

# Gaussian Process Regression for Crowdfunding Campaigns Prediction

Victor Kristof

December 16, 2015

## Table of Contents

# 1 Introduction

## 1.1 Notations

- Let us denote  $\mathbf{v}_{i:j}$ ,  $i < j$  the subvector  $\mathbf{u} = [v_i, \dots, v_j]^T$  consisting of the elements  $v_i$  to  $v_j$  of the vector  $\mathbf{v}$ .

# 2 Problems Definition

## 2.1 Single-Project Regression

### Model

We are first approaching the problem as time series regression, considering only one project. Our dataset  $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \dots, T\}$  consists of  $T$  observations, with  $x_i$  the time index corresponding to an amount of pledged money  $y_i$ . Hence, we have  $X = [1, \dots, T]^T$ , a  $(T \times 1)$  matrix of time indices, and  $\mathbf{y} = [y_1, \dots, y_T]^T$ , a vector of observed values. We model the pledged money  $f(X)$  at time indices  $X$  as a Gaussian Process (GP):

$$f(X) \sim GP(m(X), k(X, X')). \quad (1)$$

Our goal is to predict the future values of the pledged money  $\mathbf{f}_{t:T} = f(X_{t:T})$  at future time indices  $X_{t:T} = [t, \dots, T]^T$  after observing the values  $\mathbf{y}_{1:t} = [y_1, \dots, y_t]^T$  at time indices  $X_{1:t} = [1, \dots, t]^T$ . In the GP framework, we can compute this prediction using

$$\mathbf{f}_{t:T} \mid X_{1:t}, \mathbf{y}_{1:t}, X_{t:T} \sim \mathcal{N}(\bar{\mathbf{f}}_{t:T}, \text{cov}(\mathbf{f}_{t:T})) \quad (2)$$

$$\bar{\mathbf{f}}_{t:T} = K(X_{t:T}, X_{1:t}) [K_{1:t} + \sigma_n^2 I]^{-1} \mathbf{y}_{1:t} \quad (3)$$

$$\text{cov}(\mathbf{f}_{t:T}) = K_{t:T} - K(X_{t:T}, X_{1:t}) [K_{1:t} + \sigma_n^2 I]^{-1} K(X_{1:t}, X_{t:T}), \quad (4)$$

with  $K_{1:t} := K(X_{1:t}, X_{1:t})$  and  $K_{t:T} := K(X_{t:T}, X_{t:T})$ . Finally, the model's optimal hyperparameters  $\theta_*$  are learned by maximizing the *log marginal likelihood* over the first  $t$  observed values:

$$\theta_* = \arg \max_{\theta} \log p(\mathbf{y}_{1:t} \mid X_{1:t}, \theta) \quad (5)$$

$$= \arg \max_{\theta} \left\{ -\frac{1}{2} \mathbf{y}_{1:t}^T [K_{1:t} + \sigma_n^2 I]^{-1} \mathbf{y}_{1:t} - \frac{1}{2} \log \det [K_{1:t} + \sigma_n^2 I] - \frac{t}{2} \log 2\pi \right\}. \quad (6)$$

## Results

The major problem in this context was that the predictive mean  $\mathbf{f}_*$  always falls back to the mean  $m(\mathbf{x})$  very quickly. One solution is to sum two squared-exponential kernels and initializing one of them to a large length-scale in order to capture the global trend. This yields to some reasonable results. However, when using a model trained on one project to predict the future values of another one gives very poor performances.

