reasonably, we use a kernel function $K(.)$ which is specified as Gaussian for satisfying the similarity measurement.

$$K(x_j^P - x_q^P) = \frac{1}{\sqrt{2\pi}\sigma} exp[-\frac{(x_j^P - x_q^P - \mu)^2}{2\sigma^2}], \tag{21}$$

where we set $\mu = 0, \sigma = 1$ in our experiments. Since the transfer quantity between two perks is opposite, we bring in a index function $I(.)$ for representing the distance direction between each two perks, r in $I(r)$ is each value in $x_j^P - x_q^P$. Obviously, $Sim(x_j^P, x_q^P) = -Sim(x_q^P, x_j^P)$ and $v_{jq} = -v_{qj}$. For example, there are three perks in a campaign which shown in bottom right corner of Figure 4, perk j is more competitive than perk q, but less competitive than perk u . In this case, the transfer quantity v_{jq} from perk j to perk q is negative, which means a part of investment volume transfer from perk q to perk j. Similarly, the value v_{ju} is positive because of perk u is more competitive than perk j. This figure also shows that each value in matrix V is computed by vector U and Sim function, as like Eq. (19).

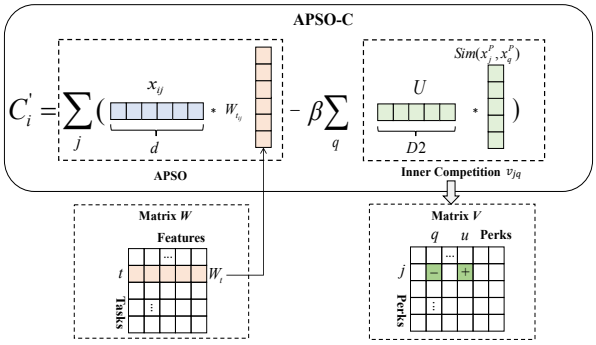


Fig. 4: An illustration of the APSO-C model, which incorporates the inner competition into the APSO model.

4. https://go.indiegogo.com/blog/2012/07/indiegogo-insight-perk-pricing-practices.html

5.3 The APSO-C Model

In this subsection, we give a description of new Return and Risk function and the changes on the original two steps on account of the inner competition factor. It should be noted that, since the APSO-C model is an extension of APSO, the task splitting part in APSO-C also adopts the automatic one which based on EM-process.

5.3.1 The Investment Volume Prediction in APSO-C model

As shown in Figure 4, by combining the investment prediction function (Eq. (3)) in APSO model (on the left) and inner competition function (Eq. (19), on the right), we predict the investment of a campaign with competition as (without loss of generality, here we just focus on the motivation about the number of investors which means $h_{ij} = 1$ for each perk of any campaign):

$$\begin{aligned} C'_i &= \sum_{j=1}^{n_i}(c'_{ij} - \beta \sum_{q=1}^{n_i} v_{jq}) \\ &= \sum_{j=1}^{n_i}(x_{ij} * W_{t_{ij}}^{\top} - \beta \sum_{q=1}^{n_i} U * Sim(x_j^P, x_q^P)), \end{aligned} \tag{22}$$

here we subtract the transferred volume from original predicted investment volume under the inner competition environment, and then we have each perk's investment volume, so the investment volume of campaign i is the sum of its perks'. Like we introduced in Section 4.1 (Eq. (4), Eq. (5)), the new return and risk function become:

$$\begin{aligned} Return_i &= C'_i, \\ Risk_i &= (C'_i - C_i)^2. \end{aligned} \tag{23}$$

Finally, the objective function is also to maximize the return and minimize the risk, so we transfer Eq. (15) and jointly learn the coefficient matrix W, competition parameter U and optimized product supply S' as:

$$\begin{aligned} \max_{W,S',U} & \sum_{i=1}^{M}\sum_{j=1}^{n_i} \rho_{ij}([s'_{ij} w_{t_{ij}}^{s'} + x_{ij}^{-s'}(W_{t_{ij}}^{-s'})^{\top}] - \beta \sum_{q=1}^{n_i} U * Sim(x_j^P, x_q^P)) \\ & - \sum_{i=1}^{M}(\sum_{j=1}^{n_i}[s'_{ij} w_{t_{ij}}^{s'} + x_{ij}^{-s'}(W_{t_{ij}}^{-s'})^{\top}] - \beta \sum_{q=1}^{n_i} U * Sim(x_j^P, x_q^P) \\ & - C_i)^2 - ||W||_*, \\ & s.t. \quad \forall i \in [1, M], \quad s'_{ij} \geq 0, \quad B'_i \leq B_i. \end{aligned} \tag{24}$$

Here, x_{ij} is represented by x_{ij}=< s'_{ij}, $x_{ij}^{-s'}$ >, where s'_{ij} is the optimized product supply and $x_{ij}^{-s'}$ stores other features of this perk except s'_{ij}. Similarly, matrix W_t is also split into two parts, $W_t^{s'}$ and $W_t^{-s'}$. Note that index t_{ij} indicates the j-th perk in campaign i belongs to the t-th task. The combination of W and U likes π in Eq. (2), which learn the connection between return and risk.

