UNIVERSITY OF CAMBRIDGE

Pulse of the World's Bicycles

Towards Data-Driven Methods for the Design and Administration of Bicycle Sharing Systems

Advait Sarkar

Emmanuel College

A dissertation submitted to the University of Cambridge in partial fulfilment of the degree of Master of Engineering
(Part III of the Computer Science Tripos)

June 13, 2013

Declaration of Originality

I, Advait Sarkar of Emmanuel College, being a candidate for Part III of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Word count: 11,980

Advait Sarkar
June 13, 2013

Abstract

Bike sharing systems are emerging all over the world as a cheap, green, and healthy mode of public transit. While there is great variety in their implementation, at their core they are all similar: they consist of bike stations, each containing a number of docking points, from which bicycles may be borrowed and to which they may be returned. These systems suffer from a number of open problems: designing effective bicycle redistribution schemes, optimising the placement of stations, and choosing station capacities, among others.

In this project, we use data from 10 such systems to develop and evaluate reusable techniques for their design and administration. Our study is the first to develop and apply uniform analytical methods across several disparate systems, enabling inferences at a previously unavailable level of generality.

In particular, we present methods for the unsupervised discovery of naturally-occurring groups of stations by three separate notions of behaviour. These methods facilitate the design of redistribution vehicle schedules, identification of candidate zones for station addition/removal, and selection of station capacities. We also investigate the prediction of station utilisation at fixed intervals in the future.

We demonstrate the utility of our techniques, which require only very simple, low-dimensional, and publicly-available data, by drawing general inferences about the behaviours of bike sharing systems. We discover evidence that relates the size of a system to its behavioural heterogeneity, and evidence to suggest that there is a significant amount of transferable knowledge between systems of similar sizes.

Acknowledgements

I would like to thank my acting supervisor, Dr Neal Lathia, for his enthusiasm, interest, and generosity with his invaluable expertise and guidance.

I would also like to thank my senior supervisor, Dr Cecilia Mascolo, for her constant support and encouragement throughout the course of this project.

Contents

Chapter 1

Introduction

A bicycle sharing system (or bikeshare) is a service that makes bicycles available for shared use to communities. The bicycles can be used for short journeys in urban areas as an alternative to private vehicles or motorised public transport. Communal benefits include reduced traffic congestion, noise and air pollution, and improved public health. Bikeshares have also been touted as a solution to the "last mile" problem, i.e., connecting users to public transit networks.

Barcelona (Bicing) Rio de Janeiro (BikeRio) London (Barclays)

Figure 1.1: Stations at bike sharing systems around the world.

These systems address many of the disadvantages of bicycle ownership, including entry price, susceptibility to theft and vandalism, parking and storage requirements, and maintenance. However, by restricting the locations where bicycles may be borrowed and returned, the service becomes a form of public transit, sometimes less convenient than a privately-owned bicycle.

There are several open problems suffered by existing systems that we need to better identify and understand in order to build bike sharing systems more effectively. For example, Transport for London has acknowledged that redistribution of bikes is a genuine problem [1], as the scale of the redistribution required was underestimated. A user satisfaction survey conducted by the London Assembly identifies a lack of bikes and docking points as a major issue [2].

The systems are difficult to model and analyse at a high level of generality because of large differences in the characteristics of individual cities. System designers and administrators would benefit [3] from general methods that enabled them to

- improve the efficiency of bicycle redistribution schemes, e.g. through better schedules and routes for effective redistribution at lower costs,
- reduce maintenance overhead per station, for instance by setting better station capacities so that unnecessary docking points are not built, and
- improve user satisfaction, for example by ensuring a high probability of
  - a station being within walking distance
  - said station having a bike available
  - destination stations having an empty space available

  by better placing their stations across the city.

Identification of common themes, if any, would help solve some of these problems in ways that were reusable and immediately applicable to new systems. Core to this activity is the performance of an *inter-city analysis*. In this project, we explore the utility of large-scale data mining techniques to uncover a better understanding of the dynamics of these systems.

1.1 Objectives & outline

The objective of this project is not to conclusively solve the problems outlined above. Rather, we aim to develop data-driven analytical frameworks that are applicable in the general context of bike sharing systems.

