

Estimating Early Fundraising Performance of Innovations via Graph-Based Market Environment Model

Likang Wu,¹ Zhi Li,¹ Hongke Zhao,² Zhen Pan,¹ Qi Liu,¹ Enhong Chen^{1*}

¹Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China

²College of Management and Economics, Tianjin University

{wulk, zhili03, pzhen}@mail.ustc.edu.cn, {qiliuql, cheneh}@ustc.edu.cn, hongke@tju.edu.cn

Abstract

Well begun is half done. In the crowdfunding market, the early fundraising performance of the project is a concerned issue for both creators and platforms. However, estimating the early fundraising performance before the project published is very challenging and still under-explored. To that end, in this paper, we present a focused study on this important problem in a market modeling view. Specifically, we propose a Graph-based Market Environment model (GME) for estimating the early fundraising performance of the target project by exploiting the market environment. In addition, we discriminatively model the *market competition* and *market evolution* by designing two graph-based neural network architectures and incorporating them into the joint optimization stage. Finally, we conduct extensive experiments on the real-world crowdfunding data collected from Indiegogo.com. The experimental results clearly demonstrate the effectiveness of our proposed model for modeling and estimating the early fundraising performance of the target project.

Introduction

Recent years have witnessed the rapid development of online crowdfunding platforms, such as Indiegogo¹ and Kickstarter². More and more entrepreneurs and individuals create projects on these platforms to demonstrate their innovative ideas and solicit funding from the public.

The prevalence of crowdfunding platforms triggers many research problems, such as project success prediction (Li, Rakesh, and Reddy 2016), recommendations (Zhang et al. 2019) and funding dynamic tracking (Zhao et al. 2017b; 2019). Most of these existing studies focus on modeling the fundraising process after the project published. However, for creators and platform operators, they all want to understand and estimate the early fundraising performance before the project setup. Actually, according to the official statistics of online crowdfunding platforms, the fundraising ability of the project reached its peak in the first 24 hours³, and projects

with early funding achieve 30% of corresponding goals are most likely to succeed⁴. But it is hard for the founder to ensure that the project has a good start due to the complex and dynamic market environment. Thus, modeling the market environment is an essential issue for judging whether it is a good time to start the project. Unfortunately, less effort has been made towards this goal. To that end, in this paper, we conduct a focused study on modeling and estimating the early performance of innovations in a market view. To the best of our knowledge, this is the first attempt in this area.

Indeed, it is very challenging to estimate the early performance of unpublished innovations. As mentioned above, the project fundraising performance is largely influenced by the market environment. On the one hand, once a project is published, it will face market competition from other projects after published. That is difficult to model the complex competition relationships and estimate the competitiveness of the target project. On the other hand, market evolution can also have a huge impact on the fundraising process. Then, tracking market evolution is another challenge for estimating the early performance of the target innovation.

To address the challenges mentioned above, in this paper, we present a focused study on exploiting the project early fundraising performance in a market modeling view. Along this line, we propose a novel model, i.e., Graph-based Market Environment model (GME), to explore the market environment of online crowdfunding platforms and estimate the early performance of the target project. Specifically, GME consists of two components, i.e., Project Competitiveness Modeling (PCM) and Market Evolution Tracking (MET). In the PCM module, we apply a hybrid Graph Neural Network (GNN) to model the competitiveness pressure of the target project by aggregating competitive influences of other competitive projects in the market. Additionally, in the MET module, we design a propagation structure to track the market evolution and apply a Gated Graph Neural Networks (GG-NNs) (Li et al. 2015) to model the evolution process. Next, we optimize our GME with a joint loss to learn the fundraising performance of competitive projects and estimate the early performance of the target project, simultane-

*Corresponding Author.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://www.indiegogo.com/>

²<https://www.kickstarter.com/>

³<https://pse.is/KWP5T>

⁴<https://www.startups.com/library/expert-advice/key-crowdfunding-statistics>

ously. Finally, we conduct extensive experiments on the real-world crowdfunding data collected from Indiegogo.com. The experimental results clearly demonstrate the effectiveness of our proposed model for modeling and estimating the early fundraising performance of the target project.

Related Work

In this section, we review the research from two categories, i.e., crowdfunding and graph neural networks.

Crowdfunding

The research of online crowdfunding can be divided into two categories according to the research perspective: for the individual projects and for the entire market. For the individual projects, most researchers paid much attention to the prediction of project success (Li, Rakesh, and Reddy 2016; Jin et al. 2019). Besides, Liu et al. (2017) optimized the algorithm of production supply to reduce the redundant losses of creators, and Zhao et al. (2017b) focused on tracking the dynamics for projects in their complete funding durations. For the entire market, Lin, Yin, and Lee (2018) modeled dynamic competition on crowdfunding markets, Zhang et al. (2019) are devoted to the recommender systems on the crowdfunding market, and Janku and Kucerova (2018) propose that good early performance on the crowdfunding market is especially significant.

In spite of the importance of previous studies, they mainly focused on modeling the market reaction or investors' interest after projects launched, and were still lack in a deep and quantitative exploration of the early performance before the project setup. To the best of our knowledge, in this paper, we are the first comprehensive attempt to exploit the prediction of early performance in a market modeling view.