5.3.2 Product Supply Optimization in APSO-C Model

Now we have learned the coefficient matrix W and competition parameter U, so we transfer Eq. (16) and optimize product supply for fresh campaign i in the inner competition relationship as:

$$\begin{aligned} \max_{S'_i} & \sum_{j=1}^{n_i}[s'_{ij} w_{t_{ij}}^{s'} + x_{ij}^{-s'}(W_{t_{ij}}^{-s'})^{\top}] - \beta \sum_{q=1}^{n_i} U * Sim(x_j^P, x_q^P) \\ & s.t. \quad s'_{ij} \geq 0, \quad B'_i \leq B_i. \end{aligned} \tag{25}$$

Generally speaking, in APSO-C model, we learn the W, U based on Eq. (24) and optimize the product supply for fresh campaign based on Eq. (25). The whole process of APSO-C model are summarized in Algorithm 2, and the gradient of W, U and S' are given in Appendix B.

During the implement, we first update W for each task. For task t which contains n_t perks from all campaigns, each perk has d dimensional features, so the computational complexity is $O(n_t^2 d^2)$. Then we update U and S', their computational complexity can be regarded as $O(1)$ compared to the complexity in updating W, so the overall computational complexity is $O(n_k^2 d^2 N)$, where k is the index of the task that has the biggest n_k^2. That is, the algorithm should be stopped when the changing of W, U or S' is less than a threshold (i.e. tol_W, tol_U, $tol_{S'}$) or the iteration reaches a maximum number N. In practice, we set tol_W ($tol_{S'}$) as $1.0e^{-5}$, set tol_U as $1.0e^{-3}$ and set N as $1.0e^5$, which we think is of high-quality enough.

Algorithm 2 The Procedure of APSO-C

Require: the split task with all features X, only perk features X^P, real investment volume C, and budgets B
Ensure: The product supply S'_i for each fresh campaign i
1: **for** $k = 1$ to N **do**
2: Fix U and S', update the coefficient matrix W (Eq. (1) in Appendix B),
3: Fix W and S', update the competition parameter U (Eq. (2) in Appendix B),
4: update product supply in training data S' (Eq. (3) in Appendix B).
5: **if** stopping criteria is satisfied **then**
6: Break
7: **end if**
8: **end for**
9: compute the product supply S'_i in testing data according to Eq. (25).
10: **return** S'_i

6 Experiments

In this section, we construct comprehensive experiments on a real-world dataset that we crawled from one famous crowdfunding platform in America, i.e. Indiegogo.com, please refer to Section 2.2 for the detailed information of experimental data. Specifically, we will demonstrate that:

1) the performance of investment volume prediction, including PSO, APSO and APSO-C models,
2) the understanding of different feature integrations,
3) the correlation between split tasks based on manual method and automatic method, respectively,
4) the performance of product supply optimization.

6.1 Experimental Setup

To observe how each algorithm behaves at different sparsity levels of a dataset, we construct different sizes of training sets from 50% to 80% of the campaigns with the increasing step at 10%, and we name the four pairs of training and

TABLE 3: Investment volume prediction performance.

Methods/Metrics		Ridge	Lasso	SVR	SigmoidSVR	MTL_1	MTL_2	PSO	APSO	**APSO-C**
nMSE	D#1	1.0054	0.9991	1.0183	1.0204	0.9950	1.0042	1.0109	0.9775	**0.9741**
	D#2	1.0274	1.0199	1.0123	1.0136	1.0032	0.9927	1.0085	0.9921	**0.8788**
	D#3	1.0560	1.0533	1.0340	1.0366	1.0602	1.0602	1.0301	0.9817	**0.8844**
	D#4	1.0846	1.0631	1.0628	1.0600	1.0557	1.0578	1.0580	1.0580	**1.0022**
RMSE	D#1	164.1025	163.5917	165.1502	165.3253	163.2748	164.0274	164.5794	161.8328	**161.5491**
	D#2	176.2585	175.6107	174.9538	175.0723	174.2000	173.2835	174.6615	173.2345	**160.5738**
	D#3	93.2948	93.1742	92.3154	92.4343	93.4996	93.4996	91.4995	89.9710	**85.3976**
	D#4	73.6763	72.9428	72.7275	72.4496	72.7084	72.7816	72.3907	72.7907	**70.8429**