Concretely, this project comprises the following:

- In Chapter 2 we introduce the dataset and present some preliminary analysis. We first describe the nature of the data we gathered in §2.1. We then present our data cleaning and preprocessing pipeline in §2.3. Finally, we present an analysis of the aggregate daily behaviour of each of the bike sharing systems under investigation in §2.4. We discover intercontinental differences as well as stark differences in weekday and weekend usage.

- Our work in Chapter 3 is concerned with detecting naturally-occurring groups of stations that behave similarly to one another. We first describe our general approach to the unsupervised clustering problem in §3.1. We then describe three specific notions of station behaviour and discuss our results in §3.2, §3.3, and §3.4 respectively. We show that our clustering methodologies can be used to inform redistribution vehicle routes, improve station location positioning, and choose station capacity ranges.

- In Chapter 4 we attempt to predict the number of bicycles at a station at fixed intervals in the future. We formulate the problem and test strategy in §4.1, and present our evaluation metrics in §4.2. We describe our predictive models in §4.3 and the nature of our training data in §4.4. We discuss their performance and the implications of the results in §4.5 and §4.6.

- Finally, we present our conclusions in Chapter 5. We summarise our key analytical results and contributions, and explain the immediate applicability of this work. We discuss some limitations of our study and describe avenues for future work that are now open.

1.2 Previous work

The computational analysis of bike sharing systems is a somewhat niche area of research. This section presents an overview of the relevant work in the area.

Froehlich, Neumann and Oliver applied clustering techniques to identify patterns of behaviour in stations in Barcelona's "Bicing" system, explaining their relation to location and time of day [4]. They also built and evaluated a number of predictive models for the stations of the system.

Lathia, Ahmed and Capra used clustering techniques similar to those of Froehlich et al.'s to conduct a study on London's Barclays bike sharing scheme [5]. They leveraged the same analytical framework to assess the impact of opening up the scheme to casual users, and were able to quantify notable differences in the way the system was used prior to and after the policy change.

Guenther et al. built and validated a number of arrival forecasting models based on journey data from the Barclays Cycle Hire scheme in London [6]. The work was concerned with forecasting the cumulative arrivals in small geographic clusters of stations (falling within 500m×500m squares) during peak hours. They were able to show that their two models, based on time-inhomogeneous population CTMCs[1] and multiple linear regression with ARIMA[2] error, were able to marginally outperform previous models which did not use journey data.

Randriamanamihaga et al. used clustering techniques that exploit Poisson mixture models to find temporal clusters in the origin-destination flows of bicycles in Paris' Vélib system [7]. They showed that people's journeys could be broadly classified into "weekend joyrides", "nightlife", "morning work" and "early bird".

Similarly, Shu et al. also developed a network flow model to examine the effectiveness of various schemes for bike deployment and redistribution in Singapore [8]. They modelled origin-destination demand for 3 bicycle stations based on the corresponding flows in Singapore's mass transit land trains.

[1] Continuous Time Markov Chains

[2] AutoRegressive Integrated Moving Average

Lin and Chou explored the bicycle redistribution problem as a special case of the more general vehicle routing problem [9]. They prototyped a system which leveraged publicly-available road data to make use of actual path distance to solve the vehicle routing problem, rather than use simple Euclidean distance, which had previously been the norm. They evaluated their prototype on the Kaohsiung C-Bike and Washington Capital Bikeshare systems, and were able to generate vehicle routes with marginally lower travel distance and time compared to the routes being used otherwise.

Each of the studies mentioned in this section only look at a single system, some at even smaller subsets, and so lack the richness in data required to make higher-level generalisations. System simulations, as in Shu et al., 2011, usually make a number of simplifying assumptions (e.g. that there are only 3 stations in the system), and as a consequence the results are difficult to deploy industrially. Previous research on origin-destination flows is not reproducible across several cities because that data is not always publicly available.

Consequently, the studies conducted by Lathia et al. and Froehlich et al. form the two core texts upon which our study is built and upon which we now expand. Our analytical framework, which neither considers network flows, nor attempts to simulate the interaction between stations, requires only very simple data that is publicly available for many bike sharing systems.

