Imperial College London

Department of Computing

An Integrated London Journey Planner

Author:
Zhanzhan He

Supervisors:
Dr. Alessandra Russo
Dr. Luke Dickens

June 18, 2013
Submitted in part fulfillment of the requirements for the degree of Master of Engineering in Computing of Imperial College London

Abstract

Cycle hire schemes are an increasingly popular new mode of public transport. However, current journey planners, while able to calculate routes with respect to bus and train timetables either have no support for cycle hire or do not consider the availability of bikes or docks along the routes they find. While users can check how many bikes and docks are available at each docking station online before starting their journey, there is no way of knowing whether these bikes or docks will still be available by the time they reach their docking station. Additionally, current journey planners are unable to find routes that include cycle hire as part of a journey that includes taking a bus or train, relegating cycle hire to a second class mode of public transport.

We present a method of predicting future bike and dock availability by learning the distributions of the bike pickup and dropoff rates using a Poisson mixture model and show that it can make better predictions than previous models using a single Poisson distribution. We have integrated this prediction model into a full London journey planner for desktop and mobile devices which can find routes that include both cycling and the tube, while also considering other user preferences such as ascent averseness and road quietness.

Acknowledgements

I would like to thank the following people for their support on this project:

- Dr Alessandra Russo and Dr Luke Dickens. This project would not have been possible without their exceptional supervision.
- My housemates, friends and family for their support through thick and thin, particularly Dr David Birch for the benefit of his experience and wisdom.
- Ryszard Kaleta. For his support in helping me get started with his solid codebase.

"My flesh and my heart may fail, but God is the strength of my heart and my portion forever." Psalm 73:26

Contents

1 Introduction

Bicycle sharing schemes are a new mode of public transport with demonstrated social and environmental benefits. Londons Barclays Cycle Hire scheme has been growing in the number of journeys and is set to expand to cover more of London in 2013. See section 2 for more details.

Current London journey planning software incorporate a mixture of transport modes to help the user reach their destination. Popular planners such as Google maps [10] and Transport for London [22] are capable of incorporating Londons timetabled modes of public transport such as bus, train and underground alongside walking, driving and cycling. They are able to consider the state of the public transport network and find alternative routes if necessary.

Some journey planners are further able to take user preferences into account, especially those geared towards a specific mode of transport. For example optitrans [15] allows the user to select the specific modes of public transport they're willing to take and set the maximum distance they're willing to walk. CycleStreets [6] allows a choice between a fast route and a quiet route and provides advanced feedback such as calories burned, number of traffic lights en route and an elevation profile.

Journey planners that incorporate cycling however, tend to assume that the user owns a bike. Relatively few journey planners incorporate the Barclays Cycle Hire scheme as a mode of transport. TFL's Cycle Journey Planner [3] is a popular planner capable of this but has a number of drawbacks:

1. No other modes of transport (except walking to docking stations) can be incorporated into the route. This does not fit in with the cycle hire scheme's ethos as a solution to the last mile problem discussed in Section 2.2

2. The application does not integrate checking availability of bikes and empty spaces. The user must go to another web page and check the availability of bikes and empty spaces separately for each station along the route.

3. Some time is likely to elapse between the user planning the journey and the user arriving at the docking stations along the route. Even if they check the availability before the journey, by the time the user reaches their docking station, all the available bikes may have been picked up.

If they are able to pick up a bike, there may be no empty spaces left at the location they're meant to drop it off.

The last point is of particular interest as the cyclist suffers considerable inconvenience when they are unable to pick up or drop off a bike at a station. As described by Kaleta[30], there are some provisions to help the user when these circumstances occur:

- If there are no bicycles at the docking station, the passenger can use the docking stations map to locate other docking stations nearby. There is no guarantee there will be a bicycle available at those stations.
- If the docking station is full, the passenger can get up to 15 minutes extra time to cycle to another station before extra charges for late bicycle return start to apply. As above, there is no guarantee that there will be a parking space at the nearby stations.

The provisions are often insufficient. Delays in dropping off a bike are particularly problematic because of the potential financial penalty involved. Even if the user finds an alternative place to drop off their bike they will have to walk additional distance to their destination in a potentially unfamiliar area. As a consequence, they are likely to perceive both the journey planner software and the cycle hire scheme as unreliable. Popular mobile applications such as Cycle Hire Widget show the availability of bikes and empty spaces at nearby docking stations but this leaves the user to improvise part of the journey themselves.

To integrate Barclays Cycle Hire more comprehensively into a journey planner, we need to be able to predict whether bikes will be available at the time the user is due to reach their docking station to pick up or drop off their bike and direct the user to a docking station where there are likely to be bikes or free docks available. While it's not clear how to write a robust algorithm to do this directly, we have access to six months of Barclays Cycle Hire journey records (Section 2.4) which we use to learn an algorithm to predict the distribution of bike pickup and dropoff rates at each docking station throughout the day. Given we have observed n bikes at time t we can then sample the rate distributions to make a prediction of how many bikes will be available at time s. We build on the ideas and code presented by Kaleta [30] by presenting a Poisson mixture model capable of representing the rate distributions more accurately in a way that is statistically significant.

1.1 Contributions

1. A derivation of algorithms to learn the parameters of a regression model and a mixture model which are cable of more accurately representing the pickup and dropoff rates at each docking station (Chapter 6).

2. A statistical analysis of the data which challenges Kaleta's assumption of a constant pickup or dropoff rate within a small enough time interval and substantiates the applicability of our mixture model (Chapter 5).

3. A complete journey planner with our new prediction model integrated. We introduce new user preferences, an ascent averseness preference and an expected changeover time (Chapter 7). We demonstrate that it produces plausible routes with respect to user preferences in practice (Chapter 10).

4. New features that make the journey planner more informative, including the ability to generate route elevation profiles, the ability to check a docking station status by clicking a map marker and the marking of tube changeovers on the map (Chapter 8).

5. A mobile optimized version of the journey planner (Chapter 9) which we demonstrate works across across multiple platforms (Chapter 10).

2 Background

2.1 Terminology

- *Docking station* refers to the Barclays Cycle Hire terminals across London from which bicycles can be picked up or dropped off.
- *Pickup* refers to removing an available and functional bicycle from its docking station
- *Dropoff* refers to parking a bicycle at an available dock at a docking station
- *Transfer* refers to entering, exiting or switching between lines at a Tube station
- *Rollout* refers to the calculation of a possible future number of bikes at a docking station by sampling the probability distributions that describe the pickup and dropoff rates.

2.2 Bicycle Sharing Schemes

Bicycle sharing schemes allow users to access a shared fleet of bicycles. As of March 2011, there were 135 bicycle sharing schemes in 160 cities across Europe, Asia, North and South America operating over 235,000 shared bicycles [33]. While bicycle sharing schemes are not new, having been first introduced in Europe in 1965, they are virtually emission-free compared to personal vehicle use [33] and are increasingly viewed as a way to curb the negative social and environmental impacts of global motorization. Notably, cycle hire schemes provide an environmentally friendly solution to the 'last mile' problem, bridging the short distance between public transport stations and the home or workplace, which may otherwise be too far to walk [32].

However, many current popular journey planners such as Google maps[10], cyclestreets.net [6] and optitrans [15] do not incorporate the Barclays Cycle Hire scheme. Those that do, such as Transport for London's Cycle Journey Planner [3] either assume bike ownership or are not able to mix Barclays Cycle Hire with other modes of public transport. This presents a significant barrier to bike sharing being a fully integrated part of London's public transport system.

2.2.1 Barclays Cycle Hire

The Barclays Cycle Hire scheme has become a popular mode of public transport in London, with a total of 18 million hires made as of December 2012. This popularity is expected to increase as the scheme expands to the south of the Thames in late 2013, with an extra 300,000 hires expected to be made each month [20].

2.3 Sample stations for analysis and evaluation

We chose a sample of 20 London Barclays Cycle Hire stations to use to evaluate our machine learning model and router, as well as to use in our statistical analysis. We chose a mix of stations based on how busy they were (observed from the plots generated by the analytics module), including some busy stations like Waterloo Station 3 as well as relatively quiet stations like The Green Bridge, Mile End.

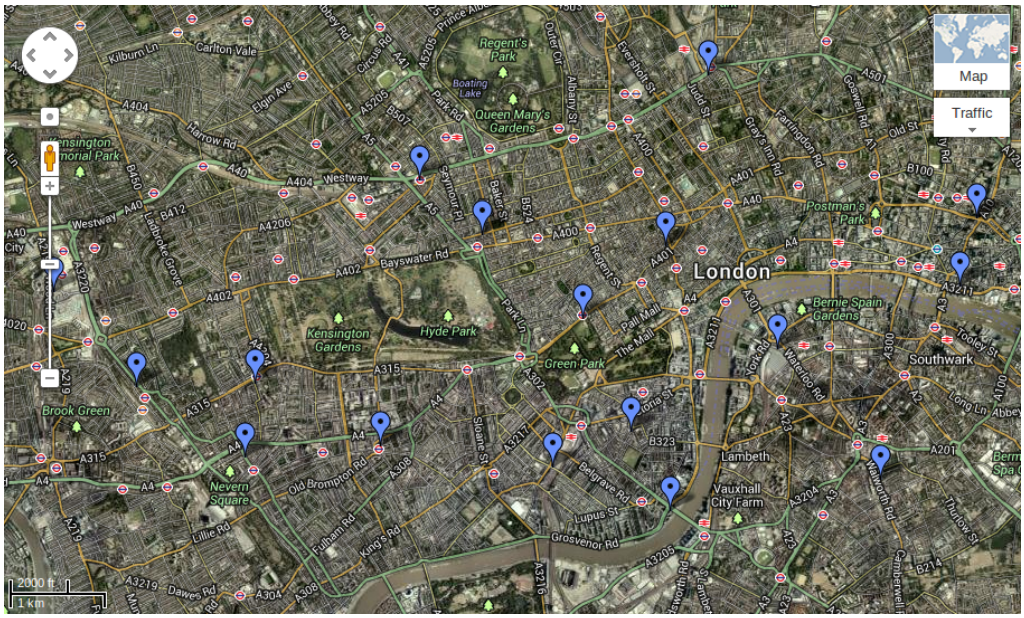


Figure 1: Sample stations across London for the evaluation of our machine learning models and router

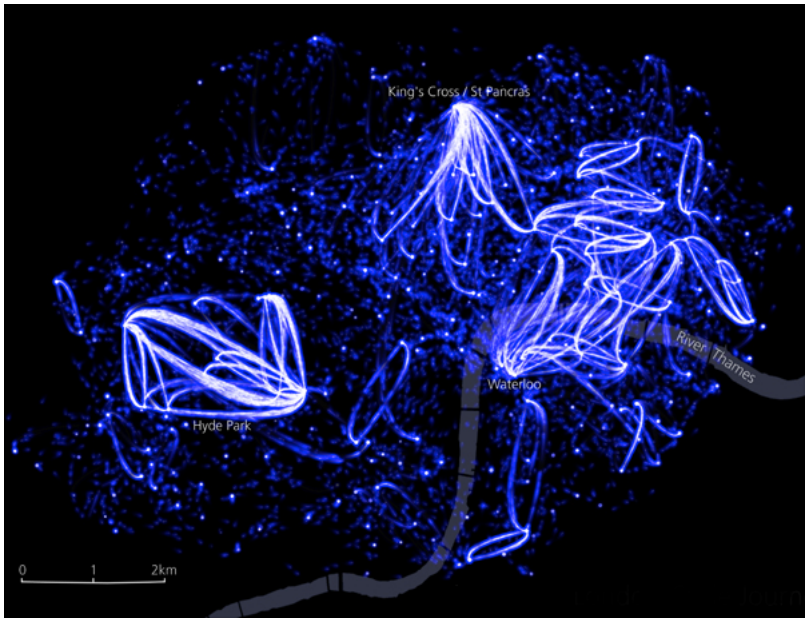


Figure 2: Example use of the Barclays Cycle Hire statistics, a visualization of London's Barclays Cycle Hire journeys by Jo Wood [23]

2.4 Available Data

2.4.1 Barclays Cycle Hire Statistics

Transport for London provide a variety of data from the London transport network for free through the developers area of their website [21]. Of particular interest are the Barclays Cycle Hire statistics, which contain the details of every cycle journey including the start time, finish time, start station and finish station. This data spans the 6 months between 1 February 2012 - 21 July 2012. From this we can extract the number of bike pickups and dropoffs at each station for every day in the date range.

In the work done by Kaleta [30] the cycle journey statistics for the first six months of operation of the Barclays Cycle Hire scheme were used. This did not include journey information for docking stations that started operation during or after these six months. Additionally the usage pattern of the Cycle Hire Scheme is likely to have evolved over the first six months as the public got used to the scheme, but this should have stabilized by the time the 2012 statistics were recorded. While it may have been useful have the first six months of data as training data for our machine learning models,

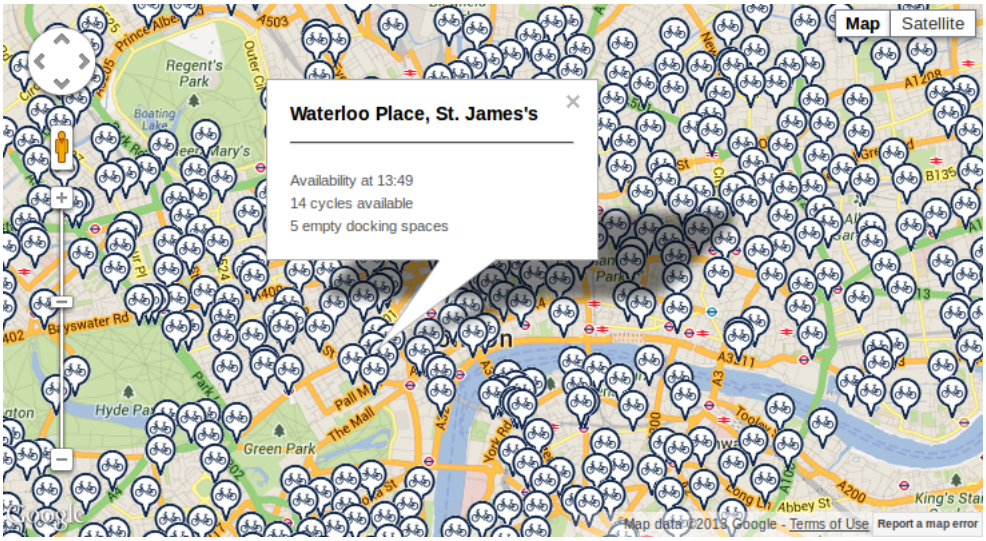


Figure 3: TFL Barclays Cycle Hire map, which shows bike availability using the live data feeds [1]

unfortunately these data are no longer available and the more recent six months of statistics should yield more representative results.

2.4.2 Barclays Cycle Hire Live Feeds

TFL provide live bicycle availability data for all the operational Barclays Cycle Hire docking stations in London, updated approximately every three minutes. This includes the number of available bikes, not including bikes that are locked or faulty and the number of available docking points.

2.4.3 London Underground Data

We continue with the same London Underground data used by Kaleta [30] from the same sources [14][7] [8]. This contains the name, latitude and longitude of every London Underground station, the lines that connect each station to the next and the travel times between them. These travel times are not very accurate, but are good enough for a proof of concept Journey Planner and can still easily be swapped for a better dataset.

We have augmented the database this to include adjustable transfer times between lines, as discussed in Section 7.3.1.

2.4.4 Greater London Data

We continued with the OpenStreetMap data used by Kaleta [30]. This is available in OSM XML *.osm* format and contains the latitude and longitudes of nodes for features of interest (e.g. junctions) and edges for the connections between them. Each edge contains information on:

- source and target nodes for the edge
- edge length
- the edge geometry, a list of (latitude, longitude) points which form the edge (edges are not necessarily straight lines)
- car accessibility
- bicycle accessibility
- foot accessibility

2.4.5 SRTM Height Data

The Shuttle Radar Topography Mission (SRTM) is an international project led by the National Geospatial-Intelligence Agency (NGA) and NASA. A radar system flew aboard Space Shuttle Endeavour for an 11 day mission in February 2000, obtaining elevation data for most of the world [17].

We were able to augment the OpenStreetMap Greater London data with SRTM elevation data using Osmosis [16], a command line application for processing OSM data. We used an SRTM plugin [19] for Osmosis which is able to interpolate the SRTM data to obtain node elevations for each OSM node.

2.4.6 Geocoding

- *Geocoding* is the process of finding geographical co-ordinates (e.g. latitude, longitude) given an address.
- *Reverse Geocoding* is the process of finding an address given geographical co-ordinates.

It is possible to augment OpenStreetMap data with raw address data and try to solve this problem ourselves, but fast, proven solutions are already available for free, such as the Google Maps API [9] and Nominatim [13].

Figure 4: Shaded Relief map of North America generated using the SRTM dataset [18]

2.5 Probability Distributions

2.5.1 Binomial Distribution

Bernoulli Trial An experiment with two possible outcomes, "success" and "failure" which has probability p of success.

Combination A way of selecting several objects from a larger group of objects, ignoring the order in which the objects are chosen.

k-combination A subset containing k distinct elements of a set S. The number of k-combinations of a set with N elements is the binomial coefficient

$$\binom{N}{k} = \frac{N!}{k!(N-k)!} \tag{1}$$

The binomial distribution is a discrete probability distribution with prob-

ability mass function

$$P(k|N,p) = \binom{N}{k} p^k (1-p)^{N-k} \tag{2}$$

It represents the probability of getting k "success" outcomes in a sequence of N Bernoulli trials with p probability of "success".

2.5.2 Poisson Distribution

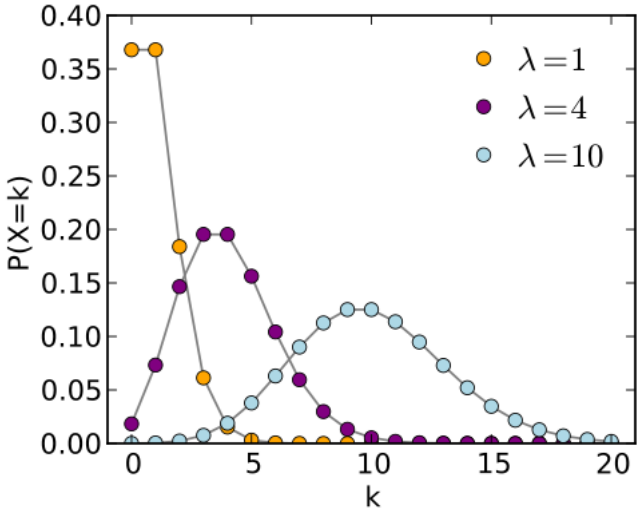


Figure 5: Plot of Poisson pmf for multiple values of k[4]

The Poisson distribution is a discrete probability distribution with a probability mass function which describes the probability of some number of events occurring within a fixed time interval. It is parameterized by the rate λ which can be thought of as the number of events expected to occur within that interval. It is the limiting case of the binomial distribution as $N \to \infty$. The probability of k events occurring in a time interval where λ event occurrences are expected is given by probability mass function:

$$P(k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{3}$$

For a discrete random variable X which follows a Poisson distribution the following holds:

$$\lambda = E(X) = var(X) \tag{4}$$

2.5.3 Gaussian Distribution

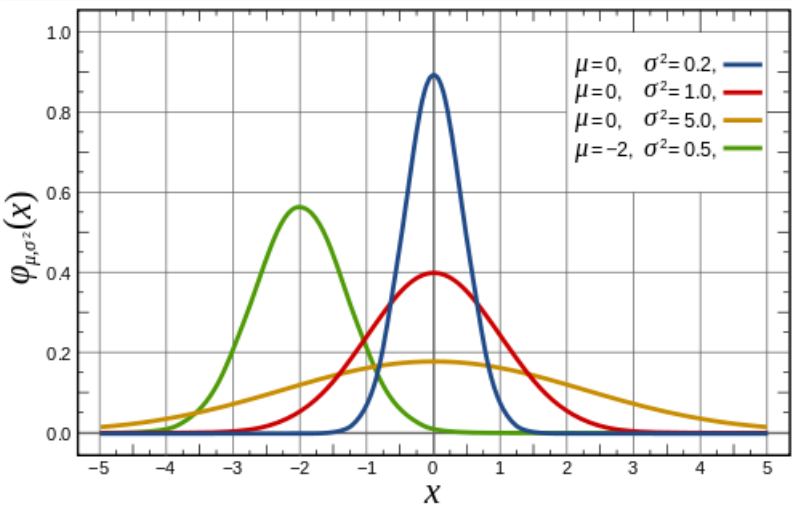


Figure 6: Plot of Gaussian pdf for multiple values of μ and σ^2 [5]

The Gaussian or normal distribution is a continuous probability distribution with probability density function:

$$\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} exp(-\frac{1}{2\sigma^2(x-\mu)^2}) \tag{5}$$

for a single variable x, where μ is the mean and σ^2 is the variance of x. In the multi variable case, the distribution takes the form:

$$\frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}} exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}))) \tag{6}$$

where x is the vector of variables, $\boldsymbol{\mu}$ is the D-dimensional vector of means and $\mathbf{\Sigma}$ is a D x D covariance matrix. The Gaussian distribution is used as a building block in many types of models. The sum of multiple random variables of any distribution, which is itself a random variable has a distribution that becomes closer to the Gaussian distribution as the number of terms in the sum increases (Central Limit Theorem) [25].

2.5.4 von Mises Distribution

Gaussian distributions may not be appropriate for models involving periodic variables. A example of a periodic variable in our journey planner would be

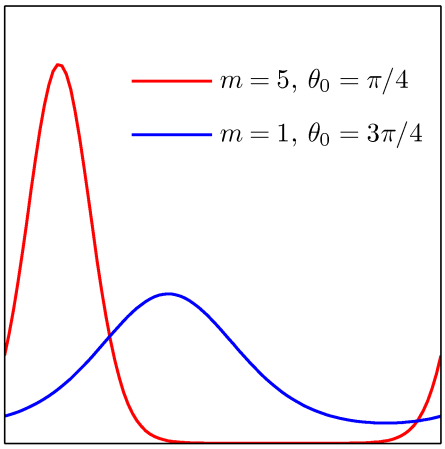


Figure 7: Plot of von Mises distribution for different m and θ_0 [25]

the number of bikes at a docking station over 24 hour periods. If we try to fit a Gaussian model to periodic data (see Section 2.7.1), the goodness of the fit will depend on the choice of origin for the data, which is arbitrary. The von Mises distribution is a periodic generalization of the Gaussian distribution which satisfies the following properties [25]:

$$p(\theta) \geq 0 \tag{7}$$

$$\int_0^{2\pi} p(\theta)d\theta = 1 \tag{8}$$

$$p(\theta + 2\pi) = p(\theta) \tag{9}$$

The distribution is given by:

$$p(\theta|\theta_0, m) = \frac{1}{2\pi I_0(m)} exp(mcos(\theta - \theta_0)) \tag{10}$$

where θ is an angle, θ_0 is the mean and m is analogous to the inverse variance of the Gaussian distribution.

2.6 Density Estimation

Density estimation aims to create a model of a probability distribution of a random variable given observations of that random variable. Kaleta's [30] model split each day into time intervals. Within each time interval, he used

density estimation to model the density of the bike pickups and dropoffs. He treated pickups and dropoffs as coming from a Poisson distribution used maximum likelihood estimation over the TFL statistics to estimate the pick-up/dropoff rate. We would like to experiment with more different models and also check whether the pickups and dropoffs within each interval really follow a poisson distribution.

2.6.1 Gaussian mixture models and expectation maximization

When we examine the cycle statistics data the frequency of pickups/dropoffs in an interval contains more than one peak for some docking stations. In these cases the density cannot be accurately modelled by a single Poisson or Gaussian distribution as these distributions have only one peak.

In this situation it may be appropriate to use a mixture model. A mixture model is a linear combination of more basic distributions. We could use a combination of Gaussians, which we can combine into a complex model if we can effectively adjust their means and covariances as shown in Figure 8. This is also a way of accounting for *latent* or *hidden* variables, which are not directly observed but can be inferred.

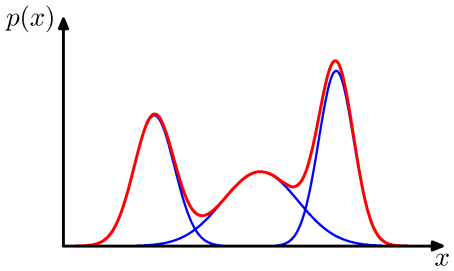


Figure 8: A Gaussian mixture model showing the three scaled Gaussian components in blue and the sum in red [25]

A *Gaussian mixture model* can be expressed as:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k G(\boldsymbol{x}|\mu_k, \Sigma_k) \tag{11}$$

Each Gaussian in the model has its own mean and covariance and the scaling factors π_k are known as mixing coefficients where K is the number of

Gaussians in the mixture. The π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are parameters to the mixture model which could be set using the expectation maximization algorithm [25]:

1. Choose initial values for the parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$.

2. Evaluate the responsibilities with the current parameters (π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$) using the equation:

$$\gamma(z_{nk}) = \frac{\pi_k G(\boldsymbol{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j G(\boldsymbol{x}_n|\mu_j, \Sigma_j)} \tag{12}$$

3. Update the parameters using the calculated responsibilities

$$\mu_{new}^k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \tag{13}$$

$$\sigma_{new}^k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_{new}^k)(\boldsymbol{x}_n - \boldsymbol{\mu}_{new}^k)^T \tag{14}$$

$$\pi_{new}^k = \frac{N_k}{N} \tag{15}$$

4. If parameters or the log likelihood has converged, stop. Otherwise return to step 2.

2.6.2 Model Comparison

Kaleta's [30] model only requires a single parameter, which may not necessarily fit the data as well as a complex model, but is unlikely to overfit. We need some method of comparing the models we come up with with the old one. One way to do this is using the *bayes factor* which can compare models with different numbers of parameters. Given training data set D and models M_1 and M_2 the *bayes factor* is defined by:

$$\frac{p(D|M_1)}{p(D|M2)} \tag{16}$$

where $p(D|M_1)$ is the likelihood of the data given model i. If the ratio is greater than 1 M_1 is favoured, otherwise there is more evidence for M_2

2.7 Regression

Regression [25] aims to predict the value of one or more target variables continuous target variables t given some input vector $x \in R^D$. Regression is a supervised learning problem where we have training data which consists of N observations of input vectors $\boldsymbol{x}_n$ with their corresponding target values t_n. The simplest approach is to create a function $y(\boldsymbol{x})$ which predicts the value of t for a given input vector $\boldsymbol{x}$. We aim to model a predictive distribution $p(t|\boldsymbol{x})$ which reflects our uncertainty about the value of t for all $\boldsymbol{x}$ and use it to make predict t for any value of $\boldsymbol{x}$.

We have the statistics of every cycle journey (See section 2.4) from which we can extract the number of bikes picked up or dropped off at every station over time. We can then use the number of pickups or dropoffs as a target value, and time as our input and use a regression model to predict the number of pickups or dropoffs for any time interval of a day.

