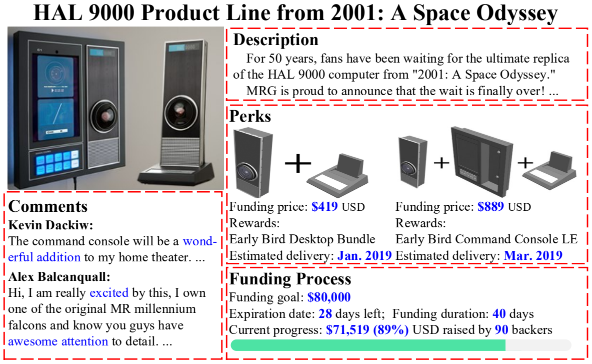
Estimating the Days to Success of Campaigns in Crowdfunding: A Deep Survival Perspective

Binbin Jin,[†] Hongke Zhao,[†] Enhong Chen,[†§*] Qi Liu,[†§] Yong Ge[‡]
[†]Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China
[§]School of Data Science, University of Science and Technology of China
[‡]Eller College of Management, University of Arizona
{bb0725, zhhk}@mail.ustc.edu.cn, {cheneh, qiliuql}@ustc.edu.cn, yongge@email.arizona.edu

Abstract

Crowdfunding is an emerging mechanism for entrepreneurs or individuals to solicit funding from the public for their creative ideas. However, in these platforms, quite a large proportion of campaigns (projects) fail to raise enough money of backers' supports by the declared expiration date. Actually, it is very urgent to predict the exact success time of campaigns. But this problem has not been well explored due to a series of domain and technical challenges. In this paper, we notice the implicit factor of distribution of backing behaviors has a positive impact on estimating the success time of the campaign. Therefore, we present a focused study on predicting two specific tasks, i.e., *backing distribution prediction* and *success time prediction* of campaigns. Specifically, we propose a *S*eq2seq based model with *M*ulti-facet *P*riors (*SMP*), which can integrate heterogeneous features to jointly model the backing distribution and success time. Additionally, to keep the change of backing distributions more smooth as the backing behaviors increases, we develop a linear evolutionary prior for backing distribution prediction. Furthermore, due to high failure rate, the success time of most campaigns is unobservable. We model this censoring phenomenon from the survival analysis perspective and also develop a non-increasing prior and a partial prior for success time prediction. Finally, we conduct extensive experiments on a real-world dataset from Indiegogo. Experimental results clearly validate the effectiveness of *SMP*.



HAL 9000 Product Line from 2001: A Space Odyssey

Description
For 50 years, fans have been waiting for the ultimate replica of the HAL 9000 computer from "2001: A Space Odyssey." MRG is proud to announce that the wait is finally over! ...

Perks

Funding price: **$419** USD
Rewards:
Early Bird Desktop Bundle
Estimated delivery: **Jan. 2019**

Funding price: **$889** USD
Rewards:
Early Bird Command Console LE
Estimated delivery: **Mar. 2019**

Comments
Kevin Dackiw:
The command console will be a wonderful addition to my home theater. ...

Alex Balcanquall:
Hi, I am really excited by this, I own one of the original MR millennium falcons and know you guys have awesome attention to detail. ...

Funding Process
Funding goal: **$80,000**
Expiration date: **28** days left; Funding duration: **40** days
Current progress: **$71,519 (89%)** USD raised by **90** backers

Figure 1: A campaign example from Indiegogo.

Introduction

In recent years, crowdfunding has gradually become a popular way for entrepreneurs and individuals to solicit funding from the public for their creative ideas. Among various types of crowdfunding platforms, reward-based ones (e.g., Indiegogo) are considered as the most popular (Bannerman 2013). Nowadays, thousands of entrepreneurs and individuals create *campaigns* for funding and demonstrate their practical productions on these platforms every day. They expect to attract enough backers to meet their financial demands so that they can witness the birth of their ideal productions.

For example, Figure 1 shows an example of the campaign from Indiegogo. Generally, a campaign contains a detailed description, several perks with different funding prices and rewards (often vowing future productions), and its funding process with a pledged goal, expiration date, etc. Anyone interested in the project can choose any perk to support and expect the corresponding rewards in return. Among these heterogeneous features, it is challenging to comprehensively analyze and integrate them to predict the success time of campaigns (i.e., *success time prediction*). Fortunately, Indiegogo also reveals the backers' comments which generally affect the following backing behaviors (i.e., funding the campaign or not; selecting specific perks) and the speed of funding process. For example, in Figure 1, the positive comments (highlighted in blue) encourage this campaign to quickly raise 89% of its funding goal in a short term. Actually, with the help of comments, creators can more accurately estimate the backing distribution in the future (i.e., the ratio of backing behaviors in different price ranges or perks), which is related to the backers' preference or perk popularity. Intuitively, more backing behaviors with higher funding prices lead to rapid fundraising. Therefore, *backing distribution prediction* is significant for creators to assess and update the campaign design in time. Additionally, due to the limited backing behaviors on each campaign (around 3 backing behaviors per day on average), the change of backing distributions is slow and smooth so that creators could further estimate the success time of campaigns based on the predicted backing distribution. However, how to construct the predictions of backing distribution and success time remains to be open issues.

[*]Corresponding Author.

In this area, some researchers focused on predicting the success rate of campaigns (Lu et al. 2014; Zhang et al. 2016). However, the problems of backing distribution prediction and success time prediction of campaigns are more difficult and complicated. Indeed, our focus is more challenging and dominant to the prior studies for the following reasons. First, the two prediction tasks are influenced by various factors, such as campaign description, perks information, comments. Analyzing these heterogeneous features and integrating them for accurate prediction are nontrivial. Second, the backers' feedback (e.g., comments) can greatly affect the following backing behaviors. Therefore, it is necessary to model the relations between comments and backing distributions. In the meantime, keeping the change of backing distributions smooth is quite difficult. Third, we notice the implicit factor of backing distribution has a strong impact on the speed of fundraising. In other words, it is significant to take the predicted backing distribution into account for our senior goal, i.e., predicting the success time of campaigns. Last but not the least, around 60% of campaigns failed in fully funding (Liu et al. 2017). As a consequence, their success time is unobservable, which is referred to as *censoring phenomenon* (Li, Rakesh, and Reddy 2016). This censored data imposes a significant challenge to our study.