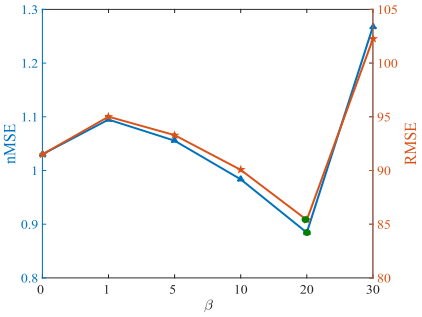


Fig. 5: Investment volume prediction performance on APSO-C

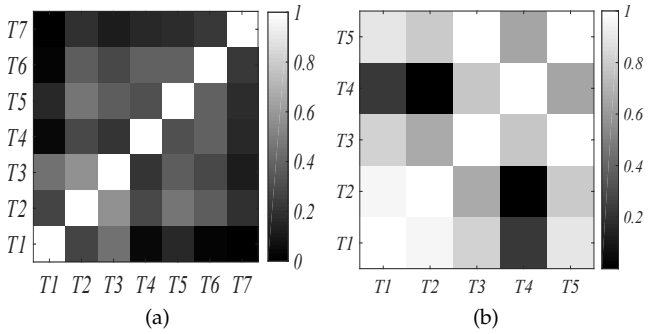


Fig. 6: Correlation about two kinds of split task. (a) Correlation between manual split tasks. (b) Correlation between automatic split tasks.

testing sets as $D\#1$, $D\#2$, $D\#3$ and $D\#4$. The detailed features have been introduced in Section 2.

Parameter Setting. Firstly, we show the way of computing budgets B_i' and B_i. As we can see from Figure 1, the reward e_{ij} in crowdfunding is usually the product or a thanks card of the project, therefore, it is very hard to directly compute B_i' and B_i based on e_{ij}. Luckily, in the literature of marketing, it is usually assumed that the price positively influences the perception of product (reward) quality, that is $e_{ij} \propto p_{ij}$ [17], and the unit cost of the product with a quality e_{ij} is $e_{ij}^2/2$ [3], [18] which can be further represented as $\gamma p_{ij}^2/2$, where γ is a parameter. Therefore, in the following experiments we possibly define $B_i' = \frac{1}{2}\gamma \sum_{j=1}^{n_i} s_{ij}' p_{ij}^2$ and $B_i = \frac{1}{2}\gamma \sum_{j=1}^{n_i} s_{ij} p_{ij}^2$.

Parameter ρ_{ij} represents the risk preferences of each creator, here we simply define ρ_{ij} as a uniform value since it is hard to quantify risk preference for each creator with limited data records. Parameter λ is learned by cross-validation.

In APSO-C model, β controls the effect of competition. As the training records are much larger than the parameters, we set β in a reasonable range (e.g., from 1 to 30) for better knowing the effect of competition. Obviously, if $\beta = 0$ then APSO-C model goes back to APSO model. For conveniently comparing, we compare them in different β values.

6.2 Evaluations on Investment Volume Prediction

Before proving the effectiveness of the entire framework of product supply optimization, we first show the performance of **PSO**, **APSO** and **APSO-C** models on measuring the risk (predicting the future investment volume) of each campaign. As shown in Eq. (15) and Eq. (24), we get learned W from training step, then we can use it to predict investment volume in testing data. We adopt RMSE and nMSE as the metrics since they are widely used in MTL [10], [19]. For the RMSE and nMSE, the smaller the value, the better the performance. We choose several state-of-the-art regressions and two MTL methods which using k-means [20] to task splitting for comparison [10] :

- **Ridge**: Linear regression with L2-norm regularizer [21].
- **Lasso**: Linear regression with L1-norm regularizer [22].
- **Support Vector Regression (SVR)**: training a Support Vector Regression model [23], [24].
- **SVR with sigmoid kernel (SigmoidSVR)**: Support Vector Regression with sigmoid kernel [25].
- **MTL with K-means for task splitting** (**MTL**$_1$,**MTL**$_2$): **MTL**$_1$ uses all features for task splitting and **MTL**$_2$ uses partial features for task splitting.

The experimental results of our methods and the baselines on four data splits are shown in Table 3. Due to space limitation, we only show the results with $h_{ij} = 1$. We can see that the PSO method consistently performs better than the single task models on all splits in terms of two evaluation metrics, which clearly validates the effectiveness of our multi-task learning method. Also, it is a fact that MTL_1 and MTL_2 perform better than PSO upon most occasions, which shows that using more features instead of perk prices for task splitting is more reasonable. However, it is obvious that these two automatic task splitting methods perform worse than APSO method, it proves that using EM-process (shown in Section 4.2.2) to split task is more effective than k-means method and manual task partition metric, and the reason behind good performance is that EM-process can project the perk features into the same projection space.

TABLE 4: Performance on different feature integrations.

Data Split	Feature Integration	nMSE	RMSE
D#1	NP	0.9107	156.2047
	NPC	**0.8837**	**153.8720**
	NTPC	0.9741	161.5491
D#2	NP	0.8853	163.6396
	NPC	0.8909	164.1568
	NTPC	**0.8788**	**160.5738**
D#3	NP	0.9851	90.1279
	NPC	0.9336	88.0607
	NTPC	**0.8844**	**85.3976**
D#4	NP	1.1344	75.3700
	NPC	1.0786	73.4946
	NTPC	**1.0022**	**70.8429**

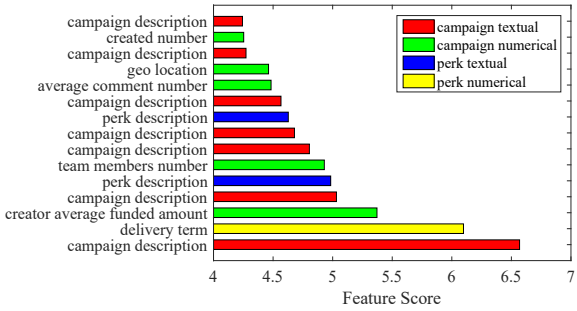


Fig. 7: An illustration of several important features.