2.7.1 Linear Regression

This a basic model that uses some linear combination of the input vector to predict the target values:

$$\begin{aligned} y(\boldsymbol{x}, \boldsymbol{w}) &= w_0 + w_1 x_1 + ... + w_D x_D \\ &= \boldsymbol{w}^T \boldsymbol{x} \end{aligned} \tag{17}$$

This fits some straight line through the data. To find the best fitting line we can define or derive a cost function and find the weights $\boldsymbol{w}$ that minimizes the value of some cost or error function using gradient descent. We can also form a likelihood model and find the $\boldsymbol{w}$ the likelihood of the model with respect to the training data.

2.7.2 Basis Functions

Real world data, such as bike dropoffs over time often cannot easily have a straight line fit through it. We can instead use a linear combination of non-linear functions of the inputs:

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\boldsymbol{x}) \tag{18}$$

Where M is the number of parameters in the model. The ϕ_j are basis functions and we can define $\phi_0 = 1$ to get a more convenient form:

$$\begin{aligned} y(\boldsymbol{x}, \boldsymbol{w}) &= \sum_{j=0}^{M-1} w_j \phi_j \\ &= \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \end{aligned} \tag{19}$$

Examples of basis functions we could use are polynomials, sigmoids and Gaussians.

2.7.3 Gradient Descent

To find the a weight vector w which will yield a reasonable prediction, we can define a cost function and try to minimize the value of the cost function over the training data. An example is the sum of squares error function, which is derived by treating the data as having come from a 'noisy sampling' of a function of the inputs. For all our N observations of pairs of corresponding inputs x and target values t we sum the squared difference between our prediction and the actual observation:

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, \boldsymbol{w}) - t_n)^2 \tag{20}$$

We can initialize $\boldsymbol{w}$ to some arbitrary value and optimize it with respect to the chosen cost function using gradient descent. The gradient is obtained by taking derivative of the cost function with respect to $\boldsymbol{w}$. Gradient descent takes a step along the negative gradient at each iteration until $\boldsymbol{w}$ converges:

$$\boldsymbol{w}^{i+1} = \boldsymbol{w}^i + \eta \nabla E(\boldsymbol{w}^i) \tag{21}$$

η is the learning rate which helps control the size of the step taken on each iteration. Provided $\eta > 0$, the algorithm will converge, but it may fail to find the optimum $\boldsymbol{w}$ as it can get stuck in a local minimum. Deriving a closed form solution using (Section 2.7.4. However sometimes it's very difficult or not possible to derive a closed form solution, and gradient descent can be faster for large datasets.

2.7.4 Maximum Likelihood Estimation

The least squares error function discussed in section 2.7.3 is derived by forming a likelihood model for the target values given the inputs. For example we can treat our observed data as being sampled from a function $y(\boldsymbol{x}, \boldsymbol{w})$ with noise added:

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon \tag{22}$$

If we treat the noise as Gaussian with precision β, we can derive a likelihood function:

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = G(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) \tag{23}$$

We can try and maximize the likelihood function over all the data. This quantity is a product of likelihoods. As log is a monotonic increasing function, we usually maximize the log likelihood instead. A closed form for solution for the maximum likelihood $\boldsymbol{w}$ that maximizes the likelihood of the model can be found by differentiating the likelihood function, setting the derivative equal to 0 and solving for $\boldsymbol{w}$. If we do this we get the *normal equation*:

$$\boldsymbol{w}_{ml} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \tag{24}$$

Where $\boldsymbol{\Phi}$ is the design matrix, of size $N \times M$ where N is the number of observations and M is the number of basis functions and $\Phi_{nj} = \phi_j(\boldsymbol{x_n})$.

In our journey planner there is good evidence that the number of bike pickups or dropoffs follow a Poisson distribution within small time intervals during the day. We could assume Gaussian noise and use the normal equation directly to fit some linear combination of basis functions to the data. Alternatively, we can form a likelihood model that treats the arrivals as having come from a noisy sampling of a Poisson distribution instead of a Gaussian and derive a closed form solution using maximum likelihood instead. Kaleta [30] found a closed formed solution using maximum likelihood to estimate the rate of a Poisson distribution in his prediction model.

2.7.5 Overfitting

More complex models have more parameters and can more accurately describe the underlying function that the training data was noisily sampled from. However, if the model is too complex it can overfit the data as shown

in Figure 9. A high order polynomial runs through every point in the training data and the value of the cost function will be 0. However, it is a poor representation of the function the data were sampled from. As a consequence it will generalize poorly and give poor predictions for inputs that are not in the training data set.

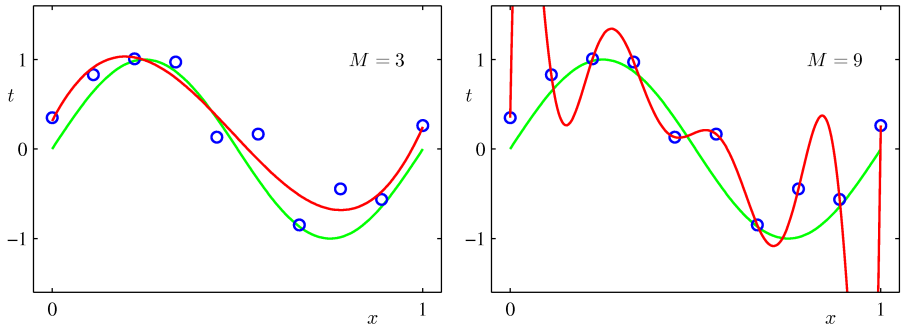


Figure 9: Illustration of overfitting. A noisy sampling of the green function generated the data points (in blue). The red function is plot of a model created using polynomial basis functions of order M [25]

.

2.7.6 Logistic Regression

Logistic regression is used for classification, by mapping the value predicted by linear model into the interval $[0, 1]$ using the logistic sigmoid function defined by:

$$\sigma(a) = \frac{1}{1 + e^{-a}} \tag{25}$$

If we have 2 classes C_1, C_2 in our data we can express the probability of C_1 as:

$$p(C_1|\phi(\boldsymbol{x})) = \sigma(\boldsymbol{w}^T \phi(\boldsymbol{x})) \tag{26}$$

One way to predict bike availability is to treat it as having 2 classes, a bike being available and no bike being available. We could form a linear model for the number of bikes over time, and use this linear model with logistic regression to get a probability of a bike being available.

2.8 Lagrange Multiplier

Using a Lagrange Multiplier [12] is a way to find local maxima and minima of a function $f(x, y, z)$ subject to some equality constraint, for example $g(x, y, z) = c$. If f and g both have continuous first partial derivatives, we introduce the Lagrange multiplier $\lambda_{lagrange}$ and consider the function:

$$\Lambda(x, y, z, \lambda_{lagrange}) = f(x, y, z) + \lambda_{lagrange}(g(x, y, z) - c) \tag{27}$$

We can then take partial derivatives with respect to x, y or z and setting them to zero finds us the maxima and minima in terms of $\lambda_{lagrange}$ which we can eliminate. We use this in our derivation of an expectation maximization algorithm for our Poisson mixture model discussed in Section 6.2.

3 Literature

3.1 An Integrated London Journey Planner

Kaleta [30] built a journey planner integrating walking, the Barclays Cycle Hire scheme and London Underground. Kaleta split each day into fixed time intervals and used the Poisson distribution to model the pickup and dropoff rate within each interval.

This assumes that the arrivals and departures within each time interval follow a Poisson distribution, and that the rate of that distribution does not change in the 15 minute intervals. Also, in a given time interval, it assumes that all arrivals occur first, followed by all the departures and does not take into account the different possible orderings of arrivals and departures within an interval.

One strength of approach is that it uses a model with only one parameter for each time interval. This reduces the problems of the Curse of Dimensionality and overfitting.

3.1.1 Learning the rate parameters

To predict whether bikes will be present, the day is split into 15 minute time intervals. The pickup and dropoff rate within each interval is assumed to follow a Poisson distribution. The maximum likelihood estimator for the rate parameter of a Poisson distribution turns out to be the sample mean of the observations of the number of events in a fixed time interval. We can calculate this separately for the pickups and dropoffs at each station using the Barclays Cycle Hire statistics. We can use these rates to predict the availability of bikes in the future.

3.1.2 Predicting future bicycle availability

When a prediction for the availability of bikes at time s is requested at time t, the number of bikes n available at time t is known. For each time interval between t and s, the Poisson distribution is sampled to estimate the number of pickups and dropoffs in that interval. For each interval between t and s we add the sampled number of pickups to and subtract the number of dropoffs from the number of bikes at the start of the interval (which is n at time$t$). The result represents a possible future state of the docking station which we call a *rollout*.

To make a prediction, Kaleta computes 1000 rollouts and divides the number of rollouts for which a bike was available at the end by 1000 to get the probability of a bike (Algorithm 1) being available. The same principle is applied to calculate the probability of an empty space being available at a docking station.

Algorithm 1 Kaleta's sampling of the Poisson distribution to predict bicycle availability

```
function PROB_BIKE_AVAILABLE(station_id, request_dt, journey_start_dt)
    counter = 0
    num_iterations = 1000
    timestep = 15 minutes
    next_interval_start_dt = request_dt
    curr_num_bikes = get_curr_num_bikes(station_id)
    for i = 1 → num_iterations do
        while next_interval_start_dt < journey_start_dt do
            num_pickups = get_scaled_pickups_mean(station_id, request_dt)
            num_dropoffs= get_scaled_dropoffs_mean(station_id, request_dt)
            drawn_pickups = poisson.rvs(num_pickups, size=1)
            drawn_dropoffs = poisson.rvs(num_dropoffs, size=1)
            curr_num_bikes = max(min(curr_num_bikes + drawn_dropoffs
                             - drawn_pickups, num_docks_all), 0)
            next_interval_start_dt += timestep
        end while
        if curr_num_bikes > 0 then
            counter += 1
        end if
        i += 1
    end for
end function
```

3.1.3 Cycle journey planning

In Kaleta's journey planner, the user sets inputs their risk averseness, a value between 0 and 1 where 1 is the most risk averse, along with the time they want to start their journey.

To calculate a complete cycle route with respect to the user's risk averseness, Kaleta's journey planner searches for the nearest docking stations to

the start and end points of the journey. It then uses the A* algorithm to find a cycling route between the docking stations, a walking route from the start point to the start docking station and a walking route from the end docking station to the end point. If the probability of a bike being available is less than the user's risk averseness, the route is rejected and the search is tried again from the next nearest docking stations to the start and end points.

3.1.4 Mixed route calculation

Kaleta calculates a route that mixes cycling and walking with respect to the time allowed by the user for the journey, input as a user preference. To compute routes with a cycling and tube portion, Kaleta began by calculating a tube route. The tube route is assumed to be the fastest possible route between any two points. If the overall time taken by the tube route is less than the time allowed by the user, the routing algorithm tries to create a slower route by taking stations off each end of the tube route (getting on the tube later or off it earlier). The extra distance is made up by cycling. The longest route that falls within the time allowed by the user is given to the user as the preferred mixed route.

3.2 Testing for a Poisson distribution

Kaleta [30] treated the number of pickups and dropoffs within a 15 minute interval of the day as following a Poisson distribution. We would like to verify statistically that this is indeed the case. Two hypothesis tests described by Brown [27] could provide evidence that the data follows a Poisson distribution.

3.2.1 Likelihood ratio test applied to a Poisson distribution

The *likelihood ratio test*'s null hypothesis is that all the observed data points came from the *same* Poisson distribution, and its alternative hypothesis is that each data point comes from *some* Poisson distribution with $\lambda > 0$. This can show that the data doesn't come from a Poisson distribution if the null hypothesis is rejected. However if the null hypothesis is accepted, we can only say there is not enough evidence to prove the data doesn't follow a

Poisson distribution. The test statistic is:

$$T_{LR} = 2\sum_{i=1}^{n} X_i ln\left(\frac{X_i}{\bar{X}}\right) \tag{28}$$

It is distributed as a Chi-squared variable with $n-1$ degrees of freedom, so we reject the null hypothesis when $T_{LR} > \chi^2_{n-1;1-\alpha}$.

3.2.2 Conditional Chi-squared test applied to a Poisson distribution

The *Conditional Chi-squared test*'s null hypothesis states that the data came from a uniform multinomial distribution, whereas its alternative hypothesis is that the data came from a Poisson distribution. This shows there is evidence to support the data coming from a Poisson distribution as opposed to a uniform distribution if the null hypothesis is rejected. The test statistic is:

$$T_{CC} = \frac{(\sum X_i - \bar{X})^2}{\bar{X}} \tag{29}$$

It is also distributed as a Chi-squared variable with $n-1$ degrees of freedom. We reject the null hypothesis if $T_{CC} > \chi^2_{n-1;1-\alpha}$. Neither test can prove beyond doubt that the data follows a Poisson distribution but can provide us with good evidence to believe whether it is or not.

3.2.3 Inter event arrival times

Another approach described by Neiman and Loewenstein [31] is based on the fact that the time between events follows an exponential distribution if the number of events in an interval follows a Poisson distribution. They check how well the inter event times fits an exponential distribution as well as how well they fit a straight line on a logarithmic scale.

3.3 Poisson mixture models

Church and Gale [28] describe how Poisson mixtures can be applied to estimate the probability distributions of words in text. Gaussian mixture models as discussed in section 2.6.1 provide a continuous probability distribution, but the frequency of bike pickups or dropoffs are discrete. A Poisson mixture

model might be more appropriate. Expectation maximization can be generalized to work with Poisson distributions. This could account for hidden variables such as day of week, weather and holidays that might affect the number of bikes picked up or dropped off in our prediction model.

3.4 Call Centre Literature

Most of the cost in running a call centre comes from paying the agents to answer calls. Queueing networks are commonly used to predict the performance of telephone call centres in advance so the agents can be allocated to provide the best service for the lowest cost. Queueing network models depend on knowing the arrival rate of calls. As arrival rates are unlikely to stay the same over a whole day, common practice is to split a day into time intervals, treating arrivals as following a Poisson distribution with a constant rate in each interval. The literature on predicting call arrival rates attempts to improve on this common model, which is of particular interest to us as this Poisson distribution based model is familiar from Kaleta's [30] project. The call arrival data they learn their models from is the same in principle as TFL's historical cycle journey statistics (See Section 2.4).

3.4.1 Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective

Brown et al. [26] treat call arrival rate as an inhomogeneous Poisson process. The calls are grouped into types and a hypothesis test is formulated to check whether the historical data on arrivals of each type of call really fit an inhomogeneous Poisson process. They fit a smoothing spline to the daily arrival rate data. The advantage of this approach is that it provides a continuous estimate for the arrival rate. This allows us to make the interval size adaptive. At points in the day where the arrival rate is changing a lot, smaller intervals can be used so that the rate changes less within each interval. However, splines require many parameters to fit, which requires more data to avoid overfitting and our data is likely to be insufficient.

3.4.2 Managing Uncertainty in Call Centres using Poisson Mixtures

Jongbloed and Koole [29] note that the assumption that the call arrival rate remains constant in a small time interval is incorrect. Some uncertainty in the arrival rate can be explained by weekly variations and changes, but this cannot explain all the variability. It is possible to try and account for more variables, but this complicates the analysis and may not be useful in practice. An example is weather, which could be accounted for but can only be predicted a week in advance whereas agent rosters may need to be published further in advance. Instead of assuming a single Poisson distribution with a fixed rate within each time interval, a Poisson mixture model is used. This helps to account for hidden parameters and the *overdispersion* in the data caused by the change in rate within that interval. Again though, it is not clear whether the improvement in fit is worth the increase in Model complexity, especially in situations like ours where data is scarce.

4 Software architecture

We continued with the same technologies used by Kaleta and built our journey planner on top of his journey planner implementation. One of the major challenges we faced in this project was understanding the substantial existing codebase. We introduced a set of end-to-end tests of the router, which calculate 13 routes of varying length in London and check them for sanity. This allowed us to refactor existing code and introduce new features with confidence that we weren't breaking the existing routing engine.

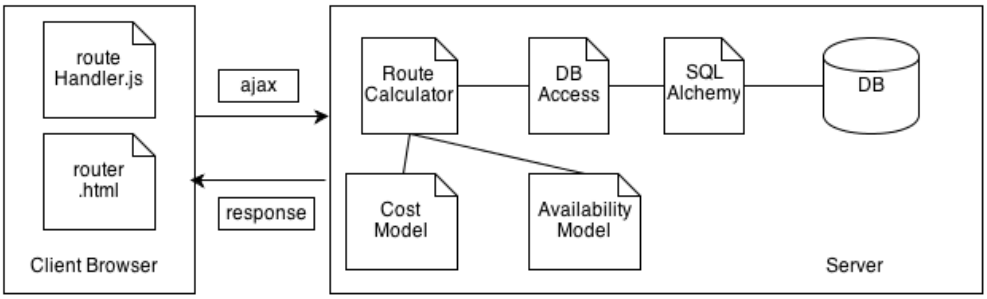


Figure 10: Simplified architecture of the existing system

.

The existing system used the SQLAlchemy object relational mapping to query with the database. A database access layer was built on top of it to select the relevant data and map it from SQLAlchemy objects into python data structures. The route calculation was built on top of the database access layer, with abstractions for the edge cost models and the bike availability prediction models used for routing (Figure 10).

Displaying the map and marking the routes was done on the client side in Javascript, which makes an AJAX request to the server with the user's preferences gathered from the user interface.

We added what new database schema and data selection functions we needed to the database access layer, but our primary modifications were to the route calculation, the edge cost and bike availability modules. We also added a new client webpage for the mobile application which largely shared the same code for marking routes on the map as the main webpage (routeHandler.js).

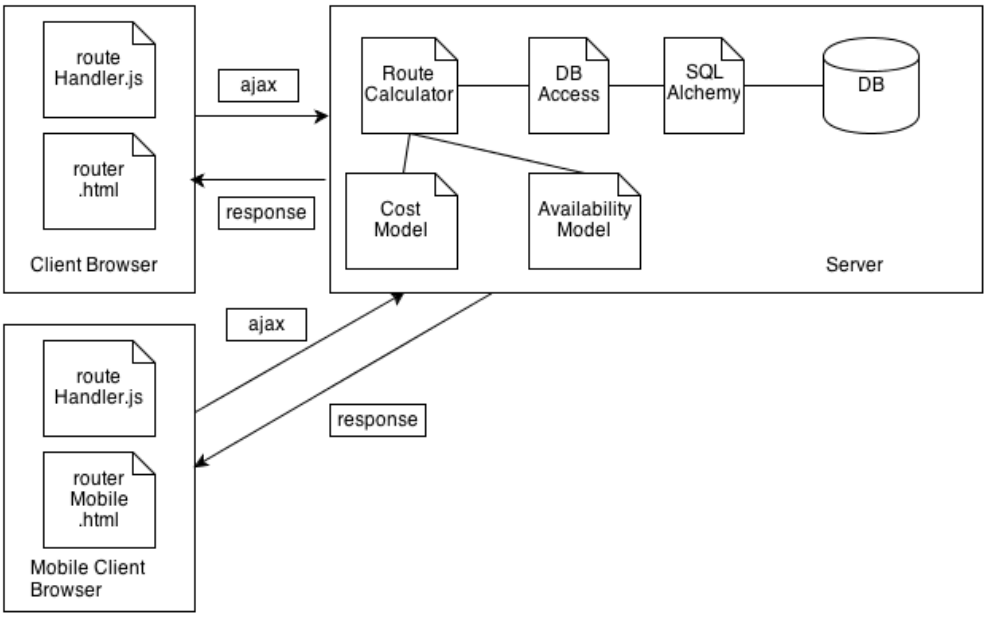


Figure 11: Adding the mobile client

.

5 Analysis of the Barclays Cycle Hire statistics

5.1 Plotting Module

We created a simple analytics module able to generate plots over the Barclays Cycle Hire statistics. This was made with a web based interface so anyone could generate and view plots over the Barclays Cycle Hire data. See Appendix E to see how it is used. These plots helped us familiarize ourselves with the shape of the data and inspired our on Machine Learning models.

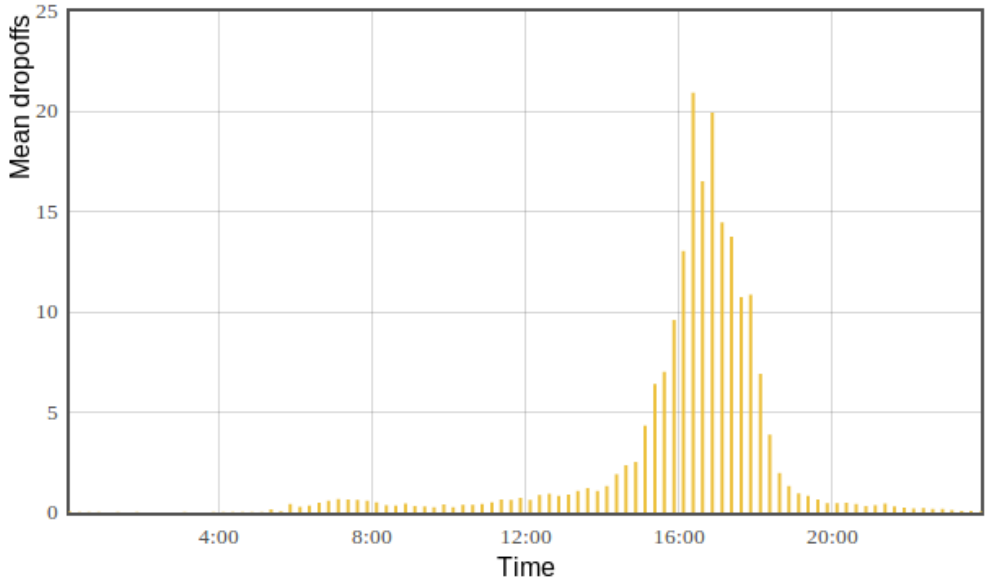
5.1.1 Mean number of events

We made it possible for users to generate plots of the mean number of pickups or dropoffs at any station over a single day or a week. Kaleta's [30] Machine Learning model relied on the assumption that the pickups and dropoffs within a 15 minute interval followed a Poisson distribution with a constant rate λ. Kaleta's evaluation had already noted that this wasn't necessarily true for every 15 minute time interval at every station.

Plotting the mean numbers of pickups and dropoffs over all the days of data, we saw some patterns emerge. Many stations have a pronounced peak in the number of dropoffs and number of pickups at a particular time of day. For some stations, such as Vauxhall Bridge, Pimlico and Waterloo Station 3, the number of pickups peak in the morning and the number of dropoffs peaks in the evening, as shown in Figure 12 and 13.

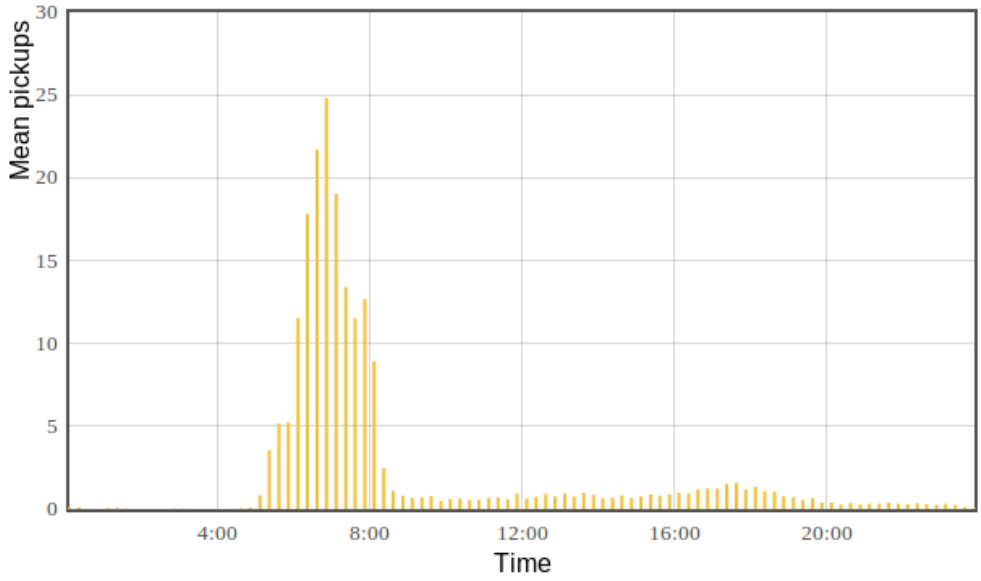
The stations that follow this usage pattern include the major transport hubs such as Waterloo Station and Belgrove Street, King's Cross. This provides evidence that Barclays Cycle Hire is being used as a solution to bridge the 'last mile' (Section 2.2) in their commute to and from work.

Figure 12: Waterloo Station 3 mean dropoffs over all the days of data



.

Figure 13: Waterloo Station 3 mean pickups over all the days of data



.

5.1.2 Frequency Density of events in a given interval

For a given time interval, specified by the user we plotted the frequency density with which different numbers of events occurred.

As the interval size is increased, we observed the overdispersion described by Jongbloed and Koole [29]. However even in a 15 minute interval, as used by Kaleta, the number of pickups and dropoffs often did not look like it followed a Poisson distribution.

5.2 Testing for a Poisson Distribution

The shape of the frequency density data in Section 5.1.2 and the results of our regression model described in Section 6 gave us reason to doubt that it was reasonable to assume that the pickup and dropoff rate in a 15 minute time interval would follow a Poisson distribution. We did the likelihood ratio test and the conditional chi-squared test described in section 3.2 on the Barclay Cycle Hire Statistics. We used 15 minute time intervals over the 20 sample stations discussed in section 2.3. We chose 3 times of day to do the tests and chose a 1% significance level:

- *Morning* 8am-9am
- *Evening* 5pm-6pm
- *Afternoon* 2pm-3pm

These are the times of day that are interesting to us from looking at the daily trends in pickup and dropoff rate. The morning and afternoon periods are the ones which tend to have a high pickup and dropoff rate and where there is likely to be a change in rate even within a 15 minute interval. The afternoon period is less busy for most stations and is more likely to have a constant rate within each time interval.

For 59.6% of the dropoff likelihood ratio tests and 58.8% of the pickup likelihood ratio tests, we rejected the null hypothesis that the data is from a single Poisson distribution with rate λ. Additionally, for 28.8% of the dropoff chi-squared tests and 25% of the pickup chi-squared tests we accepted the null hypothesis that the data came from a multinomial uniform distribution rather than a Poisson distribution. From this evidence we decided to drop the assumption that the data was from a single Poisson distribution and

formulated the Poisson mixture model described in section 6.2. A complete listing of the results is available in Appendix A.