Chapter 2

Preliminary Analysis

Before we began our in-depth analysis of bike sharing systems, it was first necessary to gather, preprocess, and conduct preliminary analysis of the data. In this chapter we first specify the nature of our data (§2.1), then describe our preprocessing and cleaning pipeline (§2.3), and finally conduct a preliminary discussion of the usage patterns of the bike sharing systems on an aggregate level (§2.4).

2.1 Data gathering

A bike sharing system contains a number of stations. Each of these stations hosts a number of docks/slots where a bicycle may be parked. These systems typically provide a web service where a user can check how many bicycles and vacant slots are currently available at any given station in the system. From January 1st, 2011 onwards, we sampled these numbers every two minutes by scraping the web services of the 10 bike sharing systems in Table 2.1.

The web services also provide an interactive map displaying the locations of the stations, from which we gathered latitude and longitude coordinates for each station at every two-minute interval. Since these coordinates are generally static over time, new coordinates for a given station were appended to the record only if they did not match the previously recorded coordinates for that station. For the majority of bike stations we had a single recorded location, but for some there were multiple locations. In §2.3 we offer some possible reasons for this, and explain how we resolved multiple locations.

City	**System**	**Country**
Joao Pessoa	SAMBA	Brazil
Siracusa	GoBike	Italy
Taipei	YouBike	Taiwan
Girona	GiroCleta	Spain
Rio de Janeiro	BikeRio	Brazil
Rome	Roma‘n’Bike	Italy
Miami	DecoBike	U.S.A.
Denver	Denver B-Cycle	U.S.A.
London	Barclays Cycle Hire	England
Barcelona	Bicing	Spain

Table 2.1: Bike sharing systems from which data was gathered.

We gather no other data. It is important to note that we sample only the numbers of bicycles and vacant docking points at each station, but *not* the origin and destination of bicycles, because that is not available. As data about the trajectory of individual journeys is not made available on these web services, we are unable to reason about the bicycle sharing systems as networks. However, as we found, even focusing on the number of bikes and vacancies at each station as univariate time series yields considerable insight.

2.2 Terminology

The following terminology is used throughout the report.

- **Observation**: a sample of the number of bikes and vacancies at a single station at a certain timestamp. An observation is generally a tuple of the form (b,v). When observations are part of a series, individual observations are indexed using subscripts; the i^{th} observation is referred to as (b_i,v_i).
- **Day**: a period of 24 hours starting at 00:00:00 and ending at 23:59:59.
- **Capacity**: the number of docks (thus, the maximum number of bicycles it is possible to deposit) at a particular station.

- **Occupancy**: the fraction of the station's total docking points occupied by bicycles at a given point in time; the "fullness" of a station. For example, if a station has a capacity of 20, and there are 5 bikes at the station, the station has an occupancy of 0.25. A station with occupancy=1.0 is completely full and a station with occupancy=0.0 is completely empty.

  We use occupancy to describe the instantaneous state of stations throughout as it is capacity-independent. It allows us to meaningfully compare stations with different capacities. Across our 10 systems, there are 996 stations in total, which range from small (capacity $\approx$ 10) to quite large (capacity $\approx$ 50).

2.3 Data preprocessing

The data was not in a readily usable format and suffered from a few types of noise. We now describe our preprocessing and cleaning pipeline, which is loosely based upon the procedures used by Froehlich et al. 2009, and Lathia et al. 2012.

We attempted to sample the number of bicycles and vacancies at each bicycle station every two minutes starting from January 1st, 2011. However, the periodic downtime of the server running the web crawler and unexpected modifications to the web services caused gaps in the dataset. Consequently, we restricted the period under investigation to lie between the 23rd of March, 2011 and the 6th of August, 2011, inclusive. This period spanning approximately 4.5 was the largest contiguous region of uncorrupted data common to all cities.

The timestamps recorded for each observation were in UTC. To relate our temporal analysis to local contexts, the timestamps were converted into local times, taking into consideration regional changes due to daylight savings-like schemes.