Graph Neural Network

Graph Neural Network (GNN) has been proven successful in modeling the structured graph data due to its theoretical elegance (Bronstein et al. 2017). Extending neural networks to work with graph-structured data was first proposed by Gori, Monfardini, and Scarselli (2005). With the proposal of spectral GNN (Bruna et al. 2013) and its improvements (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016), the research in this area began to develop rapidly. Besides, Zhou, Huang, and Schölkopf (2007) regularized the learning procedure of the target task with a hypergraph that captures the higher-order relations among entities. Gilmer et al. (2017) proved that applications of the neural networks to graphs could be seen as specific instances of a learnable message passing framework on graphs.

These years, graph neural network has been applied to many tasks without explicit graph structure. For instance, Cucurull, Taslakian, and Vazquez (2019) addressed the compatibility prediction problem using a graph neural network that learns to generate product embeddings conditioned on context. Feng et al. (2019) created the influence graph model by building the edges between related stocks in the stock prediction task. In this paper, we organize the projects

in the crowdfunding market as a market graph. And we further propose the GNN-based networks to model the market competitiveness and market evolution, which is the first attempt in this domain.

Methodology

In this section, we propose a framework to estimate the project's early fundraising performance in a market modeling view. First, we introduce the research problem and the constructed features we use. Then, we present the technical details of our proposed model, i.e., Graph-based Market Environment model (GME).

Preliminaries

In online crowdfunding services, different start-up time has a great impact on the early fundraising performance due to the different market environments (Janku and Kucerova 2018). Along this line, we aim at estimating the early performance in a market modeling view. However, the early fundraising amounts are improper to directly measure the early performance, since the same fundraising amount in early-stage means different performance for projects with different fund goals. So in this paper, we use the achievement of goal in the early stage to measure our target value. More specifically, For the target project i , we can get the quantitative definition of the early performance y_i as:

$$y_i = \log_2 \left(1 + \frac{\alpha_i}{g_i} \right), \quad (1)$$

where y_i denotes the normalized value, α_i is the amount of soliciting funding during the first 24 hours, g_i is the declared pledged goal, so $\frac{\alpha_i}{g_i}$ represents the percentage of funding in early stage to declared pledged goal. There are two reasons for using function $\log_2(\cdot)$ to normalize y_i : one is suppress the large variations between minimum and maximum (Wang et al. 2018), the other is that the original percentage values mostly distribute in a small value range, such as $[0, 0.5]$, and the $\log_2(\cdot)$ is an increasing function with a decreasing slope which can map small value intervals to larger value intervals to make more discriminative prediction.

Next, the studied problem can be defined as follows:

Definition 1 Early Fundraising Performance Estimating. Given the target project g which may be launched at time T_g and the static features \mathbf{X}^{T_g} of all the related projects with the dynamic fundraising sequence S^{T_g} in the market before T_g , our task is to estimate the early performance y_g for target project g .

Static Features. Given N projects in the whole dataset, the static feature $\mathbf{x}_i \in \mathbb{R}^m$ of project i concatenates the vector of basic information published by creators such as project descriptions, perks information, pledged goals, etc. Table 1 lists the detailed information of static features. Specifically, we discretize numerical values by one-hot encoding to categorical type. And, following the example of Jin et al. (2019), text data is converted into numerical vectors by the method of doc2vec (Le and Mikolov 2014). Moreover, we use an