5.3 Finding the optimal number of Poissons to use in the mixture model

We wanted to verify statistically whether a Poisson mixture model would be an improvement on using a single Poisson distribution to model the number of pickups and dropoffs within a time interval. A mixture of k Poisson distributions will always explain the data at least as well as a mixture of $k-1$ Poissons, which is a special case of a mixture of k Poissons. To test whether the improvement in likelihood between a mixture of j Poissons and a mixture of $j+1$ Poissons is statistically significant, we first fit both models to the data and compute the likelihood of the data being generated by each model. We then compute the test statistic for the observations within a given time interval:

$$T_{LR} = -2ln\left(\frac{likelihoodofmixtureofjPoissons}{likelihoodofmixtureofj+1Poissons}\right) \tag{30}$$

which we compare to a chi-squared distribution with 2 degrees of freedom, as a mixture of $j+1$ Poissons has 2 more parameters than a mixture of j parameters. If the result indicates the difference is significant at the 1% level, we repeat this for $j+1$ and $j+2$ Poissons and so on, until we find some k such that the Poisson mixture with $k+1$ parameters is not significantly more likely than the Poisson mixture with k parameters at the 1% level. k is then the optimal number of Poissons to use for that time interval. Note that a mixture of 1 Poisson is the same as Kaleta's method. We performed this process for our 20 sample stations using 15 minute time intervals and the same morning, afternoon and evening time periods discussed in section 5.2. We found that 54% of the pickup intervals and 55% of the dropoff intervals could have their pickups and dropoffs explained significantly better by a mixture of more than 1 Poisson distribution. Furthermore, busy stations like Belgrove Street, King's Cross and Waterloo Station 3 had an optimal mixture of up to 4 Poissons. This fits with the idea that the data is overdispersed due to the change in rate within an interval, as discussed in section 3.4.2. A complete listing of the results is available in Appendix B.

6 Predicting Bicycle Availability

As discussed in Section 3.1, Kaleta [30] built a prediction model for the availability of bikes by modeling both arrivals and departures of bikes as a Poisson process with a constant rate within 15 minute intervals. These rates were learned using the Barclays Cycle Hire statistics which contained details of every bike journey made in the first six months of the scheme. We aimed to use the more recent set of Barclays Cycle Hire statistics to build a more representative model for better predictions.

6.1 Regression

Our first idea to refine Kaleta's approach was to try linear regression on the bicycle pickup and dropoff rates across the day. Continuing with the assumption that the pickup and dropoff rates follow a Poisson distribution within some small time interval, we wanted to be able to predict the rate parameter λ for any given time of day. Instead of having fixed 15 minute intervals, this would allow us to take samples in smaller intervals for times of day where the rates are changing more quickly.

Instead of treating the data points as having been generated by a Gaussian distribution with some mean μ for any given time of day, we treat them as having come from a Poisson distribution with mean λ. This allows us to find predict the rate for any input time of day.

6.1.1 Derivation

Treat the target variable t as coming from a Poisson distribution with rate λ where λ comes from a linear model $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})$. The likelihood of target variable t given inputs $\boldsymbol{x}$ and weight vector $\boldsymbol{w}$:

$$p(t|\boldsymbol{x}, \boldsymbol{w}) = Poi(t|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})) \tag{31}$$

If we have N training observations, the likelihood over all the data (represented N-dimensional target value vector $\boldsymbol{t}$ and matrix of inputs $\boldsymbol{X}$) is given by:

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}) = \prod_{n=1}^{N} Poi(t_n|\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)) \tag{32}$$

Expanding with using the pmf of the Poisson distribution leads to:

$$p(\boldsymbol{t}|\boldsymbol{X},\boldsymbol{w}) = \prod_{n=1}^{N} \frac{(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))^{t_n} e^{-\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)}}{t_n!} \tag{33}$$

The likelihood is always conditioned on $\boldsymbol{X}$ so we drop it for convenience. As ln is a monotonic increasing function, maximising the likelihood is equivalent to maximising the log-likelihood.

$$\begin{aligned}
ln(p(\boldsymbol{t}|\boldsymbol{w})) &= \sum_{n=1}^{N} ln\left(\frac{(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))^{t_n} e^{-\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)}}{t_n!}\right) && (34)\\
&= \sum_{n=1}^{N} ln\left((\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))^{t_n} e^{-\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)}\right) - ln(t_n!) && (35)\\
&= \sum_{n=1}^{N} t_n ln(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)) - (\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))ln(e) - ln(t_n!) && (36)\\
&= \sum_{n=1}^{N} t_n ln(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)) - \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n) - ln(t_n!) && (37)
\end{aligned}$$

Formula for differentiation with respect to a vector $\boldsymbol{x}$

$$\frac{\partial f}{\partial \boldsymbol{x}} = \left(\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}\right) \tag{38}$$

Differentiate $ln(p(\boldsymbol{t}|\boldsymbol{w}))$ with respect to $\boldsymbol{w}$ to get the gradient of the log likelihood. M is the number of basis functions.

$$\begin{aligned}
\frac{\partial t_n ln(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))}{\partial \boldsymbol{w}} &= \left(\frac{\partial t_n ln(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))}{\partial w_1}, ..., \frac{\partial t_n ln(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n))}{\partial w_m}\right)\\
&= \left(\frac{\partial t_n ln(\sum_{k=1}^{M} w_k\phi_k(\boldsymbol{x}_n))}{\partial w_1}, ..., \frac{\partial t_n ln(\sum_{k=1}^{M} w_k\phi_k(\boldsymbol{x}_n))}{\partial w_m}\right)\\
&= \left(\frac{t_n\phi_1(\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)}, ..., \frac{t_n\phi_m(\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)}\right)\\
&= \frac{t_n\boldsymbol{\phi}(\boldsymbol{x}_n)^T}{\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_n)} && (39)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)}{\partial \boldsymbol{w}} &= \left(\frac{\partial \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)}{\partial w_1}, ..., \frac{\partial \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)}{\partial w_m} \right) \\
&= \left(\frac{\partial \sum_{k=1}^{M} w_k \phi_k(\boldsymbol{x}_n)}{\partial w_1}, ..., \frac{\partial \sum_{k=1}^{M} w_k \phi_k(\boldsymbol{x}_n)}{\partial w_m} \right) \\
&= (\phi_1(\boldsymbol{x}_n), ..., \phi_m(\boldsymbol{x}_n)) \\
&= \boldsymbol{\phi}(\boldsymbol{x}_n)^T \qquad (40)
\end{aligned}$$

$$\frac{\partial ln(t_n!)}{\partial \boldsymbol{w}} = \mathbf{0} \qquad (41)$$

Combining results 38, 39 and 40, we get:

$$\nabla ln(p(\boldsymbol{t}|\boldsymbol{w})) = \sum_{n=1}^{N} \frac{t_n \boldsymbol{\phi}(\boldsymbol{x}_n)^T}{\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)} - \boldsymbol{\phi}(\boldsymbol{x}_n)^T \qquad (42)$$

6.1.2 Results

This model did not give us the improved predictions we expected, doing worse than Kaleta's prediction algorithm (Section 10.1). This led us to challenge the assumption that the data comes from a Poisson distribution and do the statistical analysis discussed in section 3.2.

6.2 Poisson mixture model

From our statistical analysis, we saw that for most stations there was good evidence that the data did not follow a Poisson distribution, so the assumption that all the data was generated by a Poisson distribution with some mean λ was not reasonable. Jongbloed and Koole [29] note that for call centre data, the rate of call arrivals changes even within short intervals. This results in the number of calls within each time interval being overdispersed. They used a Poisson mixture model to account for this. With further testing, we found that a mixture of more than one Poisson distribution explained the data better than a single Poisson distribution statistically. We created a Poisson mixture model and learned its parameters within each 15 minute interval instead. We present our own derivation of an expectation maximization algorithm to find the parameters of a k Poisson mixture model.

6.2.1 Derivation

Treat the arrivals and departures in a time interval as a mixture of Poisson distributions in the form:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k Poi(\boldsymbol{x}|\lambda_k) \tag{43}$$

Let $\boldsymbol{z}$ be a K-dimensional binary random variable represented by a one of k vector where a particular z_k is equal to 1 and all other elements are zero. The mixing coefficients π_k specify the marginal distribution over $\boldsymbol{z}$

$$p(z_k = 1) = \pi_k \tag{44}$$

Where the following properties hold:

$$0 \leq \pi_k \leq 1 \tag{45}$$

$$\sum_{k=1}^{K} \pi_k = 1 \tag{46}$$

$\boldsymbol{z}$ is a one of K representation so the distribution over $\boldsymbol{z}$ can be written:

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} (\pi_k)^{z_k} \tag{47}$$

The conditional value of $\boldsymbol{x}$ for a particular value of $\boldsymbol{z}$ is therefore:

$$p(\boldsymbol{x}|z_k = 1) = Poi(\boldsymbol{x}|\lambda_k) \tag{48}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} Poi(\boldsymbol{x}|\lambda_k) \tag{49}$$

We can now calculate the joint distribution:

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{k=1}^{K} (\pi_k)^{z_k} Poi(\boldsymbol{x}|\lambda_k)^{z_k} \tag{50}$$

And obtain the marginal distribution of x by summing over the possible $\boldsymbol{z}$

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{z}) p(\boldsymbol{x}|\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k Poi(\boldsymbol{x}|\boldsymbol{\lambda}) \tag{51}$$

It is then possible to calculate the probability of $\boldsymbol{z}$ given $\boldsymbol{x}$ using Bayes' Theorem:

$$\begin{aligned} p(z_k = 1|\boldsymbol{x}) &= \frac{p(z_k = 1) p(\boldsymbol{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1) p(\boldsymbol{x}|z_j = 1)} && (52) \\ &= \frac{\pi_k Poi(\boldsymbol{x}|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(\boldsymbol{x}|\lambda_j)} && (53) \\ &= \gamma(z_k) && (54) \end{aligned}$$

This $\gamma(z_k)$ is referred to as the responsibility that component k of the mixture takes for explaining x.

Now if we let $\boldsymbol{X}$ be an $N \times 1$ matrix of observations of $\boldsymbol{x}$ the likelihood function is given by:

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{\lambda}, \boldsymbol{\pi}) &= \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k Poi(\boldsymbol{x}|\lambda_k) && (55) \\ ln\left(p(\boldsymbol{X}|\boldsymbol{\lambda}, \boldsymbol{\pi})\right) &= \sum_{n=1}^{N} ln\left(\sum_{k=1}^{K} \pi_k Poi(\boldsymbol{x}|\lambda_k)\right) && (56) \end{aligned}$$

This can be differentiated with respect to λ_k to find a maximum likelihood expression for λ_k

$$\frac{\partial \sum_{n=1}^{N} ln\left(\sum_{k=1}^{K} \pi_k Poi(\boldsymbol{x}|\lambda_k)\right)}{\partial \lambda_k} = \frac{\partial \sum_{n=1}^{N} ln\left(\sum_{k=1}^{K} \pi_k \frac{\pi_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!}\right)}{\partial \lambda_k} \tag{57}$$

Using the chain rule with $u = \sum_{k=1}^{K} \pi_k \frac{\pi_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!}$ and applying the product

rule to differentiate u we obtain:

$$\sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \frac{\pi_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \frac{\sum_{k=1}^{K} \pi_k \left(-\lambda_k^{x_n} e^{-\lambda_k} + e^{-\lambda_k} x_n \lambda_k^{x_n - 1}\right)}{x_n!} \tag{58}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \left(-\lambda_k^{x_n} e^{-\lambda_k} + e^{-\lambda_k} x_n \lambda_k^{x_n - 1}\right)}{\frac{1}{x_n!} \left(\sum_{k=1}^{K} \pi_k \lambda_k^{x_n} e^{-\lambda_k}\right) x_n!} \tag{59}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \left(-\lambda_k^{x_n} e^{-\lambda_k} + e^{-\lambda_k} x_n \lambda_k^{x_n - 1}\right)}{\sum_{k=1}^{K} \pi_k \lambda_k^{x_n} e^{-\lambda_k}} \tag{60}$$

We can divide both the top and the bottom by $x_n!$ to get a more familiar form:

$$\sum_{n=1}^{N} \frac{\frac{\pi_k \left(-\lambda_k^{x_n} e^{-\lambda_k} + e^{-\lambda_k} x_n \lambda_k^{x_n - 1}\right)}{x_n!}}{\frac{\sum_{k=1}^{K} \pi_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{61}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \frac{\left(-\lambda_k^{x_n} e^{-\lambda_k} + e^{-\lambda_k} x_n \lambda_k^{x_n - 1}\right)}{x_n!}}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{62}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{e^{-\lambda_k} x_n \lambda_k^{x_n - 1}}{x_n!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{63}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{\lambda_k^{x_n - 1} e^{-\lambda_k}}{(x_n - 1)!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{64}$$

$$= \sum_{n=1}^{N} \frac{\pi_k \left(-Poi(x_n|\lambda_k) + Poi(x_n - 1|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{65}$$

We can see that the first term inside the sum is the responsibility γ for

x_n

$$\sum_{n=1}^{N} \frac{\pi_k \left(-Poi(x_n|\lambda_k) + Poi(x_n - 1|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{66}$$

$$= \sum_{n=1}^{N} -\frac{\pi_k \left(Poi(x_n|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} + \frac{\pi_k \left(Poi(x_n - 1|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{67}$$

$$= \sum_{n=1}^{N} -\gamma(z_{nk}) + \frac{\pi_k \left(Poi(x_n - 1|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{68}$$

Now we can set the first derivative to zero to derive an expression for λ_k:

$$0 = \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{\lambda_k^{x_n - 1} e^{-\lambda_k}}{(x_n - 1)!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{69}$$

$$\tag{70}$$

Multiplying both sides by λ_k we get:

$$0 = \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{\lambda_k^{x_n} e^{-\lambda_k}}{(x_n - 1)!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{71}$$

$$0 = \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{x_n \lambda_k^{x_n} e^{-\lambda_k}}{x_n (x_n - 1)!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{72}$$

$$0 = \sum_{n=1}^{N} \frac{\pi_k \left(\frac{-\lambda_k \lambda_k^{x_n} e^{-\lambda_k}}{x_n!} + \frac{x_n \lambda_k^{x_n} e^{-\lambda_k}}{x_n!}\right)}{\sum_{k=1}^{K} \pi_k \frac{\lambda_k^{x_n} e^{-\lambda_k}}{x_n!}} \tag{73}$$

This can again be expressed in terms of the Poisson probability mass function:

$$0 = \sum_{n=1}^{N} \frac{\pi_k \left(-\lambda_k Poi(x_n|\lambda_k) + x_n Poi(x_n|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{74}$$

$$0 = \sum_{n=1}^{N} -\lambda_k \frac{\pi_k \left(Poi(x_n|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} + x_n \frac{\pi_k \left(Poi(x_n|\lambda_k)\right)}{\sum_{k=1}^{K} \pi_k Poi(x_n|\lambda_k)} \tag{75}$$

$$0 = \sum_{n=1}^{N} -\lambda_k \gamma(z_{nk}) + x_n \gamma(z_{nk}) \tag{76}$$

Now we denote $\sum_{n=1}^{N} \gamma(z_{nk})$ as N_k

$$\sum_{n=1}^{N} \lambda_k \gamma(z_{nk}) = \sum_{n=1}^{N} x_n \gamma(z_{nk}) \tag{77}$$

$$\lambda_k \sum_{n=1}^{N} \gamma(z_{nk}) = \sum_{n=1}^{N} x_n \gamma(z_{nk}) \tag{78}$$

$$\lambda_k = \frac{1}{N_k} \sum_{n=1}^{N} x_n \gamma(z_{nk}) \tag{79}$$

Next, we can derive an expression for π_k by differentiating with respect to π_k. A Lagrange multiplier is used to take care of the constraint that the mixing coefficients π_k sum to one. We maximise:

$$ln(p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\lambda})) + \lambda_{lagrange} \left(\left(\sum_{k=1}^{K} \pi_k \right) - 1 \right) \tag{80}$$

To do this we first compute the derivative:

$$\frac{\partial}{\partial \pi_k} ln(p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\lambda})) + \lambda_{lagrange} \left(\left(\sum_{k=1}^{K} \pi_k \right) - 1 \right) \tag{81}$$

Using the chain rule, we get:

$$\sum_{n=1}^{N} \frac{Poi(x_n|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(x_n|\lambda_k)} + \lambda_{lagrange} \tag{82}$$

If we set the derivative to zero, we can solve for $\lambda_{lagrange}$

$$0 = \sum_{n=1}^{N} \frac{Poi(x_n|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(x_n|\lambda_k)} + \lambda_{lagrange} \tag{83}$$

Multiply through by π_k to get:

$$0 = \sum_{n=1}^{N} \frac{\pi_k Poi(x_n|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(x_n|\lambda_k)} + \pi_k \lambda_{lagrange} \tag{84}$$

$$\pi_k \lambda_{lagrange} = -\sum_{n=1}^{N} \frac{\pi_k Poi(x_n|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(x_n|\lambda_k)} \tag{85}$$

$$\pi_k \lambda_{lagrange} = -\sum_{n=1}^{N} \gamma(z_{nk}) \tag{86}$$

Sum both sides over all k to get:

$$\sum_{k=1}^{K} \pi_k \lambda_{lagrange} = -\sum_{k=1}^{K} \sum_{n=1}^{N} \frac{\pi_k Poi(x_n|\lambda_k)}{\sum_{j=1}^{K} \pi_j Poi(x_n|\lambda_k)} \tag{87}$$

$$\lambda_{lagrange} \sum_{k=1}^{K} \pi_k = -\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma(z_{nk}) \tag{88}$$

$$\lambda_{lagrange} \sum_{k=1}^{K} \pi_k = -\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \tag{89}$$

$\sum_{k=1}^{K} \pi_k = 1$ by definition of π_k. $\gamma(z_{nk}) = p(z_k = 1|x_n)$ so the sum of this over all possible states k must be one. We get:

$$\lambda_{lagrange} = -\sum_{n=1}^{N} 1 \tag{90}$$

$$\lambda_{lagrange} = -N \tag{91}$$

We can then eliminate $\lambda_{lagrange}$:

$$\lambda_{lagrange}\pi_k = -\sum_{n=1}^{N}\gamma(z_{nk}) \tag{92}$$

$$-N\pi_k = -\sum_{n=1}^{N}\gamma(z_{nk}) \tag{93}$$

$$\pi_k = \frac{N_k}{N} \tag{94}$$

Now we have the basis for an expectation maximization algorithm to find the parameters of our Poisson Mixture Model.

1. Initialize the rates λ_k and mixing coefficients π_k and evaluate the log-likelihood

$$ln\left(p(\boldsymbol{X}|\boldsymbol{\lambda},\boldsymbol{\pi})\right) = \sum_{n=1}^{N} ln\left(\sum_{k=1}^{K}\pi_k Poi(\boldsymbol{x}|\lambda_k)\right)$$

2. **E-Step** Evaluate the responsibilities using the current parameter values

$$\gamma(z_k) = \frac{\pi_k Poi(\boldsymbol{x}|\lambda_k)}{\sum_{j=1}^{K}\pi_j Poi(\boldsymbol{x}|\lambda_j)}$$

3. **M-Step** Re-estimate the parameters using the current responsibilities

$$\lambda_k = \frac{1}{N_k}\sum_{n=1}^{N} x_n\gamma(z_{nk})$$

$$\pi_k = \frac{N_k}{N}$$

4. Evaluate the log-likelihood and check if either the log-likelihood or the parameters have converged

7 Routing

7.1 Cost function

7.1.1 Exponential cost

Kaleta [30] used the A* search algorithm to find the shortest path in the graph of London. We need to combine the attributes of each graph edge has such as length, bike accessibility and ascent, into a single cost. Kaleta defined the overall cost of $C(a, b)$ traversing an edge from node a to its neighbour b as a weighted sum of some cost function over each attribute, $c_i(a, b)$.

$$C(a,b) = \sum_{i=1}^{\#edgeattributes} w_i \times c_i(a,b) \tag{95}$$

Kaleta used the cost function:

$$1 - e^{\frac{-x_i}{d_i}} \tag{96}$$

where x_i is the value of the i^{th} attribute and d_i is the average value of the i^{th} attribute across all the edges in the graph. The weights w_i are input as user preferences. This returns a cost between 0 and 1 for each edge and prevents costs with with high absolute values from overshadowing other costs. An example is the length verses ascent. Edges in the graph of London have an average length of 1340m [30], but there is no edge with more than 100m of ascent.

If we scale them using Kaleta's cost function, both become values between 0 and 1. This has the nice property that if we weight the costs equally $w_{length} = w_{ascent}$, ascent contributes as much to the cost as the length does.

However when we used this algorithm in practice we found unexpected results. Even when all attributes except for the edge length were given weight 0, this cost function resulted in a longer path than the simplest cost function $c(a, b) = x_i$ for all the paths we tried.

We chose instead to use the simplest cost function $c(a, b) = x_i$ and set $w_{length} = 1$. All the other attributes could then be reasoned about in terms of length by setting sensible allowed ranges for the weight of each attribute, for example we could allow w_{ascent} to be in the range 0-500. If $w_{ascent} = 100$ for example, it would capture the idea that the user would be willing to go 500 metres further to avoid 1 metre of ascent.

7.2 Ascent Averseness Preference

Using NASA's SRTM dataset described in Section 2.4.5, we augmented all the nodes in our graph of London with an SRTM elevation. We were then able to add an *elevation delta* to all of the graph's edges, representing the amount of ascent required to traverse the edge. This allowed us to add a new user preference ascent averseness. This capture the idea that the user might be willing to travel a longer distance to avoid some ascent. Our cost function becomes:

```
cost = length + ascent_averseness * elevation_delta
```

7.3 Transfer Time

We define a *transfer* as either entering or exiting a tube station or changing lines at a tube station. We added a new feature to allow the user to enter the *transfer time*, the amount of time they would expect transfers to take. Any tube or mixed route is calculated with respect to this user defined transfer time. This allows users who are averse to changing lines avoid find a route that avoids changeovers as much as possible by entering a large transfer time.

7.3.1 Augmented Graph

The existing tube graph consisted of a single node for each tube station. For each pair of directly connected tube stations there was an edge, with the following attributes:

- length in metres
- time in minutes
- the lines that directly connect the tube stations

To allow changeover penalties to be applied, we used a new format for the graph. For each station instead of a single node, we have one node representing each line's platform and one node representing the station entrance. We introduce edges between all of these to form a strongly connected network of nodes for each station. Each of these edges has a time cost, representing the time taken to transfer between platforms, or to enter or leave the station. This time cost is currently set to the user defined transfer time

for all the edges that connect the different parts of a station. Edges representing tube journeys connect different stations together by connecting their platform nodes.

7.4 Mixed route algorithm

Kaleta's mixed route algorithm assumed that the fastest mixed route would contain no bike portion (Section 3.1.4). This algorithm calculated the shortest tube route, and if the user allowed more time than the duration of this route, the algorithm would try to add a cycling portion to the start or end of the route. As soon as there is enough cycling to fill up the time allowed, the search. There are a number of problems with this:

- Having two cycling sections in a tube route is inconvenient and we can always find a route with one cycling portion that takes just as long
- We found in practice this algorithm was slow and often an unreasonable route was returned.
- The assumption that taking the tube is always faster a route that takes both tube and bike isn't warranted, especially now that the user is able to set their own expected transfer time, which could result in the calculated tube journey being a lot longer in duration than the bike journey.

We present a simplified algorithm (Algorithm 2 which we show works well in practice in section 10. This evaluates potential mixed routes by adding bike portions from each station along the tube route to the end point. It chooses the journey that most closely matches the time the user allows.

Algorithm 2 Psuedocode for the mixed route calculation algorithm

```
function calculateMixedRoutes(startPoint, endPoint, allowedTime)
    startStation ← getNearestTubeStation(startPoint)
    endStation ← getNearestTubeStation(endPoint)
    tubeRoute ← getTubeRoute(startStation,endStation)
    startPortion ← getWalkingRoute(startPoint,startStation)
    possibleRoutes ← []
    for i = 1 → length(tubeRoute) do
        tubePortion ← tubeRoute[0 → i]
        tubeStationPoint ← tubeRoute[i].latlng
        ▷ Get the fastest cycle hire or walking route to the end point
        endPortion ← calculateCycleHireRoute(tubeStationPoint,endPoint)
        route ← concatenate(startPortion,tubePortion,endPoint)
        possibleRoutes.append(route)
    end for
    returngetRouteClosestDuration(possibleRoutes,allowedTime)
end function
```

8 Journey Planner

8.1 Docking Station Status Overlay

The ability to display the live docking station statuses alongside any calculated routes is not currently a feature present in any journey planners known to us. We added a a layer of markers indicating the positions of all the active Barclays Cycle Hire docking stations. This layer can be toggled on or off, as they might otherwise hide important features on the map.

Using the Barclays Cycle Hire live feeds we were able to additionally display the live status of each docking station upon clicking a marker, including the docking station name, the number of bikes, the number of docks and the time of the last status update.

8.2 Elevation Profiles

To go with the new ascent averseness preference, we wanted a way for the user to see for themselves the ascent and descent in each of the routes and judge for themselves whether the calculated route is satisfactory. We added a button to each route which opens a graph of the elevation above sea level along the entire route. The total ascent along the route is also shown to the user.

8.3 Tube Route Changeover Display

Kaleta's journey planner was able to display a simple outline of the tube routes it calculated, drawing a line between each station along the calculated tube route. This leaves the user to work out for themselves which lines to take and where to changeover. We added markers to the map to inform the user what line they should take at each station.

Additionally, in Kaleta's journey planner the tube was only used as part of calculating a mixed route. It was not possible to request the calculation of a tube route without potentially including a bike portion. We have added the calculation of a pure tube route to every routing request as it is relatively fast and provides a useful point of comparison for the mixed routes and bike routes we calculate.

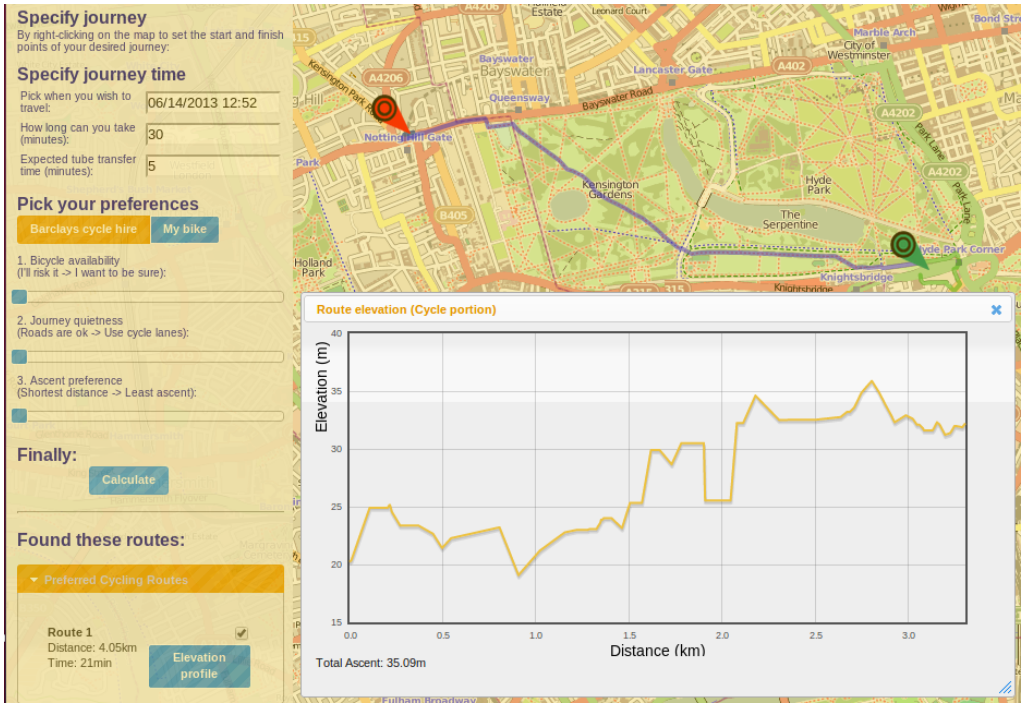


Figure 14: A route through Hyde Park marked on the map, along with its corresponding elevation profile

8.4 Own bike

There are still currently large portions of London with no Barclays Cycle Hire docking stations, most notably south of the Thames. No sensible routes involving the Barclays Cycle Hire scheme can be calculated when the start point or end point is placed in these areas. Some of our routing preferences can be applied if the user owns a bike, most notably route busyness and ascent averseness. We added the ability to calculate a bike route assuming bike ownership while taking road busyness and ascent averseness into account. Additionally, TFL allow taking a folding bike anywhere on the tube and a non-folding bike on many lines during less busy times [2]. We added the ability to calculate mixed routes for a user owned bike to help accommodate this.