2.3.1 Inferring station capacity & filtering observations

While the capacity of a station is not directly reported by the web services we scraped, it is possible to infer a station's capacity by adding together the number of available bikes and the number of available vacancies at any given time. One might expect that this sum, this "inferred" capacity, would remain constant over time, given that an increase in the number of available bicyles directly results in an equal decrease in available vacancies, and vice versa. In practice, however, this number is occasionally off-by-one or negative, a phenomenon likely attributable to malfunctioning bicycle/dock sensors.

In Lathia et al. 2012, these errors were accounted for by picking the 95th percentile of observed station capacities for each station and setting that value as the assumed capacity. This fixes a single capacity across the entire period being studied. However, this method does not allow for genuine changes in station capacity, such as those caused by the addition or removal of docks, which is a frequent occurrence in systems with active maintenance schemes.

We allowed for genuine fluctuations in capacity by calculating a capacity for every observation. That is, for an observed tuple (b,v), the capacity *at the time of observation* is assumed to be $b + v$. To account for erroneously reported capacities, we removed observations where the calculated capacity appeared fewer than 720 times across the entire dataset. At two-minute intervals, we make 720 observations per station per day. Therefore, any calculated station size must be reported for at least a day's worth of observations in order to be considered a genuine reflection of the station size.

A high number of invalid observations in a single day signals anomalous station behaviour and potential problems with the remaining observations of the day. In consistency with previous work, we removed all days with fewer than 504 (70% of 720) observations. Finally, we removed all stations with fewer than 62 remaining days, i.e. 45% of our total 4.5 month period.

2.3.2 Resolving station locations

Recall that in addition to the timestamped observations, we also recorded the geographic coordinates of each station as reported by the web service. These are not static, however, as bike stations are often renamed or moved. Occasionally the location data was also corrupt, perhaps due to malfunctioning GPS sensors or due to errors in the transport authority databases, causing coordinates that were reported as zero or were out of the allowed range of values.

As we recorded all unique locations reported for each station, we had multiple observed locations for many stations. As a station's behaviour is highly dependent on its location, data from a station that has been genuinely moved (i.e., that has been dismantled and reconstructed elsewhere, e.g. in an adjacent street) cannot be trusted, as the station is now effectively an entirely new station, despite being referred to using the same station ID by the bike sharing system administrators.

We detected genuine location changes as follows: the pairwise ground distances between all locations recorded for a single station were calculated using the Haversine formula with the spherical-earth assumption [10]:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{2.1}$$

where d is the distance between two points whose latitude/longitude coordinates are given by (ϕ_1, λ_1) and (ϕ_2, λ_2) respectively, and r is the radius of the earth.

If n locations $l_1, ..., l_n$ were observed for a station, we calculated the distance between each pair of locations l_i and l_j, $i \neq j$.[1] If any of these distances was larger than 10m, the station was deleted altogether. If all distances were less than 10m, the most recently reported location was assumed the truth.

The 10m threshold was chosen on the basis that a change in the reported position of a station by less than 10m is not likely to reflect a change in the actual location of the station, as most stations are themselves larger than 10m in size, and the accuracy of civilian GPS positioning at the time of writing is ~10m [11].

[1] For instance, if a station was reported to be at locations a, b, and c, the ground distances between a and b, b and c, and a and c were calculated.

Miami's reported locations were particularly noisy, and this step caused the removal of nearly 50% of its observation data. A more rigorous methodology would optimise for the fraction of stations lost by varying this threshold. This was beyond the scope of the current investigation, but is suitable for further study.

The data loss as a consequence of our cleaning is presented in Table 2.2.

City	Observns., Pre-Clean	Stations, Pre-Clean	Observns., Post-Clean	Stations, Post-Clean	Observations Retained (%)
Joao Pessoa	389260	4	381760	4	98.07
Girona	956510	10	945154	10	98.81
Taipei	1081630	11	975540	10	90.19
Siracusa	1750500	18	1740400	18	99.42
Rio de Janeiro	2132812	22	2091210	22	98.05
Rome	2941530	30	2924098	30	99.41
Denver	4907697	52	4787219	50	97.55
Miami	8158542	99	4589939	53	56.26
London	37954996	410	35414070	390	93.31
Barcelona	38674016	415	37087351	409	95.90

Table 2.2: Data loss due to cleaning.

