Project Success Prediction in Crowdfunding Environments

Yan Li
Dept. of Computer Science
Wayne State University
Detroit, MI - 48202.
rock_liyan@wayne.edu

Vineeth Rakesh
Dept. of Computer Science
Wayne State University
Detroit, MI - 48202.
vineethrakesh@wayne.edu

Chandan K. Reddy
Dept. of Computer Science
Wayne State University
Detroit, MI - 48202.
reddy@cs.wayne.edu

ABSTRACT

Crowdfunding has gained widespread attention in recent years. Despite the huge success of crowdfunding platforms, the percentage of projects that succeed in achieving their desired goal amount is only around 40%. Moreover, many of these crowdfunding platforms follow "all-or-nothing" policy which means the pledged amount is collected only if the goal is reached within a certain predefined time duration. Hence, estimating the probability of success for a project is one of the most important research challenges in the crowdfunding domain. To predict the project success, there is a need for new prediction models that can potentially combine the power of both classification (which incorporate both successful and failed projects) and regression (for estimating the time for success). In this paper, we formulate the project success prediction as a survival analysis problem and apply the censored regression approach where one can perform regression in the presence of partial information. We rigorously study the project success time distribution of crowdfunding data and show that the logistic and log-logistic distributions are a natural choice for learning from such data. We investigate various censored regression models using comprehensive data of 18K Kickstarter (a popular crowdfunding platform) projects and 116K corresponding tweets collected from Twitter. We show that the models that take complete advantage of both the successful and failed projects during the training phase will perform significantly better at predicting the success of future projects compared to the ones that only use the successful projects. We provide a rigorous evaluation on many sets of relevant features and show that adding few temporal features that are obtained at the project's early stages can dramatically improve the performance.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications-Data Mining; I.2.6 [**Artificial Intelligence**]: Learning; H.3.3 [**Information Search and Retrieval**]: Information filtering

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.
© 2016 ACM. ISBN 978-1-4503-3716-8/16/02…$15.00
DOI: http://dx.doi.org/10.1145/2835776.2835791

Keywords

Prediction, project success, survival analysis, regression, crowdfunding.

1. INTRODUCTION

Crowdfunding has emerged as "the next big thing" in entrepreneurial financing. It aims at providing the seed capital for many start-up companies, creating job opportunities and reviving lost business ventures. Crowdfunding websites helped companies and individuals worldwide raise $89 million from the public in 2010 and explosively grown to $5.1 billion in 2013. The concept of crowdfunding is similar to micro-financing where the required funds are collected by pooling relatively small amounts of money from several individuals instead of a single venture capitalist. Over the past few years, crowdfunding platforms have raised several billion dollars worldwide, thereby becoming a viable alternative for people seeking the help of banks, brokers, and other financial intermediaries to jump-start their business ventures.

In spite of the tremendous success in crowdfunding, statistics show that only around 40% of the projects succeed in reaching their pledged goal (KickStarterStats) [15]. Even small amounts of improvement in the projects' success can bring potentially millions of dollars in the overall revenue for the creators. This can potentially lead to better innovation and provide more job opportunities since most of these projects will not receive any funding from other sources at such early stages of product development.

Project success is an extremely vital component of crowdfunding which, if correctly estimated, can provide some guideline to the project creators and backers about the progress and potential of the project. In addition, this information can guide future algorithms for recommending projects that are more likely to succeed for the backers. In other words, having a good prediction model can aid the individuals to invest in projects that are more likely to succeed in the future. Since many of the crowdfunding domains follow an "all-or-nothing" policy (which means the pledged money is collected only if the goal amount is reached in a certain predefined time duration), it becomes annoying to the users who invest in the projects that eventually do not succeed because if the investors fund projects that eventually fail, then it will waste their time (with no returns) and increase the "opportunity cost".

However, merely estimating whether a project will be successful or not using its corresponding goal date cannot provide a proper guideline to the backers who want to invest in popular projects. To illustrate the weakness of the classification based approaches to solve this problem and motivate our work, let us consider the following real-world example shown in Figure 1. If we just build a model to predict whether a project will succeed or not, then projects 1, 4, and 5 will be labeled as “failed”, and projects 2, 3, and 6 will be labeled as “successful”. Project 1 absorbed almost \$40K during 30 days period and might have the potential to attract a lot of attention in few more days, while project 4 only absorbed \$2K during 60 days. It obviously states that project 1 is much more attractive and valuable to investigate comparing to project 4, so that it is unfair for project 1 to put these two projects in one class which will mislead the investigators. Thus, classification methods are not suitable for project recommendation in the crowdfunding domain.

Typically, investors would like to invest in projects which can succeed as soon as possible. Our goal in this paper is to rank the projects based on their expected success date, and thus the investors can choose some interesting projects from the pool of highly-ranked projects. If the investor’s behavior is influenced by our ranking result, then they will fund those highly-ranked projects (such as project 1 in the above example) which will then help the projects become successful eventually. It can also help the project creators to have an idea about where they stand with respect to other projects in terms of achieving the goal amount even before the project begins. Hence, a good success prediction model can help both investors (to choose some valuable and potentially successful projects) and creators (by providing some guidelines on the chance of success).

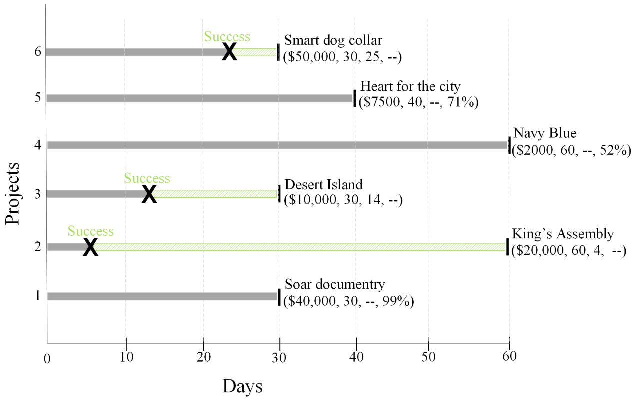


Figure 1: An example of 6 projects from the crowdfunding data. X-axis represents the project duration (in days). The projects with complete solid lines correspond to the failed projects (1,4, and 5 in this example). The remaining projects (2,3, and 6) are successful since they achieved the project goal amount within the goal date (which is marked by ’**X**’ in the figure). Each project is indicated in the following format: project title (w, x, y, z), where w indicates the project goal amount, x indicates the project duration, y indicates the number of days taken to achieve success, and z indicates the percentage of amount received by the goal date. It should be noted that while w and x are available for all the projects, y is available only for the successful projects and z is available only for the failed projects.