9 Mobile Application

The existing journey planner web page was not suited to use on a handheld device. To set the start and finish positions, it is necessary to right click to bring up a context menu. While a right click can be translated into a long press of the touch screen, it is much harder to be precise. Much less of the map is visible on the smaller screen, so a user on the move could end up wasting their time and data allowance panning the map searching for their start and finish positions. The user interface for setting the user's preferences on the left of the screen is too verbose and takes up too much horizontal screen space for the application to be used effectively on a mobile device. We included a subset of the functionality of the full journey planner for the mobile application to keep the user interface clean and simple to use on the move.

9.0.1 Geocoding

To solve the problem of not being able to right click, we introduced setting the journey start and end points by typing in an address. A request is then sent to Nominatim's geocoding API, which returns a list of possible locations matching the address as a JSON array. We present these possible locations to the user, who selects one of them to set the start or end point.

9.0.2 User's location

To get the user's location, we use the HTML5 Geolocation API, which allows us to access whatever location the device running the browser is capable of providing, including the GPS location on mobile devices (subject to user consent). If the user's location is available we mark it on the map and allow the user to choose 'My Location' as the start or end point.

10 Results and Evaluation

10.1 Machine Learning Models

We first learned the parameters of our regression model. We then plotted the predictions of our regression model across the day against the mean number of pickups and dropoffs in small intervals across the day. From inspecting these plots, the regression appears to fit the data well for stations with a variety of usage patterns without overfitting.

Figure 15: Predictions of regression model for pickups at South Kensington station. Mean pickups over the data for small time intervals across the day are marked in red as a reference

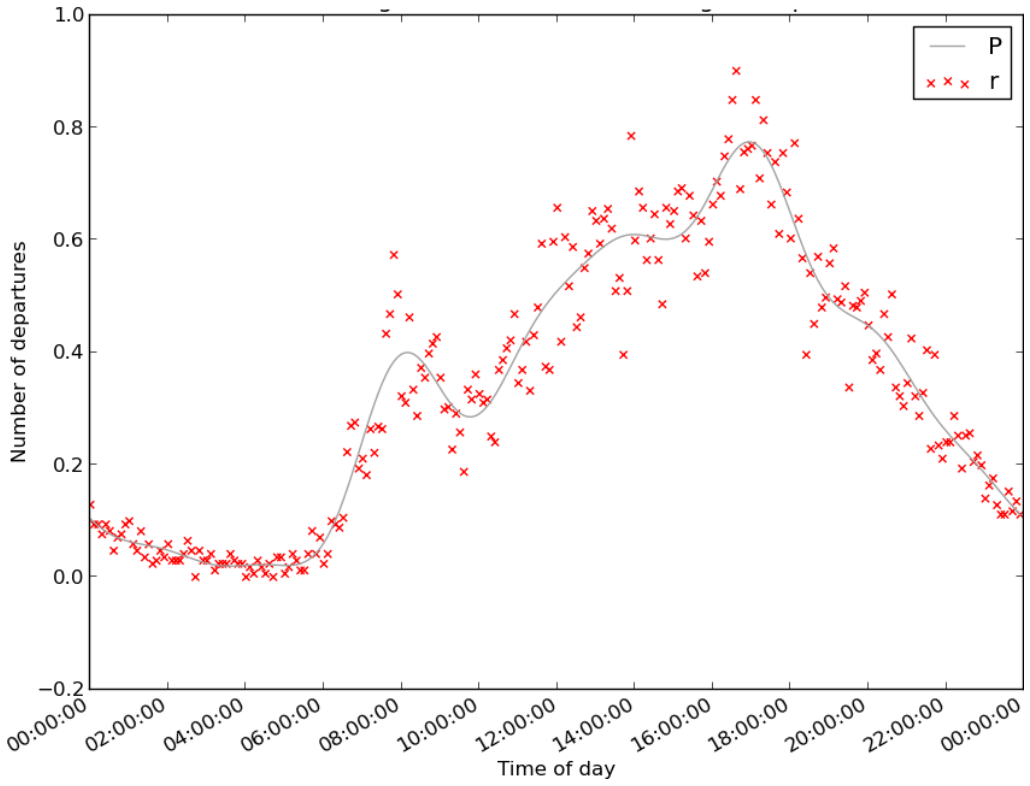


Figure 16: Predictions of regression model for dropoffs at Vauxhall Bridge, Pimlico. Mean dropoffs over the data for small time intervals across the day are marked in red as a reference

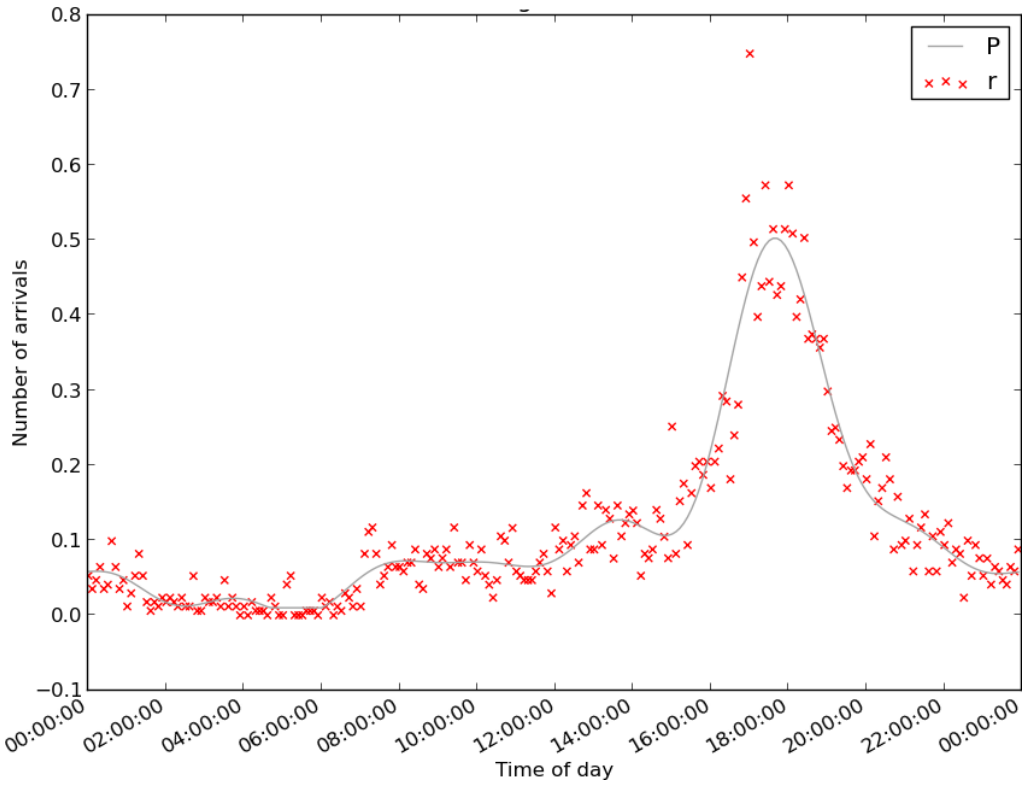
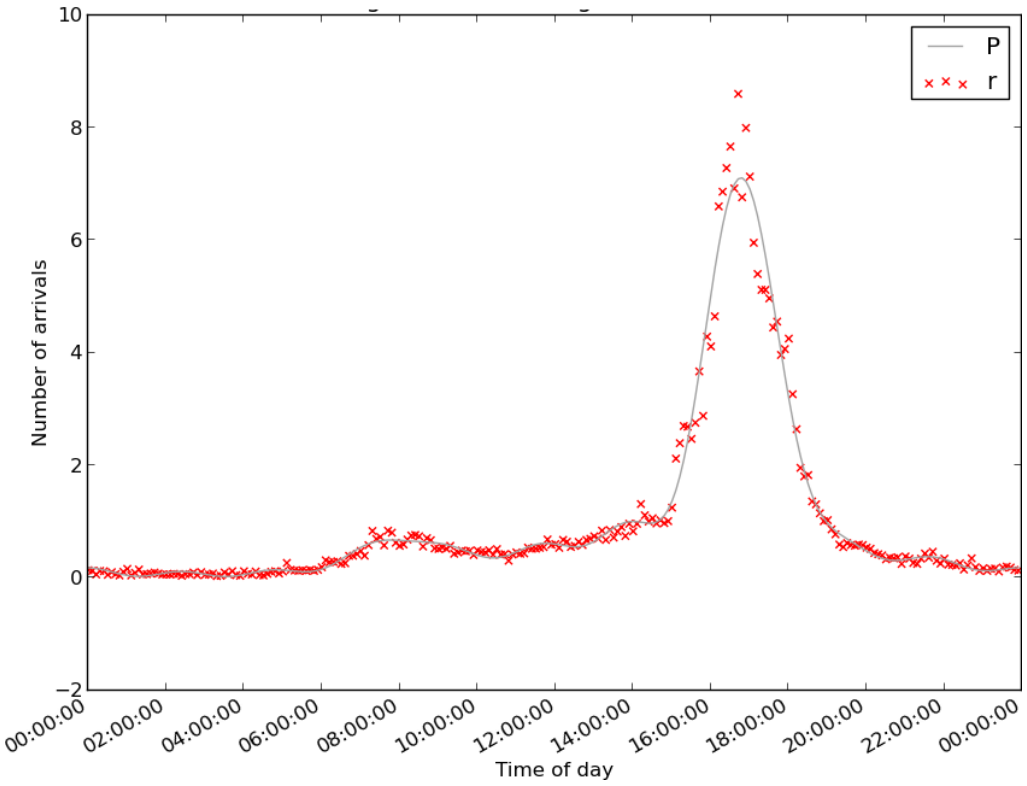


Figure 17: Predictions of regression model for dropoffs at Belgrove Street, King's Cross. Mean dropoffs over the data for small time intervals across the day are marked in red as a reference



We have already discussed how we have statistically validated that our mixture model fits the data better than a single Poisson distribution (Section 5.3). We can also similarly examine the fit of our mixture model by learning its parameters, plotting the probability of all possible numbers of pickups or dropoffs the mixture can produce and comparing it with the observed frequency density of pickups or dropoffs in the same time interval. We can see that the probability distribution of a learned Poisson mixture model for Waterloo Station 3 (Figure 18) closely matches the shape of the observed frequency densities over all the data (Figure 19).

Figure 18: Probability distribution of Poisson mixture model over numbers of pickups between 1700 and 1800

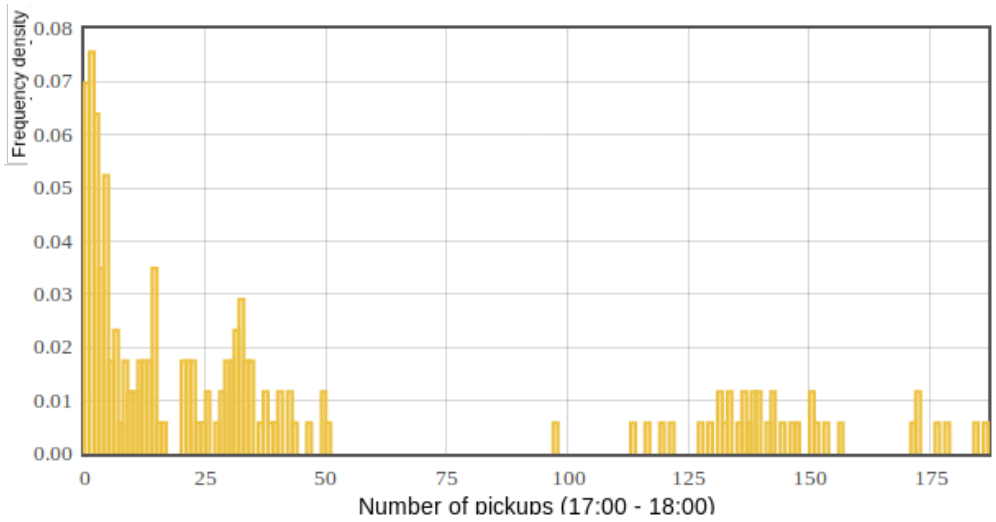
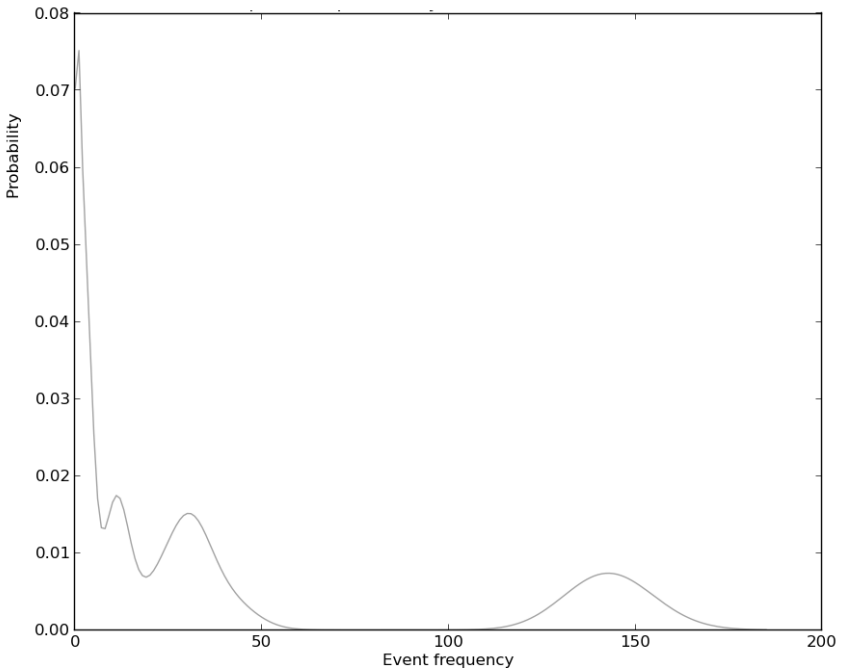


Figure 19: Observed frequency density of pickups at Waterloo Station 3 between 1700 and 1800 across all the days of data



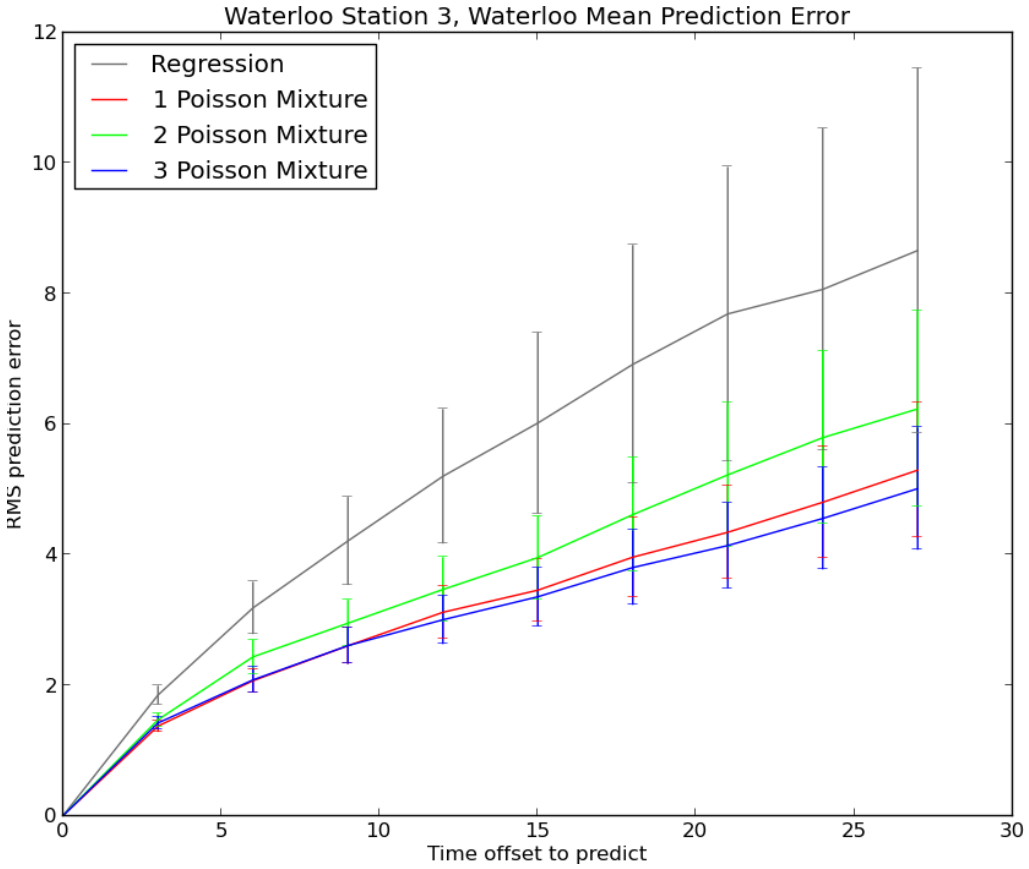
10.1.1 Prediction

Though the model has been shown to fit the data, this does not necessarily mean we have improved predictions. To test the predictions, we collected a month of the live feed data discussed in section 2.4.2. We then took a uniform sample of start times across the day, spread 20 minutes apart. For each sample start time, we make predictions up to 30 minutes in the future. We compare these predictions with the observed number of bikes from the collected data. We then plotted the root mean square error in the predicted number of bikes over time. We found for busy stations, like Belgrove Street and Waterloo Station 3, the 3 Poisson mixture model performed best especially as the predictions went further into the future.

For less busy stations like Old Quebec Street, all the methods performed very similarly, which is what we'd expect as the absolute change in the number of bikes is very small even over 30 minutes.

If we examine the frequency density graph (Figure **??**) we see its peaks (which correspond roughly to the location of the Poisson distributions for

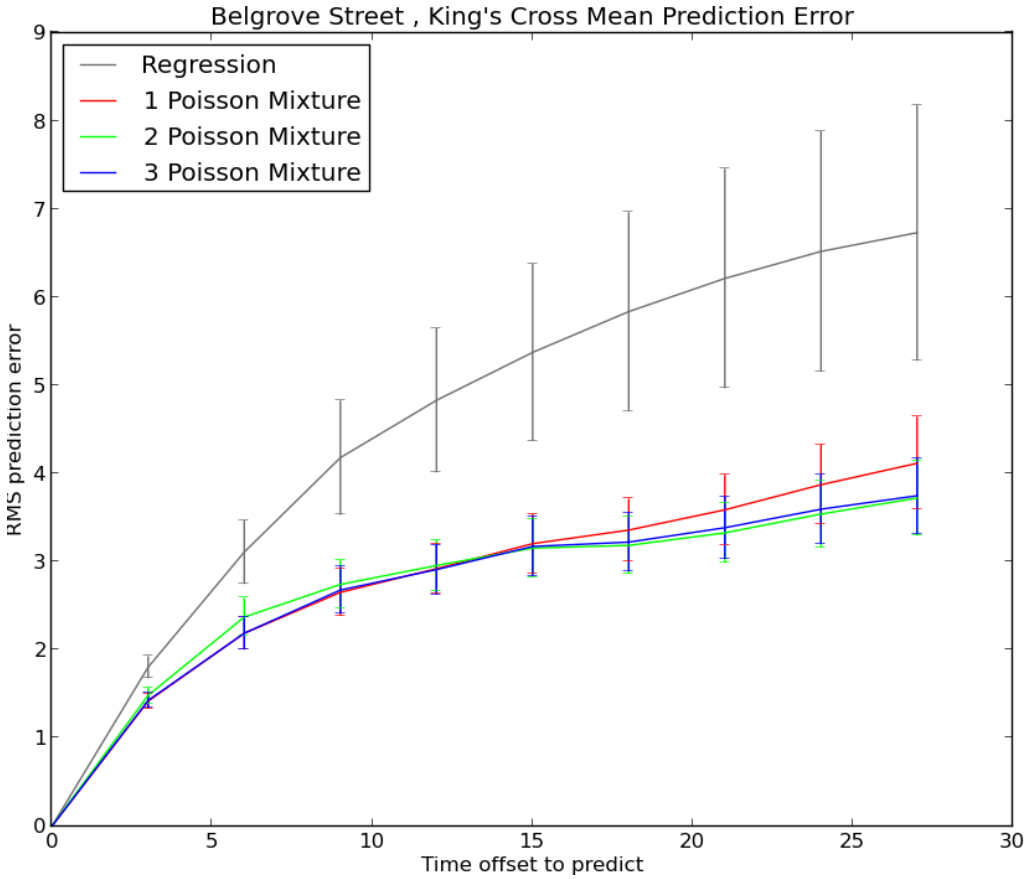
Figure 20: Root Mean Square error of prediction models for Waterloo Station 3



our mixture model) are far apart. If the data were overdispersed due to a change in the pickup rate within the time interval, we would expect peaks that are relatively close together, such as the cluster of peaks we can see from 0 to 25. However there are two peaks at 135 and 175 which are distant from the rest. This suggests that there are other hidden variables that cause more variability in the observed pickups and dropoffs than we'd expect from overdispersion of the data.

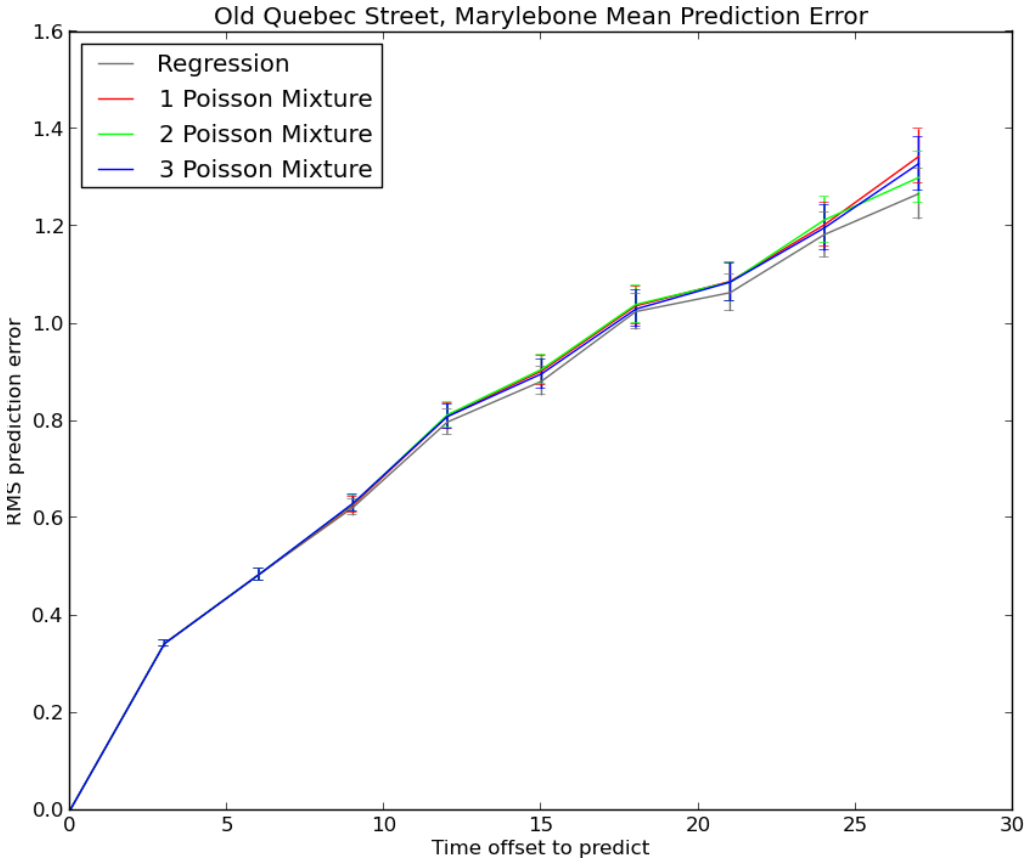
One plausible explanation is the difference between weekday and weekend usage patterns. Saturdays and Sundays could be the cause of the peaks close to 0, as we'd expect less usage of the cycle hire scheme for a commuter hub like Waterloo on those days. The weekdays could be the cause of the non-zero peaks.

Figure 21: Root Mean Square error of prediction models for Belgrove Street, King's Cross



Our algorithm works by sampling the mixture model, taking an average over all the component Poisson distributions. If it is the case that the variability is explained by overdispersion, this is the best we can do. However if there are other hidden variables, we could potentially make significantly better predictions by finding out what the hidden variable is and splitting up the data based on observations of this variable. For example, if the variability is caused by the difference between weekdays and weekends, we can split the data into weekdays and weekends and learn a separate Mixture Model for each. Then we can use the appropriate model depending on whether it's a weekday or weekend when we get a journey planning request.

Figure 22: Root Mean Square error of prediction models for Old Quebec Street, Marlyebone



10.2 Routing

We wanted to be able to demonstrate that the router would sensibly adjust the route based on the user's preferences in a number of scenarios.

The figures in this section show only the key preference settings for each scenario alongside the route returned, marked on the map. For a more complete demonstration of the user interface, see Appendix C

10.3 Scenario 1 - Hill avoidance

We can see from picking any route in London, as we increase the ascent averseness the sum of ascent over the route decreases.

To show that this preference is useful, we wanted to be able to demonstrate that on a route where the shortest route goes over a hill, but would avoid the hill if ascent averseness was non-zero. This is a feasible case in which the user would prefer a longer route. This behaviour is demonstrated at Parliament Hill, one of the highest peaks in London.

The shortest route passes straight over the hill.

Once ascent averseness is set to a value greater than zero, a route around the hill is found. From the elevation profile we can see the total ascent is 60.68 metres as opposed to the 76.88 metres of the shortest route. In absolute terms, the maximum height above sea level is around 110 metres as opposed to 115 metres.

