

Addressing Redistribution in Bikeshare Using Cox Processes and Online Supervised Learning

Walter Dempsey
University of Chicago
5734 S. University Avenue
Chicago, IL
wdempsey@galton.uchicago.edu

ABSTRACT

A bicycle sharing system (bikeshare) is a service in which individuals can check-out bicycles from stations for a short period of time and drop them off at any of the stations in the system. The main purpose of bikeshare is to provide transportation for short trips, and it has become common in many cities including New York, Paris, and Beijing. One of the issues with bikeshare is the possible imbalance of bikes across stations, specifically that some stations may have no bikes to check-out while other stations have no slots for arriving bikes. Many cities employ re-distribution techniques on an ad hoc basis. In order to improve the redistribution process, we must estimate the expected number of bikes at a station at some point in the future, as well as the chance that the station becomes empty or full in that time window. To do this, we model the arrival and departure of bikes as a log Gaussian Cox Process. We build a block approximation of the underlying Gaussian processes by clustering neighborhoods, which will allow for parallel estimation. Moreover, we build online methods for updating the parameters from streaming station data. In order to assess performance, we apply these techniques to the Washington D.C. bikeshare data. We end with a brief discussion of how our approach can be used in analyzing new stations without sufficient data and as well as new networks/cities.

General Terms

Log Gaussian Cox Processes, Block-Approximate Gaussian Processes, Parallel Algorithms, Online Learning

1. INTRODUCTION

Bicycle Sharing systems have become ubiquitous over the past several years, providing an alternative mode of transportation and improving the connectivity of their respective cities. These networks (typically referred to as bikeshare) consist of a number of stations distributed throughout the city with individuals able to check-out bicycles from stations for a short period of time before dropping them off at an-

other station in the system. A common issue with bikeshare is that traffic patterns commonly result in bike imbalance across stations, with some stations being either empty or full. We call these **extreme** stations. This leads to delays as bikeshare users must find alternate stations, while also lowering overall ridership due to reliability concerns. Bikeshare systems tackle this issue by some form of redistribution; however, the methods for redistribution of bikes is ad hoc and may be sub-optimal.

For a given time in the future, we wish to predict the number of bikes at the station as well as the probability of the station becoming extreme at some point in this time window given its current state as well as auxiliary variables. This will allow us to predict extreme stations as well as provide an estimated optimal number of bikes per station during redistribution to avoid extreme events in the next time window.

We model arrival and departure of bikes at each station as a pair of correlated Poisson Processes. We propose then modeling the entire bikeshare system as a set of correlated Poisson Processes, where two pairs are correlated based on their spatial proximity. This set is therefore modeled as a log-Gaussian Cox Process in which we capture station-level, temporal, and spatial random effects.

This approach is computationally intensive, and therefore we propose approximation methods for estimation, which allow us to run estimation in parallel and provide necessary speedups at the cost of some inexactness. The bikeshare data analyzed is also streaming, and therefore we provide an online algorithm which updates parameter estimates given the new observations. This allows us to update models given new data without re-fitting the entire model everytime. We end with an application of our techniques to Washington D.C. bikeshare data. We analyze the performance, and provide methods for prediction of new stations given estimation from surrounding stations.

2. ARRIVAL AND DEPARTURE PROCESSES

We start by considering the temporal process at a given station. Let $Y_i(t)$ denote the number of bikes at the station at time t . The bike count is an integer-valued process with jumps. We use $Y(t)$ to define a pair of counting processes which we then model.

The **arrival counting process**, $N_i^a(t)$ can be defined as the number of bikes arriving at the station up to time t . As-

suming a finite number of bike arrival times in any interval, we can then define these counting processes as:

$$N^a(t) = \sum_{s < t} |Y_i(s) - Y_i(s-)| \mathbf{1}[Y_i(s) - Y_i(s-) > 0] \quad (1)$$

The **departure counting process**, $N_i^d(t)$ can be defined likewise for bikes leaving the station. A consideration is that a station which is empty cannot have departures. Therefore, $Y_i(t) = 0$ in some interval $[t, t+\delta]$ implies $N_i^d(t+\delta) - N_i^d(t) = 0$. Let A_d be the set of all such windows. Then, we assume the departure counting process to be an inhomogenous Poisson point process on the complement of this set, $\mathbb{R}^+ - A_d$. We consider the same to be true of the arrival process and the window of all times for which the station is full, A_f .