Due to the dynamic nature of the projects and the fact that the data contains information about both success and failed projects, it becomes non-trivial to build a prediction model for this crowdfunding data. Especially, the presence of both successful and failed projects in the training data along with the time information presents a complex environment for the prediction task. For the projects that are successful (projects 2,3, and 6 in Figure 1), the true value of the time taken for achieving success is exactly known. However, for the failed projects (projects 1,4, and 5 in Figure 1), the only information available is the amount they received until the project goal date and there will be no information about when those projects can receive the entire goal amount (become successful).

Standard regression and ranking models ignore the data about failed projects since the failed ones do not have the information about the actual success date. Hence, these standard regression approaches can only consider the successful projects (for which the number of days to achieve success is known and is a positive integer value). However, the failed projects carry important piece of information that they are not successful until a certain time point (project goal date). This information is vital and ignoring such information will reduce the model performance. Hence, *in this crowdfunding domain, there is a need for regression or ranking models that can potentially combine the information of both successful and failed projects.* In spite of the importance of the problem, this area of research is relatively unexplored in the data mining and machine learning communities.

In this paper, we will demonstrate that using the failed projects in the prediction model can provide significantly better results compared to the model that will only use successful projects for training. It should be noted that standard regression models suffer from two drawbacks when applied on this data. Since time can only be a positive number, the model should output only positive values. In addition, the model should accommodate the failed projects for which there is only partial information available. To effectively solve these challenges, we formulate the project success prediction task as a suvival analysis problem, where the failed projects can be viewed as the censored instances and successful projects as the uncensored instances, and utilize the censored regression approaches to slove the prediction task.

The main contributions of our work are summarized as follows:

- We formulate the project success prediction task as a survival analysis problem and show that the censored regression models that take complete advantage of both successful and failed projects during the training phase will perform significantly better at predicting the success of future projects compared to the ones that use only the successful project information.

- Show that the logistic and log-logistic distributions are a natural choice for fitting the parametric models for crowdfunding (Kickstarter) data and rigorously evaluate and compare these two models with various other censored regression models available in the literature.

- Evaluate the most optimal set of features that need to be extracted from the real crowdfunding domain (Kickstarter dataset) for predicting the project success.

- Demonstrate that adding few temporal features that are obtained after the project begins (at the early stages such as first 3 days) can dramatically improve the prediction performance.

The rest of this paper is organized as follows: Section 2 provides the related work in crowdfunding and prediction problems. The Kickstarter dataset and the formal definition of the prediction problem are described in Section 3; our analysis and solution of the prediction problem are given in Section 4. A detailed discussion of the experimental results is provided in Section 5 and Section 6 concludes our discussion.

2. RELATED WORK

2.1 Crowdfunding and Kickstarter

Since crowdfunding is still an emerging platform, most works in this domain are relatively new. The most popular form of crowdfunding is the reward-based, where the individuals fund a project in exchange for a variety of rewards. Kickstarter has become one of the most popular reward-based crowdfunding platforms. Raising a whopping $529 million in pledged amount and 22,252 successfully funded projects, the year 2014 was extremely successful for projects in the Kickstarter domain. Kickstarter terms the investors as backers, and the founders of a project as creators. The creators project their ideas by posting a detailed description about their projects. Usually, the description contains videos, images and textual information that explains the novelty of the project. In addition to this, the creators provide a detailed timeline, funding goal, and the rewards for different pledge levels. Even though the progress of the Kickstarter domain has been outstanding, the success rate of projects has not been very impressive. Recent statistics report a success rate of less than 50%. Being relatively new, very few studies have explored this domain from a data mining perspective [32, 10, 27].

In [20], the authors examine the dynamics of the Kickstarter domain. To understand the factors that motivate users to invest in crowdfunding projects, [12] and [17] perform the real-world analysis on crowdfunding platforms. The work in [22] delineates the impact of social network on Kickstarter projects. In their work, the authors leverage social network based features such as: promotional activity in Twitter, effect of weakly connected components, network diameter, triadic closures, etc. to predict the number of backers and funding amount that will be accrued by a project. The authors of [7] propose a Maximum-entropy distribution model and show the impact of team behaviors in the *Kiva.org* domain. There were also some prior efforts in exploring the effects of the internet on micro-financing, and peer-to-peer lending transactions [4, 2]. Studies on microfinance decision-making have discovered that lenders favor lending opportunities not only to the entities that are similar to themselves but also to individuals in situations that trigger an emotional reaction [1, 11]. While the domain is interesting and can potentially bring huge financial impact, it is surprising to see that most of the computational techniques proposed are relatively naive. To solve the problem described in the previous section, we need to have more sophisticated approaches that can provide insightful information about the prediction of project success.

2.2 Background on Prediction Methods

Before describing the work related to our approach, we will first highlight the drawbacks of the standard classification and regression models for solving the success prediction problem stated previously.

- Modeling crowdfunding data poses a new challenge in terms of incorporating the projects where we know the success date and the projects where we have only partial information that they did not succeed until a certain project goal date. Such projects are termed as *censored* ones. In traditional regression/classification setting, these projects are simply treated as missing data and they do not contribute any information unless one makes quite stringent assumptions followed by heavy computation (e.g. multiple imputation). However, using censored regression models, a likelihood function is constructed using the partial information.
- In regression problems, the outcome variable is continuous and will be any real number, while time by it's very nature will strictly be non-negative. The standard machine learning methods such as linear and logistic regression cannot be used to predict survival times. This is due to the fact that one cannot enforce linear and logistic regression algorithms to *predict non-negative outcomes.* The censored regression models can inherently handle this non-negative constraint and build models that predict only non-negative outcome variables.

It is clear from the above discussion that the censored regression models have some critical advantages compared to standard regression/classification. Albeit it is not to be seen as a competitor to the standard regression analysis, rather, such censored models are applicable to more specialized and complex modeling scenarios, namely, modeling Time-to-Event data. In this paper, we consider the event of interest to be the project success and the goal is to predict when a project can potentially become successful compared to the other ones that are available. Hence, in such problems, one will have complete information about the events for successful projects only. The failed projects will not have the event occurred and will be observed only until the project goal date. *The critical difference between our formulation and the standard regression approaches is the fact that our work incorporates both successful and failed projects simultaneously as opposed to using only the successful projects as done in regression based formulations.*

We will now introduce more details about the censored regression models that will be used in this paper. They mainly contain two components: (i) Time-to-event, i.e. time taken for a specific event of interest (project success) to occur and (ii) Censoring, i.e. partial information of projects where success did not occur. The form of censoring that is seen in our problem is the right censoring where the survival time is known to be longer than a certain value but its precise value is unknown. Additionally, there are also features that one needs to relate with time to explain time-to-event phenomenon (such as project success time). Such models test for differences in success times for two or more projects of interest, while allowing to adjust for the project features. More recently, few problems in computational advertising have been effectively tackled using such survival models [6]. In order to model the censored data, some of these approaches use an approximation of the likelihood function called the partial log likelihood, to train the survival model [8, 23].