## 2.2 Multi-Project Regression

### Model

Our next idea is to consider  $P$  projects and try to learn the hyperparameters  $\theta$  over various time series at the same time. In this case, for a given project  $p$ , we have a dataset  $\mathcal{D}^{(p)} = \{(x_i^{(p)}, y_i^{(p)}) \mid i = 1, \dots, T\}$ . Again,  $y_i^{(p)}$  is the amount of pledged money at each time index  $x_i^{(p)}$ . Note that we have  $x_i^{(p)} = x_i = i$ , for all  $p = 1, \dots, P$  and all  $i = 1, \dots, T$ . We combine the projects together to obtain the full dataset  $\mathcal{D} = \{\mathcal{D}^{(p)} \mid p = 1, \dots, P\}$  (*multi-task learning*). We then have the input  $X = [1, \dots, T]^T$ , a  $(T \times 1)$  matrix of time indices, and the output  $Y = [\mathbf{y}^{(p)}]_{p=1}^P$ , a  $(T \times P)$  matrix of observed values per project. For a given project  $p$ , we are trying to predict the pledged money  $\mathbf{f}_{t:T}^{(p)} = f(X_{t:T}^{(p)})$  at future time indices  $X_{t:T}^{(p)} = [t, \dots, T]^T$  after observing the values  $\mathbf{y}_{1:t}^{(p)}$  at time indices  $X_{1:t}^{(p)} = [1, \dots, t]^T$ . To do so in the GP framework, we have

$$\mathbf{f}_{t:T}^{(p)} \mid X_{1:t}^{(p)}, \mathbf{y}_{1:t}^{(p)}, X_{t:T}^{(p)} \sim \mathcal{N}(\bar{\mathbf{f}}_{t:T}^{(p)}, \text{cov}(\mathbf{f}_{t:T}^{(p)})), \quad (7)$$

with  $\bar{\mathbf{f}}_{t:T}^{(p)}$  and  $\text{cov}(\mathbf{f}_{t:T}^{(p)})$  obtained similarly to Equations ?? and ??. The model's optimal hyperparameters  $\theta_*$  are learned similarly to Equation ?? by maximizing the log marginal likelihood over all the projects, that is

$$\theta_* = \arg \max_{\theta} \sum_{p=1}^P \log p(\mathbf{y}^{(p)} \mid X^{(p)}, \theta). \quad (8)$$

Note that in this case, we consider the “full” projects, that is  $\mathbf{y}^{(p)} = \mathbf{y}_{1:T}^{(p)}$  and  $X^{(p)} = X_{1:T}^{(p)}$ .

## Results

Again, we couldn't obtain good results with this approach, as the predictions always fall back very quickly to the mean of the GP. [MORE DETAILS]

## 2.3 Project Classification

### Model

We then decide to perform a simpler task. Instead of trying to predict future amounts of pledged money after some observations, we try now to classify whether a project  $p$  will be successful ( $c^{(p)} = 1$ ) or not ( $c^{(p)} = 0$ ). Indeed, by separating the dataset in two classes (*successful* and *failed*), we notice that they have a very different profiles, as shown in Figure ??, and therefore should be easy to discriminate. To do so, we train one GP using the successful projects and one using the failed projects. We then try to determine whether a new, partially observed project will be successful or not. We learn  $\theta_s$  the hyperparameters