Above all, our extended model APSO-C performs the best on all data splits, what demonstrates the effectiveness and shows there really exist inner competition which affects the funding results.

According to Eq. (22), parameter β controls the effect of competition, it is obviously that $\beta = 0$ means the model without competition (i.e., APSO). Here we compare the performance over different β values which is shown in Figure 5. The blue line is the performance under nMSE metric and the orange one is under RMSE over different β values. As can be seen from this figure, our proposed APSO-C model performs the best when $\beta = 20$ (marked as green square) under two metrics. Specifically, among all the range of β values, the result in $\beta = 10$ is also better than APSO result ($\beta = 0$), but opposite in other situations, this phenomenon states that there is a kind of relatively reasonable competition degree in crowdfunding market, and creator should carefully consider that when posting campaigns. Besides, it also proves that backers' choices are really influenced by the inner competition, and our automatic splitting task method is effective.

Until now, we have already compared the results in two splitting task methods (PSO and APSO), now, we analyze the utilities of them. To intuitively illustrate the utilities of PSO and APSO, we compute the Jaccard similarity[5] among the rows (tasks) in W. The results are shown in Figure 6(a) and 6(b). From Figure 6(a) we can see that the tasks with the similar price ranges are usually more similar, of course, task $T1$ and task $T7$ are the most distinctive ones. On the other hand, we can see that the APSO automatically divides all perks into 5 tasks in Figure 6(b), most tasks are similar expect task $T2$ and $T4$ which are the most dissimilar ones. It is clear that the tasks in PSO have weaker relation than in APSO, it shows that the strong ties of tasks are helpful to learn relationships among all perks.

5. The similarity is computed by summarizing the number of features, whose absolute value in two rows of W are both larger than the mean value.

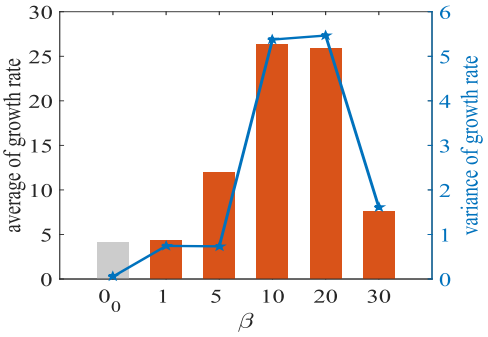


Fig. 8: Product supply performance on APSO-C.

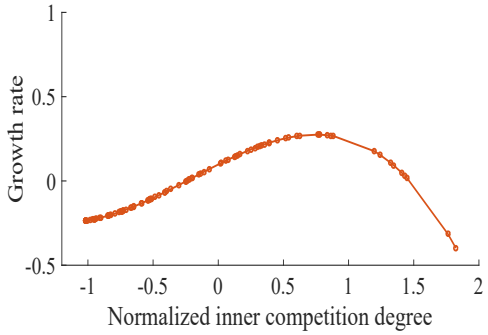


Fig. 9: The impact of inner competition.

6.3 Analysis on Different Feature Integrations

To explore how these features affect the investment volume prediction, we group them into three integrations:

- **NP** mainly contains numerical perk features.
- **NPC** includes the numerical campaign features, besides features in NP.
- **NTPC** includes campaign textual features and perk textual features, besides features in NPC.

Therefore, the relationship among the feature integrations is $NP \subseteq NPC \subseteq NTPC$, and generally, the feature matrix X of the following experiments is constructed based on all of the features, i.e. $NTPC$. We should also note that the proposed product supply optimization approach is a general framework and it is open to some other features.

We show the performance of APSO-C in terms of different feature integrations, and the results are given in Table 4. We can have the following observations. First, NTPC feature integration almost always has the best performance, which indicates the effectiveness of our feature extraction. In our opinion, the exception exists on D#1 because of the limited training data. Second, in most feature integration except D#2, adding numerical campaign features to NP features will improve the performance, and further adding textual features seems to have produced a noticeable improvement. Therefore, we can argue that the features are internally correlated, and our model can have the best performance only when combining all kinds of contextual features.

One step further, Figure 7 gives several important features in W, where feature importance is measured based on the summary of the absolute value of each feature at all the tasks in APSO-C. Since we use vectors to express the textual features, there are multiple dimensions belong to one feature. Indeed, the features about campaign/perk description are generally more important than others for predicting the investment volume of the campaign, as shown in Figure 7.