3. DATASET AND PROBLEM

3.1 Dataset Description

(i) Data sources: For our experiments, we collected data from three different sources namely: Kickstarter, Twitter and Facebook. Our dataset was prepared using a two step approach. First, we obtained the Kickstarter projects and removed irrelevant projects. Second, we used these filtered set of projects to fetch their promotional activities from Twitter and Facebook. We describe this process in detail in this section.

Kickstarter Database: We obtained six months of Kickstarter data from *kickspy.*[1] Our dataset spans from 12/15/13 to 06/15/14, which consists of 27,270 projects. We removed projects that were canceled or suspended as well as those with less than one backer and $100 as pledged amount. In this manner, we obtained 18,093 projects.

Table 1: Basic statistics of our Kickstarter data consisting of 18,093 projects collected from Dec 2013 - Jun 2014.

Attribute	Mean	Min	Max	StdDev
Goal Amt	26,531.2	100	100,000,000	758,366.5
Pledged Amt	11,023.6	100	6,224,955	78,550.8
Duration(days)	31	1	60	10.05

Promotions from Twitter: Social media sites such as Twitter and Facebook are often used as means to promote Kickstarter projects. Researchers have shown that such promotional activities have a very strong impact on the success of Kickstarter projects [26, 22]. Therefore, in this paper, we built our database by retrieving tweets that contain the term *http://kck.st* in their URL field [2]. By expanding these short URLs, we eliminated tweets that did not map to our project database. Using this method, we obtained 106,738 unique tweets, which covered 55% of our projects. The remaining 45% were never promoted using Twitter.

Promotions from Facebook: Since Facebook does not allow us to fetch the data using their API, we simply scraped the following information from the Kickstarter website: *number of facebook shares for a project*, and *facebook friends of creators.*

(ii) Feature extraction: The various kinds of extracted features were successfully used in our recent work [26], which can be summarized as follows:

Project based features: We extracted 15 different features for every project in our database. The numerical features include the duration of project, the goal amount, the number of images, the presence of videos and the number of comments about the project. The duration of a project ranges anywhere between 1-60 days with an average of 30 days and the comments in Kickstarter are posted by those who are interested in the status of the ongoing projects; for every project, we count the number of comments posted by these users. The textual features such as project description, risks and challenges, and FAQs were converted into numerical values by counting the number of words in the respective feature. The categorical feature for a project is based on the topic of the project and it's geo-location. Kickstarter classifies the topical category of a project into 15 different groups such as art, comics, music, technology, publishing etc. For a detailed description of these features, the readers are encouraged to go through [26, 22].

Features from the project creators: This includes the number of projects created, projects backed by the creator, success ratio of the creator, and features obtained from creator's facebook profile (7 features).

Social network features (obtained from Twitter): These are network-based measures that were created using Twitter users who promoted the projects in our database. The features include tie-strength between the promoters of projects, number of bi-connected components, and the PageRank scores of Twitter users who promoted the projects in the first three days of the project duration (3 features). The details about the creation of these features can be found in our recent work [26].

Temporal features: The accumulation over the first three days in terms of the number of backers, the funding amount, the number of Twitter promotions and the number of Facebook shares (12 features).

(iii) Distribution of Projects: The maximum time period a project can last is 60 days. In other words, the creator can choose anywhere between 1 and 60 days for the project duration. In this crowdfunding problem, for each project, its starting day is considered to be the first day of our study time scale; thus, in the study time scale, the maximum value of the actual observed successful or failed day is 60. Figure 2 shows the overall statistics of all the 18,093 Kickstarter projects where the X-axis is the actual number of observed days and Y-axis is the logarithm (base 10) of the number of projects. We can observe that the successes and failures of projects occur at every possible day, and a relatively large amount of project creators choose 30 days as their project duration.

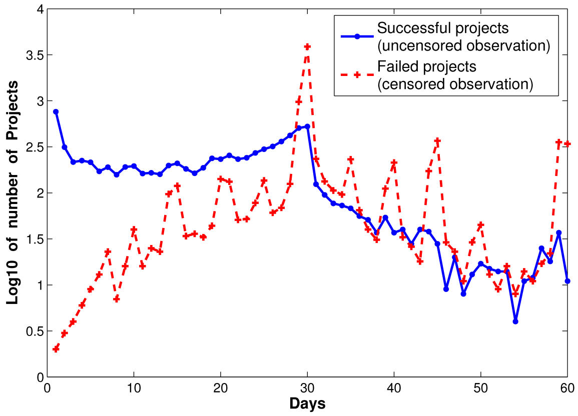


Figure 2: Distribution of the successful (in blue) and failed projects (in red) during the 60 days time window.

3.2 Problem Formulation

We formulate the problem of estimating project success and ranking in crowdfunding as a survival analysis problem and employ censored regression models which can simultaneously leverage both successful and failed projects. Survival analysis is one of the most important methods in the field of statistics [19, 25] which aims at modeling the time to a particular event of interest (project success in our

[1] www.kickspy.com

[2] we used the query API available at www.topsy.com

case). In such longitudinal studies, the observation starts from a certain starting timepoint and will continue until the occurrence of project success or the predefined project duration is reached (in which case the project success is not observed). This notion of having only partial information available about the project behaviour is also known as censoring [19, 28] and those failed projects are considered as censored observations. Given the historical database of successful and failed projects, the goal here is to estimate the time taken for the project success for a new project and recommend the project based on the result of our prediction. The problem can be formulated as follows:

Problem Statement:

For the i^{th} project, let us consider its predefined project duration to be U_i and it takes T_i days to reach the project goal amount. It should be noted that T_i *is a latent value for failed projects because it did not reach its goal amount during the predefined project duration.* Each project can be represented by a triplet (X_i, y_i, δ_i), where X_i is $1 \times m$ project feature vector, and δ_i is the project failure indicator ($\delta_i = 1$ for a successful project and $\delta_i = 0$ for a failed project). The *observed time* y_i for a project is then defined as follows:

$$y_i = \begin{cases} T_i & \text{if} \quad \textit{project is successful } (\delta_i = 1) \\ U_i & \text{if} \quad \textit{project is failed } (\delta_i = 0) \end{cases} \qquad (1)$$

The final goal of our work is to estimate T_j for a new (j^{th}) project whose feature descriptors are represented by X_j. It should be noted that T_j will be a non-negative continuous value in this case.