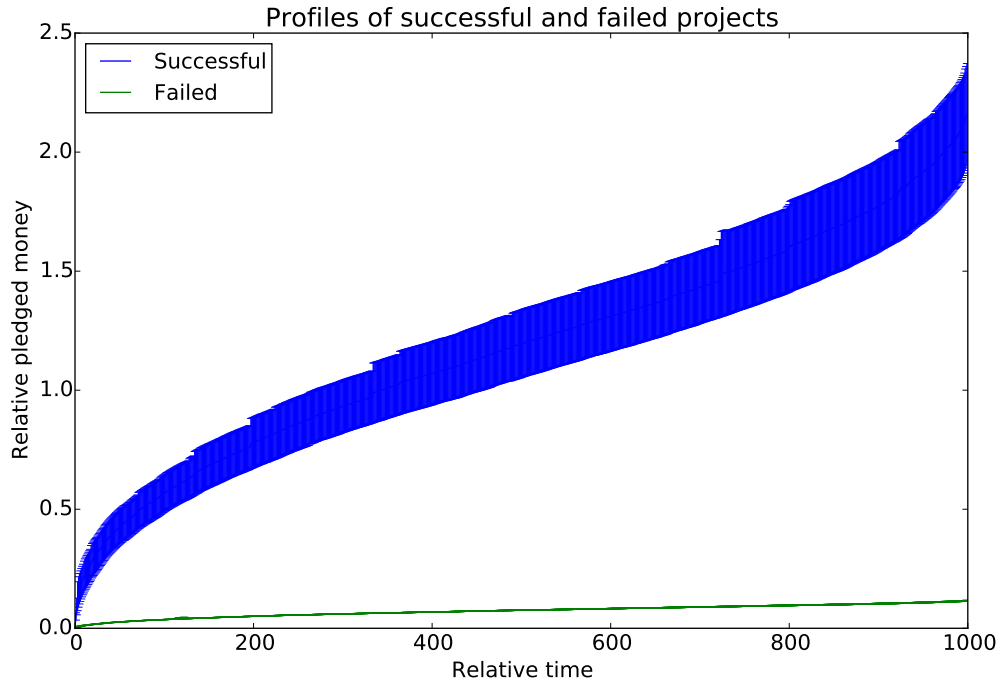


Figure 1: Average and normalized standard deviation for both classes displaying two very distinct profiles that suppose the possibility to classify the projects.

of a GP over  $P_s$  successful projects and  $\theta_f$  the hyperparameters of a GP over the  $P_f$  failed projects. That is, we obtain the two sets of hyperparameters by maximizing the log marginal likelihoods

$$\theta_s = \arg \max_{\theta} \sum_{p=1}^{P_s} \log p(\mathbf{y}_s^{(p)} | X^{(p)}, \theta)$$

$$\theta_f = \arg \max_{\theta} \sum_{p=1}^{P_f} \log p(\mathbf{y}_f^{(p)} | X^{(p)}, \theta),$$

where  $\mathbf{y}_s$  and  $\mathbf{y}_f$  denote the observations of the successful and failed projects respectively. We then determine the success state  $c^{(p)}$  of a new project  $p$  with partial observations  $\mathbf{y}_{1:t}^{(p)}$  as

$$c^{(p)} = \begin{cases} 1, & \text{if } \log p(\mathbf{y}_{1:t}^{(p)} | X_{1:t}, \theta_s) > \log p(\mathbf{y}_{1:t}^{(p)} | X_{1:t}, \theta_f) \\ 0, & \text{otherwise} \end{cases}.$$

## Results

[TO BE COMPLETED]

### 2.4 Output as Input

#### Model

We change completely the set up. Instead now of using the time as input and trying to predict the pledged money at new time indices, we now consider the pledged money at each time step as input and the last time index as the output. That is, we have now a dataset  $\mathcal{D} = \{(\mathbf{x}^{(p)}, y^{(p)}) | p = 1, \dots, P\}$  with  $\mathbf{x}^{(p)} = \mathbf{y}_{1:t}^{(p)}$  and  $\mathbf{y}^{(p)} = y_T^{(p)}$ . We then have  $X = [\mathbf{x}^{(p)}]_{p=1}^P$ , a  $(P \times t)$  matrix of pledged money, and  $\mathbf{y} = [y_T^{(p)}]_{p=1}^P$ , a  $(P \times 1)$  vector corresponding to the total amount of money at the end of the campaign. The difference with the previous approach is that now the features for each project are the amount of pledged money at different time steps and not the same (shared) input values  $[1, \dots, T]^T$ . Note that  $t$  can be arbitrarily set to any value up to  $T - 1$  and that the number of samples taken in  $[1, t]$  can also be determined arbitrarily.

