Enhancing Campaign Design in Crowdfunding: A Product Supply Optimization Perspective

Qi Liu[†], **Guifeng Wang**[†], **Hongke Zhao**[†], **Chuanren Liu**[‡], **Tong Xu**[†], **Enhong Chen**[†*]
[†]Anhui Province Key Lab. of Big Data Analysis and Application, Universitys of S&T of China
[‡]Decision Sciences & MIS Department, Drexel University
qiliuql@ustc.edu.cn, {wgf1109, zhhk}@mail.ustc.edu.cn, chuanren.liu@drexel.edu,
tongxu@mail.ustc.edu.cn, cheneh@ustc.edu.cn

Abstract

Crowdfunding is an emerging Internet application for creators designing campaigns (projects) to collect funds from public investors. Usually, the limited budget of the creator is manually divided into several perks (reward options), that 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 campaign. To this end, in this paper, we aim to enhance the funding performance of the newly proposed campaigns, with a focus on optimizing the product supply of perks. Specifically, given the expected budget and the perks of a campaign, we propose a novel solution to automatically recommend the optimal product supply to every perk 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 risk of each campaign is measured by a multi-task learning method. Finally, experimental results on the real-world crowdfunding data clearly prove that the optimized product supply can help improve the campaign performance significantly, and meanwhile, our multi-task learning method could more precisely estimate the risk of each campaign.

1 Introduction

With the rapid development of the Internet, crowdfunding, which provides a revolutionary way to support ideas and campaigns across a wide range of domains (e.g. technology, film, art), has rapidly risen in popularity [Gerber and Hui, 2013]. It was estimated that the global crowdfunding industry has raised more than US$34 billion for millions of campaigns in 2015, and this market share may have surpassed venture capital in the year of 2016 [Barnett, 2015].

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

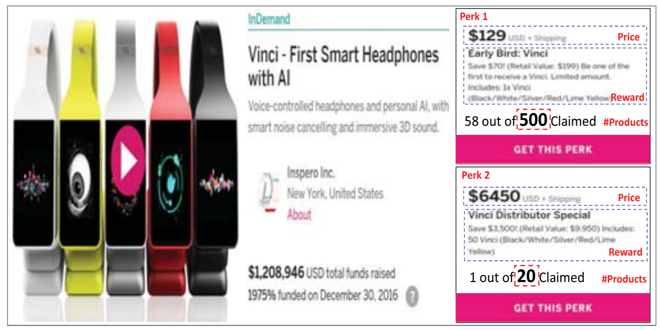

Figure 1: An example of the campaign (Indiegogo.com).

(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 [Li *et al.*, 2016]. Therefore, predicting the success rate of a campaign and inferring the impacts of specific factors on investor decision (e.g., the campaign descriptions, the social networks of creators) have become research hotspots.

Since the investors in crowdfunding are sufficiently heterogeneous in product valuations, a campaign usually offers a variety of rewards in the form of *perks* [Hu *et al.*, 2015] 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. $129), product rewards (e.g. one headphone) and the claimed product supply (e.g. 500 for Perk 1) to maximize the expected funding or awareness. However, 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, the choice of the product supply for each perk is limited due to the entire budget of one campaign. Second, it is difficult to estimate the return and risk before the campaign is finished, since the investments are potentially affected by many static and temporal factors, such as the perk or campaign descriptions and the funding dynamics.

To address these challenges, in this paper, 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, simul-

*Corresponding author.

taneously. Along this line, inspired by the modern portfolio theory [Markowitz, 1952; Wang and Zhu, 2009], 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 the perks when attracting investments, we propose a trace-norm multi-task learning method to estimate the future return 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. Finally, we conduct extensive experiments on the real-world crowdfunding data that was crawled from Indiegogo.com. The results clearly prove that the optimized product supply can help improve the future performance of campaigns significantly, and meanwhile, our multi-task learning method could more precisely estimate the investment (risk) of each campaign setting. 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 Related Work

The related studies can be grouped into three categories: Crowdfunding, Portfolio Selection and Multi-task Learning.