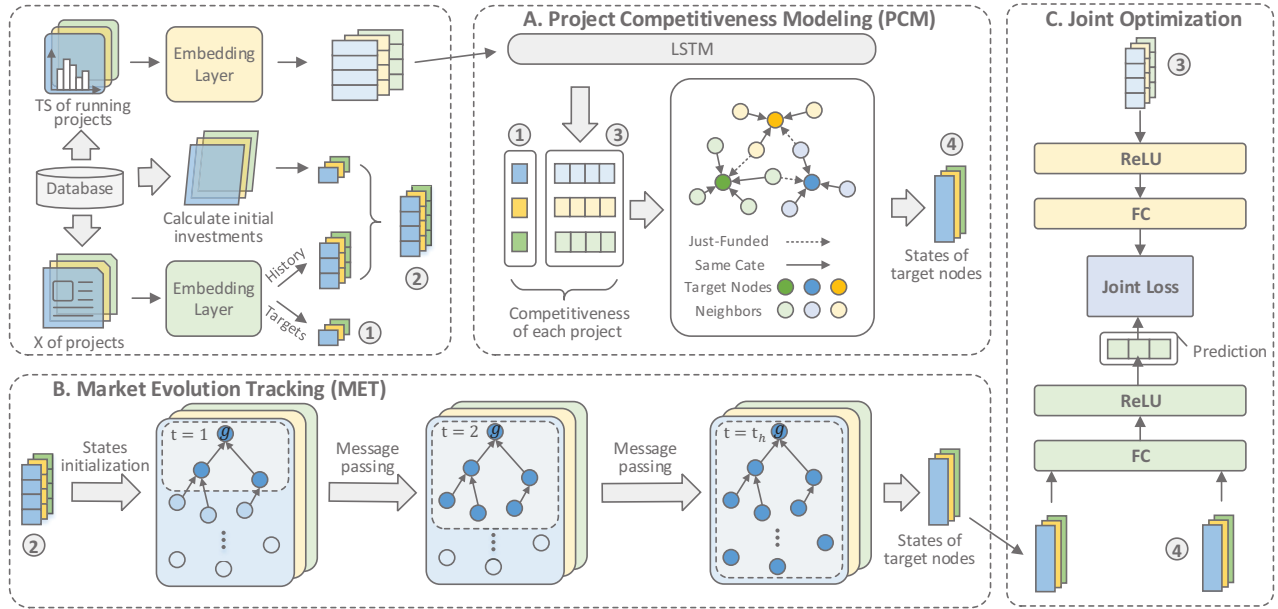


Figure 1: The overview of Graph Market Environment model (GME) : A. Project Competitiveness Modeling (PCM) captures the competitive relationships between the target projects and others. B. Market Evolution Tracking (MET) tracks the market evolution. C. Joint Optimization jointly trains the model with the joint loss of target predicted loss and competitiveness loss.

Table 1: The information of static features.

Features	Type
Project Description	Text
Project Category	Categorical
Creator Type	Categorical
Currency	Categorical
Declared Funding Duration	Numerical
Declared Pledged Goal	Numerical

instance matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$ to present all the existing projects and each row is one project feature.

Market Dynamic Features. Unlike the traditional prediction like success prediction based on the historical time series of the target project, we cannot do that since our task reduce the risk before the target project publish. But we need to model the market evolution, the historical fundraising processes of the projects in the market are important to reflect the project competitiveness and market evolution. For the fundraising process of project i in the market, we can generate a fundraising sequence $S_i = \{\langle v_1, t_1 \rangle, \langle v_2, t_2 \rangle, \dots, \langle v_{|S_i|}, t_{|S_i|} \rangle\}$, where v presents the investment amount of the project at time t .

Project Competitiveness Modeling

Once a project is published, it would face fierce competition in the market. Then, how to model the competitiveness pressure of the target project in the current market environment is particularly essential. Along this line, we propose the Project Competitiveness Modeling (PCM) module to model the current competitive environment. Specifically, we first utilize

sequential modeling to predict the funding status as the competitiveness of published projects in the market. Then, we propose a competitiveness graph to aggregate the competitive influence of published projects.

In detail, we first select the competitive projects. Considering the fund resource in the market is limited, all the funding projects are competitive with each other (Lin, Yin, and Lee 2018). Thus, given the target project g , whose funded time is T_g . For each project i running on the platform at time T_g , we quantify the competitiveness of i as:

$$\mathbf{h}_i^c = \text{LSTM}(\mathbf{TS}_i), i \in \Psi_g, \quad (2)$$

where we use Long Short-Term Memory networks (LSTM) (Hochreiter and Schmidhuber 1997) to predict the funding status of each project in the future, Ψ_g is the set of running projects. $\mathbf{TS}_i = [\xi_0, \xi_1, \dots, \xi_{23}]$ denotes the time series of project i during the past 24 hours before T_g . Since the investment sequence of project i is $S_i = \{\langle v_1, t_1 \rangle, \langle v_2, t_2 \rangle, \dots, \langle v_{|S_i|}, t_{|S_i|} \rangle\}$, so the hourly amount is:

$$\xi_k = \log_2 \left(\sum v_j \right), \quad (3)$$

where $v_j \in S_i, T_g - (k+1) * \Delta \leq t_j < T_g - k * \Delta, k = 0, 1, \dots, 23, \Delta$ represents the time interval of one hour. To build the influence relations from running projects to the target project, we construct all the directed edges $\langle i, g \rangle, i \in \Psi_g$, and apply the embedding vectors \mathbf{h}_i^c as the state of i . Note that, the state of node g is generated from full-connected layer with \mathbf{x}_g , since g does not have fundraising time series.

Considering the computational pressure of the platform, it is not necessary to meet the demands of creators for pre-

cise time. Along this line, we segment one day into six periods according to the general lifestyle of humans (Wu et al. 2017), i.e. “8:00~12:00”, “12:00~14:00”, “14:00~17:00”, “17:00~20:00”, “20:00~24:00” and “0:00~8:00”. We define a *target set* \mathcal{G} which contains unpublished projects whose pre-funded time in the same period of one day, and we train the target projects in \mathcal{G} simultaneously on one graph. Note that, to prevent the leakage of information, that is, the information after T_g is used in advance, the current time-point of the target set \mathcal{G} is defined as $T_{\mathcal{G}} = \min \{T_i \mid i \in \mathcal{G}\}$ when calculating the competitiveness.