6.4 Evaluations on Product Supply Optimization

As shown in Eq. (16) (from PSO and APSO model) and Eq. (25) (from APSO-C model), we optimize the product supply for fresh campaigns in testing data. Indeed, there are no related studies on product supply optimization in crowdfunding, and we treat the investment volumes of the campaigns (e.g. $\sum_{j=1}^{n_i} c_{ij}h_{ij}$) under the manually claimed numbers S as a baseline. Since the aim of our framework is using S' to improve these investment volumes (e.g. $\sum_{j=1}^{n_i} c'_{ij}h_{ij}$), the gain between these two different kinds of investment volumes can be adopted as metrics. For instance, we define the metric "Growth Num" as $\sum_{i=1}^{M}(\sum_{j=1}^{n_i} c'_{ij}h_{ij} - \sum_{j=1}^{n_i} c_{ij}h_{ij})/M$, and define "Growth Rate" as the average of $(\sum_{j=1}^{n_i} c'_{ij}h_{ij} - \sum_{j=1}^{n_i} c_{ij}h_{ij})/(\sum_{j=1}^{n_i} c_{ij}h_{ij})$.

TABLE 5: Product supply results (ρ=0.05).

Metrics	Growth Num	Growth Rate
Num of Money	\$208.09 (229370)	2.79% (0.36)
Num of Investors	6.05 (107.50)	4.43% (0.11)

Without loss of generality, we only report the results on the 60%-40% data split (D#2), and our optimization algorithm converges quickly by only 26 iterations (on average). The PSO results with $\rho_{ij} = 0.05$ are shown in Table 5, where we take both the two motivations/goals of the creators into consideration. The "Num of Money" motivation measures the amount of raised money when $h_{ij} = p_{ij}$ and the "Num of Investors" motivation stands for the number of investors when $h_{ij} = 1$. From this table, we can see that the optimization of the product supply structure does improve the return of the campaigns, given so many other features fixed. For instance, after optimization, each campaign is expected to attract an average of extra \$208 (or 6 investors), which accounts for 2.79% (4.43%) of its current return. The number in each bracket $(.)$ in Table 5 is the variance. After investigating the data carefully, we found that $(\sum_{j=1}^{n_i} c'_{ij}h_{ij} - \sum_{j=1}^{n_i} c_{ij}h_{ij}) > 0$ for more than 87.1% of the campaigns, which means most of the product supply of the campaigns should be optimized.

In the same way, we compare PSO and APSO-C model in different β values, here PSO result is shown under $\beta = 0_0$. Because of the huge variations on the raised money, here we just choose the other motivation ($h_{ij} = 1$) for a fair comparison, and the result is shown in Figure 8. In this figure, the bar is the average of growth rate, and the blue line is the variance of growth rate. We can see that, in the beginning, when the degree of competition β becomes larger, the number of investors increases greatly, but it will start to decrease after reaching an upper boundary. The

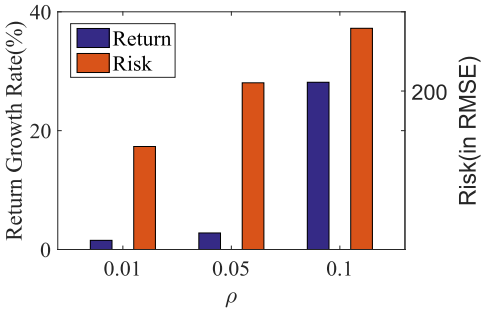


Fig. 10: Return *vs* risk.

figure shows that APSO-C model performs better than PSO, further, it also proves that the inner competition would influence final investments, so it exposes the importance of designing proper perks in one campaign.

Specifically, to know how a campaign's inner competition influences the investment volumes under the same β, we exhibit the growth rate on investment volumes of campaigns and their inner competition degrees under $\beta = 20$ in Figure (9), and all of the values in x-axis and y-axis are normalized by z-score for better exhibition. Here, we define the inner competition degree of a campaign by the average value of their own matrix V (e.g. $\sum_{j=1}^{n_i}\sum_{q=1}^{n_i} v_{jq}/(2n_i)$). From this fitting curve, we know that if perks in one campaign have low correlation, their low inner competition degree contribute little improvement of investments. On the other side, the fierce competition among perks also decrease the investments. The result shows that there is an appropriate inner competition degree for campaign designing, neither the lower nor the higher inner competition degree is better.

Finally, we also fix $\beta = 20$ in APSO-C model to explore the relationship between return and risk. Actually, the expected return for each creator can be even more impressive if we simply change the setting of ρ_{ij} (in Eq. (24)) when selecting portfolios. However, high return always associates with high risk. As shown in Figure 10, when ρ_{ij} becomes larger, not only the expected return rate but also the risk of the optimized product supply will go higher. In practice, the creators can select this parameter manually based on their risk preferences, or we can automatically make a recommendation based on historical records.