Crowdfunding. Since the vast majority of crowdfunding platforms follow the “all or nothing” rule, most of the studies in this category focus on predicting the funding results, i.e., whether a campaign will succeed or not [Li *et al.*, 2016], what factors influence the result [Lu *et al.*, 2014; Mitra and Gilbert, 2014], and the contribution behaviors [Zhao *et al.*, 2017]. For instance, [Li *et al.*, 2016] formulated the campaign success prediction as a survival analysis problem and applied the censored regression-based solution. To explore the influenced features, [Mitra and Gilbert, 2014] 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. Recently, [Zhao *et al.*, 2017] used a sequential approach to model market state of funding projects (e.g., hot and cold), and further predicted the bidding behaviors. 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 of the major motivations of creators, according to the interviews given by [Gerber and Hui, 2013]. In [Hu *et al.*, 2015] , the authors also claimed that the investors are sufficiently heterogeneous in their product valuations, and the creator should offer a line of products in the 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[1], 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.

[1] https://www.indiegogo.com/partners

Table 1: 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
L	number	the number of learning tasks
P_i	vector	p_{ik} is the price of the k-th perk in campaign i
S_i	vector	s_{ik} is the given product supply of the perk
E_i	vector	e_{ik} is the reward of the k-th perk in campaign i
C_i	vector	c_{ik} is the number of investments under the setting s_{ik}
h_i	vector	h_{ik} is 1 or the price of the k-th perk in campaign i
C_t	vector	each entry is the label (e.g. c_{ik}) of one perk
X_t	matrix	each row stores the feature vector of one perk
C'_i	vector	c'_{ik} is the number of investments under the setting s'_{ik}
C'_t	vector	the estimated number of investments (e.g. c'_{ik}) of one perk
S'_i	vector	s'_{ik} is the optimized product supply
W_t	matrix	L rows matrix, contains the importance of each feature to every task

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 [Markowitz, 1952]. 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 [Shen *et al.*, 2015], and the portfolio analysis has become an important method in finance and economics. For instance, [Luo *et al.*, 2011] viewed each investee as a portfolio of investors, and evaluated the risk of an investee based on risk preferences of investors. Similar ideas inspired by portfolio selection have also been adopted in other domains, e.g., solving hard computational problems [Silverthorn and Miikkulainen, 2010], information retrieval [Wang and Zhu, 2009] and loan recommendation [Zhao *et al.*, 2016].

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 [Caruana, 1998; Xu *et al.*, 2015]. Among existing MTL methods, the regularization-based multi-task learning 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 [Zhou *et al.*, 2011]. Since MTL usually results in improved learning efficiency and prediction accuracy, it has been used in various fields, i.e., stock selection [Ghosn and Bengio, 1997], dynamic trajectory regression [Huang *et al.*, 2014], real estate prediction [Zhu *et al.*, 2016], and natural language processing [Collobert and Weston, 2008].

3 Product Supply Optimization

In this section, we first detail the problem of product supply optimization in crowdfunding, and then show the way of solving this problem by a constrained portfolio selection and multi-task learning. For better illustration, Table 1 lists some mathematical notations, the transverse line distinguishes the input and output variables, where the variables in upper part are given (input variables), and the rest variables need to be learnt (output variables).

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:

Problem Formulation. 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 $P_i = \{p_{i1}, p_{i2}, ..., p_{in_i}\}$, the manually set/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 descriptions), our goal is to get the optimal product supply $S'_i = \{s'_{i1}, s'_{i2}, ..., s'_{in_i}\}$ of the perks, which can bring the maximum expected return with minimum risk for the target campaign.

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

where ρ_i is a heuristic parameter, 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 the cost B'_i after optimizing could not exceed the expect cost 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 the two perks in Figure 1 are \$129 and \$6450, respectively), and e_{ij} in E_i is the reward/payback that the creator should provide to the investors (e.g. e_{i1} and e_{i2} for the two perks in Figure 1 are 1x Vinci and 50x Vinci, 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 $P_i = \{\$129, \$6450\}$ and the product numbers preset by the creator are $S_i = \{500, 20\}$, respectively. We hope to recommend a more suitable product number setting S' (e.g. $\{400, 30\}$) to replace this S_i for making the campaign even more successful.