To address the challenges above, we present a focused study on two joint tasks, i.e., backing distribution prediction and success time prediction. Specifically, we propose a *S*eq2seq based model with *M*ulti-facet *P*riors (*SMP*), which can integrate heterogeneous features to jointly model the two prediction tasks. Particularly, we apply an encoder to track and predict the dynamics of backing distributions by integrating both static features (e.g., campaign description, perks information) and dynamic features (e.g., comments). Then, the final prediction together with the representation of the encoder is fed into the decoder which predicts the success time of the campaign. Additionally, to keep the change of backing distributions smooth, we develop a linear evolutionary prior for the backing distribution task. Furthermore, we model the censoring phenomenon from the survival analysis perspective and further develop a non-increasing prior and a partial prior for the success time prediction task. Finally, we conduct extensive experiments on a real-world dataset collected from Indiegogo. Experimental results clearly validate the effectiveness of *SMP*.

Related Work

In our study, the related work can be grouped into two categories: crowdfunding and survival analysis.

Crowdfunding

As an emerging research area, the development of crowdfunding has attracted much research attention in academics. Although there are some studies focusing on recommender systems (An, Quercia, and Crowcroft 2014; Rakesh, Lee, and Reddy 2016) or production supply optimization (Liu et al. 2017), most researchers pay much attention to predicting whether the campaign succeed or not (Li, Rakesh, and Reddy 2016; Zhang et al. 2016). Among them, some factors are turned out to be valuable (Lu et al. 2014; Mollick 2014). For instance, Mollick (2014) showed some campaigns which could reflect the underlying culture would be associated with funding success. Lu et al. (2014) found the social media could help crowdfunding projects succeed. There was a strong correlation between a project's early promotional activities and the final outcomes. Additionally, project creators' backing history (Zvilichovsky, Inbar, and Barzilay 2013) and project description (Zhou et al. 2018) were also effective for improving the prediction accuracy.

Recently, instead of predicting the final results of campaigns, some researchers try to solve more difficult but valuable problems such as tracking the dynamics for campaigns in their complete funding durations (Zhao et al. 2017; 2018). However, these studies mainly explore the influence of explicit factors while ignoring the positive effect of implicit ones, such as the backing distribution, on the prediction tasks. Furthermore, few studies have settled the success time prediction problem which can help creators assess and update the campaigns in time. To the best of our knowledge, this is the first attempt to predict backing distribution, and we also exploit that for success time prediction in crowdfunding.

Survival Analysis

Survival analysis aims to predict the occurrence of specific events of interest at a future time, such as the success of campaigns in our study. Actually, this problem has been studied in various application fields such as student dropouts (Ameri et al. 2016), health care (Li et al. 2016a) and check-in location prediction (Yang, Cai, and Reddy 2018). In this domain, one of the unique challenges is the presence of censored instances which do not experience the event of interest. Since the exact time of these censored instances is not observable, traditional regression models are not appropriate. To deal with this problem, survival analysis techniques have been developed to make use of such censored instances. In literature, survival analysis is a subfield of statistics and most related methods can be divided into two categories: semi-parametric and parametric methods. Cox proportional hazard regression model (Cox 1992) is one of the earliest and most widely used semi-parametric methods. To efficiently handle high dimensional data, some regularization methods are proposed such as LASSO-Cox (Tibshirani 1997), Elastic-Net Cox (Yang and Zou 2013). Parametric methods provide an important alternative to the Cox-based models. They assume the event occurrence time follows a particular distribution such as Weibull, Logistic or Log-normal (Li et al. 2016a; Li, Rakesh, and Reddy 2016).

Recently, more advanced machine learning methods, such as ensemble learning (Hothorn et al. 2004), transfer learning (Li et al. 2016b), multi-task learning (Li et al. 2016a) and active learning (Vinzamuri, Li, and Reddy 2014), have been developed to make the prediction from censored data. In addition, there are a few studies combining deep learning techniques with survival analysis (Jing and Smola 2017; Yang, Cai, and Reddy 2018). In this paper, we employ the seq2seq technique on survival analysis and further propose *SMP*, which is the first attempt in this domain.