Our goal now is to predict, for a new project  $p$ , the final amount of pledged money  $y_T^{(p)} = f(\mathbf{y}_{1:t}^{(p)})$  given  $\mathbf{y}_{1:t}^{(p)}$  the money received up to time  $t$  after observing the total pledged money for all projects  $\mathbf{y} = [y_T^{(p)}]_{p=1}^P$  and the money they received up to time  $t$ ,  $X = [\mathbf{y}_{1:t}^{(p)}]_{p=1}^P$ . In the GP framework, we can compute this prediction using

$$y_T^{(p)} | X, \mathbf{y}, \mathbf{y}_{1:t}^{(p)} \sim \mathcal{N}(\bar{y}_T^{(p)}, \text{cov}(y_T^{(p)})) \quad (9)$$

$$\bar{y}_T^{(p)} = K(\mathbf{y}_{1:t}^{(p)}, X) [K(X, X) + \sigma_n^2 I]^{-1} \mathbf{y} \quad (10)$$

$$\text{cov}(y_T^{(p)}) = K(\mathbf{y}_{1:t}^{(p)}, \mathbf{y}_{1:t}^{(p)}) - K(\mathbf{y}_{1:t}^{(p)}, X) [K(X, X) + \sigma_n^2 I]^{-1} K(X, \mathbf{y}_{1:t}^{(p)}). \quad (11)$$

As seen in Figure ??, it seems impossible to do a regression considering all the projects together. However, the two modes corresponding to the successful and failed projects are

clearly distinguishable. Therefore we try to perform the regression on each of the two classes separately.

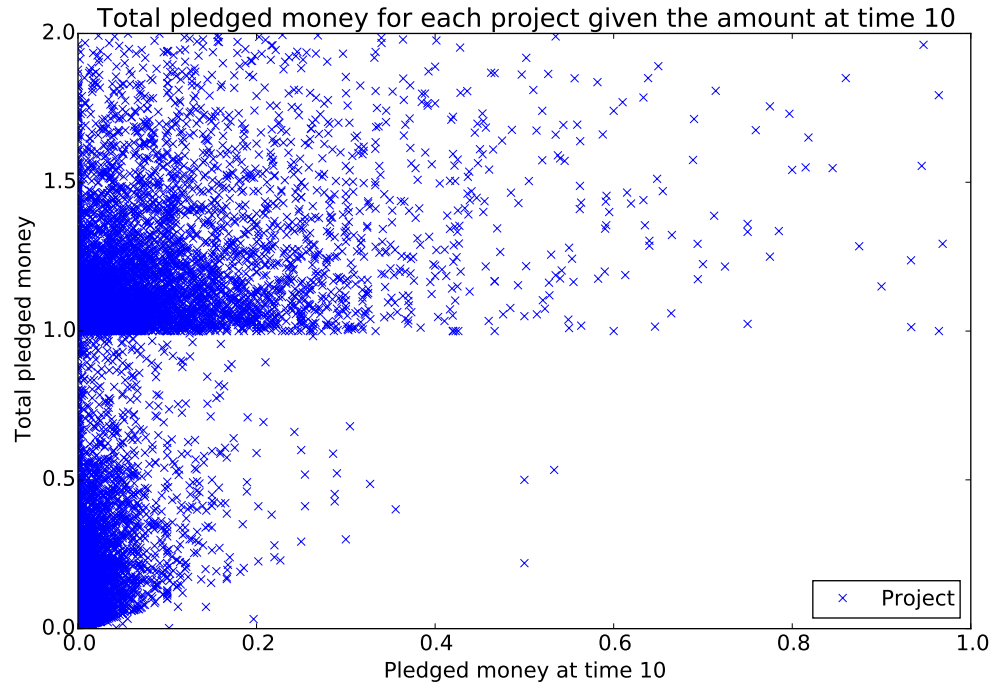


Figure 2: Total pledged money for each project versus the amount they had at time 10.

## Results

[TO BE COMPLETED] Mixture of models can probably be used as other (latent) modes may exist.

### 2.5 Mixture of Gaussian Processes

#### Model

#### Results

### 2.6 Mixture of Least Squares

#### Model

#### Results