Computation of $Return_i$. 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) [Brown *et al.*, 2016; Gerber and Hui, 2013]. Thus, the expected return of campaign i can be measured by:

$$Return_i = \sum_{j=1}^{n_i} c'_{ij} h_{ij}, \quad h_{ij} \in \{p_{ij}, 1\}, \tag{2}$$

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. 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.

Computation of $Risk_i$. 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 can not be perfectly estimated, the future return of a new campaign is actually unknown and Eq. (2) is computed with uncertainty/risk. In finance, risk is usually measured by *(co)variance* of the investments (e.g. stocks) [Markowitz, 1952; Wang and Zhu, 2009]. Correspondingly, in crowdfunding, the risk of setting product supply as S_i for campaign i can be estimated by the variance of the returns:

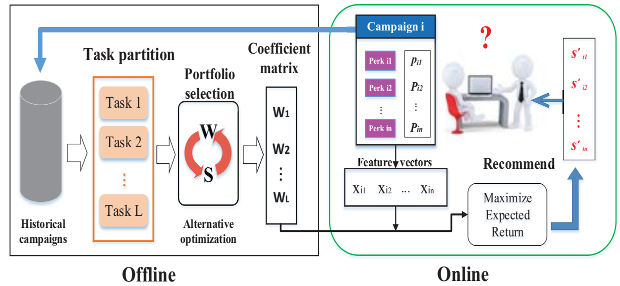

Figure 2: The flowchart of product supply optimization.

$$Risk_i \approx \sum_{j=1}^{n_i} (c'_{ij} h_{ij} - c_{ij} h_{ij})^2. \tag{3}$$

Therefore, summarizing Eq. (1) for all the campaigns:

$$\max_{S'_i} \sum_{i=1}^{M} \rho_i Return_i - \sum_{i=1}^{M} Risk_i, \quad s.t. \quad B'_i \leq B_i, \quad \forall i \in [1, M], \tag{4}$$

where M is the number of campaigns. Obviously, as a portfolio of investment choices in the platform, every campaign (perk) is related to others (e.g. the competitors will have an influence on the final investment volume). Thus, instead of computing independently, the risk in Eq. (4) should be measured more precisely by exploiting this kind of relevance.

Multi-task Learning. 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:

$$\sum_{i=1}^{M} Risk_i = \sum_{t=1}^{L} ||(X_t W_t^\top - C_t) \times h_t||_F^2 + \lambda ||W||_*, \tag{5}$$

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 $c'_{ij} = X_{ij} W_{t_{ij}}^\top$. Different choices of regularization terms may reflect different task relationships [Zhu *et al.*, 2016]. 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.

We can rewrite the $Return_i$ in Eq. (4) in this multi-task learning way. Thus, the optimization problem becomes:

$$\begin{aligned} \max_{W,S'} \quad & \sum_{i=1}^{M}\sum_{j=1}^{n_i} \rho_{ij}[s'_{ij} w_{t_{ij}}^{s'} + X_{ij}^{-s'} (W_{t_{ij}}^{-s'})^\top] h_{ij} \\ & - (\sum_{t=1}^{L} ||(X_t W_t^\top - C_t) \times h_t||_F^2 + \lambda ||W||_*), \\ s.t. \quad & s'_{ij} \geq 0 \quad and \quad B'_i \leq B_i, \quad \forall i \in [1, M]. \end{aligned} \tag{6}$$

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 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.

Product Supply Optimization. In summary, given the campaigns/perks and their features X, for solving the product supply optimization problem, we first put these perks into different learning tasks, and then learn the W and S' based on the training ones by solving Eq. (6). Thus, the optimal S'_i for each testing campaign can be computed by only maximizing the expected return.

$$\begin{aligned} \max_{S'_i} \quad & \sum_{j=1}^{n_i} \rho_{ij}[s'_{ij} w_{t_{ij}}^{s'} + X_{ij}^{-s'} (W_{t_{ij}}^{-s'})^\top] h_{ij}, \\ s.t. \quad & s'_{ij} \geq 0, \quad B'_i \leq B_i. \end{aligned} \tag{7}$$

Please note the difference between Eq. (6) and (7), i.e. for the new campaign in test, the risk is unknown without any investment records. Finally, S'_i (i.e. the output of Eq. (7)) is recommended to the creator when she is publishing this campaign. The entire flowchart of the framework is illustrated in Figure 2. Now, we show the way of solving Eq. (6).