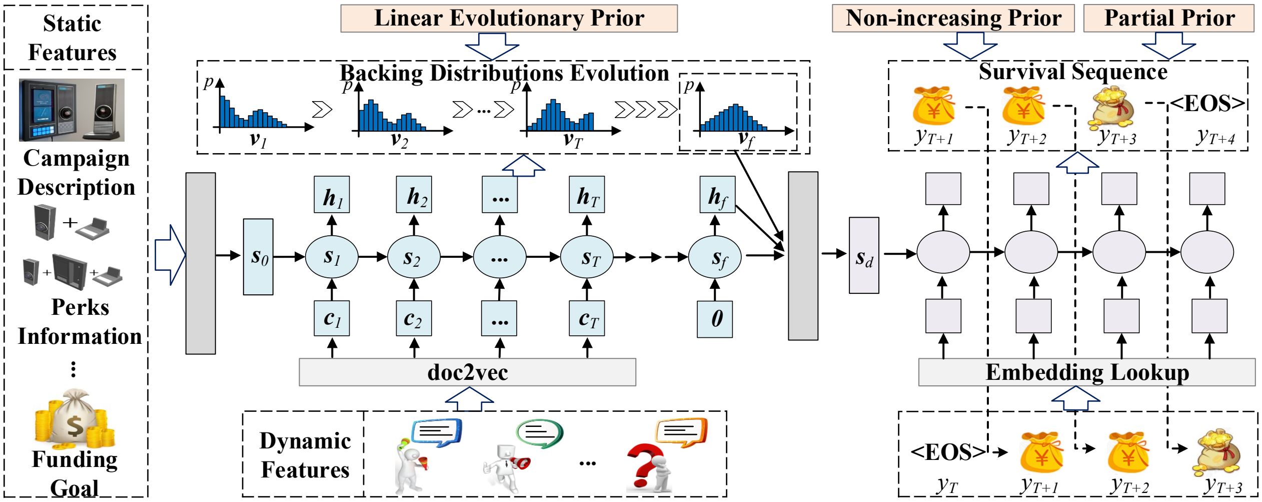


Figure 2: The architecture of *SMP*.

Table 1: The information of features.

Level	Features	Type
Static	campaign description	text
	perks description	text
	campaign's category	categorical
	creator's type	categorical
	declared funding duration	numerical
	declared pledged goal	numerical
	number of perks	numerical
	max/min/avg price of perks	numerical
	max/min/avg delivery term	numerical
Dynamic	comments	text

Methodology

In this section, we first formally define the studied problem. Then we introduce the technical details of *SMP*, i.e., the input preprocessing, the architecture of *SMP*, the multi-facet priors and the model training method.

Problem Overview

For the campaign i, it can be represented by a tuple $(\boldsymbol{X}^i, \boldsymbol{C}^i, \boldsymbol{V}^i, d^i, \delta^i)$. Specifically, $\boldsymbol{X}^i$ and $\boldsymbol{C}^i$ are static features and dynamic features, respectively. The former consists of information created by the creator such as campaign description, perks information, a pledged goal, etc, while the latter contains a sequence of comments ($\boldsymbol{C}^i = \{\boldsymbol{c}_1^i, \boldsymbol{c}_2^i, ..., \boldsymbol{c}_T^i\}$ where T is the number of days since the creation of the campaign i, and all comments posted on day t are aggregated into $\boldsymbol{c}_t^i$). $\boldsymbol{V}^i = \{\boldsymbol{v}_1^i, \boldsymbol{v}_2^i, ..., \boldsymbol{v}_T^i\} \cup \{\boldsymbol{v}_f^i\}$ consists of a sequence of backing distributions from the creation of the campaign (i.e., $\boldsymbol{v}_1^i$) to the observable day T (i.e., $\boldsymbol{v}_T^i$), and a final backing distribution (i.e., $\boldsymbol{v}_f^i$) when the campaign reaches its pledged goal or expiration date. Note that we use a vector $\boldsymbol{v}_t^i$ to denote a probability distribution of all backing behaviors before t-th day so that each element in $\boldsymbol{v}_t^i$ is non-negative and $\|\boldsymbol{v}_t^i\|_1 = 1$. δ^i is the censoring indicator where $\delta^i = 1$ denotes the campaign i has reached its pledged goal; $\delta^i = 0$ otherwise. d^i denotes the observed time which is defined as:

$$d^i = \begin{cases} O^i, & \text{if the campaign } i \text{ is successful } (\delta^i = 1), \\ U^i, & \text{otherwise } (\delta^i = 0), \end{cases} \tag{1}$$

where O^i is the number of days that the campaign i takes to succeed; U^i is the number of days when the campaign i has reached its expiration date. For censored instances (i.e., $\delta^i = 0$), O^i is unobservable. Then, the studied problem is defined as follows.

Problem Formulation. For the campaign i, given its static and dynamic features (i.e., $\boldsymbol{X}^i, \boldsymbol{C}^i$), our goal is to estimate the final backing distribution $\widetilde{\boldsymbol{v}}_f^i$ and use it as extra features to further estimate its exact time to success O^i.

Input Preprocessing

For each campaign, time to success is influenced by various features in the form of categorical, text and numerical data. Table 1 lists the detailed information of features. For consistency, we carefully preprocess them. Specifically, for categorical data, we adopt the one-hot encoding (Liu et al. 2017). For text data, it is converted into numerical vectors by the doc2vec method (Le and Mikolov 2014). For numerical data, it is normalized by Z-score transformation (Mendenhall, Sincich, and Boudreau 2016).

Seq2seq Model on Two Prediction Tasks

In our study, we first integrate heterogeneous features to track and predict the dynamics of backing distributions from the creation of the campaign to the observable day T. Then, based on the prediction, we estimate the exact success time of the campaign. To model the relations between our two tasks, we design a seq2seq model for them. Specifically, it aims to model the conditional probability $p(\widetilde{\boldsymbol{V}}^i, \widetilde{d}^i|\boldsymbol{X}^i, \boldsymbol{C}^i)$, which can be divided into two components: (a) an encoder which tracks and predicts the dynamics of backing distributions (i.e., $\widetilde{\boldsymbol{V}}^i$) and computes a representation $\boldsymbol{s}_d^i$ for the campaign i; (b) a decoder which predicts the success time of the campaign $\widetilde{d}^i$. Hence, the conditional probability $p(\widetilde{\boldsymbol{V}}^i, \widetilde{d}^i|\boldsymbol{X}^i, \boldsymbol{C}^i)$ can be decomposed as:

$$p(\widetilde{\boldsymbol{V}}^i, \widetilde{d}^i|\boldsymbol{X}^i, \boldsymbol{C}^i) = p(\widetilde{\boldsymbol{V}}^i|\boldsymbol{X}^i, \boldsymbol{C}^i)p(\widetilde{d}^i|\boldsymbol{s}_d^i). \tag{2}$$