7 Related Work

The related studies can be grouped into four categories:

Crowdfunding. In a broad way, crowdfunding is a specific practice of crowdsourcing [26] [27] in finance or business. In crowdsourcing, the foundational research is how to allocate micro-tasks to suitable crowd workers [28], [29], [30], but not for crowdfunding. Since the vast majority of crowdfunding platforms follow the "all or nothing" rule [31], most of the studies in this category focus on predicting the funding results and recommending [32], i.e., whether a campaign will succeed or not [2], what factors influence the result [33], [34], and the contribution behaviors [35]. For instance, Li et al. formulated the campaign success prediction as a survival analysis problem and applied the censored regression-based solution [2]. To explore

the influenced features, T. Mitra et al. found that the description language used in the campaign also has surprising predictive power, and even accounting for 58.56% of the variance around successful funding [34]. Recently, Zhao et al. used a sequential approach to model market state of funding projects (e.g., hot and cold), and further predicted the bidding behaviors [35]. With the success of crowdfunded campaigns, it is important to understand what drives people to either create or fund these campaigns. For instance, the desire to raise funds and expand awareness of the products are two major motivations of creators [1]. In [3], the authors also claimed that the investors are sufficiently heterogeneous in their product valuations, and the creator should offer a line of products in their campaign. These studies all contribute some novel insights on campaign/perk design. Though it is also possible for the creators to get help from the manual instructions[6], to the best of our knowledge, the problem of how to automatically help creators design more attractive campaigns according to current market status, e.g., by optimizing the product supply of each perk, remains pretty much open.

Portfolio Selection. Modern portfolio theory is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given risk, and it is built upon the seminal work of Markowitz [5]. Indeed, researchers are very much interested in investigating new methods from diverse perspectives (e.g. developing novel approaches for quickly selecting portfolios) to extend/improve this theory [36], and the portfolio analysis has become an important method in finance and economics. For instance, Luo et al. viewed each investee as a portfolio of investors, and evaluated the risk of an investee based on risk preferences of investors [37]. Similar ideas inspired by portfolio selection have also been adopted in other domains, e.g., solving hard computational problems [38], information retrieval [6] and loan recommendation [39].

Multi-task Learning. Multi-task learning (MTL) performs well in classification and regression by considering the related tasks simultaneously and utilizing the cross-task information [19], [40]. Among existing MTL methods, the regularization-based MTL is one of the main research directions. These methods share the similar framework but choose different regularization terms (e.g. L1-norm) according to the task relationships [41]. Since MTL usually results in improved learning efficiency and prediction accuracy, it has been used in various fields, i.e., stock selection [42], dynamic trajectory regression [43], real estate prediction [10], biological image analysis [44], and natural language processing [45]. Besides, there are some multi-task learning works about task clustering. For example, Bakker et al. use a mixture of Gaussians to cluster tasks based on Bayesian Multi-task learning [46]. To the best of our knowledge, most existing MTL method based on the split tasks, and there are no works focus on splitting task automatically.

Competition Mining. With the availability of large online data, some researchers are concerned with competitive information mining problem [47], [48], In [48], the authors design a data-driven system which mining competitors from the general web automatically. Some other researchers focus on the competitive relationship identification problem, and their competitors mainly come from web data sources, such as online news [49], online reviews [50], [51], [52] and search log analysis [48], [53]. Recently, Wu et al. designed factor-based models to learn the competitive degree between different products based on users' behaviors [54].

In crowdfunding platforms, a fraction of researches pay attention to mining competition relationships, for example, P. Ly et al. explore the effect of competition between microfinance NGOs, and they find that competition has a negative impact on projects' funding speed [55]. J. Viotto da Cruz uses a two-sided market approach to analysis the competition in different crowdfunding platforms [56]. Kim et al. are the first to investigate the competition among projects within same categories [57]. In [58], the authors propose a probabilistic generative model to capture the competitiveness of projects. Different from the above competition researches, we focus on exploring how inner competition in products from one funding campaign influence the campaign design and funding results.

8 Concluding Remarks

In this paper, we presented a focused study on enhancing the funding performance of the newly proposed campaigns in competitive crowdfunding by optimizing the product supply of perks. Inspired by the modern portfolio theory, we first defined it as a constrained portfolio optimization problem. Under this definition, we then proposed a multi-task learning way (PSO and APSO model) to estimate the future return for each campaign and measure the risk of the product supply settings, by considering the relevance among the perks when attracting investments. Furthermore, we presented how to leverage inner perk competition effect into the APSO model. Meanwhile, we designed the APSO-C model by simultaneously learning inner competition, the coefficient matrix for investment volume prediction and the optimized product supply. Finally, the solutions for the optimization problem were recommended to creators as the optimal product supply setting. The experimental results on a real-world dataset showed that the optimized product supply can help the campaign get more investments. We hope this study could lead to more future work on optimizing other important features for campaign design in crowdfunding.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grants No. 61672483 and U1605251), the Anhui Provincial Natural Science Foundation(No 1708085QF140), and Qi Liu gratefully acknowledges the support of the Young Elite Scientist Sponsorship Program of CAST and the Youth Innovation Promotion Association of CAS (No. 2014299).