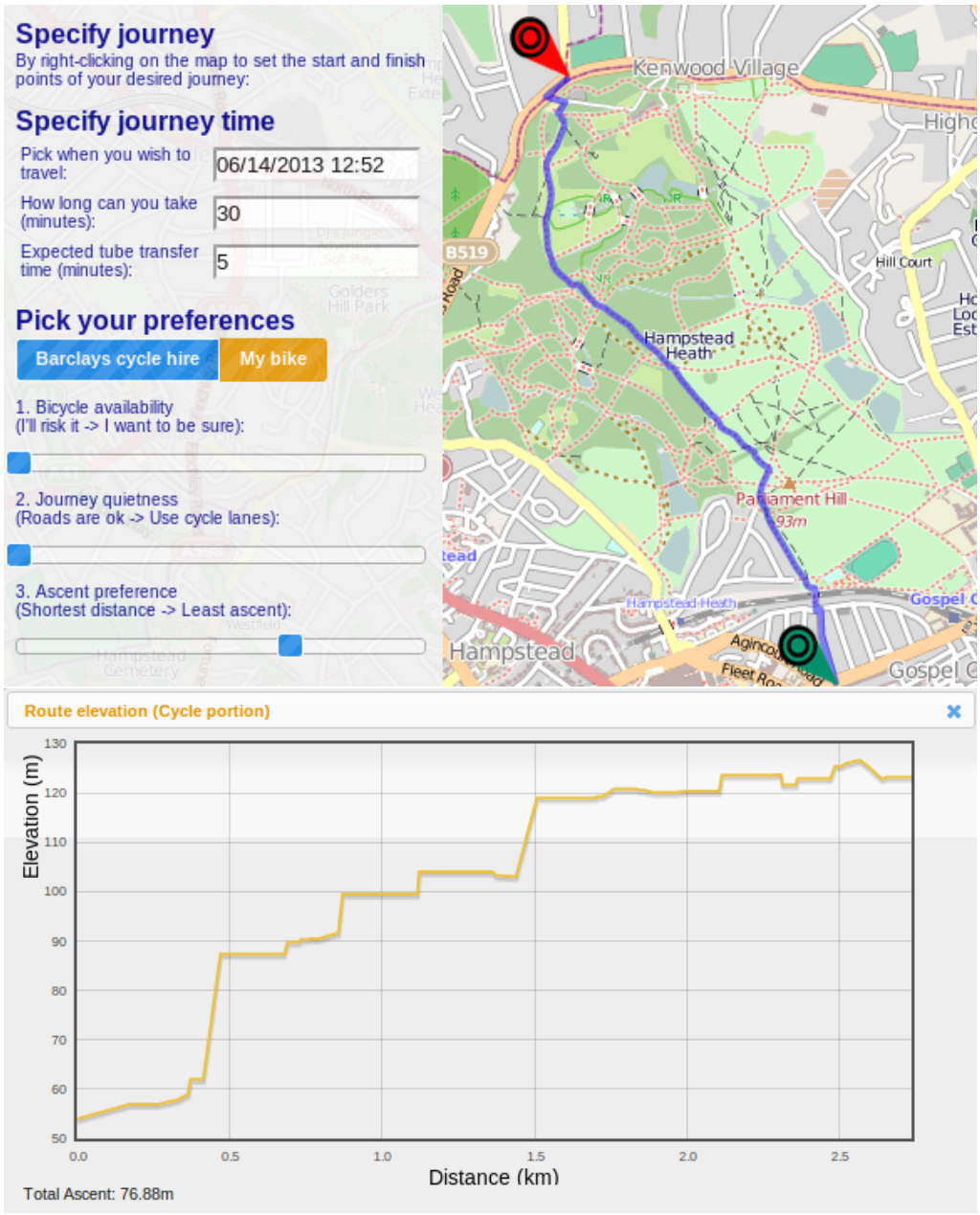


Figure 23: Fastest route over Parliament Hill

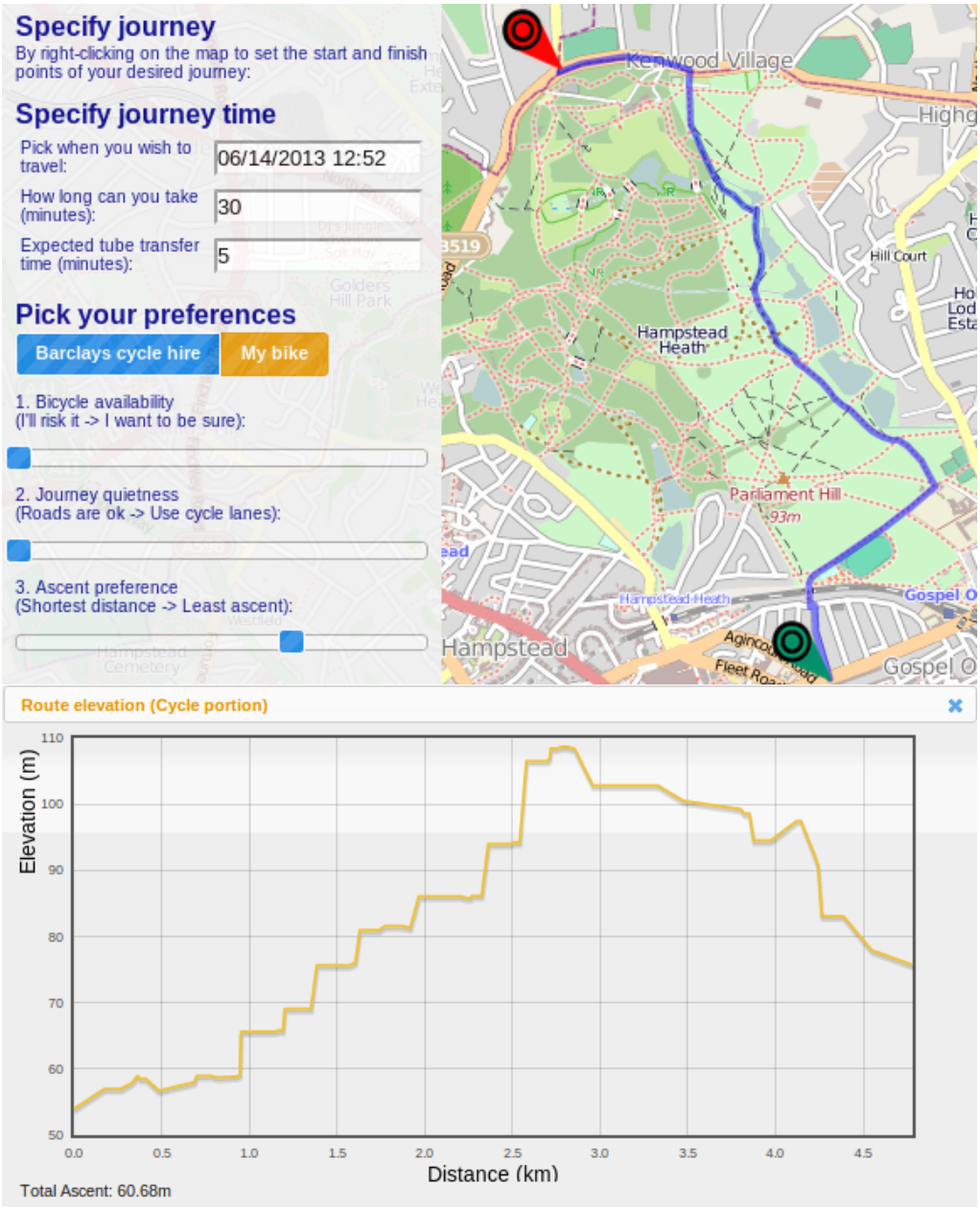


Figure 24: Preferred route around Parliament Hill

10.4 Scenario 2 - Changeover avoidance as transfer time increases

We wanted to be able to demonstrate that as the user increases the expected tube transfer time, the number of changeovers on the tube route would decrease. We set the start point to South Kensington and the end point to Liverpool Street.

This is a route through the central London tube network and many possible London Underground lines could be used to get between the two stations. First we set the expected transfer time to zero, which means changeovers have no penalty applied. The resulting route is shown in figure 25. A route with 3 changeovers and using 4 London Underground lines is calculated.

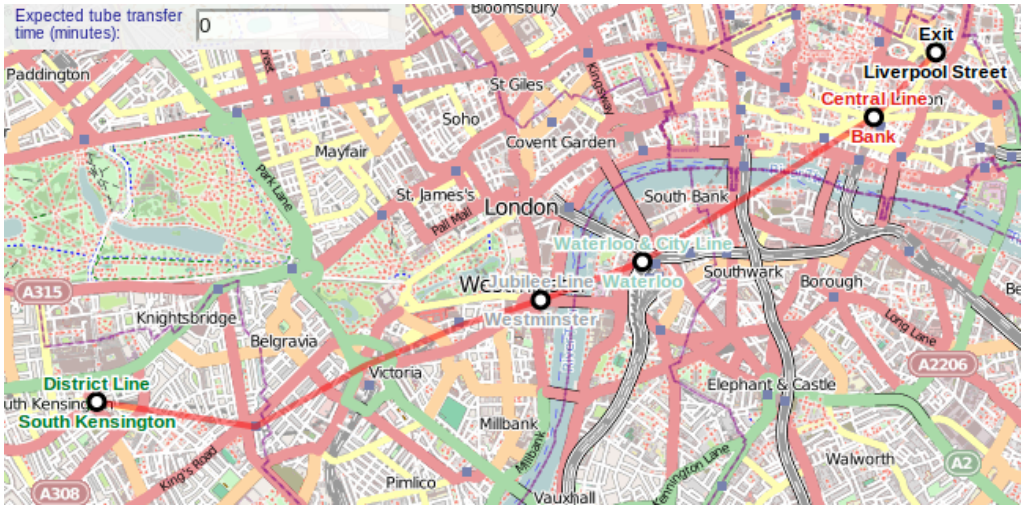


Figure 25: Tube route with preferences and transfer time set to 0

We then set a more realistic expected transfer time, 5 minutes for the same stations. This found a more sensible route with only a single changeover at Holborn (Figure 26).

Finally, we set the transfer time to 7 minutes, a feasible changeover time for busy times of day or a setting for users who are averse to changeovers. This results in a longer route along the Circle line without any changeovers at all (Figure 27).

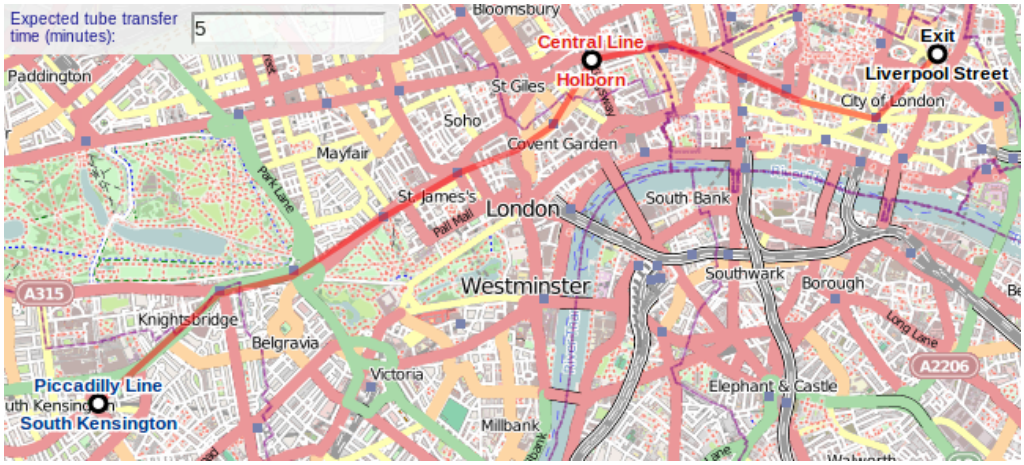


Figure 26: Tube route with preferences set to 0 and transfer time set to 5 minutes

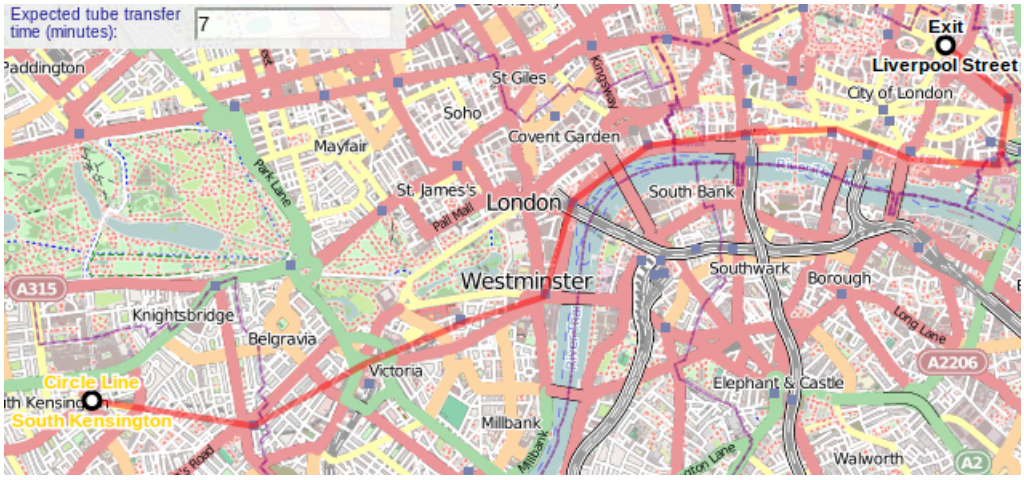


Figure 27: Tube route with preferences set to 0 and transfer time set to 7 minutes

10.5 Scenario 3 - Selecting a different docking station depending on availability

We wanted to show that our journey planner with our Poisson mixture model integrated could be capable of choosing different starting and ending stations for a Barclays Cycle Hire journey based on availability. We consider a journey from around St James' Park tube station, for a journey to start at the current time (Mon 18 Feb 8:04am). When we set the availability preference to non-zero (which means we are averse to the risk of not being able to pickup or dropoff a bike) we get a different route, shown in Figure 28 to the shortest route shown in Figure 29 where we drop off our bike at Eaton Square instead of Eccleston Place.

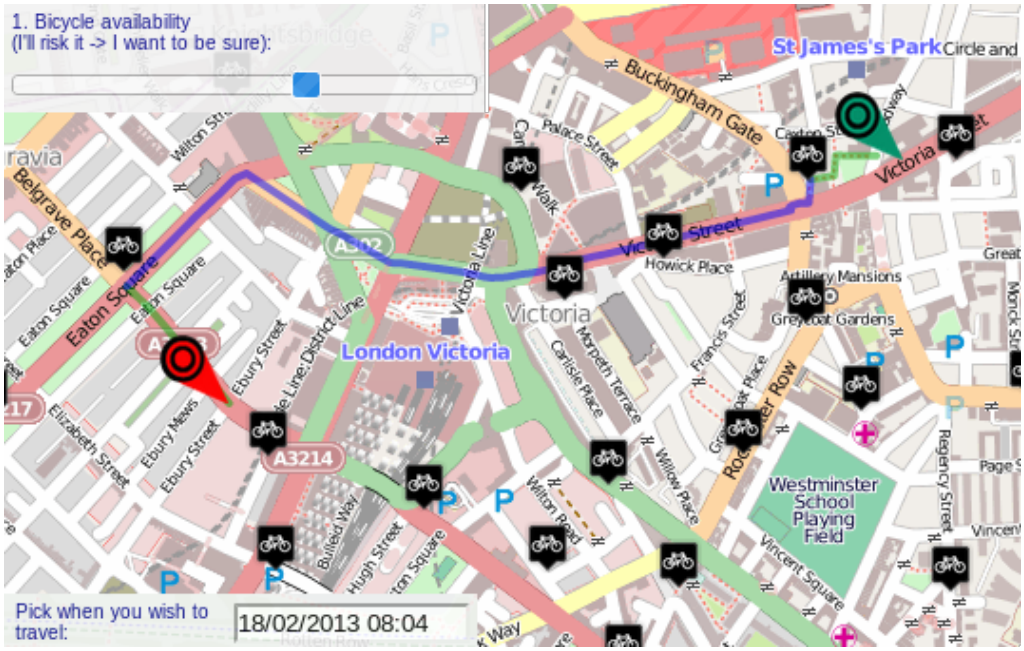


Figure 28: Safe route found when the availability preference is non zero at 8:04am

If we examine the docking station status at Eaton Square (Figure 30) and Eccleston Place (Figure 31), this seems to be a sensible decision. There are no docks available at Eccleston Place, while there are plenty of docks and bikes at Eaton Square.

If we instead plan to start the journey one hour in the future instead of immediately we get the same result, shown in Figure **??**. However if plan

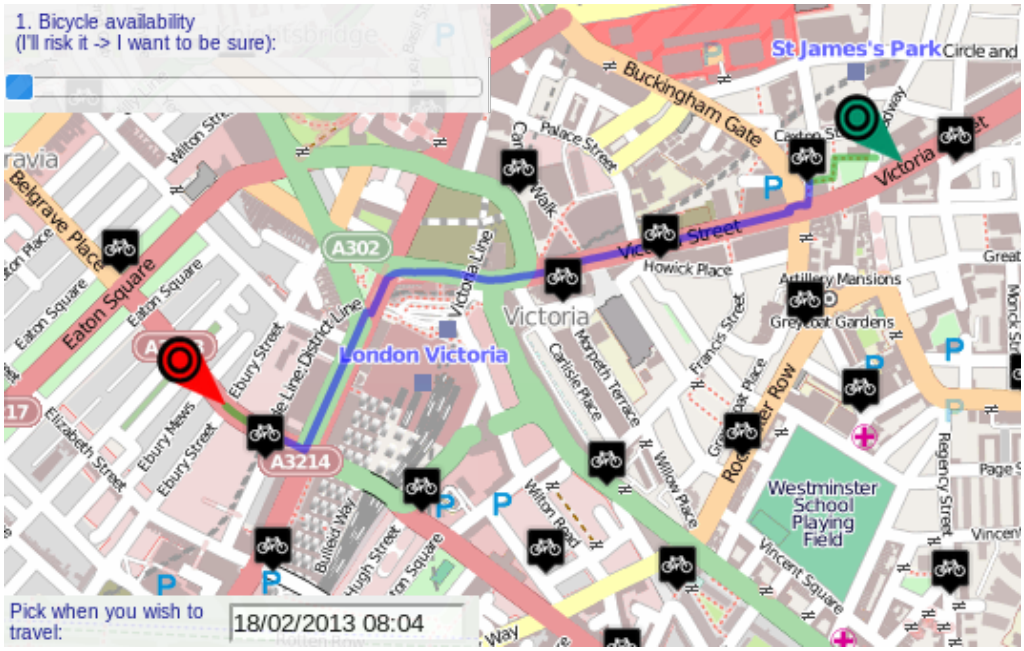


Figure 29: Fast route found when availability is not taken into consideration at 8:04am

to start the journey two hours in the future, the router decides that going through Eccleston Place is safe, shown in Figure 33.

We can examine the pickup and dropoff data for Eccleston Place to determine whether this is a sensible result From figure 34, we can see that after the one hour interval, we would expect just over one bike to have been picked up. After two hours the expected change is close to two bikes.

This change of less than two bikes seems like a small change, but it is enough to explain the change in route given our current sampling algorithm. As discussed in section 2, our algorithm rolls out many possible futures and considers only whether or not there are bikes available at the end of each possible future, not how many are available. For a given rollout as long as there is at least one bike at the end, that rollout will be considered as evidence that bikes are available. However it could be argued that it is still risky to direct the user to go to Eccleston Place if on average only one bike will be picked up from that station before the user reaches it. We leave finding a better method to evaluate risks to future work.

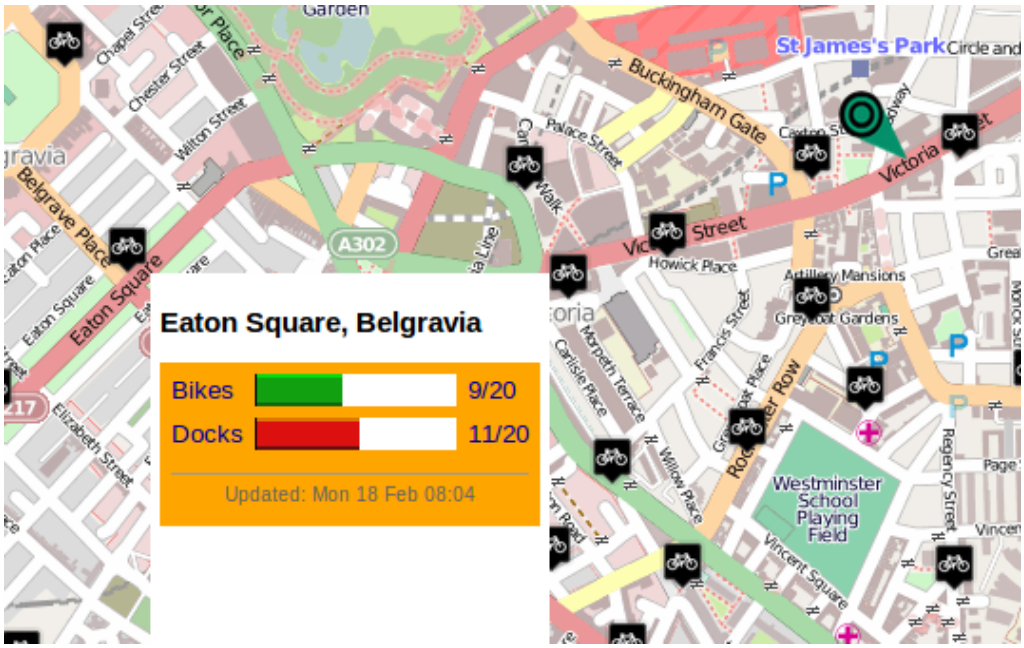


Figure 30: Docking station status at Eaton Square, Belgravia

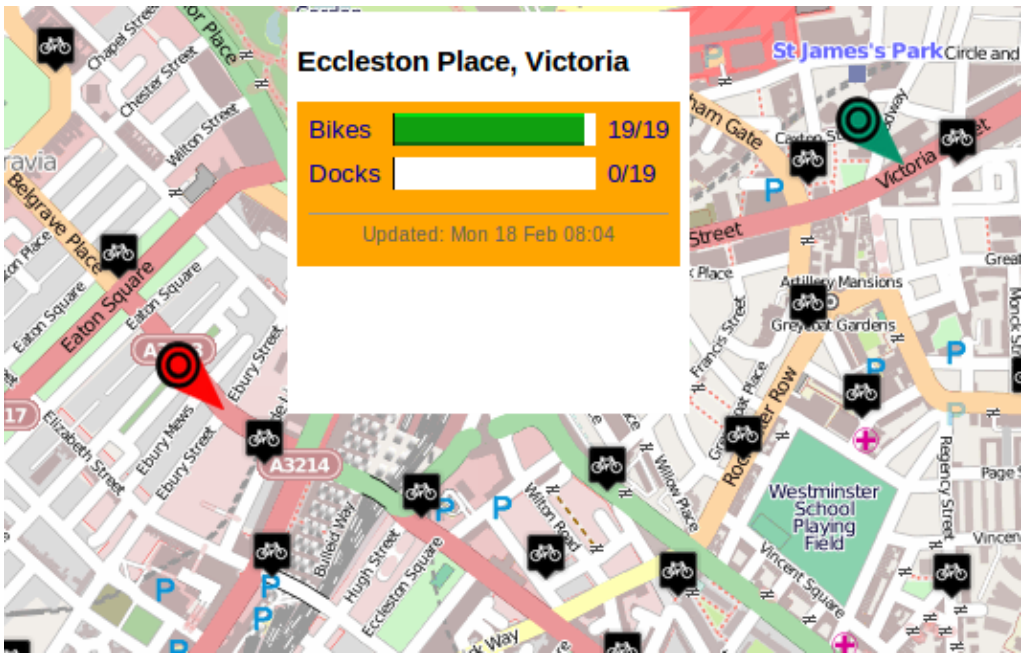


Figure 31: Docking station status at Eccleston Place, Victoria

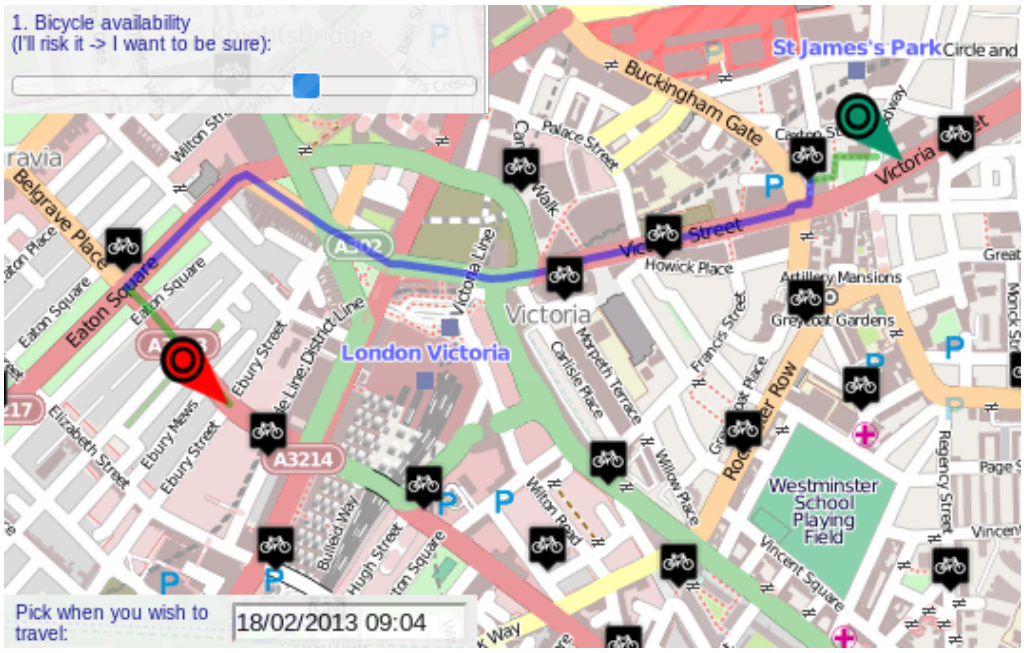


Figure 32: Safe route found when the availability preference is non zero and the journey is to start one hour in the future

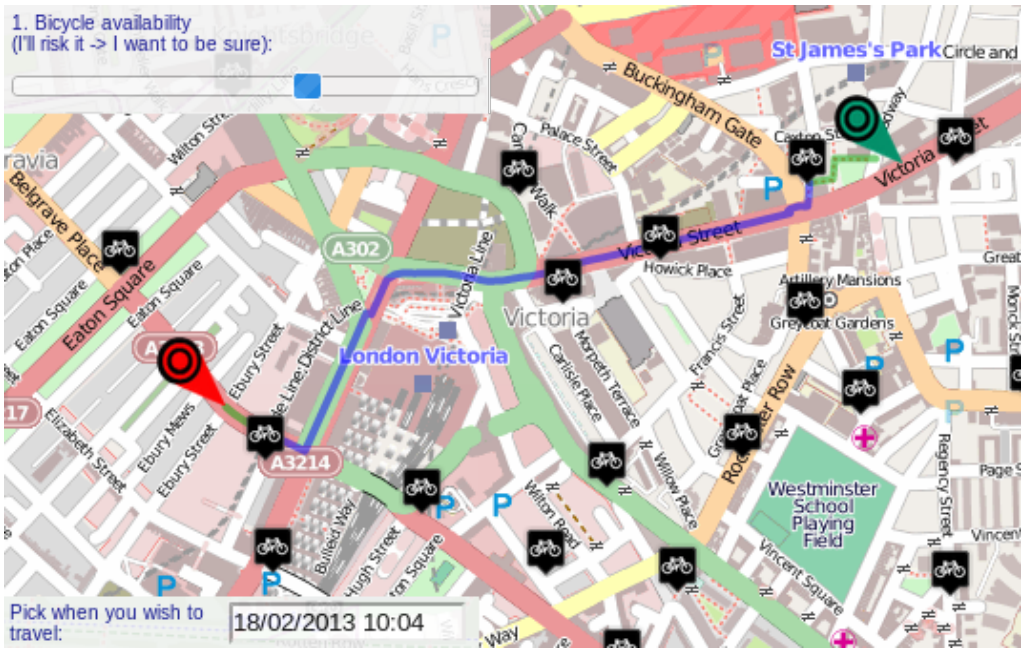


Figure 33: Safe route found when the availability preference is non zero and the journey is to start two hours in the future

	8:00-8:15	8:15 -8:30	8:30-8:45	8:45-9:00	9:00-9:15	9:15-9:30	9:30-9:45
Mean dropoffs	1.6	1.25	1.2	1.2	0.75	0.8	0.6
Mean pickups	2.25	1.55	1.5	1.3	1.1	0.8	0.5
Expected change	-0.65	-0.3	-0.3	-0.1	-0.35	0	0.1
Cumulative expected change	-0.65	-0.95	-1.25	-1.35	-1.7	-1.7	-1.6

Figure 34: Mean pickups and dropoffs at Eccleston Place, Victoria in 15 minute intervals over all the Barclays Cycle Hire statistics

10.6 Scenario 4 - Mixed routes that adapt to the time the user allows

We wanted to show that our new trip chaining algorithm would be able to produce longer journeys as the user allows more time. We consider a route between South Kensington and Holborn. First we allow 20 minutes journey time, with expected tube transfer time set to 1 and all other preferences set to 0. This returns the fastest route our journey planner can find, a 27 minute route which consists of only the tube and walking, shown in figure 35.