For the encoder, it mainly focuses on the backing distribution prediction task. We assume whether a backer will support a campaign or not depends on the following process. First, she scans the campaign's static information such as campaign description, perks information, etc. Then, when she shows a little interest but has not made her decision, she will seek others' advice so that comments on the campaign play an important role in this process. After that, the backer will decide how much to support. According to this assumption, we can easily track the dynamics of backing distributions. Since the backing distribution is influenced by the daily comments, a natural choice to model this change is Long Short-Term Memory (LSTM) (Graves, Mohamed, and Hinton 2013). Given the static features $\boldsymbol{X}^i$, we first use a single hidden layer to get the overall representation of the campaign $\boldsymbol{s}_0^i$ which is used to initialize the memory cell of LSTM. Then, given the dynamic features on t-th day $\boldsymbol{c}_t^i$ ($t \in \{1, 2, ..., T\}$), $(t-1)$-th memory cell $\boldsymbol{s}_{t-1}^i$ and hidden state $\boldsymbol{h}_{t-1}^i$, the t-th hidden state $\boldsymbol{h}_t^i$ is computed as:

$$\boldsymbol{h}_t^i = LSTM(\boldsymbol{c}_t^i, \boldsymbol{s}_{t-1}^i, \boldsymbol{h}_{t-1}^i). \tag{3}$$

In order to estimate the final backing distribution $\widetilde{\boldsymbol{v}}_f^i$, we also let LSTM output an additional hidden state $\boldsymbol{h}_f^i$ where the corresponding input is a zero vector. After getting the sequence of hidden states, we can estimate the probability distribution of all backing behaviors before t-th day (i.e., $\boldsymbol{v}_t^i$) and the final backing distribution when the campaign reaches its pledged goal or expiration date (i.e., $\boldsymbol{v}_f^i$) as:

$$p(\widetilde{\boldsymbol{v}}_t^i|\boldsymbol{s}_{t-1}^i, \boldsymbol{h}_{t-1}^i, \boldsymbol{c}_t^i) = softmax(\boldsymbol{W}\boldsymbol{h}_t^i + \boldsymbol{b}), \tag{4}$$

$$p(\widetilde{\boldsymbol{v}}_f^i|\boldsymbol{s}_T^i, \boldsymbol{h}_T^i) = softmax(\boldsymbol{W}\boldsymbol{h}_f^i + \boldsymbol{b}), \tag{5}$$

where $t \in \{1, 2, ..., T\}$; $\{\boldsymbol{W}, \boldsymbol{b}\}$ are a weight matrix and a bias vector, respectively. Finally, we define the logarithm of conditional probability of the encoder (i.e., $\log p(\widetilde{\boldsymbol{V}}^i|\boldsymbol{X}^i, \boldsymbol{C}^i)$) as:

$$\mathcal{L}_{en}^i = \sum_{t=1}^{T} \boldsymbol{v}_t^i \log p(\widetilde{\boldsymbol{v}}_t^i|\boldsymbol{s}_{t-1}^i, \boldsymbol{h}_{t-1}^i, \boldsymbol{c}_t^i) + \boldsymbol{v}_f^i \log p(\widetilde{\boldsymbol{v}}_f^i|\boldsymbol{s}_T^i, \boldsymbol{h}_T^i). \tag{6}$$

For the decoder, it aims to predict the days to success of the campaign O^i. However, due to the censoring phenomenon, the success time of censored instances (more than 60% of campaigns) is unobservable. To tackle this problem, we first design a survival sequence for campaign i (i.e., $Y^i = \{y_T^i, y_{T+1}^i, ..., y_{d^i}^i, y_{d^i+1}^i\}$) and then propose a non-increasing prior and a partial prior which will be illustrated in the next subsection. For the survival sequence, each variable y_*^i consists of three different states (i.e., $\{1, 0, \text{<EOS>}\}$). In particular, $y_*^i = 1$ means the campaign i has still been raising money; $y_*^i = 0$ means it has reached the pledged goal; <EOS> is a special symbol which is required to mark the *E*nd *O*f the *S*equence (i.e., $y_T^i, y_{d^i+1}^i =$ <EOS>). Note that for the successful campaigns, their survival sequences are $\{\text{<EOS>}, 1, 1, ..., 1, 0, \text{<EOS>}\}$, while for the others, their survival sequences are similar but without 0 (i.e., $\{\text{<EOS>}, 1, 1, ..., 1, \text{<EOS>}\}$). Then, the decoder generates one state y_*^i at a time. Specifically, we first randomize a vector for each state, which can be tuned during the training stage, and then employ another LSTM to automatically generate the target sequence. Formally, given the last hidden state and memory cell of the encoder (i.e., $\boldsymbol{h}_f^i, \boldsymbol{s}_f^i$) and the estimated backing distribution (i.e., $\widetilde{\boldsymbol{v}}_f^i$), we first initialize the memory cell by a vector $\boldsymbol{s}_d^i$ which is computed by the concatenation of $\boldsymbol{h}_f^i$, $\boldsymbol{s}_f^i$ and $\widetilde{\boldsymbol{v}}_f^i$ followed by a single hidden layer. Then, the logarithm of conditional probability of the decoder (i.e., $\log p(\widetilde{d}^i|\boldsymbol{s}_d^i)$) is defined as:

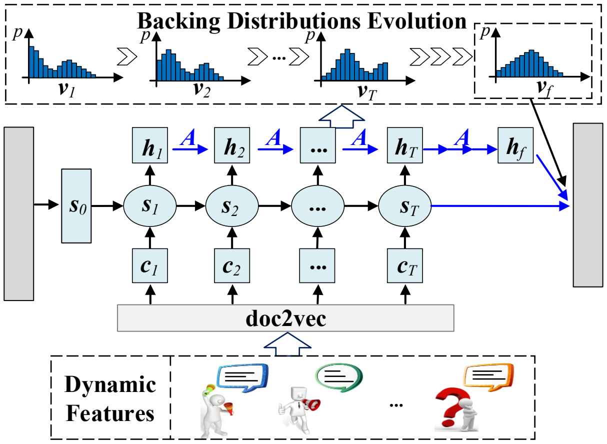


Figure 3: The encoder with linear evolutionary prior.

$$\mathcal{L}_{de}^i = \sum_{t=T+1}^{d^i+1} \log p(y_t^i|y_{t-1}^i, \boldsymbol{s}_d^i). \tag{7}$$

After getting the target sequence, we can make a prediction that the campaign i will reach its pledged goal on day $\widetilde{d}^i$ where the variable $y_{\widetilde{d}^i}^i \in \{0, \text{<EOS>}\}$ and $y_k^i = 1$ $(k < \widetilde{d}^i)$.