However, there is a serious problem that when the running projects set Ψ is large, the algorithm is time-consuming for modeling all the time series by LSTM. To solve this problem, we propose a pruning method. We think that for a project that just starts less than 24 hours, it would compete for one investor’s attention with other projects that are also observed by this investor. And, the investors are most likely to find the newly created projects in the *just-funded* section and the corresponding *category* section. Because on these sections with specific restrictions, our target projects is most likely to appear at the first few pages. According to these, we define the adjacent matrix $\mathbf{A}^G \in \mathbb{R}^{|\mathcal{G}| \times |\Psi_g|}$ as:

$$\mathbf{A}_{\mathcal{M}_i, \mathcal{M}_j}^G = \begin{cases} 1, & \text{if } T_i - T_j \leq 3 \text{ days} \\ 1, & \text{if } C_i = C_j \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where \mathcal{M} is a function which maps the original id of a project to the unique column id in \mathbf{A}^G between 0 and $|\Psi_g|$, $\mathbf{A}_{\mathcal{M}_i, \mathcal{M}_j}^G$ indicates whether j is connected to i , C_i and C_j express the categories of i and j . Simply, the pruning approach not only reduce the number of time series modelled by LSTM but also suppress some noise, and we would design ablation experiment to verify this approach can promote the performance of our model.

Due to the project with strong fundraising ability or it’s content closer to g may have a greater impact on the target project (Lin, Yin, and Lee 2018), we implement the method like Graph Attention Network (GAT) (Veličković et al. 2017) to aggregate neighborhoods information of the target node g as:

$$e_{gi} = \mathbf{V}^T [\mathbf{W}\mathbf{x}_g \parallel \mathbf{W}\mathbf{x}_i], \quad (5)$$

$$\begin{aligned} \alpha_{gi} &= \text{softmax}(e_{gi}) \\ &= \frac{\exp(\text{LeakyReLU}(e_{gi}))}{\sum_{j \in \mathcal{N}_g} \exp(\text{LeakyReLU}(e_{gj}))}, \end{aligned} \quad (6)$$

where \cdot^T represents transposition, \mathcal{N}_g extract from $\mathbf{A}_{\mathcal{M}_g}^G$: (all nodes pointing to node g) are the neighbors of g , and \parallel is the concatenation operation. α_{gi} obtained as the normalized attention coefficients are used to compute a linear combination of the features corresponding to them, to serve as the final output features for node g :

$$\mathbf{H}_g^c = \sum_{i \in \mathcal{N}_g} (\alpha_{gi} \mathbf{W}_h \mathbf{h}_i^c), \quad (7)$$

where α_{gi} is calculated from static content features, but unlike GAT, we use the product of attention weights α_{gi} and

estimated financing state \mathbf{h}_i^c not \mathbf{x}_i . In this way, we can give consideration to two influences include fundraising ability and contents of projects at the same time.

Market Evolution Tracking

After modeling the projects competitiveness, we then consider the market evolution process. Actually, the market environment is dynamic and the investment attention is changed over time (Zhao et al. 2017a). For example, in summer, the investors may pay more attention to the innovation of intelligent fan and in winter, the attention of investors may change to self-heating wear. Along this line, we design a Market Evolution Tracking (MET) module to model the dynamic evolution of the crowdfunding market. More specifically, since the number of published projects in the historical market can reach hundreds in a few days, using chained sequential models like LSTM to model the dynamic environment with such a large step size would lead to the performance degradation. Therefore, we design a propagation tree-structured GNN with message passing to track the evolution of market.

Indeed, market environment is the context of a project in the market. To consider the impact of context on fundraising process, for the project i in market, we should refer to the early fundraising states of other projects in historical market. Therefore, the state of published project j is initialized as $\mathbf{h}_j = [\mathbf{x}_j \parallel r_j]$, r_j is the amount of early fundraising of j ,

$$r_j = \log_2 \left(1 + \sum_{t_i < T_j + n_h \Delta} v_l \right), \quad v_l \in S_j, \quad (8)$$

where n_h denotes 24 hours of one day, and there is a constraint: $T_i - T_j > n_h * \Delta$. Only in this way i can learn the early fundraising performance of published project j by the message propagation through the edge $\langle j, i \rangle$, we call j the *observable node* of i . If the historical data contains past t_h days, the set of published projects is $\Phi_i = \{j \mid j < N, n_h * \Delta < T_i - T_j < n_h * t_h \Delta\}$. Easily the simplest structure to refer to historical market is to calculate the early funding of all nodes in Φ_i firstly, and then create the directed edges from nodes in Φ_i to i . But there is an irrational issue is that we put all the observable projects at the same time level. In an actual environment, there are long-term and short-term effects in forecasting task, the reference values for states with different distances to the pre-startup timepoint should be different. In this paper, we apply the tree-structured graph neural network with message passing to deal with the problem.

Example 1 To illustrate how to represent the different time levels of projects in the market, let us consider the example in Figure 2a. When we considering all the edges between nodes and their observable nodes, there are three edge pairs $\langle a, g \rangle$, $\langle b, g \rangle$, $\langle b, a \rangle$, the length of each edge is greater than n_h hours. If we delete the edge $\langle b, g \rangle$, the depths of a , b are 1 and 2 on the tree $_g$ with root g , which can represent the different time spans to T_g . Moreover, the process of message passing from node b to a to g similar to the step-by-step passing of LSTM.