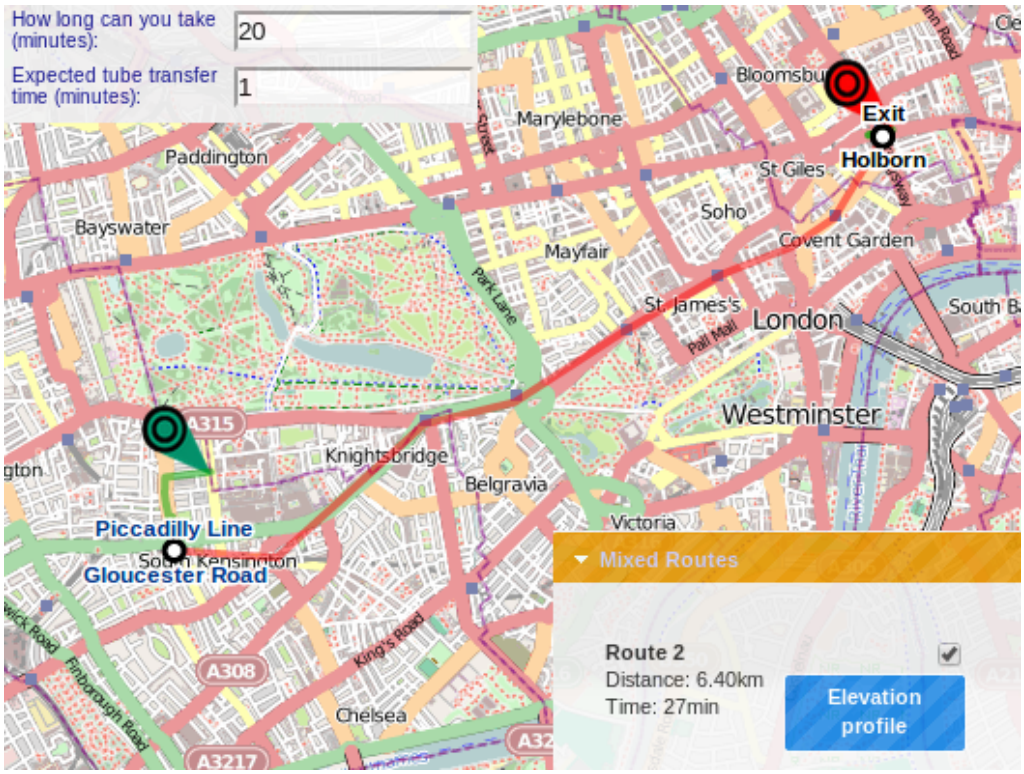


Figure 35: Route between Imperial College and Holborn, allowing 20 minutes journey time

We then adjusted the journey time to allow 30 minutes (Figure 36) and 32 minutes (Figure 37). This finds longer routes with a larger cycling portion mixed in.

Finally when we allowed 35 minutes for the journey, the planner returned a cycling only route (Figure 38).

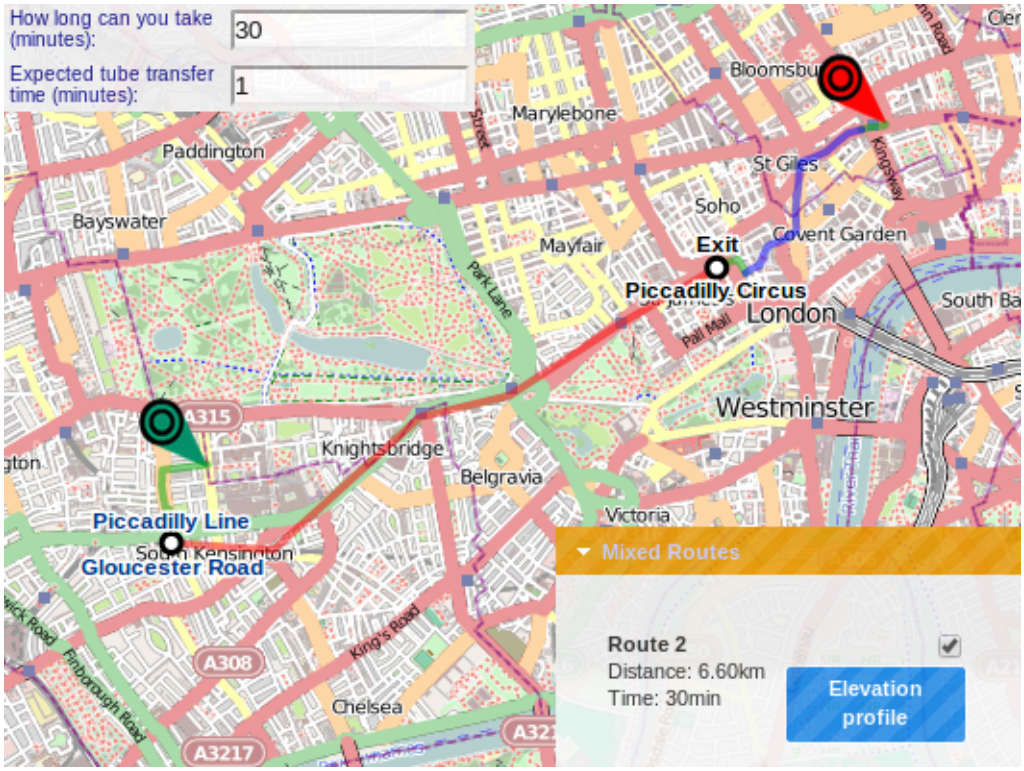


Figure 36: Mixed route between Imperial College and Holborn, allowing 30 minutes journey time

We were also able to show our preferences worked well together. If we consider again the case where we allow 20 minutes journey time. The journey planner found a 27 minute route, all on the tube. The expected changeover time that had been specified was one minute. Now if we increase the expected transfer time to 5 minutes, the tube is no longer as fast as taking a Barclays Bike directly and the route with cycling is the fastest route (Figure 39).

Consider again the case where we allowed 32 minutes to travel between South Kensington and Holborn. These preferences resulted in a mixed route where we get of the tube at Hyde Park Corner and cycle to Holborn (Figure 37). If we now increase our ascent averseness we get the route shown in Figure 40, which gets off at Piccadilly Circus instead.

If we look at the elevation profile for the cycling portions of both routes, the route which takes a bike from Piccadilly Circus (Figure 42) has less ascent than the route which takes a bike from Hyde Park Corner (Figure 41). The

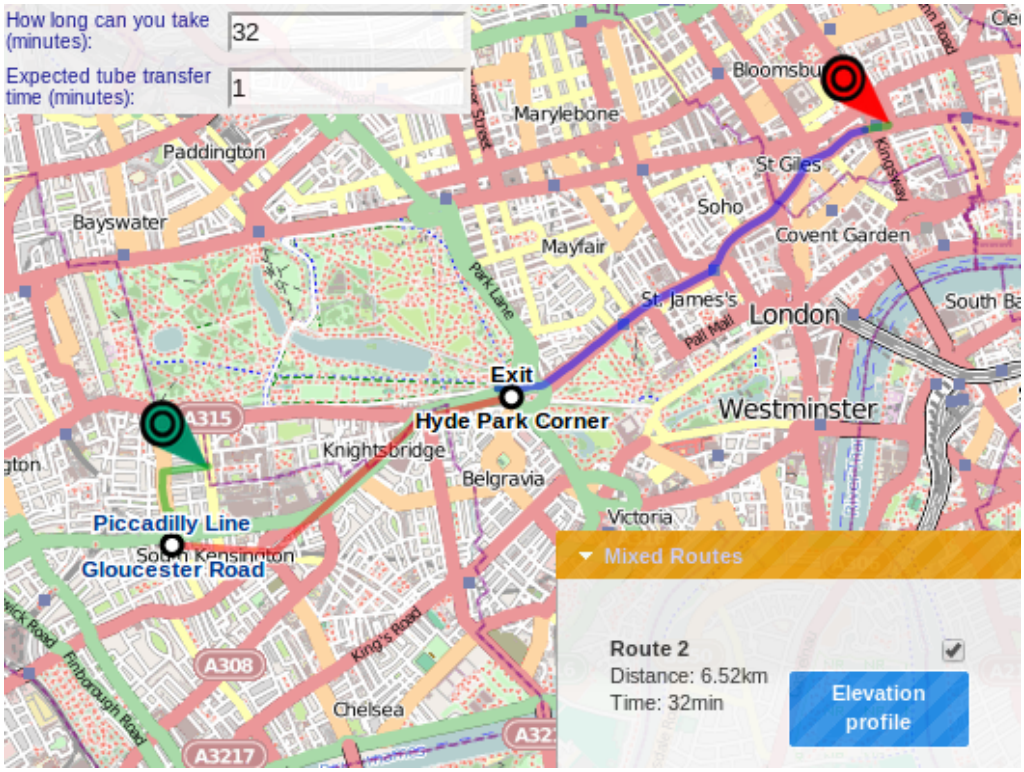


Figure 37: Mixed route between Imperial College and Holborn, allowing 32 minutes journey time

cycling portion of his route has 7.36m of ascent as opposed to the 22.43m of ascent if we take the route from Hyde Park Corner, avoiding approximately 15m of ascent.

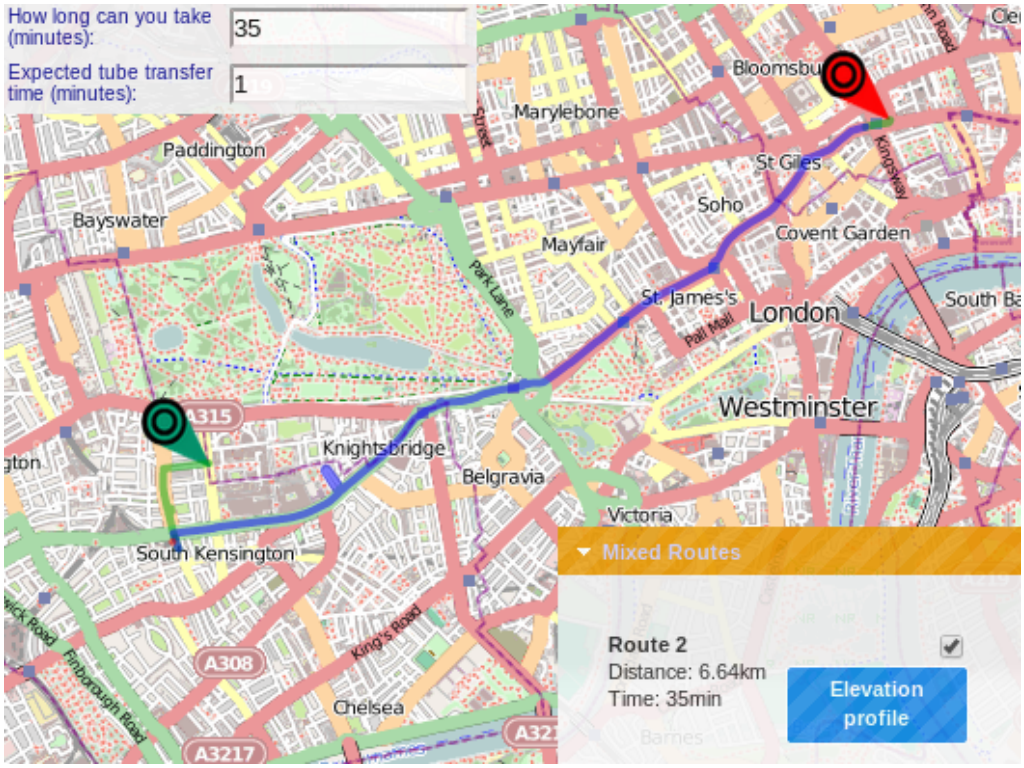


Figure 38: Mixed route between Imperial College and Holborn, allowing 35 minutes journey time

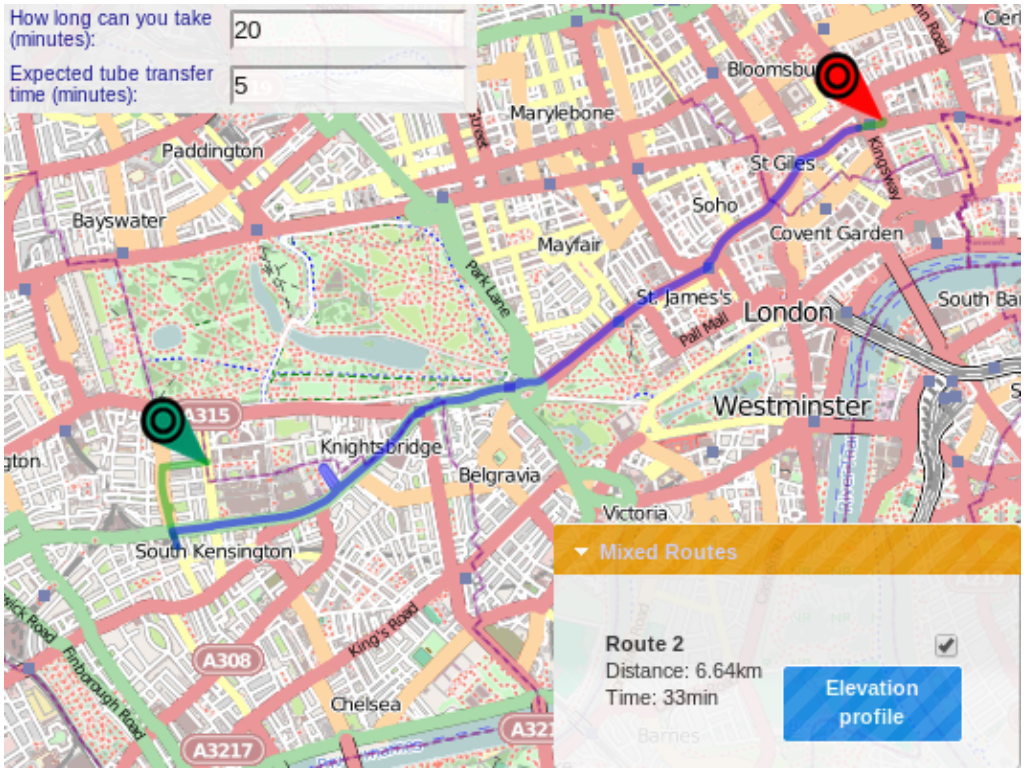


Figure 39: Mixed route between Imperial College and Holborn, allowing 20 minutes journey time but now with expected transfer time set to 5 minutes

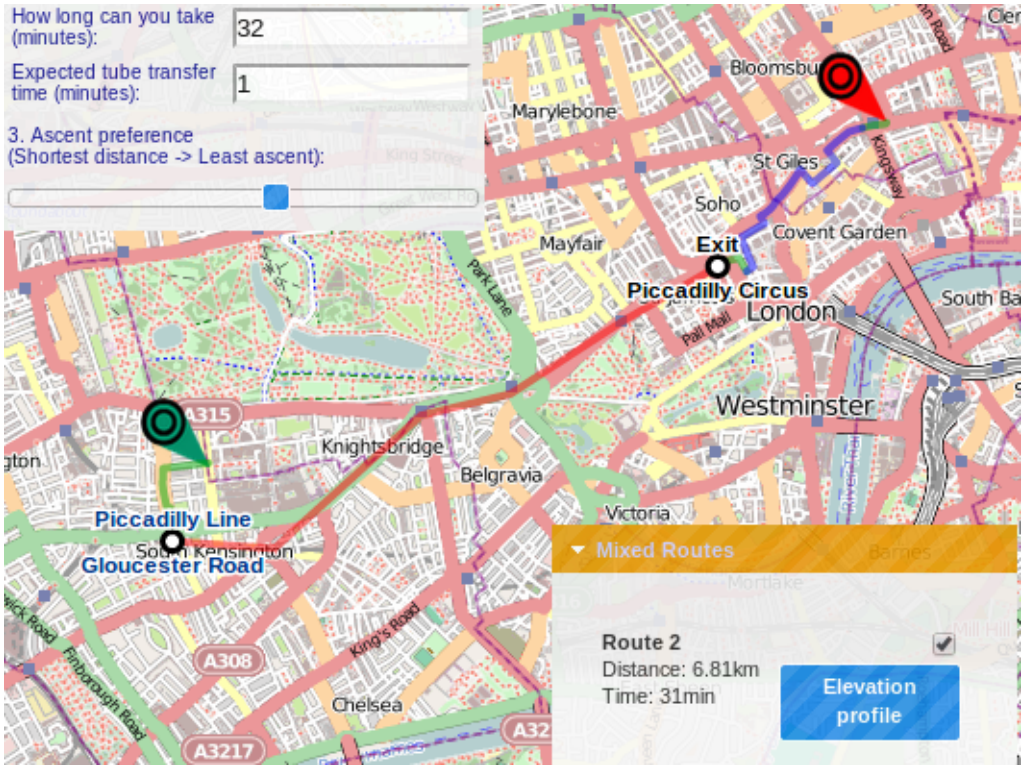


Figure 40: Mixed route between Imperial College and Holborn, allowing 20 minutes journey time but now with expected transfer time set to 5 minutes

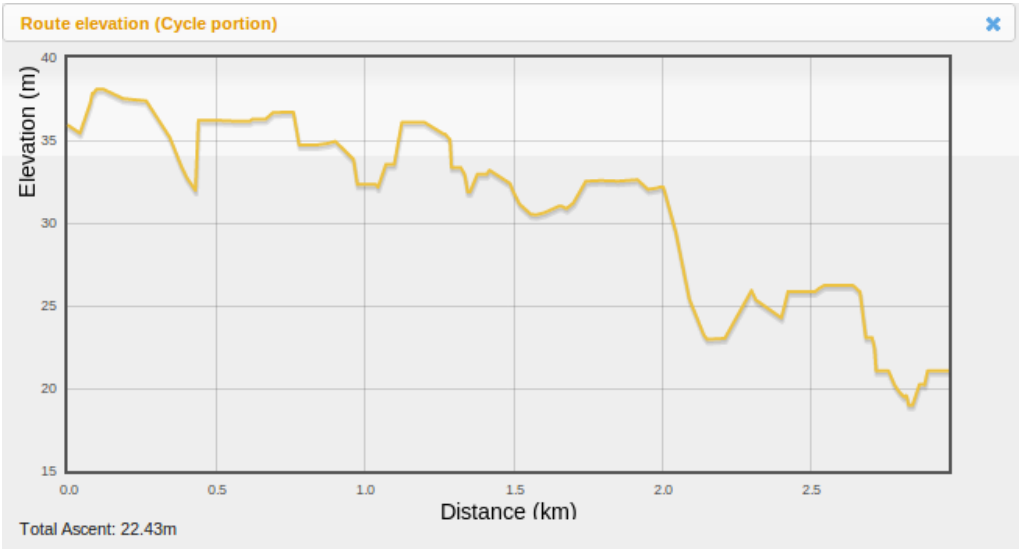


Figure 41: Elevation profiles for the cycling portion of the mixed route between Imperial College and Holborn where we get on a bike at Hyde Park Corner

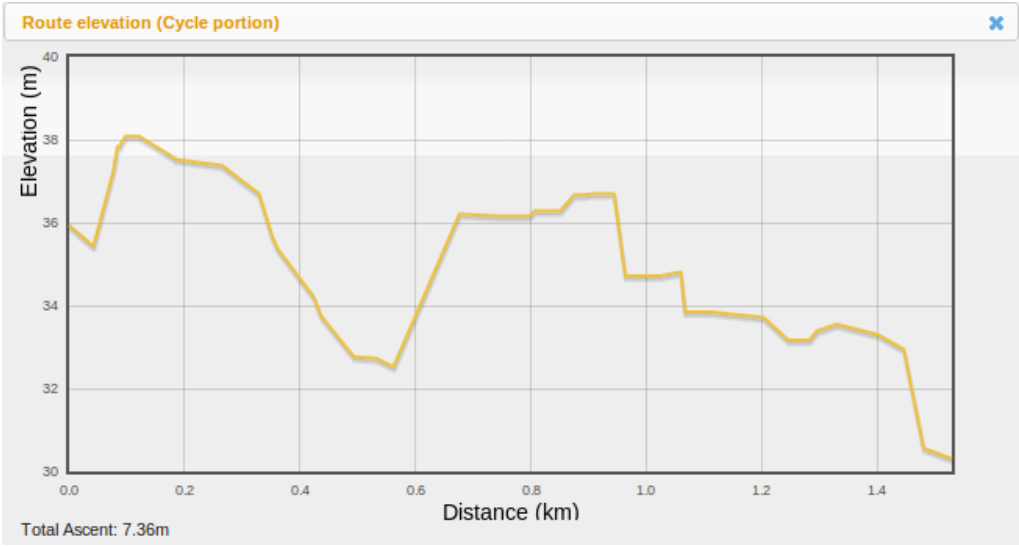


Figure 42: Elevation profiles for the cycling portion of the mixed route between Imperial College and Holborn where we get on a bike at Picadilly Circus

10.7 Mobile application

As discussed in section **??**, our mobile web application includes a subset of the functionality of the desktop application and shares the same code.

Thus for our evaluation of the mobile application, we were mostly concerned about how well the application be rendered across different devices and platforms. We tested our mobile application on a real android device, as well as emulators and simulators for Apple and Android devices.

We performed the following tests across the platforms, which cover the functionality of our the Android application.

1. Setting preferences by opening the preferences tab and adjusting the sliders

2. Setting the start and end points of the journey using address search and the user's current location, if available

3. Searching for a journey and marking the route found on the map

We made adjustments to the mobile version of the application until all these tasks could be performed on each platform and the website rendered without problems.

10.7.1 Android devices

We performed the tests on a Samsung Galaxy Europa, a device which falls into the smallest category of Android screen sizes. We also used the android emulator to test the application on the standard and tablet screen sizes. We did not find any issues.

10.7.2 iPhone

For iPhone we used the popular iPhone simulator TestiPhone.com [11].

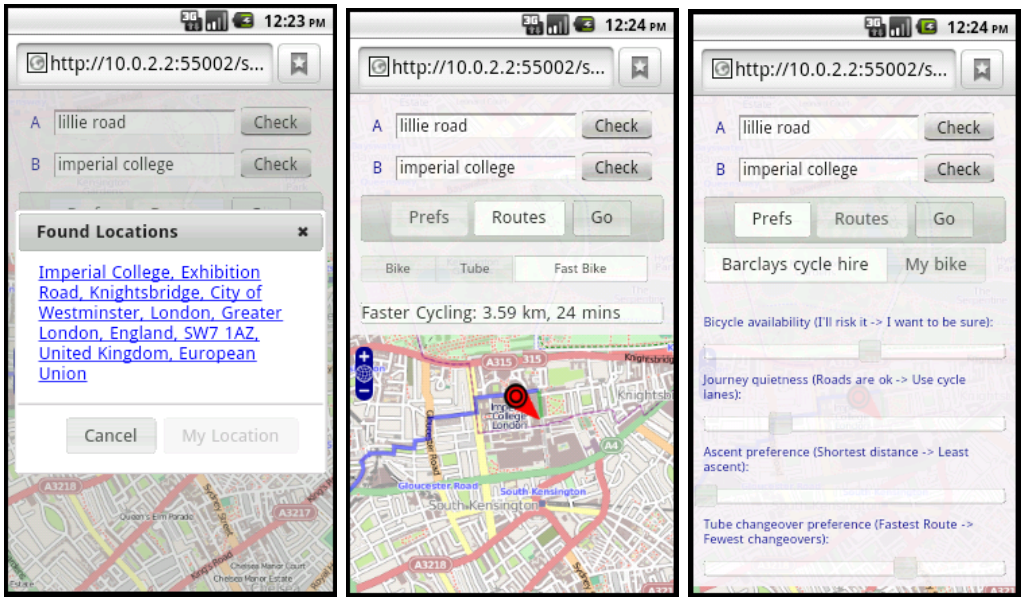


Figure 43: Marking start and finish locations, selecting a route to show and setting preferences on an Android device in the smallest screen size category

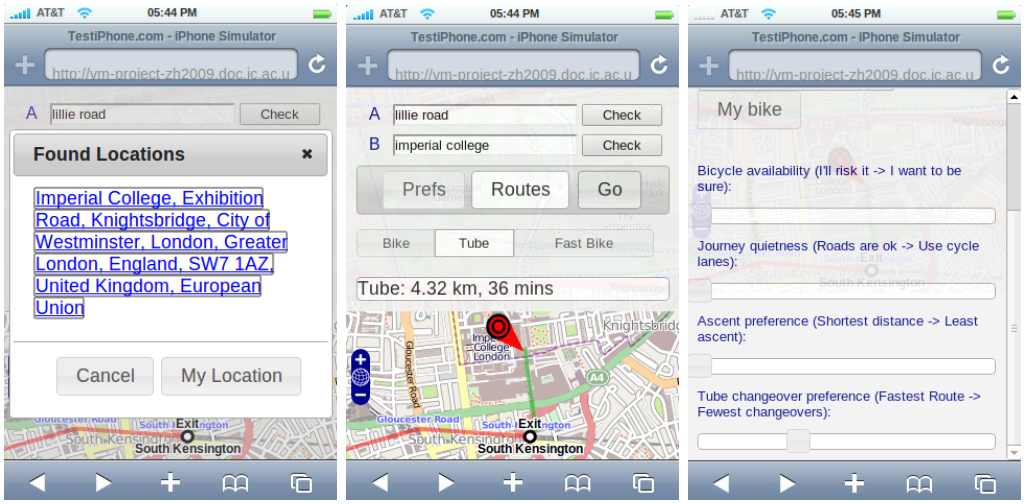


Figure 44: Marking start and finish locations, selecting a route to show and setting preferences on the iPhone

10.7.3 iPad

For iPad we used the ipadpeek.com [11].