References

[1] E. M. Gerber and J. Hui, "Crowdfunding: Motivations and deterrents for participation," *ACM Trans. Comput.-Hum. Interact.*, vol. 20, no. 6, pp. 34:1–34:32, Dec. 2013. [Online]. Available: http://doi.acm.org/10.1145/2530540

6. https://www.indiegogo.com/partners

[2] Y. Li, V. Rakesh, and C. K. Reddy, "Project success prediction in crowdfunding environments," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 247–256.

[3] M. Hu, X. Li, and M. Shi, "Product and pricing decisions in crowdfunding," *Marketing Science*, vol. 34, no. 3, pp. 331–345, 2015.

[4] Q. Liu, G. Wang, H. Zhao, C. Liu, T. Xu, and E. Chen, "Enhancing campaign design in crowdfunding: a product supply optimization perspective," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 695–702.

[5] H. Markowitz, "Portfolio selection," *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[6] J. Wang and J. Zhu, "Portfolio theory of information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 115–122.

[7] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[8] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[9] T. E. Brown, E. Boon, and L. F. Pitt, "Seeking funding in order to sell: Crowdfunding as a marketing tool," *Business Horizons*, 2016.

[10] H. Zhu, H. Xiong, F. Tang, Y. Ge, Q. Liu, E. Chen, and Y. Fu, "Days on market: Measuring liquidity in real estate markets," in *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[11] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, "Personalized travel package recommendation," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 407–416.

[12] G. Celeux and G. Govaert, "A classification em algorithm for clustering and two stochastic versions," *Computational statistics & Data analysis*, vol. 14, no. 3, pp. 315–332, 1992.

[13] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 1989–1992.

[14] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.

[15] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, pp. 67–68, 1999.

[16] A. Mukherjee *et al.*, "Characterizing product lifecycle in online marketing: Sales, trust, revenue, and competition modeling," *arXiv preprint arXiv:1704.02993*, 2017.

[17] W. B. Dodds and K. B. Monroe, "The effect of brand and price information on subjective product evaluations," *NA-Advances in Consumer Research Volume 12*, 1985.

[18] L. Guo and J. Zhang, "Consumer deliberation and product line design," *Marketing Science*, vol. 31, no. 6, pp. 995–1007, 2012.

[19] L. Xu, A. Huang, J. Chen, and E. Chen, "Exploiting task-feature co-clusters in multi-task learning." in *AAAI*, 2015, pp. 1931–1937.

[20] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[21] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[22] F. Li, Y. Yang, and E. P. Xing, "From lasso regression to feature vector machine," in *Advances in Neural Information Processing Systems*, 2006, pp. 779–786.

[23] J.-T. Jeng, "Hybrid approach of selecting hyperparameters of support vector machine for regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 3, pp. 699–709, 2005.

[24] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 438–447, 2010.

[25] O. A. Omitaomu, M. K. Jeong, and A. B. Badiru, "Online support vector regression with varying parameters for time-dependent data," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 1, pp. 191–197, 2011.

[26] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: challenges, techniques, and applications," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.

[27] D. C. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence*, vol. 14, no. 1, pp. 75–90, 2008.

[28] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 49–60.

[29] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "Slade: a smart large-scale task decomposer in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[30] Y. Z. Xiang Yu, Guoliang Li. Crowdota: An online task assignment system in crowdsourcing. [Online]. Available: https://icde2018.org/index.php/program/demo-track/

[31] H. Zhao, Y. Ge, Q. Liu, G. Wang, E. Chen, and H. Zhang, "P2p lending survey: Platforms, recent advances and prospects," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 6, p. 72, 2017.

[32] H. Zhang, H. Zhao, Q. Liu, T. Xu, E. Chen, and X. Huang, "Finding potential lenders in p2p lending: A hybrid random walk approach," *Information Sciences*, vol. 432, pp. 376–391, 2018.

[33] C.-T. Lu, S. Xie, X. Kong, and P. S. Yu, "Inferring the impacts of social media on crowdfunding," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 573–582.

[34] T. Mitra and E. Gilbert, "The language that gets people to give: Phrases that predict success on kickstarter," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014, pp. 49–61.

[35] H. Zhao, Q. Liu, H. Zhu, Y. Ge, E. Chen, Y. Zhu, and J. Du, "A sequential approach to market state modeling and analysis in online p2p lending," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[36] W. Shen, J. Wang, Y.-G. Jiang, and H. Zha, "Portfolio choices with orthogonal bandit learning." in *IJCAI*, 2015, p. 974.

[37] C. Luo, H. Xiong, W. Zhou, Y. Guo, and G. Deng, "Enhancing investment decisions in p2p lending: an investor composition perspective," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 292–300.

[38] B. Silverthorn and R. Miikkulainen, "Latent class models for algorithm portfolio methods." in *AAAI*, 2010.

[39] H. Zhao, Q. Liu, G. Wang, Y. Ge, and E. Chen, "Portfolio selections in p2p lending: A multi-objective perspective," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2075–2084.

[40] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[41] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Advances in neural information processing systems*, 2011, pp. 702–710.

[42] J. Ghosn and Y. Bengio, "Multi-task learning for stock selection," *Advances in Neural Information Processing Systems*, pp. 946–952, 1997.