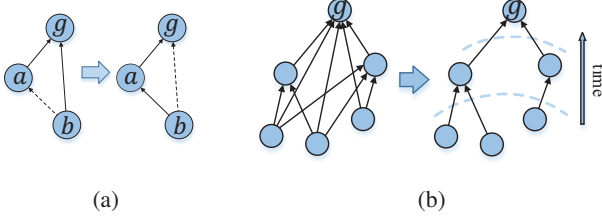


Figure 2: The propagation tree structure with message passing. Node a and b are the observable nodes of target g .

Considering a more complex situation, in Figure 2b, we apply the idea in Example 1 to all the observable nodes of target g . To prevent information redundancy caused by the states of sub-nodes propagate to g through multiple paths, we build the tree structure with root g . This similar idea is appeared in Tree-Lstm to predict the semantic relatedness of sentences in NLP tasks (Tai, Socher, and Manning 2015). Our Propagation Tree Construction algorithm is shown in **Algorithm 1**. In this algorithm, we train the target set \mathcal{G} simultaneously as in the competition module, and each T_g is $\min\{T_i | i \in \mathcal{G}\}$. So, each $tree_g$ has the same structure except for the root (target node). The adjacency matrix Γ of Algorithm 1 represents the tree structure. The depths of nodes represent different time levels. Since the market environment is the context of target projects, and the equal interval sampling is beneficial for long-term effects propagation in contextual history learning (Wu et al. 2017). Therefore, when creating the edges between nodes in our algorithm, for $i \in U^k - U^{k-1}$, we reserve the shortest edge in $\{\langle j, i \rangle | \forall j \in U^{k-1}\}$ to build the tree. In this way, the time intervals between the layers of this tree are as similar as possible, it leads to each chain represented by the path from one leaf to root that approaches to evenly spaced time series.

For the message passing of the tree-structured network, we use propagation model in GG-NNs which unroll the recurrence for fixed number of steps t_h and use back propagation through time to compute gradients. We initialize the node state when the time order $t = 1$:

$$\begin{aligned} \mathbf{h}_i^1 &= [\mathbf{x}_i \parallel \mathbf{r}_i], i \in \Phi, \\ \mathbf{h}_g^1 &= [\mathbf{x}_g \parallel 0], g \in \mathcal{G}. \end{aligned} \quad (9)$$

And in the procedure of message passing, since the target nodes and other nodes update in the same way, we use v denotes the id of each node. The information aggregation in each time step as:

$$\mathbf{a}_v^t = \Gamma_{\mathcal{M}_v}^T \cdot \left[\mathbf{h}_1^{(t-1)T} \dots \mathbf{h}_{|\mathcal{G} \cup \Phi|}^{(t-1)T} \right] + \mathbf{b}. \quad (10)$$

We implement gated recurrent units to guarantee the long-step propagation of information, and the gated module for each node at t as follow:

$$\begin{aligned} \mathbf{z}_v^t &= \sigma(\mathbf{W}_z \mathbf{a}_v^t + \mathbf{U}_z \mathbf{h}_v^{t-1}), \\ \mathbf{r}_v^t &= \sigma(\mathbf{W}_r \mathbf{a}_v^t + \mathbf{U}_r \mathbf{h}_v^{t-1}), \\ \mathbf{h}_v^{t'} &= \tanh(\mathbf{W}_a \mathbf{a}_v^t + \mathbf{U}(\mathbf{r}_v^t \odot \mathbf{h}_v^{t-1})), \\ \mathbf{h}_v^t &= (1 - \mathbf{z}_v^t) \odot \mathbf{h}_v^{t-1} + \mathbf{z}_v^t \odot \mathbf{h}_v^{t'}. \end{aligned} \quad (11)$$

Algorithm 1 Propagation Tree Construction

Input:

- Set of target projects \mathcal{G} ;
- Set of published observable projects Φ ;
- Historical days t_h ;
- Mapping function $\mathcal{M}_i \rightarrow$ unique id in $\{0, 1, \dots, |\Phi \cup \mathcal{G}|\}$, $i < N$;

Output:

Output adjacency matrix Γ

- 1: Set matrix Γ zeros, define infinite matrix **Len**
 - 2: $U^1 \leftarrow \mathcal{G}$
 - 3: **for** $k = 1, \dots, t_h$ **do**
 - 4: $B \leftarrow \Phi \cup \mathcal{G} - U^k$
 - 5: **for** $i \in B$ **do**
 - 6: **for** $j \in U^k$ **do**
 - 7: **if** $n_h * \Delta < T_j - T_i < n_h * 2\Delta$ **then**
 - 8: $\mathbf{Len}_{j,i} \leftarrow T_j - T_i$
 - 9: **if** $k = 1$ **then**
 - 10: $\Gamma_{\mathcal{M}_j, \mathcal{M}_i} \leftarrow 1$ ▷ connect with roots
 - 11: $J \leftarrow \arg \min_{j \in U^k} (\mathbf{Len}_{j,i})$
 - 12: $\Gamma_{\mathcal{M}_j, \mathcal{M}_i} \leftarrow 1$
 - 13: $U^{k+1} \leftarrow U^k \cup i$
 - 14: **return** Γ
-

After $t = t_h$, we get the final state of target g which learns the influence of propagating from entire market,

$$\mathbf{H}_g^e = \mathbf{h}_g^{t_h}, g \in \mathcal{G}. \quad (12)$$

To sum it up, comparing with the chain-like sequential models restricted by time step size, the layers of tree structure is much less, it can model the influence propagation of entire market. Therefore, our propagation tree structure is more suitable for modeling the evolution of entire market.