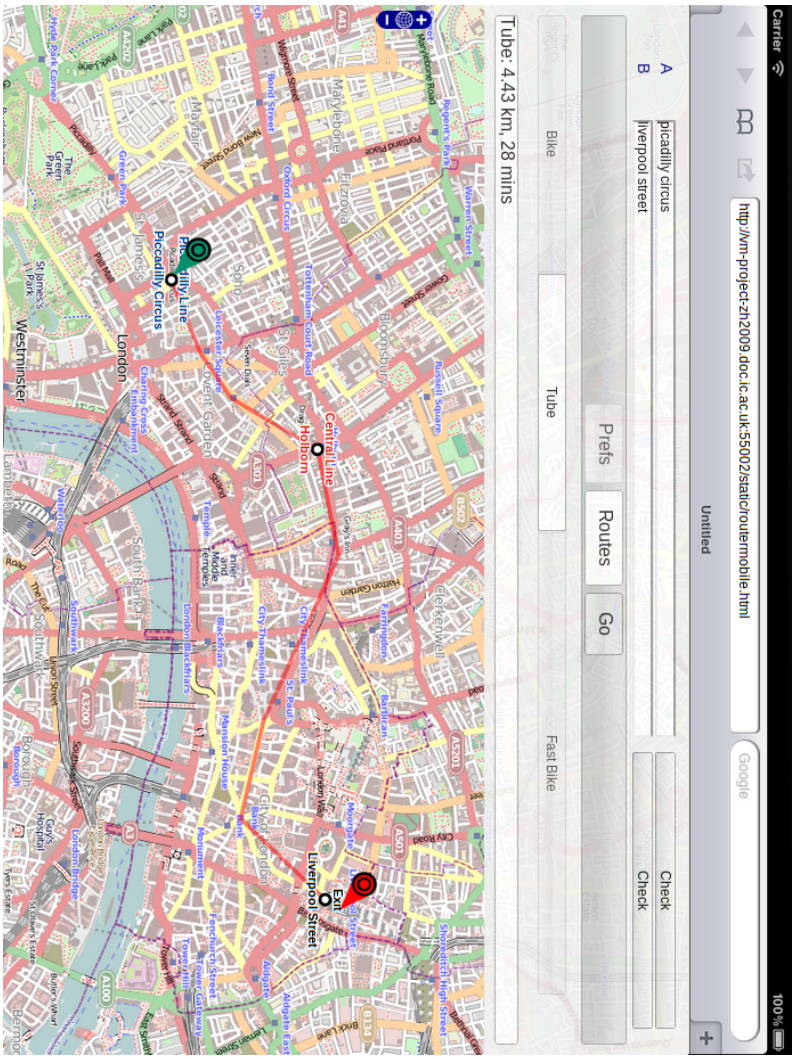


Figure 45: Underground route marked on the iPad in landscape mode

11 Conclusion and Future Work

Through the course of this investigation, we have:

1. Challenged the assumption made by Kaleta's prediction model (Chapter 5) that the pickup and dropoff rate within each interval follows a Poisson distribution and demonstrated that using a mixture of Poisson distributions results in predictions with a lower root mean squared error (Chapter 6).

2. Integrated the mixture model of pickups and dropoffs into a complete journey planner based on the one built by Kaleta .

3. Added the ability to calculate routes with respect to a user defined tube transfer time and ascent averseness preference (Chapter 7) and demonstrated that the journey planner is capable of finding sensible routes in practice (Chapter 10).

4. Added new features to make the journey planner more informative, including the docking station status overlay, the route elevation profiles and the display of tube changeovers on the map (Chapter 8).

5. Improved journey planner usability by developing a handheld optimized version of the web application capable of running across different handheld platforms (Chapter 9).

In the process of devising our machine learning models, we found there was a statistically significant case for saying that the data did not follow a Poisson distribution within a 15 minute time interval. At least some of this this could be explained by the overdispersion of the data due to the change in rate within each interval, but the distance between peaks in the data suggests that there are other hidden variables at work.

From looking at the weekly trends in pickups and dropoffs, there seems to be a case to split the data into weekday and weekend data. We found most stations have less usage on weekends than weekdays. However, even after the data is split into weekday and weekend data in relatively small time intervals, a mixture model still explains the data significantly better statistically than a single Poisson distribution, which suggest there are still one or more hidden variables which cause this variability.

We also found that whatever prediction model we used, for docking stations with low usage there was very little difference in root mean square error between them. Future work could focus on improving the prediction error for the busy time periods at busy stations, which we are most interested in.

During our modifications to the routing, we found that Kaleta's exponential cost function for combining edge attribute costs didn't work as well as a simple weighted sum in practice, but we were able to do without the benefits of this cost function by reasoning about every attribute in terms an additional cost in length.

Finally, we found that our sampling method only considers whether a bike is available at the end of each rollout, not how many are available which could potentially direct the user to a docking station with only one or two bikes on average. This could result in the user taking a greater risk than they might expect considering the way the risk averseness preference is set. Future work could additionally consider the number of bikes available at the end of each rollout.

11.1 Investigating the cause of hidden variability

A mixture model explains the data significantly better than a single Poisson distribution by accounting for the hidden variables, but this is of limited use in making improved predictions for the reasons discussed in section 10.1. This could be improved by investigating the data more thoroughly and finding out:

1. Whether there is a common cause of the hidden variability across all the docking stations, or at least enough docking stations so that we can significantly improve our predictions

2. The cause of the hidden variability

This could improve our prediction. For example if we looked at weather records and found that wet days and dry days had significantly different distributions of bike pickups or dropoffs within each time interval. We could then partition our data into wet days and dry days and learn the model parameters of each type of day separately. Then, when the user requests a route, we could check the current days weather from the BBC's weather RSS feeds and choose the model to use accordingly.

11.2 Improving the search for mixed modes of transport

Finding mixed routes requires searching for multiple cycle routes, from each stop along the underground to the end point. This means the number of A* searches grows with the length of the tube route between the start and end point, which is potentially extremely expensive A* in the worst case can expand an exponential number of nodes in the length of the solution. These routes are then sorted to find the one closest to the time the user allows.

Another problem with our mixed route algorithm is that it is inflexible. There is no way to find a journey that has two tube portions with a bike portion in between, or to find a journey that involves a walk between tube stations that are close together but serve different lines, as is the case for Bank and Monument. This kind of flexibility is even more important if we are to add more modes of transport. One approach to doing this in the literature is to use a generalized form of Dijkstra's algorithm, as described by Barrett, Jacob and Marathe [24]. Each edge in the graph is augmented with a label. For example bus edges could be labeled with b, tube edges with t, cycling edges with c and w for walking. It is then possible to find the shortest path subject to the constraint that the edge labels along the path form a valid sentence in a context free grammar that we define. This easily generalizes to many modes of transport while allowing us to keep the constraints we need, like having to drop off a cycle hire bike at a docking station before reaching the destination and is solvable in polynomial time. A simple grammar for a cycle hire portion of a route could be:

$$
\begin{aligned}
\mathit{CYCLEHIRE} &\rightarrow \mathit{WALKING}\ \mathit{CYCLING}\ \mathit{WALKING} \\
\mathit{WALKING} &\rightarrow w\ \mathit{WALKING} \\
\mathit{WALKING} &\rightarrow w \\
\mathit{CYCLING} &\rightarrow c\ \mathit{CYCLING} \\
\mathit{CYCLING} &\rightarrow c
\end{aligned}
$$

A route with edge labels $wwwccwwww$ would be a valid sentence in this grammar, while $wwwccc$ would not.

11.3 Memory Usage

The NetworkX library allocates a python dictionary for the attributes of each edge in the graph. This is not memory efficient - python dictionaries

are implemented as hash tables which are grown dynamically so that there are enough empty buckets to avoid key collisions. These dictionaries are the largest source of memory usage in representing the graphs. We do not need the flexibility of dictionaries as we know the number of attributes and the type of each attribute in advance and don't need to add new attributes at runtime. The edge attributes could be instead stored efficiently in a class or python *namedtuple*. NetworkX doesn't allow this, but if we were to design a new routing algorithm from the ground up as discussed in Section 11.2 it would be feasible to simultaneously write NetworkX out of the application and either use a different library or our own data structures for the graph.

11.4 Improving tube journey planning

Currently, we allow the user to enter the expected transfer time, which we treat as the cost of all transfers in the graph. We could instead have a separate transfer cost for each possible transfer at each station. We could add some mechanism to get user feedback on journeys they've made and update the costs of transfers accordingly. It could be possible to have different transfer costs depending on the time of day.

TFL have released a new real time tube data feed, which was not available at the beginning of this project which includes a train prediction service with station and line status. This could be integrated into the routing so that any time spent waiting for a train could be accounted for.

Our current London Underground data only includes the shortest travel time between each pair of adjacent stations. We could make more accurate predictions if we had travel times between stations for each London Underground line, for each time of day.

We could use this real time tube data feed to find expected tube journey times between adjacent stations for each London Underground line at each time of day. It is possible to obtain a full tube timetable from TFL by making a Freedom of Information Act request, which could also help us plan better routes.

11.5 Online learning

While we have demonstrated that a single Poisson distribution doesn't explain the data well, one of the strengths of using it is that it is possible to update the rate parameter using TFL's live data feeds. Kaleta [30] does

this by assuming the ratio between the pickups and all events (pickups and dropoffs) remains constant and using an iterative average.

It would be useful to evaluate whether this method results in better predictions. If so, a feasible way of extending this to our mixture model would be to weight the observed change in the number of pickup events by the responsibility of each Poisson distribution for generating the data π_k, assuming the responsibilities also remain constant.

11.6 Turn by turn navigation

This would make the journey planner more useful by naming the streets along which the user needs to travel, as well as naming where each mode of transport should be taken. This could be done using reverse geocoding, which is provided by Nominatim.

Appendices

A Hypothesis test results

Here we present the full listings of the likelihood ratio tests and conditional chi-squared tests we performed. For each time interval we tested, we present the P-value calculated for each test and whether the null hypothesis was rejected or not at the 1% significance level.

Figure 46: Hypothesis test results for pickups in interval 0800-0815

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.058	False	0.023	False
The Green Bridge, Mile End	0.000	True	0.001	True
Waterloo Station 3, Waterloo	0.366	False	0.006	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.001	True	0.000	True
Wright's Lane, Kensington	0.001	True	0.002	True
Nevern Place, Earl's Court	1.000	False	0.006	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.998	False	0.374	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.001	True	0.000	True
Vauxhall Bridge , Pimlico	1.000	False	0.835	False
Wansey Street, Walworth	0.978	False	0.361	False
Castlehaven Road, Camden Town	0.979	False	0.446	False
Russell Gardens, Holland Park	1.000	False	0.594	False
Green Park Station, West End	0.054	False	0.000	True
Elizabeth Bridge, Victoria	0.003	True	0.011	False
Rochester Row, Westminster	0.000	True	0.000	True

Figure 47: Hypothesis test results for pickups in interval 0815-0830

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.557	False	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.640	False	0.028	False
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	1.000	False	0.012	False
South Kensington Station, South Kensington	0.000	True	0.001	True
Westfield Ariel Way, White City	1.000	False	0.141	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.020	False	0.000	True
Vauxhall Bridge , Pimlico	1.000	False	0.141	False
Wansey Street, Walworth	0.978	False	0.361	False
Castlehaven Road, Camden Town	0.920	False	0.171	False
Russell Gardens, Holland Park	1.000	False	0.716	False
Green Park Station, West End	0.274	False	0.000	True
Elizabeth Bridge, Victoria	0.113	False	0.186	False
Rochester Row, Westminster	0.000	True	0.000	True

Figure 48: Hypothesis test results for pickups in interval 0830-0845

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.014	False	0.033	False
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.721	False	0.003	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	1.000	False	0.578	False
South Kensington Station, South Kensington	0.015	False	0.034	False
Westfield Ariel Way, White City	1.000	False	0.001	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.000	True
Vauxhall Bridge , Pimlico	0.938	False	0.000	True
Wansey Street, Walworth	0.985	False	0.103	False
Castlehaven Road, Camden Town	0.961	False	0.076	False
Russell Gardens, Holland Park	1.000	False	0.636	False
Green Park Station, West End	0.017	False	0.000	True
Elizabeth Bridge, Victoria	0.015	False	0.003	True
Rochester Row, Westminster	0.000	True	0.000	True

Figure 49: Hypothesis test results for pickups in interval 0845-0900

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.178	False	0.001	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.622	False	0.117	False
Belgrove Street , King's Cross	0.001	True	0.014	False
Old Quebec Street, Marylebone	0.139	False	0.087	False
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	1.000	False	0.702	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.030	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.162	False
Vauxhall Bridge , Pimlico	0.909	False	0.000	True
Wansey Street, Walworth	0.925	False	0.735	False
Castlehaven Road, Camden Town	0.661	False	0.053	False
Russell Gardens, Holland Park	1.000	False	0.735	False
Green Park Station, West End	1.000	False	0.000	True
Elizabeth Bridge, Victoria	0.091	False	0.000	True
Rochester Row, Westminster	0.645	False	0.153	False

Figure 50: Hypothesis test results for pickups in interval 1400-1415

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.267	False	0.031	False
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.002	True	0.002	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.986	False	0.017	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.048	False
Wormwood Street, Liverpool Street	0.000	True	0.001	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.986	False	0.000	True
Vauxhall Bridge , Pimlico	0.978	False	0.005	True
Wansey Street, Walworth	0.240	False	0.097	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.006	True
Green Park Station, West End	0.998	False	0.000	True
Elizabeth Bridge, Victoria	0.063	False	0.021	False
Rochester Row, Westminster	0.480	False	0.040	False

Figure 51: Hypothesis test results for pickups in interval 1415-1430

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.103	False	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.929	False	0.004	True
South Kensington Station, South Kensington	0.001	True	0.007	True
Westfield Ariel Way, White City	1.000	False	0.169	False
Wormwood Street, Liverpool Street	0.003	True	0.039	False
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.235	False	0.000	True
Vauxhall Bridge , Pimlico	0.205	False	0.000	True
Wansey Street, Walworth	0.025	False	0.000	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.018	False
Green Park Station, West End	0.168	False	0.000	True
Elizabeth Bridge, Victoria	0.001	True	0.000	True
Rochester Row, Westminster	0.499	False	0.174	False

Figure 52: Hypothesis test results for pickups in interval 1430-1445

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.082	False	0.001	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.571	False	0.003	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.030	False
Wormwood Street, Liverpool Street	0.000	True	0.001	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.948	False	0.000	True
Vauxhall Bridge , Pimlico	0.233	False	0.000	True
Wansey Street, Walworth	0.324	False	0.063	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.037	False
Green Park Station, West End	0.000	True	0.000	True
Elizabeth Bridge, Victoria	0.001	True	0.000	True
Rochester Row, Westminster	0.980	False	0.155	False

Figure 53: Hypothesis test results for pickups in interval 1445-1500

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.421	False	0.017	False
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.019	False	0.011	False
Wright's Lane, Kensington	0.003	True	0.002	True
Nevern Place, Earl's Court	0.778	False	0.026	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.000	True
Vauxhall Bridge , Pimlico	0.188	False	0.000	True
Wansey Street, Walworth	0.233	False	0.001	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.993	False	0.000	True
Green Park Station, West End	0.096	False	0.000	True
Elizabeth Bridge, Victoria	0.001	True	0.000	True
Rochester Row, Westminster	0.897	False	0.352	False

Figure 54: Hypothesis test results for pickups in interval 1700-1715

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.001	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.002	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.000	True	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.006	True	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.005	True	0.106	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.093	False	0.003	True
Green Park Station, West End	0.003	True	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.000	True	0.001	True

Figure 55: Hypothesis test results for pickups in interval 1715-1730

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.002	True
Wright's Lane, Kensington	0.002	True	0.001	True
Nevern Place, Earl's Court	0.001	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.606	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.541	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.003	True	0.074	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.166	False	0.110	False
Green Park Station, West End	0.958	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.838	False	0.099	False

Figure 56: Hypothesis test results for pickups in interval 1730-1745

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.001	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.002	True	0.002	True
Wright's Lane, Kensington	0.001	True	0.004	True
Nevern Place, Earl's Court	0.010	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.458	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.301	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.000	True	0.001	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.296	False	0.017	False
Green Park Station, West End	0.004	True	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.388	False	0.012	False

Figure 57: Hypothesis test results for pickups in interval 1745-1800

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.045	False	0.272	False
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.001	True	0.018	False
Nevern Place, Earl's Court	0.009	True	0.006	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.218	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.001	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.093	False
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.000	True	0.000	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.068	False	0.000	True
Green Park Station, West End	0.201	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.811	False	0.031	False

Figure 58: Hypothesis test results for dropoffs in interval 0800-0815

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.202	False	0.429	False
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.170	False	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.001	True
Nevern Place, Earl's Court	0.999	False	0.051	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.045	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.000	True	0.000	True
Vauxhall Bridge , Pimlico	1.000	False	0.731	False
Wansey Street, Walworth	0.987	False	0.487	False
Castlehaven Road, Camden Town	0.983	False	0.024	False
Russell Gardens, Holland Park	1.000	False	0.572	False
Green Park Station, West End	0.797	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.000	True	0.000	True

Figure 59: Hypothesis test results for dropoffs in interval 0815-0830

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.699	False	0.304	False
The Green Bridge, Mile End	0.004	True	0.003	True
Waterloo Station 3, Waterloo	0.763	False	0.185	False
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.001	True	0.000	True
Wright's Lane, Kensington	0.044	False	0.139	False
Nevern Place, Earl's Court	1.000	False	0.771	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.978	False	0.005	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.000	True	0.000	True
Vauxhall Bridge , Pimlico	1.000	False	0.416	False
Wansey Street, Walworth	0.986	False	0.529	False
Castlehaven Road, Camden Town	0.991	False	0.487	False
Russell Gardens, Holland Park	1.000	False	0.657	False
Green Park Station, West End	0.825	False	0.000	True
Elizabeth Bridge, Victoria	0.015	False	0.153	False
Rochester Row, Westminster	0.000	True	0.000	True

Figure 60: Hypothesis test results for dropoffs in interval 0830-0845

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.316	False	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.712	False	0.115	False
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	1.000	False	0.097	False
South Kensington Station, South Kensington	0.000	True	0.001	True
Westfield Ariel Way, White City	1.000	False	0.416	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.028	False	0.000	True
Vauxhall Bridge , Pimlico	1.000	False	0.162	False
Wansey Street, Walworth	0.992	False	0.173	False
Castlehaven Road, Camden Town	0.960	False	0.294	False
Russell Gardens, Holland Park	1.000	False	0.735	False
Green Park Station, West End	0.035	False	0.000	True
Elizabeth Bridge, Victoria	0.083	False	0.178	False
Rochester Row, Westminster	0.000	True	0.000	True

Figure 61: Hypothesis test results for dropoffs in interval 0845-0900

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.003	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.379	False	0.170	False
Belgrove Street , King's Cross	0.000	True	0.013	False
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.001	True
Nevern Place, Earl's Court	0.996	False	0.239	False
South Kensington Station, South Kensington	0.000	True	0.006	True
Westfield Ariel Way, White City	1.000	False	0.174	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.000	True
Vauxhall Bridge , Pimlico	0.993	False	0.001	True
Wansey Street, Walworth	0.992	False	0.662	False
Castlehaven Road, Camden Town	0.873	False	0.027	False
Russell Gardens, Holland Park	1.000	False	0.677	False
Green Park Station, West End	0.945	False	0.000	True
Elizabeth Bridge, Victoria	0.060	False	0.034	False
Rochester Row, Westminster	0.000	True	0.000	True

Figure 62: Hypothesis test results for dropoffs in interval 1400-1415

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.176	False	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.004	True
Old Quebec Street, Marylebone	0.045	False	0.004	True
Wright's Lane, Kensington	0.002	True	0.002	True
Nevern Place, Earl's Court	0.961	False	0.013	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.697	False
Wormwood Street, Liverpool Street	0.014	False	0.150	False
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	1.000	False	0.031	False
Vauxhall Bridge , Pimlico	0.733	False	0.014	False
Wansey Street, Walworth	0.228	False	0.025	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.003	True
Green Park Station, West End	0.997	False	0.000	True
Elizabeth Bridge, Victoria	0.031	False	0.002	True
Rochester Row, Westminster	0.270	False	0.017	False

Figure 63: Hypothesis test results for dropoffs in interval 1415-1430

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.765	False	0.002	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.002	True	0.001	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.999	False	0.030	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.995	False	0.000	True
Wormwood Street, Liverpool Street	0.001	True	0.033	False
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.093	False	0.000	True
Vauxhall Bridge , Pimlico	0.986	False	0.000	True
Wansey Street, Walworth	0.527	False	0.065	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.000	True
Green Park Station, West End	0.945	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.586	False	0.001	True

Figure 64: Hypothesis test results for dropoffs in interval 1430-1445

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.054	False	0.001	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.986	False	0.063	False
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.169	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.422	False	0.000	True
Vauxhall Bridge , Pimlico	0.397	False	0.000	True
Wansey Street, Walworth	0.152	False	0.004	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.026	False
Green Park Station, West End	0.608	False	0.000	True
Elizabeth Bridge, Victoria	0.042	False	0.000	True
Rochester Row, Westminster	0.107	False	0.000	True

Figure 65: Hypothesis test results for dropoffs in interval 1445-1500

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.262	False	0.003	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.770	False	0.001	True
South Kensington Station, South Kensington	0.000	True	0.004	True
Westfield Ariel Way, White City	1.000	False	0.002	True
Wormwood Street, Liverpool Street	0.000	True	0.001	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.651	False	0.000	True
Vauxhall Bridge , Pimlico	0.851	False	0.000	True
Wansey Street, Walworth	0.753	False	0.496	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	1.000	False	0.071	False
Green Park Station, West End	0.009	True	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.962	False	0.376	False

Figure 66: Hypothesis test results for dropoffs in interval 1700-1715

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.003	True	0.004	True
Nevern Place, Earl's Court	0.003	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	1.000	False	0.011	False
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.033	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.000	True	0.001	True
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.004	True	0.000	True
Green Park Station, West End	0.000	True	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.372	False	0.196	False

Figure 67: Hypothesis test results for dropoffs in interval 1715-1730

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.002	True	0.018	False
Wright's Lane, Kensington	0.000	True	0.000	True
Nevern Place, Earl's Court	0.002	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.729	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.552	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.003	True	0.042	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.395	False	0.029	False
Green Park Station, West End	0.048	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.246	False	0.000	True

Figure 68: Hypothesis test results for dropoffs in interval 1730-1745

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.005	True
Wright's Lane, Kensington	0.000	True	0.001	True
Nevern Place, Earl's Court	0.009	True	0.007	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.000	True	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.862	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.001	True	0.020	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.198	False	0.018	False
Green Park Station, West End	0.021	False	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.078	False	0.000	True

Figure 69: Hypothesis test results for dropoffs in interval 1745-1800

Station	Likelihood Ratio P-value	Reject	Chi-squared P-value	Reject
Edgware Road Station, Paddington	0.000	True	0.000	True
The Green Bridge, Mile End	0.000	True	0.000	True
Waterloo Station 3, Waterloo	0.000	True	0.000	True
Belgrove Street , King's Cross	0.000	True	0.000	True
Old Quebec Street, Marylebone	0.000	True	0.000	True
Wright's Lane, Kensington	0.010	False	0.057	False
Nevern Place, Earl's Court	0.001	True	0.000	True
South Kensington Station, South Kensington	0.000	True	0.000	True
Westfield Ariel Way, White City	0.084	False	0.000	True
Wormwood Street, Liverpool Street	0.000	True	0.000	True
Moor Street, Soho	0.000	True	0.000	True
Jubilee Plaza, Canary Wharf	0.000	True	0.000	True
Monument Street, Monument	0.507	False	0.000	True
Vauxhall Bridge , Pimlico	0.000	True	0.000	True
Wansey Street, Walworth	0.001	True	0.015	False
Castlehaven Road, Camden Town	0.000	True	0.000	True
Russell Gardens, Holland Park	0.352	False	0.009	True
Green Park Station, West End	0.009	True	0.000	True
Elizabeth Bridge, Victoria	0.000	True	0.000	True
Rochester Row, Westminster	0.312	False	0.009	True

Figure 70: Total acceptances and rejections for dropoffs

Interval	Likelihood Ratio Test		Chi-squared test	
	Rejections	Acceptances	Rejections	Acceptances
0800-0815	11	9	13	7
0815-0830	9	11	11	9
0830-0845	9	11	12	8
0845-0900	10	10	12	8
1400-1415	8	12	13	7
1415-1430	11	9	17	3
1430-1445	10	10	17	3
1445-1500	12	8	17	3
1700-1715	17	3	18	2
1715-1730	15	5	17	3
1730-1745	16	4	18	2
1745-1800	15	5	18	2
Total	143.00	97.00	183.00	57.00
Percentage	59.6%	40.4%	76.3%	23.8%

Figure 71: Total acceptances and rejections for pickups

Interval	Likelihood Ratio Test		Chi-squared test	
	Rejections	Acceptances	Rejections	Acceptances
0800-0815	11	9	13	7
0815-0830	9	11	12	8
0830-0845	8	12	14	6
0845-0900	7	13	10	10
1400-1415	10	10	14	6
1415-1430	11	9	16	4
1430-1445	12	8	16	4
1445-1500	10	10	16	4
1700-1715	18	2	18	2
1715-1730	15	5	17	3
1730-1745	16	4	18	2
1745-1800	14	6	16	4

Total	141.00	99.00	180.00	60.00
Percentage	58.8%	41.3%	75.0%	25.0%

B Optimal number of Poisson mixtures

For each time interval, we present the maximum number of Poisson distributions k such that when a likelihood ratio test is performed, the mixture of k Poissons fits the data better than a mixture of $k-1$ Poissons at the 1% significance level.