Optimization Algorithm. Before solving Eq. (6), we should first put the perks into L different learning tasks. Indeed, there are a number of methods for task partition according to the features X of perks. The details of the task partition and the features of perks will be shown in the experiments. Here, we propose to solve Eq. (6) by the Alternating Optimization Method, which is similar to the Block Coordinate Descent method [Wright and Nocedal, 1999], the variable is optimized alternatively with the other variables fixed. Because Eq. (6) is continuous and separately convex, the alternating optimization algorithm is convergent. 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):

$$\nabla f(W_t) = (W_t X_t^\top X_t - 2C_t^\top X_t) - \sum_{i=1}^{M}\sum_{j=1}^{n_i} \rho_{ij} X_{ij}, \tag{8}$$

$$\begin{aligned} \nabla g(s'_{ij}) = & 2W_{t_{ij}}^{s'} s'_{ij} W_{t_{ij}}^{s'} + 2(X_{ij}^{-s'} W_{t_{ij}}^{-s'} - C_{ij}) W_{t_{ij}}^{s'} \\ & - \rho_{ij} W_{t_{ij}}^{s'}. \end{aligned} \tag{9}$$

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 stepsize policy, and we use projection method to satisfy the constrains.

During the implement, we first update W for each task. For task t which contains n_t campaigns, each campaign has D dimensional features, so the computational complexity is $O(n_t^2 D^2)$. Then we update S' and X, and the computational complexity can be regard 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.

Algorithm 1 Alternating Optimization Method

```
Input: X, C, S, B, N, tol_W, tol_S'
Output: W, S'
set S' = S, W_0 = 0
for k = 1 to N do
    Update W based on Eq. (8),
    Update S' based on Eq. (9),
    Update X by replacing S'.
    if stopping criteria is satisfied then
        Break
    end if
end for
return W, S'
```

4 Experiments

In this section, we provide empirical validation on a real-world dataset that we crawled from one famous crowdfunding platform in America, i.e. Indiegogo.com.

Dataset Description. Our experimental data includes the campaign information, perk information, and some mutual records of creators and investors[2]. For instance, it contains 14,143 launched campaigns for more than 18 billion funds (including 98,923 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.

[2]This data will be publicly available after the paper acceptance.

Table 2: 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

Here, we remove the unfinished campaigns because their investment volume still changes. To observe how each algorithm behaves at different sparsity levels, 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 testing sets as $D\#1$, $D\#2$, $D\#3$ and $D\#4$.

Feature Extraction. We extract 23 features from the campaign level and 9 features from the perk level, and the details of them are illustrated in Table 2. We can see the features are very heterogeneous, including numerical ones (e.g., perk price, goal), categorical ones (e.g., category, location) and text (e.g., campaign description, perk description). For data preprocessing and constructing the feature matrix X, we first transform categorical data into N binary-valued features (numerical ones) using one-hot encoding (i.e. dummy feature). Meanwhile, the doc2vec method [Le and Mikolov, 2014] is adopted to convert textual data into numerical vectors, e.g., the perk description is represented by a 10 dimension vector, and the campaign title and the campaign description are turned into 5 and 100 dimension vectors separately. To explore how these features affect the investment volume prediction, we group them into four integrations:

- **NP**: Mainly contains numerical perk features.
- **NTP**: Besides features in NP, textual features of perks are also included.
- **NPC**: Besides features in NP, there are also the numerical campaign features.
- **NTPC**: Besides features in NPC, campaign textual features and perk textual features are also included.

Therefore, the relationship among the feature integrations is $NP \subseteq NTP \subseteq NTPC$, $NP \subseteq NPC \subseteq NTPC$, 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.