Joint Optimization

We combine the outputs of the above two modules and design the corresponding loss function. First, we use the fully connected layer to get the output value:

$$\tilde{y}_g = \text{ReLU}(\mathbf{W}_f(\mathbf{H}_g^c + \mathbf{H}_g^e) + b_f), g \in \mathcal{G}, \quad (13)$$

where ReLU keeps prediction value $\tilde{y}_g \geq 0$. Then, the loss function can be defined as:

$$Loss_p = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} |y_g - \tilde{y}_g|. \quad (14)$$

In addition, the MAE loss $Loss_l$ of LSTM of modeling time series in the competitive module can be calculated in the same way as Equation (14) could be jointly trained with $Loss_p$, since the more accurate states fitting of the published nodes leads to the more effective information aggregation. Next, we adopt trade-off to assign the weights between $Loss_l$ and $Loss_p$, Θ represents the parameters of our model, so the objective function as follows:

$$f(\Theta) = \min_{\Theta} (\eta Loss_p + (1 - \eta) Loss_l). \quad (15)$$

Finally, we use Stochastic Gradient Decent(SGD) to update parameters Θ of the model and apply exponential decay to the initial learning rate 0.02.

Experiments

In this section, we first introduce the dataset used in experiments. Then, the experimental setup is expressed in detail. Finally, we focus on evaluating the performances of GME in report of experimental results.

Dataset Description

Specifically, we conduct experiments on Indiegogo dataset (Liu et al. 2017) to verify the effectiveness of our model in the task of early fundraising estimating. There are $N = 14,143$ projects and $|S| = 1,862,097$ investing records in the dataset. And the dataset contains a variety of heterogeneous information about projects as shown in Table 1 mentioned in preliminaries. To test the generalization ability of our model on different scale dataset, we individually sample three sub datasets consisting of different amounts of projects (7K, 10K and 14K) from the complete dataset. In our experiments, we properly preprocess the dataset and divide it into training set and test set in chronological order with a split ratio of 5:1. To build historical market data for each target set, we use moving partition strategy to get inputs of model with a sliding time window above the time order. The sliding step is the segment periods mentioned above, and the length $t_h = \{1, 2, \dots, 7\}$ of time window would be adopted in experiments.

Experimental Setup

Evaluation Protocols. In our study, we evaluated the prediction performance of our approach and the baselines through Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), which are the two standard evaluation metrics for prediction tasks (Wang et al. 2018). Specifically, \mathcal{G} represents a set includes target projects from one time window, for the project $i \in \mathcal{G}$ whose real target value y_i and predicted value \tilde{y}_i described in problem overview, the evaluation protocols denoted as:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{|\mathcal{G}|} \sum_{i=0}^{|\mathcal{G}|} (y_i - \tilde{y}_i)^2}, \\ MAE &= \frac{1}{|\mathcal{G}|} \sum_{i=0}^{|\mathcal{G}|} |y_i - \tilde{y}_i|. \end{aligned} \quad (16)$$

Implementation Details. For numerical features in static features, we convert them into the categorical type (one-hot encoder) by dispersion. Specifically, the dimensions of declared funding duration and declared pledged goal are 4 and 16. For text data, we first use word segmentation to deal with it. Then we remove all punctuations, convert all words to lowercase, and only keep words that appear more than 5 times. On this basis, we implement the doc2vec method to generate 50-dimensional vector for each description. For all models (include baselines) in our experiments, the size of the hidden state in LSTM and the state of nodes in graph are both 50, the dimension of FC layers are also 50, and the dropout after gated module are kept 0.9 empirically. The weight η of trade-off is set as 0.7.

Baselines. Since *Early Fundraising Performance Estimating* is an open issue, many sequential methods in crowdfunding are not applicable to this task. Therefore, we compare the following well-designed models which are selected from the most related research fields, and we fine-tune some models that need to be transferred to adapt to our task.

- **RFR** (*Random Forest Regression*) is a regression-based method that can be used to model hand-crafted features (Liaw, Wiener, and others 2002). We process static features for prediction by the RFR whose max depth is 4.
- **MLP** (*Multiple Layer Perceptron*) is a typical feed-forward neural network trained with back propagation (Zhang et al. 1998). For target project g , we first concatenate the static feature x_g , the average pooling vector of running projects' competitiveness at T_g and the average pooling vector of the initial state of nodes in $tree_g$ together. Then, these concatenated vectors are fed into an MLP with three FC layers for predictions.
- **LSTM** (*Long Short-Term Memory networks*) is a recurrent neural network architecture that can avoid vanishing gradient, which is a useful tool for modeling sequence. We set its step size to 15 in compared experiments (Hochreiter and Schmidhuber 1997).
- **CLSTM** (*Contextual LSTM*) takes the contextual information into account during the message propagation process (Ghosh et al. 2016), we create the contextual vector by discretized temporal variables in hours.
- **DTCN** (*Deep Temporal Context Network*) is designed for sequential image popularity prediction (Wu et al. 2017). This model learns the long-term and short-term effect of context from historical data.
- **DMC** (*Dynamic Market Competition*) can capture the competitiveness of projects in crowdfunding (Lin, Yin, and Lee 2018), we fine-tune the model slightly to make it applicable to our problem scenario.