For now, we ignore these constraints on the process. We assume that at a station, the arrival and departure counting processes are correlated. This implies that we can model the pair as a Cox process where the rate parameters includes a station block effect. That is, $N_i^a(t)$ ($N_i^d(t)$ respectively) are poisson processes with rate parameter $\lambda_i^a(t)$ ($\lambda_i^d(t)$ respectively) where

$$\log(\lambda_i^a(t)) = \mu_i^a(t) + b_i \quad (2)$$

where b_i is a random station effect. The rate parameter for the departure process can be defined similarly.

3. BIKESHARE SYSTEM

We now consider the complete bikeshare system. Stations in close geographic proximity provide auxiliary information which should be leveraged. For example, if a station has many arrivals in a given morning, this is a strong indicator that the number of arrivals at a neighboring station will also be high. We wish to introduce the spatial dependency into our current model.

We use a log-Gaussian Cox Process to do this in which spatial dependence is captured through random effects in the stochastic rate parameters. Let $N_i(t) = (N_i^a(t), N_i^d(t))$ be the pair of counting processes for station i . Then we can assume that the arrival rate parameters are associated as are the departure rate parameters. So that

$$\log(\lambda_i^a(t)) = \mu_i^a(t) + b_i + Z(t; i) \quad (3)$$

where $Z(t; i)$ is a mean-zero gaussian process with covariance that may depend on both time t and space i . We can incorporate the station effects into the gaussian process model. We then consider a simple separable variance components model :

$$\begin{aligned} \text{cov}(Z(t; i), Z(t'; j)) &= \sigma_0^2 \delta_{tt'} \delta_{ij} + \sigma_1^2 \delta_{ij} \\ &+ \sigma_2^2 \exp\left(-\frac{d((i, t), (j, t))}{\lambda}\right) \end{aligned}$$

where $d((i, t), (j, t))$ is a specified distance metric. In our work we consider the l_2 distance. The first term is measurement error at time t at station i , the second is the random station effect, and the third is a general mean associated with all the arrival parameters.

Covariates can be chosen to account for temporal trends. We can choose a factor model for $\mu^a(t)$ which would model fixed effects attributable to time of day, month, and year, or

a basis expansion in terms of low-order Fourier terms. In this case, the distance metric will only depend on the location of the stations and the third term will only be included when t and t' are equal.

We assume a similar model for the departure rates. The random block effects link the arrival and departure processes at a given station, and the third term in the gaussian process links the rate parameters for the arrival processes across stations. The departure processes across stations are linked via a similar term, albeit different parameter values, (σ_2^2, λ') .

4. MAXIMUM LIKELIHOOD ESTIMATION

Suppose that there are n total stations. Now take the observations at a particular time window, $(t, t + \delta)$ (we model half-hour windows). The general form of the Cox Process Likelihood associated with the data, $Y(t) = ((Y_1^a(t), Y_1^d(t)), \dots, (Y_n^a(t), Y_n^d(t)))$ is:

$$\begin{aligned} l_t(\theta; Y(t)) &= \int_{\Lambda} P(X, \Lambda | \theta) d\Lambda \\ &= \int \prod_{i=1}^n \frac{\lambda_i^a \lambda_i^d}{Y_i^a Y_i^d} \frac{e^{-\lambda_i^a - \lambda_i^d}}{(2\pi)^n |\Sigma|^{1/2}} \exp\left((\lambda - \mu(\theta))^T \Sigma(\theta)^{-1} (\lambda - \mu(\theta))\right) d\lambda \\ &= E_{\lambda | \theta}[l^*(\lambda; Y(t))] \end{aligned}$$

where $l^*(\lambda; Y)$ is the likelihood for the inhomogeneous Poisson process, and λ is the vector of rate parameters for the dual processes at each station. As we incorporate temporal trends through the the model for the mean, μ , the complete likelihood is the sum over all time windows. Supposing we have T total time windows, (t_1, \dots, t_T) , then the complete likelihood is:

$$l(\theta; Y) = \sum_{i=1}^T E_{\lambda | \theta}[l^*(\lambda; Y(t_i))]$$

In our setting, the likelihood for each time window consists of estimating an expectation over the finite-dimensional distribution of λ . The number of stations is of the order of several hundred. Washington D.C. has roughly 250 stations, and therefore the dimension of our expectation is approximately 500. The high dimensionality of the integration appears formidable.