Multi-facet Priors

In this subsection, we propose three innovative priors for two prediction tasks, i.e., a linear evolutionary prior, a non-increasing prior and a partial prior.

Linear evolutionary prior. Due to the limited backing behaviors on each campaign (around 3 backing behaviors per day on average), we assume the change of backing distributions is slow and smooth. In other words, the hidden state of the encoder from day t to $t+1$ can be transferred by a matrix $\boldsymbol{A}$. As shown in Figure 3, given the hidden state $\boldsymbol{h}_t^i$ ($t \in \{1, 2, ..., T-1\}$) of the encoder, we propose the following conditional distribution to guarantee our assumption:

$$\boldsymbol{h}_{t+1}^i|\boldsymbol{h}_t^i \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{h}_t^i, \boldsymbol{\Sigma}), \tag{8}$$

where $\boldsymbol{\Sigma}$ is the conditional covariance shared by all $\boldsymbol{h}_t^i$. Under the assumption, we also assume the hidden state $\boldsymbol{h}_f^i$ can be computed by $\boldsymbol{h}_f^i = \boldsymbol{A}\boldsymbol{h}_T^i$, which is used to predict the final backing distribution in Eq. (5), rather than generating it through LSTM. Therefore, we can formulate the objective function for this prior as:

$$\mathcal{L}_{c1}^i = \sum_{t=1}^{T-1} \|\boldsymbol{h}_{t+1}^i - \boldsymbol{A}\boldsymbol{h}_t^i\|_2^2 + \|\boldsymbol{h}_f^i - \boldsymbol{A}\boldsymbol{h}_T^i\|_2^2, \quad (9)$$

where $\|\cdot\|_2$ is the L_2 Norm. We will further validate the capacity of matrix $\boldsymbol{A}$ in experiments.

Non-increasing prior. In the decoder, to solve the censoring phenomenon, we design a survival sequence for each campaign. However, most variables in these survival sequences are the same (i.e., $y_*^i = 1$) so that it is difficult for the decoder to predict the exact time when the campaign will succeed or failed (i.e., $y_*^i = 0, \text{<EOS>}$). To tackle this problem, when generating the conditional probability $p(y_t^i = 1|y_{t-1}^i, \boldsymbol{s}_d^i)$ in t-th day, we consider it as the distance from the pledged goal. Intuitively, as time goes on, more backers join in the campaign so that the distance from the pledged goal will become smaller. Therefore, it makes sense that the conditional probability of the survival sequence is non-increasing (i.e., $p(y_t^i = 1|y_{t-1}^i, \boldsymbol{s}_d^i) \geq p(y_{t+1}^i = 1|y_t^i, \boldsymbol{s}_d^i), t \in \{T+1, T+2, ..., d^i - 1\}$). Inspired by the learning to rank strategy (Severyn and Moschitti 2015), we adopt a large margin objective to model this prior. Formally, given the predicted survival sequence $\{p(y_t^i = 1|y_{t-1}^i, \boldsymbol{s}_d^i)\}$, we formulate the objective function for this prior as:

$$\begin{aligned}\mathcal{L}_{c2}^i = \sum_{t=T+1}^{d^i-1} max(0, m + g^i(t+1) - g^i(t)), \\ g^i(t) = p(y_t^i = 1|y_{t-1}^i, \boldsymbol{s}_d^i),\end{aligned} \quad (10)$$

where m is the margin which is a hyper-parameter and ranges from 0 to 1.

Partial prior. With the help of the non-increasing prior, the decoder is capable of predicting the final result of the campaign (i.e., successful or not). However, how to fit the success time of censored instances is still an open problem. To this end, we propose a partial prior which can help tackle this problem even though the success time of censored instances is unobservable. Specifically, given a censored instance (campaign i) whose observed time is d^i, we first find a campaign set $\mathcal{J}$ where each of them has reached its pledged goal by day d^i. Then, compared with the distance of campaigns in $\mathcal{J}$ from their pledged goals, the distance of the campaign i is farther from its pledged goal. Therefore, we have the following large margin objective function:

$$\mathcal{L}_{c3}^i = \sum_{j \in \mathcal{J}} max(0, m + g^j(T+1) - g^i(T+1)), \quad (11)$$

where $g^i(T+1)$ is the distance of campaign i from its pledged goal on the second day of the observable day T.

Model Training

Combining the seq2seq model with our proposed multi-facet priors, in this subsection, we will integrate them into one complete objective function. Given N campaigns, we first maximize the log-likelihood functions defined in the seq2seq model (i.e., $\mathcal{L}_{en}$, $\mathcal{L}_{de}$). Then, we incorporate our priors and minimize them (i.e., $\mathcal{L}_{c1}$, $\mathcal{L}_{c2}$, $\mathcal{L}_{c3}$). Note that the partial prior is only effective for the censored instances. Therefore, we have the whole objective function as follows:

$$\mathcal{L} = \min_{\Theta} \sum_{i=1}^{N} (-\mathcal{L}_{en}^i - \mathcal{L}_{de}^i + \lambda_1 \mathcal{L}_{c1}^i + \lambda_2 \mathcal{L}_{c2}^i + \lambda_3 I[\delta^i = 0]\mathcal{L}_{c3}^i), \quad (12)$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are hyper-parameters used to balance the effects of different priors; $I[x]$ is an indicator function that equals 1 if x is true and equals 0 otherwise; Θ denotes all parameters in *SMP* and we use *Stochastic Gradient Decent (SGD)* to update them with the back propagation through time algorithm.