4. CENSORED REGRESSION FOR ESTIMATING PROJECT SUCCESS

4.1 Notations and Definitions

Let us first introduce some important probability functions in survival analysis and provide a connection between those functions and the prediction problem in the crowdfunding domain. The *survival function* $S(t) = Pr(T \geq t)$ is the probability that the time to the event of interest is no earlier than certain specified time t. In our case, the project success is the event of interest and T is the success date; hence, $S(t)$ is the probability that the project does not succeed after t days from the project starting date and we call it as the *unsuccessful probability* in short. The *cumulative death distribution function* $F(t) = 1 - S(t)$ can be renamed as the *cumulative successful probability* which represents the probability that the project achieves its goal amount within t days. The *density function* $f(t)$ is defined as $f(t) = \frac{dF(t)}{dt} = \frac{F(t+\Delta t)-F(t)}{\Delta t}$, where $\Delta t \to 0$ is a short time interval, represents the probability that a project achieves its goal amount at day t. To describe the characteristics of the crowdfunding data, both unsuccessful (survival) probability function and the density function are needed.

Let us consider a set of N projects out of which there are c failed projects and $(N-c)$ successful projects. As described earlier, the j^{th} project is represented by (X_j, y_j, δ_j). For convenience, we use the general notation $\mathbf{b} = (b_1, b_2, \cdots, b_p)$ to represent a set of parameters and assume that the project success times follow a theoretical distribution with the unsuccessful probability function $S(t, \mathbf{b})$ and density function $f(t, \mathbf{b})$. If a project j is failed, then it is not possible to obtain the actual number of days that are needed to achieve its goal; however, it will be known that the project does not reach its goal amount until the last day of the predefined project duration U_j, so $S(U_j, \mathbf{b})$ should be a probability value that is close to 1. On the contrary, if project j is a successful project which is succeeded at T_j, then $f(T_j, \mathbf{b})$ should be a high probability.

4.2 Objective Function

Using these notations, we can now use $\prod_{\delta_j=1} f(y_j, \mathbf{b})$ to represent the joint probability of $(N-c)$ successful projects and $\prod_{\delta_j=0} S(y_j, \mathbf{b})$ to represent the joint probability of c failed projects. Hence, the complete likelihood function for all the N projects is given by

$$\begin{aligned} L(\mathbf{b}) &= \prod_{\delta_j=1} f(y_j, \mathbf{b}) \prod_{\delta_j=0} S(y_j, \mathbf{b}) \\ &= \prod_{\delta_j=1} f(y_j, \mathbf{b}) \prod_{\delta_j=0} (1 - F(y_j, \mathbf{b})) \qquad (2) \end{aligned}$$

Note that $\mathbf{b}$ will not only contain the feature coefficient vector but also includes the parameters of the chosen theoretical distribution. Now, one of the problems that arise here is the determination of $f(t, \mathbf{b})$ which consists of two parts: functional form and parameter estimation. To make an efficient and accurate prediction, first an appropriate theoretical distribution has to be selected to describe the characteristic of the Kickstarter dataset.

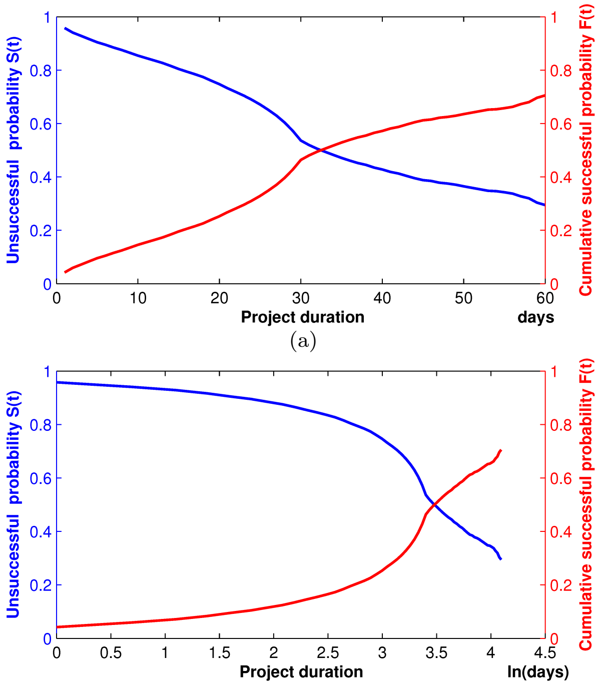


Figure 3: Plot showing the Kaplan-Meier Curves for the Kickstarter dataset. The Y-axis represents unsuccessful (survival) probability $S(t)$ and cumulative successful (death) probability $F(t)$. X-axis corresponds to the number of days in (a) and Logarithm of number of days in (b). The unsuccessful probability (in blue) and the cumulative successful probability (in red) are plotted.

We observed an interesting phenomenon that the $F(t)$ for

Kickstarter projects closely follows the cumulative distribution function (CDF) of a logistic distribution. Figure 3 shows the Kaplan-Meier curves for the Kickstarter data under two different scales on X-axis: (a) the project duration in days and (b) the logarithm of the project duration. Kaplan-Meier curve [18] is a popular non-parametric method which can be used to provide a general view of the overall distributions of $S(t)$ and $F(t)$ for a dataset with censored instances. In both figures 3(a) and 3(b), the blue curves correspond to the $S(t)$ and the red curves correspond to the $F(t)$. We see that the red curves have a shape approximately close to the CDF of a logistic function. These two figures show that *the logistic and log-logistic distributions are appropriate in modeling the probability of project success in the crowdfunding domain and hence these two distributions will be incorporated into the objective function given by Eq.(2).*

4.3 Model Learning

In this section, we will elaborate on the likelihood function by fitting both logistic and log-logistic distributions in Eq.(2). The Parameters for the objective function can be estimated using maximum likelihood estimation (MLE) [21].

Logistic distribution: The censored parametric regression with logistic distribution assumes that there exists a linear relationship between the observed time y_j and the feature vector X_j which is modeled as follows:

$$y_j = X_i\beta + \sigma\varepsilon_j \tag{3}$$

where $\beta = (\beta_1, \cdots, \beta_m)^T$ is the coefficient vector, σ is an adjusted parameter, and ε_j follows a logistic distribution. Thus, the observed time y_j follows the logistic distribution. Based on Eq.(3), the ε can be calculated as $\varepsilon = \frac{y - X\beta}{\sigma}$, and the cumulative successful function is defined as:

$$F(y, \beta, \sigma) = \frac{exp(\frac{y - X\beta}{\sigma})}{1 + exp(\frac{y - X\beta}{\sigma})} \tag{4}$$

and the density function will be

$$f(y, \beta, \sigma) = \frac{F(y)}{dy} = \frac{\frac{1}{\sigma}exp(\frac{y - X\beta}{\sigma})}{\left[1 + exp(\frac{y - X\beta}{\sigma})\right]^2} \tag{5}$$