4.1 Monte Carlo Estimation

Standard Monte Carlo methods would employ the estimate:

$$l_{MC}(\theta) = s^{-1} \sum_{j=1}^s l(\theta | X, \lambda^{(j)}) \quad (4)$$

where $\lambda^{(j)}$ are simulated realisations of λ . This is highly inefficient. Based on the work of Geyer (1999) and Diggle et al. (2013), we use the following to provide robust, efficient estimation of the log-likelihood function. Given un-normalized joint density of X and λ , $f(X, \lambda | \theta)$, then

$$\hat{L}(\theta) = \log \left\{ s^{-1} \sum_{j=1}^s r(X, \lambda^{(j)}, \theta, \theta_0) \right\} \quad (5)$$

$$- \log \left\{ s^{-1} \sum_{j=1}^s r(X^{(j)}, \lambda^{(j)}, \theta, \theta_0) \right\} \quad (6)$$

where $r(X, \lambda, \theta, \theta_0) = f(X, \lambda|\theta)/f(X, \lambda|\theta_0)$. The result provides a Monte Carlo approximation to the log-likelihood function, and consequently the maximum likelihood estimate, $\hat{\theta}$, by simulating the process at a single value, θ_0 . The accuracy is a function of the number of simulations, s , and the proximity of θ_0 to θ .

The pair $(X^{(j)}, \lambda^{(j)})$ are simulated joint realisations of X and λ at $\theta = \theta_0$. In the first term, X is held fixed and $\lambda^{(j)}$ are conditional on X . Conditional simulation of λ requires Markov Chain Monte Carlo (MCMC) methods. Design, computational issues, and design are detailed by Diggle et al. (2013).

It rests to choose the estimate $\theta_0 = (\mu_0, \sigma_0^2)$. We estimate the parameters for each Poisson process independently to get initial estimates for the mean parameters, μ_0 . We take a simple average of the estimated mean squared errors for each station to provide an initial estimate of σ_0^2 . The starting estimates for the variance components is chosen such that $\sigma_j^2 = \frac{\sigma_1^2}{2}$ for $j = 2, 3$. This is ad-hoc, but unfortunately the choice of starting estimates in this case is less clear.

5. A PARALLEL APPROXIMATE ESTIMATION ALGORITHM

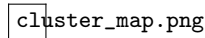
The assumed dependence among all counting processes makes maximum likelihood estimation computationally intensive. This is because, we must model the entire system of counting processes simultaneously and the dimension of the gaussian process becomes prohibitively large. For example, there are over 100 loading stations in Chicago for which we have almost a year of data. If we disaggregate the data to half-hour windows, the complete set of observations is over one-million. Given the computational cost of the high-dimensional integral, we cannot hope to estimate our model in a practical amount of time.

We start by assuming we capture the temporal dependence through a factor model. However, we still must worry about the spatial dependence as the dimension would still exceed 10,000. We do this by approximating the covariance of the gaussian process by a block-diagonal matrix. This exploits the weak dependence between stations that are far apart.

In order to approximate the gaussian process, we must determine the blocks. We do this by taking the station locations, and running a hierarchical k-means clustering on the set of locations using l_1 distances. The choice of l_1 is as a proxy for the true biking distance between points which commonly fall in a grid for cities.

In Figure ?? shows the hierarchical clustering applied to the Washington D.C. bikeshare stations. The clusters reflect the well defined neighborhoods

Figure 1: Hierarchical K-means Clustering

cluster_map.png

As we assume the clusters are independent, we can then model each separately. This allows us to run the Approximate Block Diagonal GP in parallel which provides addi-

tional computational savings.

SECTION OUTLINE

1. The Gaussian Process Leads to a Complete Graphical Model
2. However the dependency on the other far away nodes is negligible.
3. Therefore, we can approximate the Gaussian Process by an Approximate Block Diagonal GP
4. Similarity to CRFs (but we have a temporal component)
5. This allows us to fit models separately.
6. Problem is boundaries are not distinct
7. End with discussion of Estimation

6. ONLINE LEARNING: ALGORITHM AND PREDICTION

SECTION OUTLINE

1. Want a manner to update parameters in an online setting as the 'latest numbers' provide the most updated solutions
2. We want to do this in a very simple way
3. Use the online algorithms and approximate block models

7. PREDICTION

SECTION OUTLINE

1. Simple : Station with data and neighboring stations
2. Add a new Station: Need to infer the model for that station
3. Cluster Analysis
4. What about a new system? Assume the covariates are okay, then we still need to do a cluster analysis

8. EXPERIMENTAL RESULTS

8.1 Washington D.C. Bikeshare

SECTION OUTLINE

1. Explain the Complete Dataset

2. Include Maps
3. Explain how the ‘data’ is common to Chicago given the layout of the city but it will not be common to other cities
- 4.

8.2 Performance Indices

SECTION OUTLINE

1. How do we measure success?
2. We need good short term (‘15 min’ predictions) vs long term predictions of counts
3. We also want good prediction of the probability of being empty or full?
4. Need baseline and alternatives for comparison

9. SUMMARY