Experiments

In this section, we first introduce the dataset collected from Indiegogo. Then, we illustrate the experimental setup in detail. Finally, we report the experimental results of data exploration and performances on two tasks.

Dataset Description

We collect a dataset from Indiegogo including 14,143 launched campaigns with their static and dynamic information described in Table 1. Among them, there are total 98,923 perks and 240,922 comments. In addition, we also collect funding information with 1,862,097 backing behaviors. For each campaign, its funding process begins with its creation and continues until its success or declared funding duration. To observe how the model behaves at different stages, we split the funding process of each campaign into two parts at 30%, 50% or 70%. The early portion is fed into the encoder and the other is fed into the decoder.

Experimental Setup

Parameter Setting. For the encoder, from the description in Input Preprocessing, all textual inputs (i.e., the description of campaigns, perks and comments) are converted into numerical vectors by the doc2vec method. We aggregate all description and use public doc2vec lib (Gensim) to assign each text with a 50-dimensional vector. With respect to the backing distributions $\boldsymbol{V}^i$, we count the prices of all perks and empirically split them into 10 different price range (i.e., $\boldsymbol{v}_*^i \in \mathbb{R}^{10}$; the price range of the k-th element is from $k * 20$ to $(k+1) * 20$ dollars and its corresponding value (i.e., v_{*k}^i) denotes the ratio of all backings in this range). For the decoder, there are three different states in survival sequences and each of them will be randomly assigned with a 20-dimensional vector which is tuned during the training stage. In addition, the sizes of the hidden state and memory cell in the encoder (decoder) are empirically set as 50 (20).

For all parameters Θ in our model, we follow (Huang et al. 2017) and randomly initialize them with a uniform distribution in the range from $-\sqrt{6/(nin + nout)}$ to $\sqrt{6/(nin + nout)}$, where nin and $nout$ are the sizes of layers before and after the weight matrix. During the training stage, all parameters are tuned. Moreover, we use dropout with probability 0.2 to prevent overfitting. Due to the space limitation, we treat all hyper-parameters as constants. Without special illustration, m in Eq. (10) and Eq. (11) is empirically set as 0.1, and $\lambda_1, \lambda_2, \lambda_3$ in Eq. (12) are set as 1.0.

Evaluation Metrics. In our study, we focus on two prediction tasks, i.e., backing distribution prediction and success time prediction. We adopt different metrics to evaluate their performances.

For the backing distribution prediction task, we adopt three different metrics, i.e., *KL* divergence, root mean square error (*RMSE*) and mean absolute error (*MAE*). Specifically, for the campaign i, given its real backing distribution $\boldsymbol{v}_f^i$ and predicted one $\widetilde{\boldsymbol{v}}_f^i$, where $\boldsymbol{v}_f^i, \widetilde{\boldsymbol{v}}_f^i \in \mathbb{R}^{10}$, the performances can be measured by:

$$\begin{aligned} KL &= \sum\nolimits_{j=0}^{9} v_{fj}^i \log \frac{v_{fj}^i}{\widetilde{v}_{fj}^i}, \\ RMSE &= \frac{1}{10} \sum\nolimits_{j=0}^{9} (v_{fj}^i - \widetilde{v}_{fj}^i)^2, \\ MAE &= \frac{1}{10} \sum\nolimits_{j=0}^{9} |v_{fj}^i - \widetilde{v}_{fj}^i|. \end{aligned} \tag{13}$$

For the success time prediction task, due to the censoring phenomenon, the success time of a large proportion of campaigns is unobservable so that we adopt a widely used evaluation metric in survival analysis, i.e., the concordance index (CI). Specifically, given N pairs of campaigns $(d^i, \widetilde{d^i}), i \in \{1, 2, ..., N\}$, where d^i is the actual observed time, and $\widetilde{d^i}$ is the predicted one, CI can be calculated by:

$$CI = \frac{1}{num} \sum_{i \in \{1,2,...,N\} \wedge \delta^i = 1} \; \sum_{d^j > d^i} I[\widetilde{d^j} > \widetilde{d^i}], \tag{14}$$

where num denotes the number of comparable pairs.

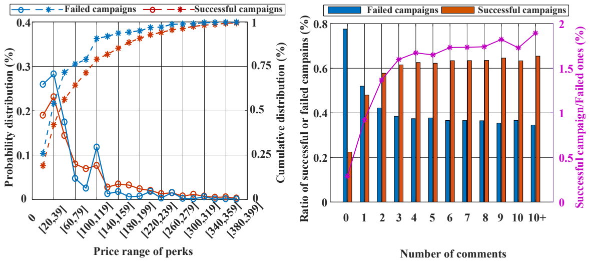
Benchmark Methods. When testing the performances on the backing distribution task, we apply the encoder part with the linear evolutionary prior (denoted as *SMP-A*) against baselines. When testing the performances on the success time prediction task, we compare our complete method (denoted as *SMP*) against baselines.

For the backing distribution task, we choose three effective methods as baselines.

- *Multinomial-LR* (Krishnapuram et al. 2005) is a classification method that generalizes logistic regression (LR) to multiclass problems.
- *MLP (Multilayer Perceptron)* (Bengio and others 2009) is a feedforward artificial neural network.
- *SMP-NA* is a variant of *SMP-A* without the linear evolutionary prior.

For the success time prediction task, we choose a series of representative models widely used in survival analysis.