Substituting Eq.(4) and Eq.(5) in Eq.(2), we obtain the likelihood function for logistic distribution

$$L(\beta, \sigma) = \prod_{\delta_j = 1} \frac{\frac{1}{\sigma}exp(\frac{y_j - X_j\beta}{\sigma})}{\left[1 + exp(\frac{y_j - X_j\beta}{\sigma})\right]^2} \prod_{\delta_j = 0} \frac{1}{1 + exp(\frac{y_j - X_j\beta}{\sigma})} \tag{6}$$

and the coefficient vector β and model parameter σ can be estimated by minimizing the negative log-likelihood

$$\begin{aligned} l(\beta, \sigma) = &\sum_{\delta_j = 1} \left\{ \frac{y_j - X_j\beta}{\sigma} - \log\sigma - 2\log\left[1 + exp(\frac{y_j - X_j\beta}{\sigma})\right] \right\} \\ &- \sum_{\delta_j = 0} \log\left[1 + exp(\frac{y_j - X_j\beta}{\sigma})\right] \end{aligned} \tag{7}$$

Log-logistic distribution: The parametric methods for censored regression with log-logistic distributions can be viewed as a special case of the accelerated failure-time (AFT) model where the logarithm of the observed time y_j is linearly related to the feature vector X_j [3]:

$$\log y_j = X_i\beta + \sigma\varepsilon_j \tag{8}$$

Similar to the logistic distribution case described above, ε_j follows a logistic distribution and can be calculated as $\varepsilon = \frac{\log y - X\beta}{\sigma}$, using Eq.(8); thus, we have

$$\begin{aligned} S(y, \beta, \sigma) &= \frac{1}{1 + exp(\frac{-X\beta}{\sigma})y^{\frac{1}{\sigma}}} \\ f(y, \beta, \sigma) &= \frac{\frac{1}{\sigma}exp(\frac{-X\beta}{\sigma})y^{\frac{1}{\sigma} - 1}}{\left[1 + exp(\frac{-X\beta}{\sigma})y^{\frac{1}{\sigma}}\right]^2} \end{aligned} \tag{9}$$

and based on the same procedure described in the logistic distribution case, the log-likelihood function is given by

$$\begin{aligned} l(\beta, \sigma) = &\sum_{\delta_j = 1} \left\{ \frac{-X_j\beta}{\sigma} + \frac{1 - \sigma}{\sigma}\log y_i \right. \\ &\left. - \log\sigma - 2\log\left[1 + exp(\frac{-X_j\beta}{\sigma})y_j^{\frac{1}{\sigma}}\right] \right\} \\ &- \sum_{\delta_j = 0} \log\left[1 + exp(\frac{-X_j\beta}{\sigma})y_j^{\frac{1}{\sigma}}\right] \end{aligned} \tag{10}$$

The coefficient vector β and model parameter σ of logistic and log-logistic distributions can be estimated by minimizing the negative of Eq.(7) and Eq.(10), respectively. These minimization problems can be solved using standard Newton-Raphson method and the gradients of the negative log-likelihood with respect to β and σ can be calculated using the chain-rule, and more details of solving it are available at [21].

5. EXPERIMENTAL RESULTS

We will now describe our experimental results including the evaluation metrics and implementation details of the methods used for experimental analysis.

5.1 Experiment setup

We evaluate the censored regression with logistic distribution and log-logistic distribution using Kickstarter data and compare these two models with other popular prediction methods that are available in the literature for handling censored observations. We used the following state-of-the-art methods for our comparison.

- **Cox proportional hazards model**: The Cox model [8] is the most commonly used semi-parametric model in survival analysis. The hazard function has the form $\lambda(t, X_i) = \lambda_0(t)exp(X_i\beta)$, where the $\lambda_0(t)$ is the common *baseline hazard function* for all individuals and β is the coefficient vector which can be estimated by minimizing the negative *log-partial likelihood* function.
- **Tobit regression**: Tobit model [30] is an extension of the linear regression $y_j = X_j\beta + \varepsilon_j, \varepsilon_j \sim N(0, \sigma^2)$, but the parameter is estimated by the maximum likelihood method rather than using the least square error. It uses the parametric method framework discussed in section 4.2 with the probability density function and the cumulative distribution function of the standard normal distribution.
- **Buckley-James estimation**: Buckley-James regression [5] is also a AFT model which uses Kaplan-Meier estimation to approximate the survival time of the censored observations as the target value, and then builds a linear model based on both the true survival times of uncensored observations and these approximated survival times.

- **Boosting concordance index**: Boosting concordance index (BoostCI) [24] is an approach where the concordance index metric (also known as the 'survival AUC' which is described in the next section) is modified into an equivalent smoothed criterion using the sigmoid function and the resulting optimization problem is solved using a gradient boosting algorithm.

The experiments in this work are performed in R programming environment. The Cox model, Tobit model, and the censored regression models with Logistic and Log-logistic distributions are implemented using the *survival* package [29]. In the survival package, the *coxph* function is employed to train the cox model and the Efron's method [9] is used to handle the tied observations. The Buckley-James Regression is fitted using the *bujar* package [31], and the BoostCI is trained based on the supporting information of [24] and the *mboost* package [16].

5.2 Evaluation metrics

Survival AUC, or the *concordance probability*, is used to measure the performance of regression and ranking models [14, 13]. Consider a pair of projects (T_1, $\hat{T}_1$) and (T_2, $\hat{T}_2$), where T_i is the actual observed day of success, and $\hat{T}_i$ is the predicted one. The concordance probability is defined as:

$$c = Pr(\hat{T}_1 > \hat{T}_2 | T_1 \geq T_2) \tag{11}$$

A high survival AUC value indicates a high concordance between predicted and observed time values. If T_i only has two possible values, then the regression models reduce to classification and the survival AUC will be the same as the standard Receiver operating characteristic (ROC) AUC. The survival AUC has the same scale as AUC, where 0.5 corresponds to random guessing and 1 indicates a perfect prediction. Survival AUC of standard regression model can be directly calculated using Eq.(11). In the Cox model, the output is related to the hazard rate and the project with a larger hazard rate should succeed earlier; the survival AUC of Cox model can be calculated by:

$$c = \frac{1}{num} \sum_{i \in \{1 \cdots N\} \delta_i = 1} \sum_{y_j > y_i} I(X_i \hat{\beta} > X_j \hat{\beta}) \tag{12}$$

where num denotes the number of comparable pairs and $I(\cdot)$ is the indicator function. The survival AUC for other methods, which directly target the time of success, should be calculated as:

$$c = \frac{1}{num} \sum_{i \in \{1 \cdots N\} \delta_i = 1} \sum_{y_j > y_i} I[S(\hat{y}_j | X_j) > S(\hat{y}_i | X_i)] \tag{13}$$

where $S(\hat{y}_i | X_i)$ is the predicted target value.

5.3 Results and Discussion

In this section we will discuss our experimental results of various censored prediction methods using different sets of features.