Figure 72: Optimal number of Poisson distributions to use to fit the number of pickups in each interval

Station	800-815	815-830	830-845	845-900	1400-1415	1415-1430	1430-1445	1445-1500	1700-1715	1715-1730	1730-1745	1745-1800
Edgware Road Stn., Paddington	1	1	2	1	1	1	1	1	2	2	2	1
The Green Bridge, Mile End	2	2	2	2	2	2	2	3	2	2	2	2
Waterloo Stn 3, Waterloo	1	1	1	1	2	2	2	2	5	4	4	4
Belgrove Street , King's Cross	2	2	2	1	2	2	2	3	4	4	4	3
Old Quebec St, Marylebone	2	2	2	1	2	2	2	1	2	2	2	2
Wright's Lane, Kensington	2	2	2	2	2	2	2	2	2	2	2	2
Nevern Place, Earl's Court	1	1	1	1	1	1	1	1	2	2	2	2
South Kensington Station	2	2	1	2	2	1	2	2	2	2	2	2
Westfield Ariel Way, White City	1	1	1	1	1	1	1	1	1	1	1	1
Wormwood St, Liverpool St	3	3	3	2	1	1	2	2	3	3	2	2
Moor Street, Soho	2	2	2	2	2	2	2	2	2	2	2	2
Jubilee Plaza, Canary Wharf	2	2	2	2	2	2	2	2	2	3	2	2
Monument St, Monument	1	1	1	1	1	1	1	1	1	1	1	1
Vauxhall Bridge , Pimlico	1	1	1	1	1	1	1	1	2	2	2	2
Wansey Street, Walworth	1	1	1	1	1	1	1	1	1	1	2	2
Castlehaven Road	1	1	1	1	2	2	2	2	3	2	2	2
Russell Gardens, Holland Park	1	1	1	1	1	1	1	1	1	1	1	1
Green Park Stn, West End	1	1	1	1	1	1	1	1	1	1	1	1
Elizabeth Bridge, Victoria	1	1	2	1	2	1	2	2	2	2	2	2
Rochester Row, Westminster	2	2	1	1	1	1	1	1	1	1	1	1

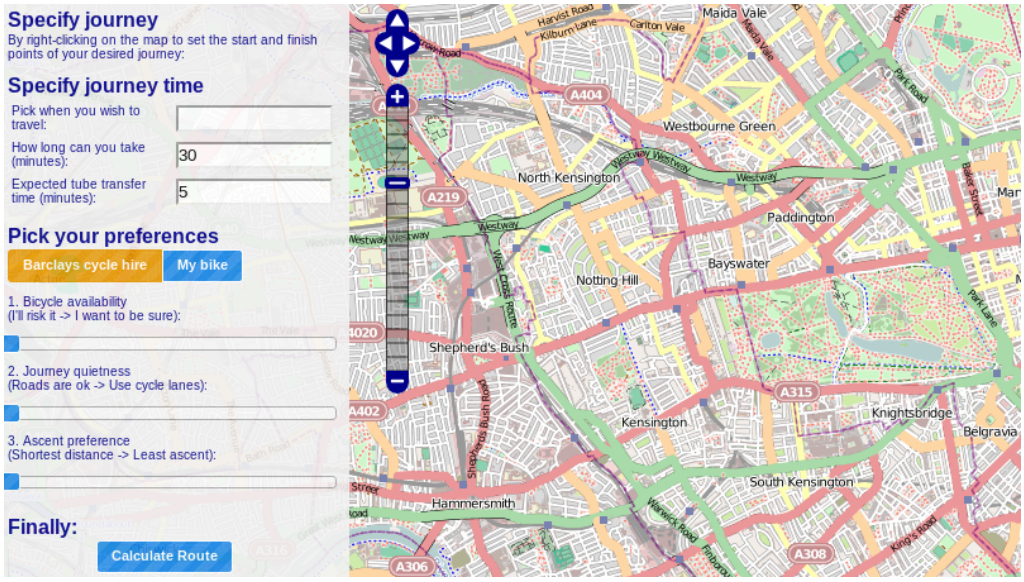
Figure 73: Optimal number of Poisson distributions to use to fit the number of dropoffs in each interval

Station	800-815	815-830	830-845	845-900	1400-1415	1415-1430	1430-1445	1445-1500	1700-1715	1715-1730	1730-1745	1745-1800
Edgware Road Stn., Paddington	1	1	1	2	1	1	2	1	2	2	2	2
The Green Bridge, Mile End	2	2	2	2	2	2	2	2	2	3	2	3
Waterloo Stn 3, Waterloo	2	1	1	1	2	2	2	2	4	4	4	3
Belgrove Street , King's Cross	2	2	2	2	2	2	2	2	3	4	4	4
Old Quebec St, Marylebone	2	2	2	2	1	2	2	2	3	1	2	2
Wright's Lane, Kensington	2	1	2	2	2	2	2	2	2	2	2	1
Nevern Place, Earl's Court	1	1	1	1	1	1	1	1	2	2	1	2
South Kensington Station	2	2	2	2	2	2	2	2	2	2	2	2
Westfield Ariel Way, White City	1	1	1	1	1	1	1	1	1	1	1	1
Wormwood St, Liverpool St	3	2	2	3	1	1	2	2	3	3	3	2
Moor Street, Soho	2	3	2	2	2	2	2	2	2	2	2	3
Jubilee Plaza, Canary Wharf	3	2	2	2	2	2	2	2	2	2	2	2
Monument St, Monument	1	1	1	1	1	1	1	1	1	1	1	1
Vauxhall Bridge , Pimlico	1	1	1	1	1	1	1	1	2	2	3	2
Wansey Street, Walworth	1	1	1	1	1	1	1	1	2	1	1	2
Castlehaven Road	1	1	1	1	2	2	2	2	2	2	2	2
Russell Gardens, Holland Park	1	1	1	1	1	1	1	1	2	1	1	1
Green Park Stn, West End	1	1	1	1	1	1	1	1	1	1	1	1
Elizabeth Bridge, Victoria	2	1	1	1	1	1	1	3	2	2	2	2
Rochester Row, Westminster	2	2	1	1	1	1	1	1	1	1	1	1

C Web journey planner user guide

Pan the map by clicking and dragging with the mouse. Zoom using the mouse wheel. This can also be done using the control overlaid on the left hand side of the map.

Figure 74: Journey planner



First select a start and end position for the journey on the map. This can be done by right clicking a point and choosing 'Set start' or 'Set finish' from the context menu (Figure 75).

The start position will be marked in green and the finish position in red (Figure 76).

Choose your preferences using the options box on the left. First choose the time you want to start your journey using the time picker(Figure 77). All the other preferences have defaults which may be adjusted. When ready, click 'Calculate Route'.

A box with routes is returned (Figure 78). There are tabs containing different types of routes:

- *Preferred cycling routes* These are the cycling routes that most closely match the preferences you set.

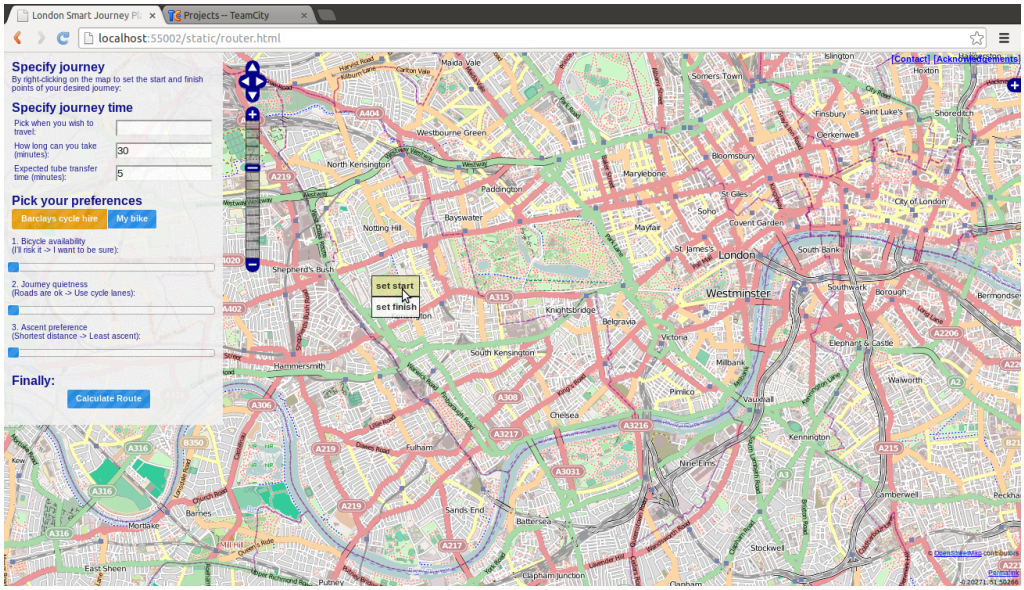


Figure 75: Setting start and finish positions

- *Tube routes* These are the tube routes that match your expected tube changeover time.
- *Mixed routes* These routes which potentially contain both tube and bike portions, which match the time you allow for the journey as closely as possible.
- *Faster cycling routes* These are faster routes which do not take any of the preferences you set into consideration.

Click any of the checkboxes to show the corresponding route on the map. The 'elevation profile' button will bring up a graph of the height along the route and give an idea of the ascent and descent involved.

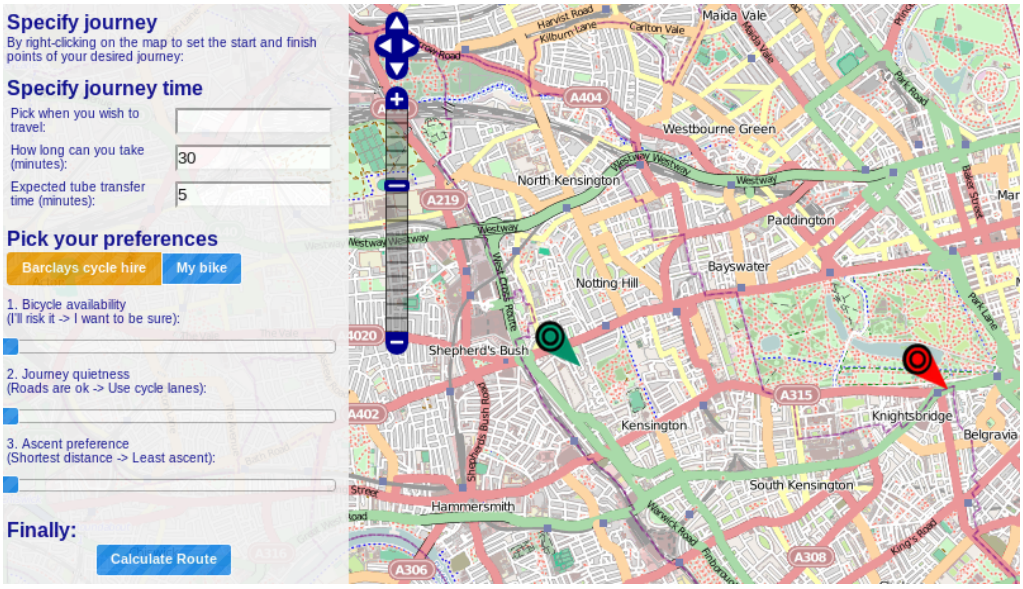


Figure 76: Setting start and finish positions

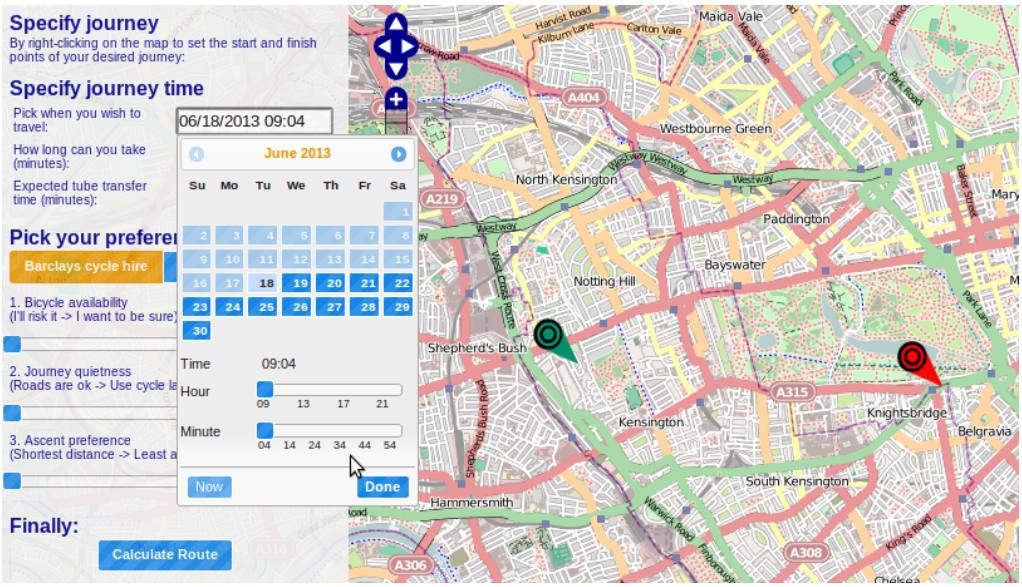


Figure 77: Setting preferences

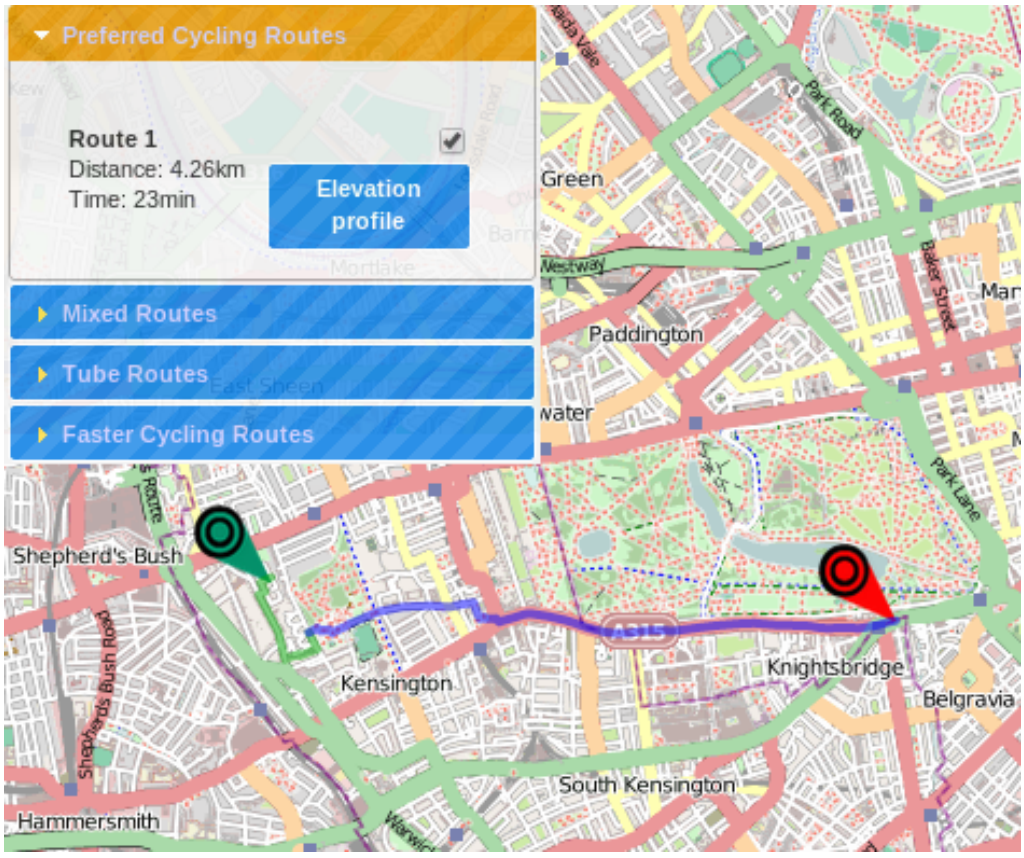


Figure 78: Found routes

D Mobile journey planner user guide

The mobile application marks your location on the map (Figure 79, if available with a red kite marker. The map can be panned by touching and the screen and dragging.

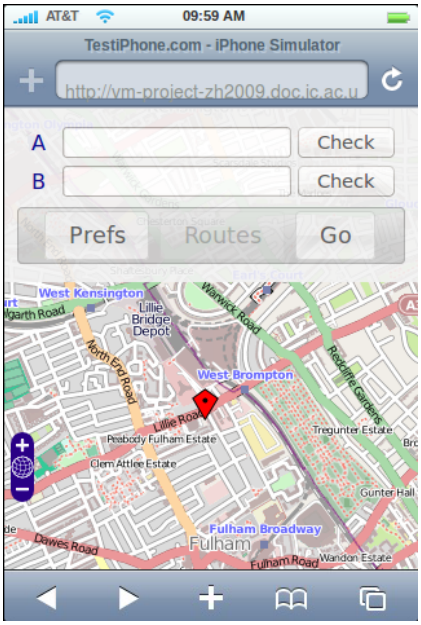


Figure 79: Main screen of the mobile application

To set the start and finish locations, type an address into the corresponding text box and click 'Check'. A list of matching addresses will pop up. Touch one to set it as the location. Alternatively, clicking the 'My Location' button will set the location to your current position, if it's available (Figure 80).

Click the 'Prefs' button to toggle the preferences screen and adjust the preference sliders as desired (Figure 81).

When ready, click 'Go' to calculate the route. A results box with buttons for the bike, tube and fast bike routes are shown. Click one to mark it on the map (Figure 82).

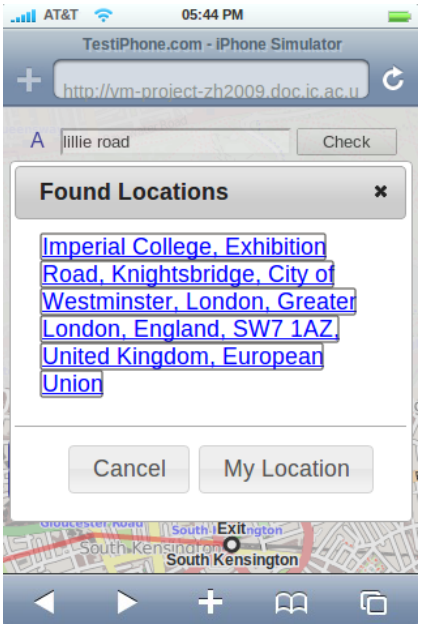


Figure 80: Marking start and finish locations

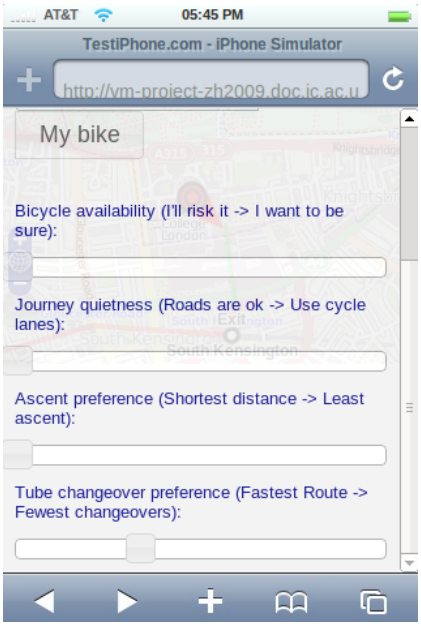


Figure 81: Setting preferences

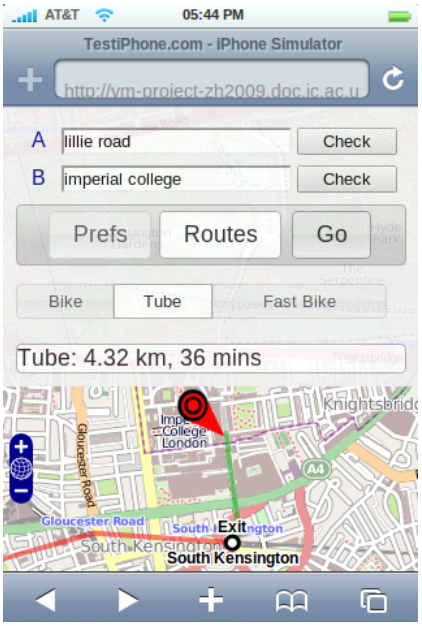


Figure 82: Setting preferences

E Plotting module user guide

From the main screen, we can choose whether to plot the pickups and dropoffs over time or the frequency of events in a given time interval by clicking the relevant link. We can plot the mean number of pickups or dropoffs over a single day or whole week for any docking station by choosing the appropriate drop down menus. Start typing in the 'Station' text box to get an autocomplete menu of available docking stations (Figure 83).

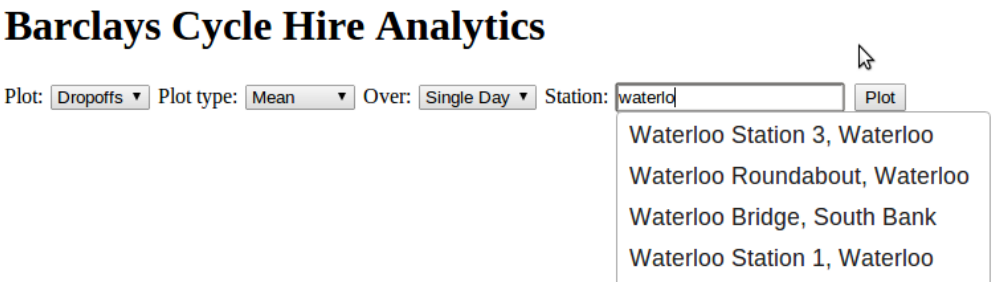


Figure 83: Selecting a station to plot

Click 'Plot' to generate the plot. The result is show in figure 84 The frequency plots work in a similar way, but give the additional option of separating the data into weekdays and weekends.

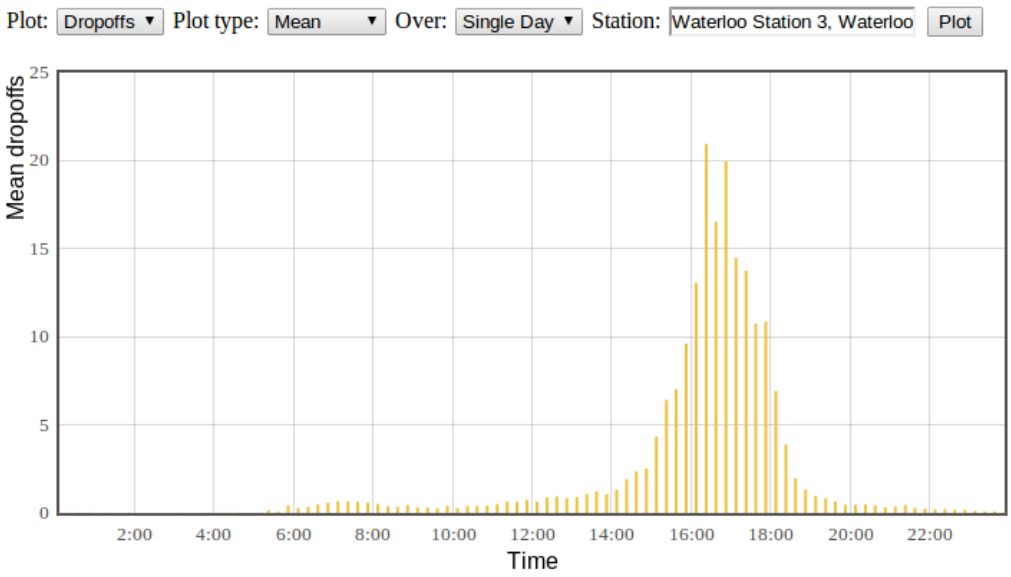


Figure 84: Selecting a station to plot

References

[1] Barclays cycle hire map by transport for london. https://web.barclayscyclehire.tfl.gov.uk/maps. Accessed: 06/06/2013.

[2] Bikes on public transport. http://www.tfl.gov.uk/roadusers/cycling/11701.aspx. Accessed: 10/06/2013.

[3] Cycle journey planner by transport for london. http://cyclejourneyplanner.tfl.gov.uk/. Accessed: 14/01/2013.

[4] Cycle journey planner by transport for london. http://en.wikipedia.org/wiki/File:Poisson_pmf.svg. Accessed: 14/01/2013.

[5] Cycle journey planner by transport for london. http://en.wikipedia.org/wiki/File:Normal_Distribution_PDF.svg. Accessed: 14/01/2013.

[6] Cyclestreets journey planner. http://www.cyclestreets.net/journey. Accessed: 14/01/2013.

[7] Geoff marshall. distance between stations. http://ni.chol.as/media/geoff-files/sillymaps/milesdistances.gif. Accessed: 03/2012.

[8] Geoff marshall. travel times between stations. http://ni.chol.as/media/geoff-files/sillymaps/travel_times.jpg. Accessed: 03/2012.

[9] Google maps geocoding api. https://developers.google.com/maps/documentation/geocoding/. Accessed: 13/06/2012.

[10] Google maps journey planner. http://maps.google.com. Accessed: 14/01/2013.

[11] iphone simulator. http://www.testiphone.com/. Accessed: 15/06/2012.

[12] Lagrange multiplier. http://en.wikipedia.org/wiki/Lagrange_multiplier. Accessed: 13/06/2012.

[13] Nominatim. http://wiki.openstreetmap.org/wiki/Nominatim. Accessed: 13/06/2012.

[14] Openstreetmap community. list of london underground stations. http://wiki.openstreetmap.org/wiki/London_Tube_Stations. Accessed: 05/06/2013.

[15] Optitrans london journey planner. http://london.optitrans.net. Accessed: 14/01/2013.

[16] Osmosis. http://wiki.openstreetmap.org/wiki/Osmosis. Accessed: 06/06/2012.

[17] Shuttle radar topography mission. http://www2.jpl.nasa.gov/srtm/northAmerica.htm. Accessed: 06/06/2012.

[18] Srtm north america images. http://www2.jpl.nasa.gov/srtm/northAmerica.htm. Accessed: 06/06/2012.

[19] Srtm plugin for openstreetmap's osmosis. http://code.google.com/p/osmosis-srtm-plugin/. Accessed: 06/06/2012.

[20] Transport for london corporate schemes. http://www.tfl.gov.uk/corporate/projectsandschemes/26296.aspx. Accessed: 05/06/2013.

[21] Transport for london developer's area. www.tfl.gov.uk/developers. Accessed: 14/01/2013.

[22] Transport for london journey planner. http://journeyplanner.tfl.gov.uk. Accessed: 14/01/2013.

[23] Tron-like map of bike journeys reveals london's hubs. http://www.newscientist.com/blogs/shortsharpscience/2012/09/map-of-bicycle-journeys-reveal.html.

[24] C. Barrett, R. Jacob, and M. Marathe. Formal language constrained path problems. Los Alamos, United States, 2000.

[25] C. M. Bishop. Pattern recognition and machine learning. New York, United States, 2006.

[26] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call centre: A queueing-science perspective. Philidelphia, United States, 2005.

[27] L. Brown and L. Zhao. A test for the poisson distribution. Pennsylvania, United States, 2005.

[28] K. W. Church and W. A. Gale. Poisson mixtures. Murray Hill, United States, 1995.

[29] G. Jongbloed and G. Koole. Managing uncertainty in call centres using poisson mixtures. Amsterdam, Netherlands, 2001.

[30] R. T. Kaleta. An integrated london journey planner. London, United Kingdom, 2012.

[31] T. Neiman and Y. Loewenstein. Reinforcement learning in professional basketball players, 2011.

[32] S. Shaheen, H. Zhang, and S. Guzman. Bikesharing in europe, the americas, and asia. Richmond, United States, 2009.

[33] S. Shaheen, H. Zhang, and E. Martin. Hangzhou public bicycle: Understanding early adoption and behavioral response to bikesharing in hangzhou, china. Richmond, United States, 2011.