Task Partition. In MTL, we should put perks into different tasks. Considering that, products with near prices are more related than distant ones, we propose to split perks into different tasks according their prices, i.e. the perks with the similar prices will be put into the same task. Without loss of generality, 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. Actually, our solution is a general framework which is open to different task partition methods[Liu *et al.*, 2011].

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 some other stuff of the project, therefore, it is very hard to directly compute B'_i and B_i based on e_{ij} (Please refer to Section 3 for detailed information). Luckily, in the marketing literature, it is usually assumed that the price positively influences the perception of product (reward) quality, that is $e_{ij} \propto p_{ij}$ [Dodds and Monroe, 1985], and the unit cost of the product with a quality e_{ij} is $e_{ij}^2/2$ [Guo and Zhang, 2012; Hu *et al.*, 2015] which can be further represented as $\theta p_{ij}^2/2$, where θ is a parameter. Therefore, in the following experiments we define $B'_i = \frac{1}{2}\theta \sum_{j=1}^{n_i} s'_{ij} p_{ij}^2$ and $B_i = \frac{1}{2}\theta \sum_{j=1}^{n_i} s_{ij} p_{ij}^2$.

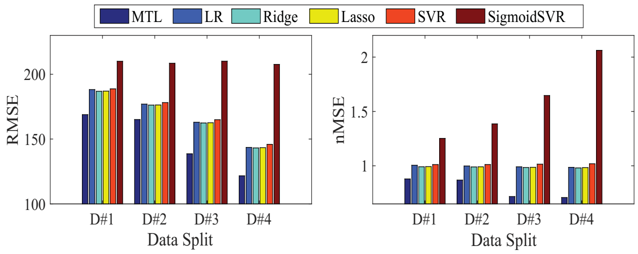

Figure 3: Investment volume prediction performance.

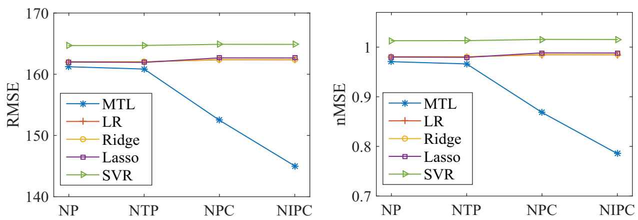

Figure 4: Performance on different feature integrations.

Parameter ρ_{ij} represents the risk preferences of creators, 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 through cross validation.

Evaluations on Multi-task Learning. Before proving the effectiveness of the entire framework of product supply optimization, we first show the performance of MTL on measuring the risk (predicting the future investment volume) of each campaign (Eq. (5)). We adopt RMSE and nMSE as the metrics as they are widely used in MTL [Xu *et al.*, 2015; Zhu *et al.*, 2016]. For the RMSE and nMSE, the smaller the value, the better the performance. We also choose several state-of-the-art regressions for comparison [Zhu *et al.*, 2016]:

- **Linear Regression (LR)**: training a linear regression model for predicting the investment volume.
- **Ridge**: LR with L2-norm regularizer.
- **Lasso**: LR with L1-norm regularizer.
- **Support Vector Regression (SVR)**: training a Support Vector Regression model.
- **SVR with sigmoid kernel (SigmoidSVR)**: Support Vector Regression with sigmoid kernel.

The experimental results of our MTL method and the baselines on four data splits are shown in Figure 3. Due to space limitation, we only show the results with $h_{ij} = 1$. We can see that MTL method consistently performs the best on all splits in terms of two evaluation metrics, which clearly validates the effectiveness of our multi-task learning method. Also, the *SigmoidSVR* is the worst which proves that investment volume prediction is more likely a linear regression instead of a nonlinear regression. We also show the performance of each method in terms of different feature integrations, and the results are given in Figure 4. We can have the following observations. First, the more features contains, the better performance of trace-norm MTL is, but this kind of phenomenon is not obvious in other methods (baselines). We think the reason is that trace-norm MTL method constrains the learning model from different tasks to share a low-dimension subspace to capture the task/campaign relevance. The more features are

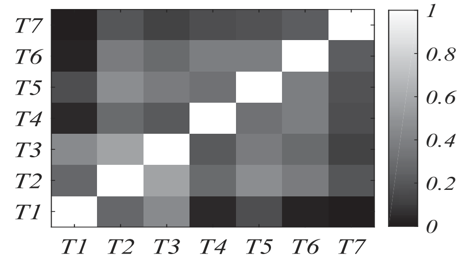

Figure 5: Correlation between tasks.