We performed experiments using different feature subsets: "Static" corresponds to the basic (static) statistical features obtained form the project description and the project creator; "Static+Social" corresponds to the basic (static) features along with the social network features obtained from Twitter; "Static+3days" denotes the basic (static) features along with the temporal features obtained at the beginning stages (first 3 days) of each project; "Static+Social+3days" denotes the complete set of features (union of all the previous three categories). In addition, for each feature set, we also generated two variants of the training data: "with censored", which includes both the successful and failed projects, and "without censored" which includes only the successful projects. Note that in the "without censored" version, the Tobit regression will become ordinary least squares (OLS) linear regression, and the other parametric censored regression models will become the corresponding uncensored regression models.

Table 2 provides the survival AUC (concordance index) values of each model on the Kickstarter data using 10-fold cross validation. For all the methods across all the feature set combinations, our results evidently show that adding the failed projects (censored observations) to the successful ones will provide significantly better prediction results compared to the corresponding data which contains only the successful projects ("without failed" version). By incorporating the failed projects, the survival AUC of all the methods was improved by 4.3% on an average. This result clearly indicates that *incorporating the failed projects (censored informations) will significantly help in building an accurate prediction model.*

In Figure 4, we show the concordance probability matrices (**C**) for four different methods using only the success projects and adding failed projects to the successful ones (containing all the features). The index of each element of the matrix plot corresponds to the actual observed days of a pair of comparable projects; in other words, $\mathbf{C}_{ij}$ is the concordance probability of all comparable project pairs whose actual observed days are represented by i and j, respectively. The term "actual observed days" used here corresponds to the number of days taken for obtaining the goal amount for successful projects and the total project time period (days until the goal date) for the failed projects. Note that this matrix is symmetric, and since we cannot calculate the concordance probability of two projects when their actual observed days are same, we had to set the value of diagonal elements to be 0. We can observe that there exists one common phenomenon among all the plots shown in Figure 4. The concordance probability of elements close to the diagonal are usually lower compared to the ones away from the diagonal.

This phenomenon reflects the fact that it is hard to predict the correct ordering when the two projects take similar number of days to succeed. In the top row, the four sub-figures are generated without using the failed projects for all features, and the four sub-figures in the bottom row are generated using both the successful and failed projects. The two sub-figures within the same column are generated using the same prediction method. The plot shows the distribution across all possible pairwise combinations and hence it will help us in visualizing and understanding the regions where the improvements are significant when using the failed projects. We can evidently see that, within same prediction method, the concordance probability is higher in the case of using the failed projects compared to the one where they are not being used.

From Table 2, we also observe that, compared to the features extracted from social network, the temporal (dynamic) features are more useful in prediction. *Significant improvements on prediction can be made if we can obtain the information from the first 3 days of the project progress.* This

Table 2: Performance comparison of various sets of features of Kickstarter projects with or without failed projects (censored observations) using Survival AUC values (along with their standard deviation).

	Static		Static+Social		Static+3days		Static+Social+3days	
	without failed	with failed	without failed	with failed	without failed	with failed	without failed	with failed
Cox	0.7322 (0.0104)	0.7727 (0.0092)	0.7463 (0.0098)	0.7942 (0.0089)	0.7667 (0.0126)	0.7965 (0.0093)	0.7724 (0.0121)	0.8098 (0.0087)
Tobit	0.7281 (0.0108)	0.7755 (0.0100)	0.7381 (0.0099)	0.7960 (0.0082)	0.7833 (0.0124)	0.8226 (0.0096)	0.7841 (0.0121)	0.8309 (0.0084)
BJ	0.7097 (0.0130)	0.7313 (0.0114)	0.7235 (0.0128)	0.7587 (0.0080)	0.8016 (0.0127)	0.8157 (0.0102)	0.8016 (0.0127)	0.8201 (0.0089)
BoostCI	0.5919 (0.0140)	0.6649 (0.0288)	0.6128 (0.0380)	0.6796 (0.0212)	0.8135 (0.0430)	0.8668 (0.0229)	0.8141 (0.0421)	0.8671 (0.0231)
Logistic	**0.7354** **(0.0106)**	0.7815 (0.0095)	**0.7457** **(0.0095)**	0.8009 (0.0086)	0.8332 (0.0097)	0.8659 (0.0075)	0.8331 (0.0094)	0.8695 (0.0067)
Log-logistic	0.7277 (0.0111)	**0.7826** **(0.0099)**	0.7411 (0.0096)	**0.8029** **(0.0081)**	**0.8800** **(0.0057)**	**0.9010** **(0.0056)**	**0.8774** **(0.0060)**	**0.9030** **(0.0057)**

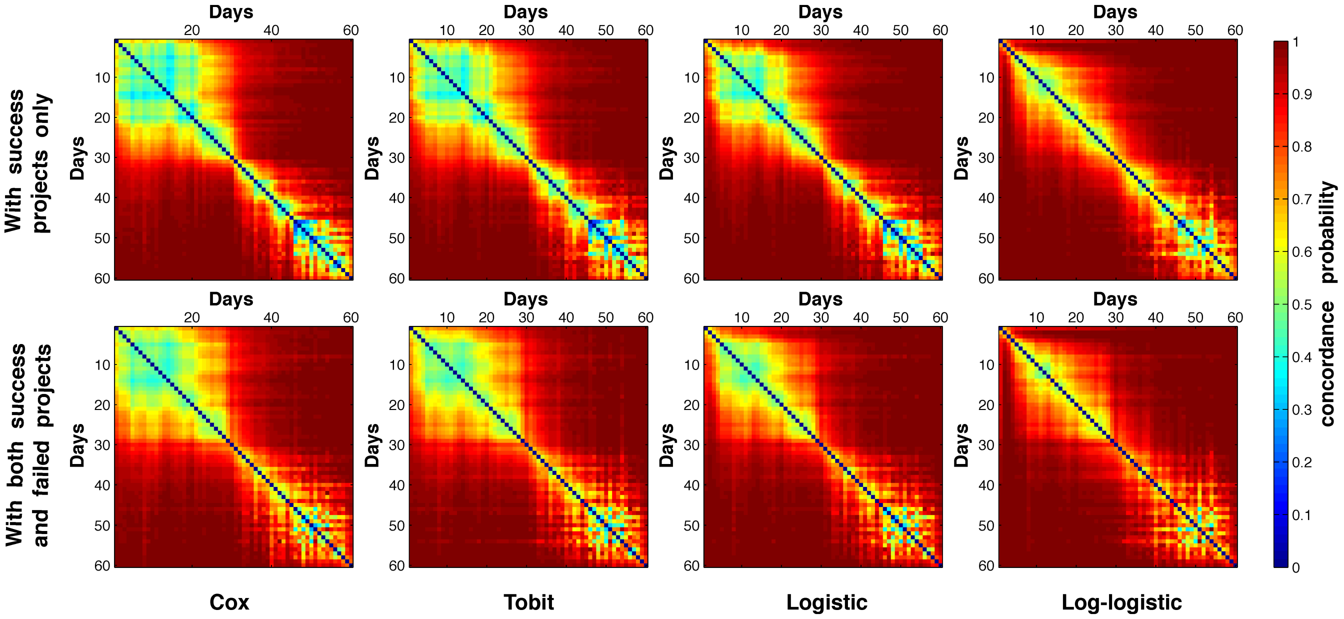


Figure 4: Concordance probability matrices for four different methods (Cox, Tobit, Logistic, and Log-logistic) using "with success projects only" (top row) and "with both success and failed projects" (bottom row). These results are based on all the features that are being studied.