[43] A. Huang, L. Xu, Y. Li, and E. Chen, "Robust dynamic trajectory regression on road networks: A multi-task learning framework," in *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 2014, pp. 857–862.

[44] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," *IEEE transactions on Big Data*, 2016.

[45] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[46] B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," *Journal of Machine Learning Research*, vol. 4, no. May, pp. 83–99, 2003.

[47] T. Lappas, G. Valkanas, and D. Gunopulos, "Efficient and domain-invariant competitor mining," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 408–416.

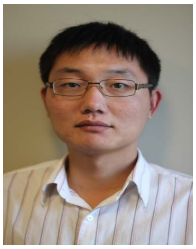
[48] S. Bao, R. Li, Y. Yu, and Y. Cao, "Competitor mining with the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 10, pp. 1297–1310, 2008.

[49] Z. Ma, G. Pant, and O. R. Sheng, "Mining competitor relationships from online news: A network-based approach," *Electronic Commerce Research and Applications*, vol. 10, no. 4, pp. 418–427, 2011.
[50] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 785–794.
[51] K. Xu, S. S. Liao, J. Li, and Y. Song, "Mining comparative opinions from customer reviews for competitive intelligence," *Decision support systems*, vol. 50, no. 4, pp. 743–754, 2011.
[52] O. Netzer, R. Feldman, J. Goldenberg, and M. Fresko, "Mine your own business: Market-structure surveillance through text mining," *Marketing Science*, vol. 31, no. 3, pp. 521–543, 2012.
[53] Q. Wei, D. Qiao, J. Zhang, G. Chen, and X. Guo, "A novel bipartite graph based competitiveness degree analysis from query logs," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 2, p. 21, 2016.
[54] L. Wu, Q. Liu, R. Hong, E. Chen, Y. Ge, X. Xie, and M. Wang, "Product adoption rate prediction in a competitive market," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 325–338, 2018.
[55] P. Ly and G. Mason, "Competition between microfinance ngos: evidence from kiva," *World Development*, vol. 40, no. 3, pp. 643–655, 2012.
[56] J. Viotto da Cruz, "Competition and regulation of crowdfunding platforms: a two-sided market approach," 2015.
[57] J. Kim, M. Lee, D. Cho, and B. Lee, "An empirical analysis of semantic network in online crowdfunding: evidence from kickstarter," in *Proceedings of the 18th Annual International Conference on Electronic Commerce: e-Commerce in Smart connected World*. ACM, 2016, p. 44.
[58] Y. Lin, P. Yin, and W.-C. Lee, "Modeling dynamic competition on crowdfunding markets," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1815–1824.

Guifeng Wang received the B.E. degree in artificial intelligence from the Xiamen University, Xiamen, China, in 2015. She is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China. Her research interests include data mining, Internet finance-based applications, such as Crowdfunding and P2P lending.

Qi Liu is an associate professor at USTC. He received the Ph.D. degree in Computer Science from USTC. His general area of research is data mining and knowledge discovery. He has published prolifically in refereed journals and conference proceedings, e.g., TKDE, TOIS, TKDD, TIST, KDD, IJCAI, AAAI, ICDM, SDM and CIKM. He is a member of ACM and IEEE. Dr. Liu is the recipient of the ICDM 2011 Best Research Paper Award, the Best of SDM 2015 Award and the KDD 2018 Best Student Paper Award (Research Track).

Hongke Zhao received the B.E. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. His current research interest includes data mining, deep learning and Internet finance-based applications, such as Crowdfunding and P2P lending. He has published papers in refereed journals and conference proceedings, such as the ACM Transactions on Intelligent Systems and Technology, the IEEE Transactions on Systems, Man, and Cybernetics: Systems, the Information Sciences, ACM SIGKDD, IEEE ICDM, AAAI and IJCAI.



Chuanren Liu received the BS degree from the University of Science and Technology of China (USTC), the MS degree from the Beijing University of Aeronautics and Astronautics (BUAA), and the PhD degree from Rutgers, the State University of New Jersey. He is currently an assistant professor in the Decision Science and Management Information Systems Department at Drexel University. His research interests include data mining and knowledge discovery, and their applications in business analytics. He has published papers in refereed journals and conference proceedings, such as the Annals of Operations Research, the European Journal of Operational Research, the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Cybernetics, Knowledge and Information Systems, and ACM SIGKDD, IEEE ICDM, SIAM SDM, etc.

Tong Xu (M'17) currently working as a Associate Researcher of the Anhui Province Key Laboratory of Big Data Analysis and Application, USTC. He has authored 20+ journal and conference papers in the fields of data mining, including KDD, AAAI, ICDM, SDM, etc.

Enhong Chen (SM'07) is a professor and vice dean of the School of Computer Science at USTC. He received the Ph.D. degree from USTC. His general area of research includes data mining and machine learning, social network analysis and recommender systems. He has published more than 100 papers in refereed conferences and journals, including IEEE Trans. KDE, IEEE Trans. MC, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, SDM. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. He is a senior member of the IEEE.