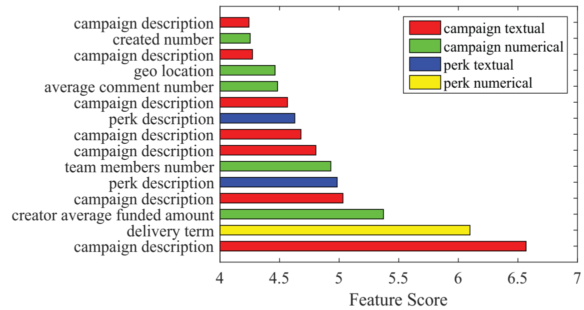

Figure 6: An illustration of several important features.

applied, the better the shared subspace can be learnt, and then the more clear the task relevance is. Second, the performance improvement is significant from *NTP* to *NPC*, and to *NTPC* on MTL method, which indicates the campaign features are very effective in improving the prediction accuracy on investment volume, and carefully describing the campaign is really crucial for the success of creator [Mitra and Gilbert, 2014].

Task Correlations and Feature Study. To intuitively illustrate the utilities of MTL, we compute the Jaccard similarity[3] among the rows (tasks) in W. The results are shown in Figure 5, where we can see that the tasks with the similar price ranges are usually more similar, and task $T1$ and task $T7$ are the most distinctive ones. One step further, Figure 6 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. 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 6.

Evaluations on Product Supply Optimization. Indeed, there are no related studies on the task of 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 difference between these two different kinds of investment volumes can be adopted as metrics. For instance, we define the metric

[3]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.

Table 3: 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)

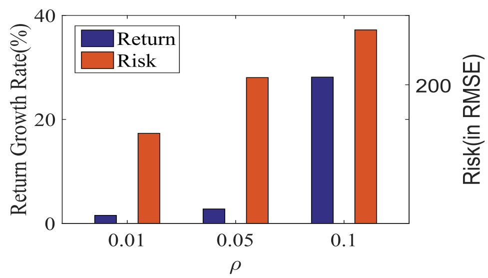

Figure 7: Return *vs* risk.

"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 $(\sum_{j=1}^{n_i} c'_{ij}h_{ij} - \sum_{j=1}^{n_i} c_{ij}h_{ij})/(\sum_{j=1}^{n_i} c_{ij}h_{ij})$.

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 final experimental results with $\rho_{ij} = 0.05$ are shown in Table 3, 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 3 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.

Actually, the expected return for each creator can be even more impressive if we simply change the setting of ρ_{ij} (in Eq. (6)) when selecting portfolios. However, high return always associates with high risk. As shown in Figure 7, 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.

Case Study. In Table 4, we present a case study of the product supply optimization results on two real campaigns, Pon[4] and The Wipy[5]. Specifically, Price(in \$) is the price of each perk, S is the claimed/given product supply number and S' is the optimized/recommended number (Without loss of generality, we set $h_{ij} = 1$ for outputting S'). Let's take campaign Pon as an exmaple. From this table we can see that the creator of Pon simply sets the same supply number (i.e. 100) to each perk. In contrast, our optimization method can detect the variance of investors (e.g. more investors will be interested in the perk with price \$35 and there are few of investors interest in the highest price \$85) and make the product supply more reasonable. Note that the number of investors can be found in the webpage of these campaigns.

Table 4: Case study on two campaigns.

Campaign Title	Pon-The Punctureless Push Pin						The Wipy: IoT of the future	
Price(in \$)	9	11	13	35	47	85	22	35
S	100	100	100	100	100	100	1500	500
S'	78	85	85	207	78	69	1556	446

5 Conclusion

In this paper, we presented a focused study on enhancing the funding performance of newly proposed campaigns in 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 of estimating the future return for each campaign and measuring the risk of the product supply settings, considering the relevance among the perks when attracting investments. 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.

Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the National Natural Science Foundation of China (Grants No. 61672483 and U1605251), the Anhui Provincial Natural Science Foundation(No 1708085QF140), and the Fundamental Research Funds for the Central Universities. Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299) and the MOE-Microsoft Key Laboratory of USTC.

[4]https://www.indiegogo.com/projects/pon-the-punctureless-push-pin-photography

[5]https://www.indiegogo.com/projects/the-wipy-the-internet-of-things-of-the-future-wifi

References

[Barnett, 2015] Chance Barnett. Trends show crowdfunding to surpass vc in 2016. *Forbes*, 2015.

[Brown *et al.*, 2016] Terrence E Brown, Edward Boon, and Leyland F Pitt. Seeking funding in order to sell: Crowdfunding as a marketing tool. *Business Horizons*, 2016.

[Caruana, 1998] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[Collobert and Weston, 2008] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

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

[Gerber and Hui, 2013] Elizabeth M. Gerber and Julie Hui. Crowdfunding: Motivations and deterrents for participation. *ACM Trans. Comput.-Hum. Interact.*, 20(6):34:1–34:32, December 2013.

[Ghosn and Bengio, 1997] Joumana Ghosn and Yoshua Bengio. Multi-task learning for stock selection. *Advances in Neural Information Processing Systems*, pages 946–952, 1997.

[Guo and Zhang, 2012] Liang Guo and Juanjuan Zhang. Consumer deliberation and product line design. *Marketing Science*, 31(6):995–1007, 2012.

[Hu *et al.*, 2015] Ming Hu, Xi Li, and Mengze Shi. Product and pricing decisions in crowdfunding. *Marketing Science*, 34(3):331–345, 2015.

[Huang *et al.*, 2014] Aiqing Huang, Linli Xu, Yitan Li, and Enhong Chen. Robust dynamic trajectory regression on road networks: A multi-task learning framework. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 857–862. IEEE, 2014.

[Le and Mikolov, 2014] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[Li *et al.*, 2016] Yan Li, Vineeth Rakesh, and Chandan K Reddy. Project success prediction in crowdfunding environments. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 247–256. ACM, 2016.

[Liu *et al.*, 2011] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. Personalized travel package recommendation. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 407–416. IEEE, 2011.

[Lu *et al.*, 2014] Chun-Ta Lu, Sihong Xie, Xiangnan Kong, and Philip S Yu. Inferring the impacts of social media on crowdfunding. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 573–582. ACM, 2014.

[Luo *et al.*, 2011] Chunyu Luo, Hui Xiong, Wenjun Zhou, Yanhong Guo, and Guishi 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*, pages 292–300. ACM, 2011.

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

[Mitra and Gilbert, 2014] Tanushree Mitra and Eric 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*, pages 49–61. ACM, 2014.

[Shen *et al.*, 2015] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *IJCAI*, page 974, 2015.

[Silverthorn and Miikkulainen, 2010] Bryan Silverthorn and Risto Miikkulainen. Latent class models for algorithm portfolio methods. In *AAAI*, 2010.

[Wang and Zhu, 2009] Jun Wang and Jianhan Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122. ACM, 2009.

[Wright and Nocedal, 1999] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35:67–68, 1999.

[Xu *et al.*, 2015] Linli Xu, Aiqing Huang, Jianhui Chen, and Enhong Chen. Exploiting task-feature co-clusters in multitask learning. In *AAAI*, pages 1931–1937, 2015.

[Zhao *et al.*, 2016] Hongke Zhao, Qi Liu, Guifeng Wang, Yong Ge, and Enhong 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*, pages 2075–2084. ACM, 2016.

[Zhao *et al.*, 2017] Hongke Zhao, Qi Liu, Hengshu Zhu, Yong Ge, Enhong Chen, Yan Zhu, and Junping Du. A sequential approach to market state modeling and analysis in online p2p lending. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[Zhou *et al.*, 2011] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in neural information processing systems*, pages 702–710, 2011.

[Zhu *et al.*, 2016] Hengshu Zhu, Hui Xiong, Fangshuang Tang, Yong Ge, Qi Liu, Enhong Chen, and Yanjie 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.