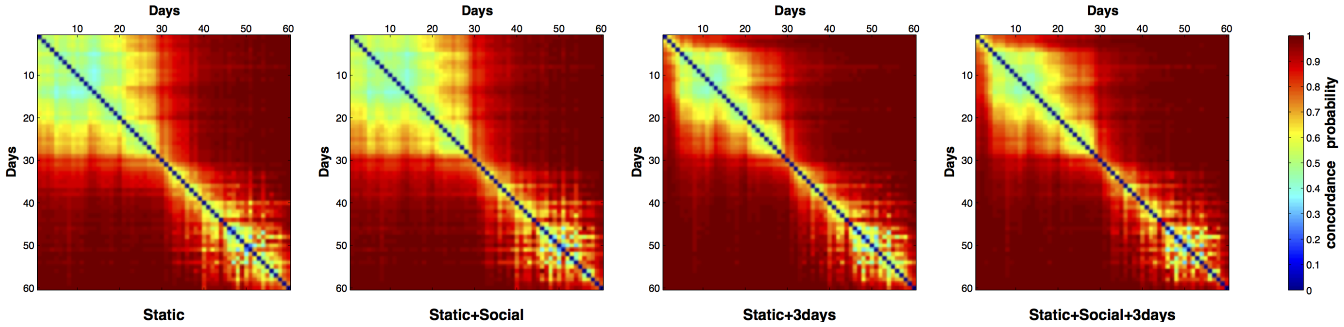


Figure 5: Concordance probability matrices obtained by the parametric censored regression with Logistic distribution using different feature subsets.

is a very useful characteristic in practice since it can guide the backers in deciding whether to invest in a particular project or not. Potentially this information can also be used in recommending projects to the backers. In Figure 5, we demonstrate the performance of the censored regression with Logistic distributions with different subset of features (using both failed and successful projects). We can clearly see that the model performance cannot be dramatically improved if we only combine the social network features with the static ones. However, adding the temporal features that are obtained at the beginning stages (first 3 days) of the project progress can dramatically help in improving the prediction performance. Overall, we can conclude that all the features obtained are useful for training the models, and the more appropriate features we collect the better the performance will become. From both Figures 4 and 5, we can also observe that all the methods show good prediction results when the actual observed days is large (greater than 20-30 days).

One of the main objectives of this paper is to demonstrate that, in the crowdfunding domain, when the goal is to predict project success, using only the successful projects will provide inferior prediction results compared to the case where failed projects are also being added. In other words, more value is added to the prediction when partial (censored) information from the failed projects is added to the successful projects (where the complete information on success is available). The partial information here refers to the fact that, in the failed projects, the information which is available is that the project did not receive success until the goal date and the information which is missing is that one does not know when the project will become successful.

While the above observations unanimously conclude that adding the failed projects is extremely useful in practice, we performed even more thorough analysis on the effects of adding such failed projects. The failed projects are censored (incomplete) observations; the data distribution of such censored observations have some correlations with the data distribution of the successful projects. The prediction result can be improved when we incorporate only a portion of the failed projects. In Figure 6, we present the prediction performance of different methods by varying the percentage of failed projects included in the model. It should be noted that the x-axis corresponds to the percentage of failed projects that are incorporated within the successful projects while building the prediction models. Hence, 0% corresponds to the case where only the successful projects were used and 100% corresponds to using all the projects (both successful and failed together). The results reported here are the average (of 10 different runs) improvements made by adding a random set of certain percentage of failed projects. From these four sub-figures we can see that the survival AUC can be improved dramatically even if only a relatively smaller portion (around 20%-30%) of the failed projects are incorporated, but the curve becomes close to a flat one when the failed projects added exceeds a certain limit.

6. CONCLUSION AND FUTURE WORK

In this paper, we solve the problem of predicting project success in a crowdfunding environment combined with innovative introduction of survival analysis based approaches. While the day of success is considered to be the time to reach an event, the failed projects are considered to be censored since the day of success is not known. We performed rigorous analysis of the Kickstarter crowdfunding domain to reveal unique insights about factors that impact the success of projects. Our experimental results show that incorporation of failed projects (censored information) can significantly help in building a robust prediction model and such censored models can perform better than standard prediction models that are available in the literature. Additionally, we also created several Twitter-based features to study the impact of social network on the crowdfunding domain. Our study shows that these social network-based features can help in improving the prediction performance. Most importantly, we found that the temporal features obtained at the beginning stage (first 3 days) of each project will significantly improve the prediction performance. In the future, we plan to implement a system which is able to rank the Kickstarter projects dynamically and help the project backers make a better decision on their investments in a real-time environment.

Acknowledgments

This work was supported in part by the US NSF grants IIS-1231742, IIS-1242304, and IIS-1527827.

7. REFERENCES

[1] J. Andreoni. Impure altruism and donations to public goods: a theory of warm-glow giving. *The economic journal*, pages 464–477, 1990.

[2] A. Ashta and D. Assadi. Do social cause and social technology meet? impact of web 2.0 technologies on peer-to-peer lending transactions. *Cahiers du CEREN*, 29:177–192, 2009.

[3] S. Bennett. Log-logistic regression models for survival data. *Applied Statistics*, pages 165–171, 1983.

[4] T. Bruett. Cows, kiva, and prosper. com: How disintermediation and the internet are changing microfinance. *Community Development Investment Review*, 3(2):44–50, 2007.

[5] J. Buckley and I. James. Linear regression with censored data. *Biometrika*, 66(3):429–436, 1979.

[6] O. Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1105. ACM, 2014.

[7] J. Choo, D. Lee, B. Dilkina, H. Zha, and H. Park. To gather together for a better world: understanding and leveraging communities in micro-lending recommendation. In *Proceedings of the 23rd international conference on World wide web*, pages 249–260, 2014.

[8] D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 187–220, 1972.

[9] B. Efron. The efficiency of cox's likelihood function for censored data. *Journal of the American Statistical Association*, 72(359):557–565, 1977.

[10] V. Etter, M. Grossglauser, and P. Thiran. Launch hard or go home!: predicting the success of kickstarter campaigns. In *Proceedings of the first ACM conference on Online social networks*, pages 177–182, 2013.