In addition, we conduct the ablation experiments to verify the effectiveness of the main components of GME.

- **GME-C** is refer to our complete model GME without the market evolution module. We compare it with other models to test the competition module.
- **GME-H** is refer to our complete model GME without the project competition module. We compare it with other models to test the market evolution module.

Experimental Results

Model Comparison. The comparison results on the fundraising prediction task with different models including GME are shown in Table 2. First, we focus on the performances of our complete approach GME, it achieves the best prediction performances on three data size settings with two evaluation metrics, and the minimum MAE is 0.1771, the minimum RMSE is 0.2857. Compared with non-GME models (except GME-H and GME-C), GME outperforms other prediction algorithms with averages of 15.6%, 14.0% and 12.9% relative MAE improvements on three data size settings. Next, we can easily see that the components of GME

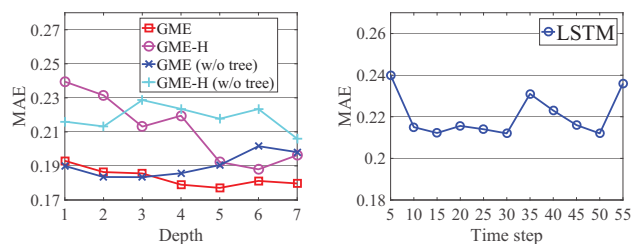
Table 2: Comparisons on the datasets.

Models	Indiegogo-7K		Indiegogo-10K		Indiegogo-14K	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
RFR	0.2175	0.3343	0.2224	0.3293	0.2251	0.3376
MLP	0.2201	0.3377	0.2209	0.3351	0.2193	0.3305
LSTM	0.2134	0.3316	0.2238	0.3324	0.2117	0.3240
CLSTM	0.2112	0.3279	0.2128	0.3266	0.2146	0.3293
DTCN	0.2024	0.3165	0.1978	0.3018	0.2044	0.3142
DMC	0.1970	0.3100	0.1956	0.3117	0.1933	0.2938
GME-C	0.1941	0.3047	0.1974	0.3141	0.2053	0.3078
GME-H	0.1880	0.3086	0.1930	0.3023	0.1914	0.2924
GME	0.1771	0.2940	0.1819	0.2894	0.1836	0.2857

are also able to get superior results. Especially, GME-C surpasses all models except GME on RMSE on Indiegogo-7K. And compared with three sequential models (LSTM, CLSTM, DTCN), the propagation tree structure of GME-H is not complicated but can effectively capture the evolution of the entire market environment. It leads to GEM-H gets more accurate results on MAE and RMSE. Finally, observing the performance of the combination of modules, there is an obvious appearance that GME can achieve higher accuracy than GME-H and GME-C. It proves that combining competition learning with market evolution learning can further boost the performance. To sum up, the effectiveness of market modeling in our approach is verified in the experiments.

Influence of History Length. In this section, first, we visualize the performance variations with the increase of the depth of propagation tree on Indiegogo-7K. The depth is the number of historical days we can observe before the pre-startup time of the target project. What stands out in Figure 3a is that the performances of GME and GME-H become better with deeper depth in the overall trend. However, at the tail of these curves, GME and GME-H show slight upward trends. These tests reveal that lengthening history length without limit does not necessarily lead to better experimental results, since it probably to bring more noise. In our experiment, the most appropriate depth for GME is 5, it achieves the best result 0.1771 on MAE. Second, compare with the classic sequential model, as shown in Figure 3b, we find that LSTM does not improve significantly with the increase of step size (the number of historical projects). Finally, to verify that the tree structure with time levels is more efficient than directly aggregating all historical nodes by GNN, we define “w/o tree” to denote not using tree structure. And we give the results of corresponding architectures GME (w/o tree) and GME-H (w/o tree). Obviously, GME (w/o tree) and GME-H (w/o tree) do not perform well, the greatest scores of them are 0.1834 and 0.2060. And they do not improve by the increase of the history length.

Influence of Pruning Approaches. In the competitiveness module, we propose the pruning approach that retains only two physical relationships: projects belong to the same category and on the just-funded section. The experimental results of pruning approaches are shown in Figure 4, where all methods are based on GME. *Unpruned* retains all edges, *Only Cate* retains the edges between the target project and



(a) MAE changes with depth. (b) Timestep influence LSTM.

Figure 3: Influence of history length.