- *COX* (Cox 1992) is the most commonly used semi-parametric model in survival analysis.
- *Logistic, Log-logistic* (Li, Rakesh, and Reddy 2016) are parametric censored regression models with logistic and log-logistic distributions respectively.



(a) Backing distribution. (b) Comment effects.

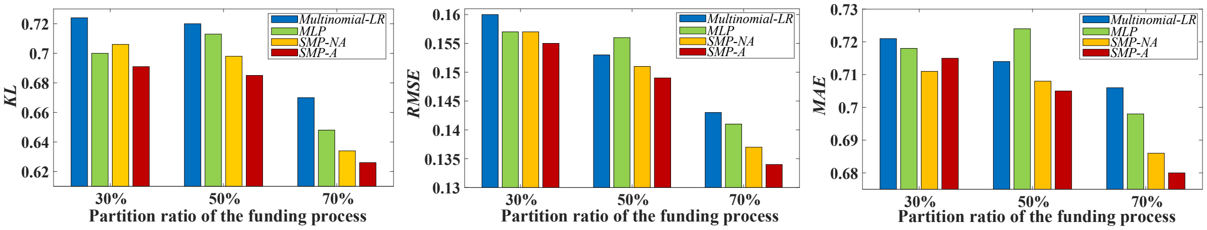
Figure 4: Data exploration.

- *Tobit* (Li et al. 2016a) is an extension of the linear regression which estimates relations between variables when there is censoring in the dependent variable.
- *Multitask_L21, Multitask_Lasso* (Li et al. 2017) are two standard multi-task learning models with $l_{2,1}$ and $LASSO$ norm penalty, respectively.
- *BoostCOX* (Hothorn et al. 2004) is an ensemble method using Gradient-boosted Cox proportional hazard loss with regression trees as base learner.
- *SurvivalSVM* (Navab and Katouzian 2015) is a ranking-based support vector machine for survival analysis
- *SMP-S* is a variant of *SMP* by removing the encoder part. We initialize the memory cell of the decoder by concatenating the static and dynamic features with a hidden layer and then continue to make a prediction.
- *SMP-T* is another variant of *SMP* by removing the backing distribution prediction task and the linear evolutionary prior, i.e., $\mathcal{L}_{en}^i$ and $\mathcal{L}_{c1}^i$ are removed in Eq. (12).

Except our models (i.e., *SMP*, *SMP-T*, *SMP-A*, *SMP-NA*), other benchmark methods cannot model the sequential features (i.e., comments in our study). To deal with it, we transform all comments into one vector through an average pooling operation and then concatenate it with static features. Note that all methods are trained with parameters which perform their best on the training data.

Experimental Results

Data Exploration. In this experiment, we explore the impact of backing distribution and backers' feedback (i.e., comments) on campaign success. For the former factor, we collect all backing behaviors and separate them into several groups according to the two rules: (1) Whether the corresponding campaign reaches its pledged goal by the expiration date? (2) How much is the backing? According to our statistics, in our dataset, there are 5,534 successful campaigns and 8,609 failed ones so that 60.9% of campaigns are censored instances. Then, we plot the backing distribution as shown in Figure 4a. In this figure, solid lines denote the probability distribution of successful campaigns (red curve) and failed campaigns (blue curve), while dotted lines denote the cumulative distribution. From the comparison of two solid lines, we find the blue one is higher than the red



(a) *KL* (b) *RMSE* (c) *MAE*

Figure 5: Performances of the backing distribution prediction.

Table 2: Performances of success time prediction.

Level	Model	Partition Ratio		
		30%	50%	70%
Semi-Para	*COX*	0.737	0.693	0.720
Parametric	*Tobit*	0.770	0.709	0.793
	Logistic	0.684	0.680	0.689
	Log-Logistic	0.695	0.702	0.701
Machine Learning	*Multitask_L21*	0.608	0.596	0.595
	Multitask_Lasso	0.610	0.622	0.600
	BoostCOX	0.836	0.776	0.85
	SurvivalSVM	0.752	0.717	0.794
Our Models	*SMP-S*	0.734	0.729	0.764
	SMP-T	0.883	0.930	0.899
	SMP	**0.910**	**0.960**	**0.914**

one at lower price range but it changes to the opposite at higher price range. That is to say, the average funding price of successful campaigns is larger than that of failed campaigns. The comparison of two dotted lines can also reflect the same conclusion. Therefore, the backing distribution is highly related to the results of campaigns.

For the second factor, we make some statistics according to the number of comments. In Figure 4b, for a fixed number of comments, the corresponding red bar denotes the percentage of successful campaigns while the blue bar denotes the failed campaigns. In addition, we also compute the ratio of successful ones to the failed ones plotted by a solid line. From this figure, we observe that the success rate of campaigns increases with the number of comments. Generally, it means the comments play a positive role in raising funds and help campaigns appeal to more backers to support so that reaching their pledged goals.

From the statistics and analysis above, we can conclude some representative features such as funding prices and dynamic comments play an important role in our two tasks.

Performances on Backing Distribution Prediction. Here, we demonstrate the performance comparisons on the backing distribution prediction task. Figure 5 provides the histogram plots on *KL*, *RMSE* and *MAE* metrics. From the results, it can be observed that our models (i.e., *SMP-A* and *SMP-NA*) outperform the other baselines in most cases. Additionally, compared with *SMP-NA*, *SMP-A* performs better which indicates that our linear evolutionary prior is effective and help the model predict the backing distribution more accurately. Furthermore, as the partition ratio of the funding process increases, the number of comments also increases and the distance between the predicted and real distribution is getting smaller. This observation once again shows the important role that comments play in our study.