[11] J. Galak, D. Small, and A. T. Stephen. Microfinance decision making: A field study of prosocial lending. *Journal of Marketing Research*, 48(SPL):S130–S137, 2011.

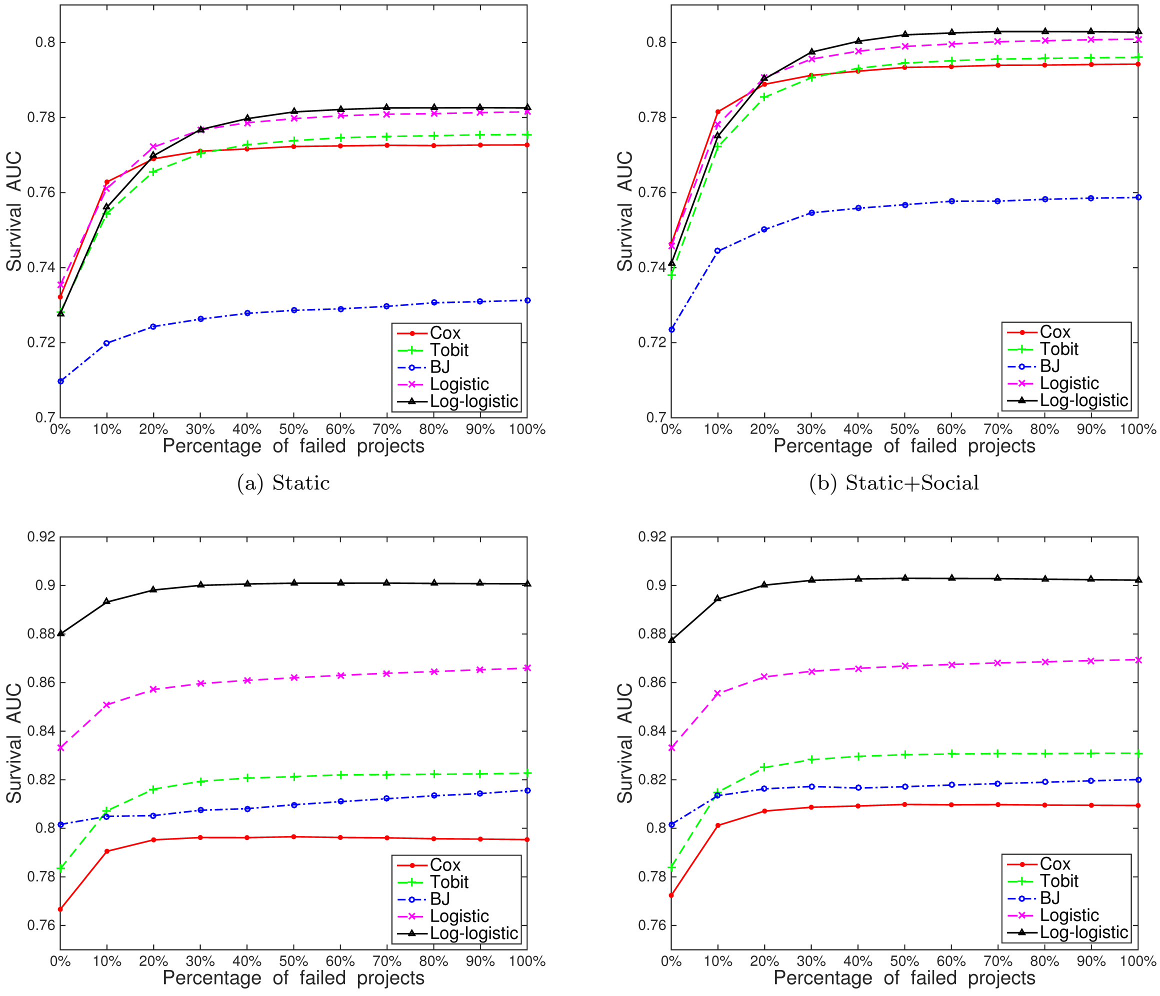
[12] E. M. Gerber, J. S. Hui, and P.-Y. Kuo. Crowdfunding: Why people are motivated to post and fund projects on crowdfunding platforms. In *CSCW Workshop*, 2012.

[13] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. Evaluating the yield of medical tests. *JAMA*, 247(18):2543–2546, 1982.

[14] F. E. Harrell, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in medicine*, 3(2):143–152, 1984.

[15] S. Heyman. Keeping up with kickstarter. *The New York Times, URL http://www.nytimes.com/2015/01/15/arts/international/keeping-up-with-kickstarter.html*, 2015.

[16] T. Hothorn, P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner. Model-based boosting 2.0. *Journal of Machine Learning Research*, 11:2109–2113, 2010.



(a) Static

(b) Static+Social

(c) Static+3days

(d) Static+Social+3days

Figure 6: Survival AUC curves for different methods obtained by varying the percentage of failed projects included along with the successful ones.

[17] J. S. Hui, M. D. Greenberg, and E. M. Gerber. Understanding the role of community in crowdfunding work. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 62–74. ACM, 2014.

[18] E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[19] J. P. Klein and M.-J. Zhang. *Survival analysis, software.* Wiley Online Library, 2005.

[20] V. Kuppuswamy and B. L. Bayus. Crowdfunding creative ideas: The dynamics of project backers in kickstarter. *SSRN Electronic Journal*, 2013.

[21] E. T. Lee and J. Wang. *Statistical methods for survival data analysis*, volume 476. Wiley. com, 2003.

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

[23] M. Lunn and D. McNeil. Applying cox regression to competing risks. *Biometrics*, pages 524–532, 1995.

[24] A. Mayr and M. Schmid. Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations. 2014.

[25] R. G. Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011.

[26] V. Rakesh, J. Choo, and C. K. Reddy. Project recommendation using heterogeneous traits in crowdfunding. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[27] V. Rakesh, W.-C. Lee, and C. K. Reddy. Probabilistic group recommendation model for crowdfunding domains. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 2016.

[28] C. K. Reddy and Y. Li. A review of clinical prediction models. In C. K. Reddy and C. C. Aggarwal, editors, *Healthcare Data Analytics*. Chapman and Hall/CRC Press, 2015.

[29] T. Therneau. A package for survival analysis in s. r package version 2.37-4. *URL http://CRAN. R-project. org/package= survival. Box*, 980032:23298–0032, 2013.

[30] J. Tobin. Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, pages 24–36, 1958.

[31] Z. Wang and C. Wang. Buckley-james boosting for survival analysis with high-dimensional biomarker data. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

[32] A. Xu, X. Yang, H. Rao, W.-T. Fu, S.-W. Huang, and B. P. Bailey. Show me the money!: an analysis of project updates during crowdfunding campaigns. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 591–600, 2014.