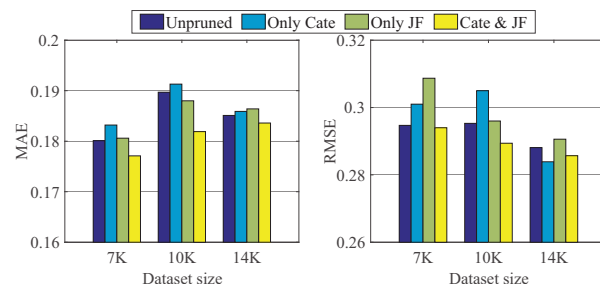


Figure 4: Influence of pruning approaches.

other nodes with the same category, *Only JF* deletes all connections except the projects on the just-funded section at time T_g . Moreover, *Cate & JF* denotes the union of *Only Cate* and *Only JF*. It can be seen in Figure 4, pruning with only one physical relationship would reduce the accuracy of the model slightly, since it loses a considerable amount of information. But the significant result is that *Cate & JF* lead to a more accurate prediction than *Unpruned* on our dataset. It proves that our pruning method is capable of avoiding more useless input for the influence aggregation.

Conclusion

In this paper, we studied the issue of estimating the early fundraising performance of the unpublished project on online crowdfunding platforms. Along this line, we proposed a Graph-based Market Environment model (GME) to explore the market environment and estimate the early performance of the target project. Specifically, we discriminatively modeled the market competition and market evolution by designing two graph-based neural network architectures and incorporating them into the joint optimization stage. We conducted extensive experiments on the real-world crowdfunding data collected from Indiegogo.com. The experimental results clearly validated the effectiveness of GME.

In the future, we will apply our models to other scenarios and applications leading to a more general framework.

Acknowledgements

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2016YFB1000904), and the National Natural Science Foundation of China (Grants No. U1605251, 61922073,

61672483). Qi Liu acknowledges the support of the Young Elite Scientist Sponsorship Program of CAST and the Youth Innovation Promotion Association of CAS (No. 2014299).

References

- Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34(4):18–42.
- Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.
- Cucurull, G.; Taslakian, P.; and Vazquez, D. 2019. Context-aware visual compatibility prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12617–12626.
- Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.
- Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; and Chua, T.-S. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37(2):27.
- Ghosh, S.; Vinyals, O.; Strope, B.; Roy, S.; Dean, T.; and Heck, L. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.
- Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1263–1272. JMLR. org.
- Gori, M.; Monfardini, G.; and Scarselli, F. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, 729–734. IEEE.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Janku, J., and Kucerova, Z. 2018. Successful crowdfunding campaigns: The role of project specifics, competition and founders’ experience. *Czech Journal of Economics and Finance (Finance a uver)* 68(4):351–373.
- Jin, B.; Zhao, H.; Chen, E.; Liu, Q.; and Ge, Y. 2019. Estimating the days to success of campaigns in crowdfunding: A deep survival perspective. In *Thirty-Third AAAI Conference on Artificial Intelligence*.
- Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, 1188–1196.
- Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.
- Li, Y.; Rakesh, V.; and Reddy, C. K. 2016. Project success prediction in crowdfunding environments. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 247–256. ACM.
- Liaw, A.; Wiener, M.; et al. 2002. Classification and regression by randomforest. *R news* 2(3):18–22.
- Lin, Y.; Yin, P.; and Lee, W.-C. 2018. Modeling dynamic competition on crowdfunding markets. In *Proceedings of the 2018 World Wide Web Conference*, 1815–1824.
- Liu, Q.; Wang, G.; Zhao, H.; Liu, C.; Xu, T.; and Chen, E. 2017. Enhancing campaign design in crowdfunding: A product supply optimization perspective. In *IJCAI*, 695–702.
- Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wang, W.; Zhang, W.; Wang, J.; Yan, J.; and Zha, H. 2018. Learning sequential correlation for user generated textual content popularity prediction. In *IJCAI*, 1625–1631.
- Wu, B.; Cheng, W.-H.; Zhang, Y.; Huang, Q.; Li, J.; and Mei, T. 2017. Sequential prediction of social media popularity with deep temporal context networks. *arXiv preprint arXiv:1712.04443*.
- Zhang, Z.; Lyons, M.; Schuster, M.; and Akamatsu, S. 1998. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings Third IEEE International Conference on Automatic face and gesture recognition*, 454–459. IEEE.
- Zhang, L.; Zhang, X.; Cheng, F.; Sun, X.; and Zhao, H. 2019. Personalized recommendation for crowdfunding platform: A multi-objective approach. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, 3316–3324. IEEE.
- Zhao, H.; Liu, Q.; Zhu, H.; Ge, Y.; Chen, E.; Zhu, Y.; and Du, J. 2017a. A sequential approach to market state modeling and analysis in online p2p lending. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48(1):21–33.
- Zhao, H.; Zhang, H.; Ge, Y.; Liu, Q.; Chen, E.; Li, H.; and Wu, L. 2017b. Tracking the dynamics in crowdfunding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 625–634. ACM.
- Zhao, H.; Jin, B.; Liu, Q.; Ge, Y.; Chen, E.; Zhang, X.; and Xu, T. 2019. Voice of charity: Prospecting the donation recurrence & donor retention in crowdfunding. *IEEE Transactions on Knowledge and Data Engineering*.
- Zhou, D.; Huang, J.; and Schölkopf, B. 2007. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, 1601–1608.