Performances on Success Time Prediction. Here, we show the performance comparisons on the success time prediction task among all methods. As shown in Table 2, our complete method (i.e., *SMP*) significantly performs better than the baselines at three different partitions. Compared with the best baseline apart from our variant models (i.e., *BoostCOX*), *SMP* respectively performs improvements with 8.9%, 7.9% and 6.9% on *CI*. This indicates the effectiveness of *SMP* on the success time prediction task. Then, we turn to the comparison results of *SMP-S*, *SMP-T* and *SMP*. Specifically, *SMP-S* integrates heterogeneous features by a hidden layer while *SMP-T* employs LSTM to model their relations. As a consequence, *SMP-T* dominates *SMP-S* on this task which indicates the encoder part in *SMP-T* has the ability on learning the complex relations of heterogeneous features. In addition, compared with *SMP-T*, *SMP* takes the backing distribution into account when predicting the success time of the campaign. And the results also show that *SMP* performs better than *SMP-T* which reflects the predicted backing distribution has the positive effects on this task.

Conclusion

In this paper, we presented a focused study on predicting both backing distribution and success time of campaigns by integrating heterogeneous features. Inspired by the sequence learning and survival analysis, we first proposed a seq2seq based model to jointly model two prediction tasks. Then, according to our observation and analysis, we made some assumptions and also developed three innovative priors, i.e., a linear evolutionary prior, a non-increasing prior and a partial prior. In experiments, we analyzed the influence of comments and backing distributions on the results of campaigns and validated the capacity of *SMP* and its variants on two tasks. The experimental results clearly validated the effectiveness of *SMP*.

Acknowledgements

This research was supported by grants from the National Natural Science Foundation of China (Grants No.

U1605251, 61727809 and 61672483). 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

Ameri, S.; Fard, M. J.; Chinnam, R. B.; and Reddy, C. K. 2016. Survival analysis based framework for early prediction of student dropouts. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 903–912. ACM.

An, J.; Quercia, D.; and Crowcroft, J. 2014. Recommending investors for crowdfunding projects. In *Proceedings of the 23rd international conference on World Wide Web*, 261–270.

Bannerman, S. 2013. Crowdfunding culture. *Journal of Mobile Media* 7(01):1–30.

Bengio, Y., et al. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

Cox, D. R. 1992. Regression models and life-tables. In *Breakthroughs in statistics*. Springer. 527–541.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6645–6649. IEEE.

Hothorn, T.; Lausen, B.; Benner, A.; and Radespiel-Tröger, M. 2004. Bagging survival trees. *Statistics in medicine* 23(1):77–91.

Huang, Z.; Liu, Q.; Chen, E.; Zhao, H.; Gao, M.; Wei, S.; Su, Y.; and Hu, G. 2017. Question difficulty prediction for reading problems in standard tests. In *AAAI*, 1352–1359.

Jing, H., and Smola, A. J. 2017. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 515–524.

Krishnapuram, B.; Carin, L.; Figueiredo, M. A.; and Hartemink, A. J. 2005. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (6):957–968.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.

Li, Y.; Wang, J.; Ye, J.; and Reddy, C. K. 2016a. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD*, 1715–1724. ACM.

Li, Y.; Wang, L.; Wang, J.; Ye, J.; and Reddy, C. K. 2016b. Transfer learning for survival analysis via efficient l2, 1-norm regularized cox regression. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 231–240.

Li, H.; Ge, Y.; Zhu, H.; Xiong, H.; and Zhao, H. 2017. Prospecting the career development of talents: A survival analysis perspective. In *Proceedings of the 23rd ACM SIGKDD*, 917–925. ACM.

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.

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 *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 695–702. AAAI Press.

Lu, C.-T.; Xie, S.; Kong, X.; and Yu, P. S. 2014. Inferring the impacts of social media on crowdfunding. In *Proceedings of the 7th ACM international conference on Web Search and Data Mining*, 573–582. ACM.

Mendenhall, W. M.; Sincich, T. L.; and Boudreau, N. S. 2016. *Statistics for Engineering and the Sciences, Student Solutions Manual*. Chapman and Hall/CRC.

Mollick, E. 2014. The dynamics of crowdfunding: An exploratory study. *Journal of business venturing* 29(1):1–16.

Navab, N., and Katouzian, A. 2015. Fast training of support vector machines for survival analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 243–259.

Rakesh, V.; Lee, W.-C.; and Reddy, C. K. 2016. Probabilistic group recommendation model for crowdfunding domains. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 257–266. ACM.

Severyn, A., and Moschitti, A. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 373–382. ACM.

Tibshirani, R. 1997. The lasso method for variable selection in the cox model. *Statistics in medicine* 16(4):385–395.

Vinzamuri, B.; Li, Y.; and Reddy, C. K. 2014. Active learning based survival regression for censored data. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, 241–250. ACM.

Yang, Y., and Zou, H. 2013. A cocktail algorithm for solving the elastic net penalized cox's regression in high dimensions. *Statistics and its Interface* 6(2):167–173.

Yang, G.; Cai, Y.; and Reddy, C. K. 2018. Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

Zhang, Y.; Xiong, Y.; Kong, X.; and Zhu, Y. 2016. Netcycle: Collective evolution inference in heterogeneous information networks. In *Proceedings of the 22nd ACM SIGKDD*, 1365–1374. ACM.

Zhao, H.; Zhang, H.; Ge, Y.; Liu, Q.; Chen, E.; Li, H.; and Wu, L. 2017. Tracking the dynamics in crowdfunding. In *Proceedings of the 23rd ACM SIGKDD*, 625–634. ACM.

Zhao, H.; Liu, Q.; Zhu, H.; Ge, Y.; Chen, E.; Zhu, Y.; and Du, J. 2018. 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.

Zhou, M. J.; Lu, B.; Fan, W. P.; and Wang, G. A. 2018. Project description and crowdfunding success: an exploratory study. *Information Systems Frontiers* 20(2):259–274.

Zvilichovsky, D.; Inbar, Y.; and Barzilay, O. 2013. Playing both sides of the market: Success and reciprocity on crowdfunding platforms. *Social Science Electronic Publishing* 4.