2.4 Systemic occupancy series analysis

Our dataset is low-dimensional, but voluminous. With 720 daily observations for ∼1000 stations over ∼150 days, we have over 10^8 data points. We first analysed this data at an aggregate level showing how daily usage varies across cities.

Recall that we define occupancy to be the fraction of the station's total slots currently occupied by bicycles. We divided each 24-hour day into 240 6-minute bins; 00:00–00:06, 00:06–00:12, ..., 23:54–00:00. We then averaged the observations within each bin on a per-station basis.

The averaged observations are tuples of the form $(b_i,\ v_i)$, $1 \leq i \leq 240$, where b_i and v_i are the average number of bikes and vacancies in the i^{th} bin respectively. These tuples are then expressed as occupancies: $o_i = \frac{b_i}{b_i+v_i}$. This occupancy series was then averaged across all stations in a system to create two series representing the system, one with weekdays only, and the other with weekends only.

We separated weekdays and weekends because Froehlich et al., 2009 and Lathia et al., 2012 both reported a clear difference in usage patterns between weekdays and weekends. Our data confirms that this is true in general for bike sharing systems, and not just for the specific system studied in those cases. There is variation within the weekdays themselves, but it is small, and studying separate series for each day does not yield significantly greater analytical value over simply separating weekdays and weekends. This is a subject suitable for further research.

Consequently, for each system, we are left with two 240-point occupancy series. These represent a picture of the system's stations' average occupancy in each 6-minute bin over the course of a 24-hour day. We now present graphs of these series. On the x-axis is the hour of day, and on the y-axis is the average occupancy of the city's stations. The red line depicts the series for weekends, and the blue line depicts weekdays. We present the systems in increasing order of size.

It is important to bear in mind that occupancy, or "fullness of stations", is to be interpreted as the *inverse* of usage. Occupancy is low when few bikes remain in the stations, indicating that the bikes are being highly used. Conversely, occupancy is high when many bikes are idle in stations, indicating that the bikes are not being used as much. A negative slope in the occupancy series corresponds to increasing usage, and a positive slope corresponds to decreasing usage.

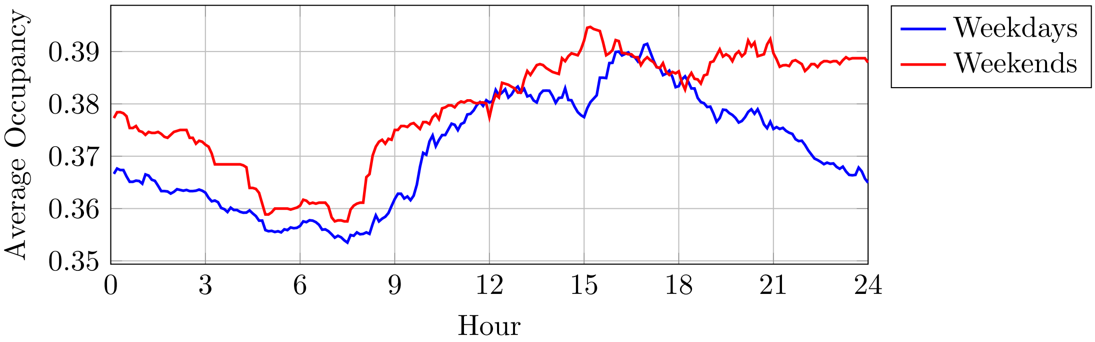


Figure 2.1: Joao Pessoa occupancy series.

We begin with Joao Pessoa (Fig. 2.1), capital of the Brazilian state of Paraiba, and our smallest system with only 4 stations. In both series there is a drop in occupancy (increase in usage) at around 7:30AM. Occupancies gradually increase until a peak in the afternoon, which occurs at 5:00PM on weekdays and 3:00PM on weekends, after which occupancies start declining again. The difference in weekend and weekday behaviour clearly corresponds to typical hours of work.

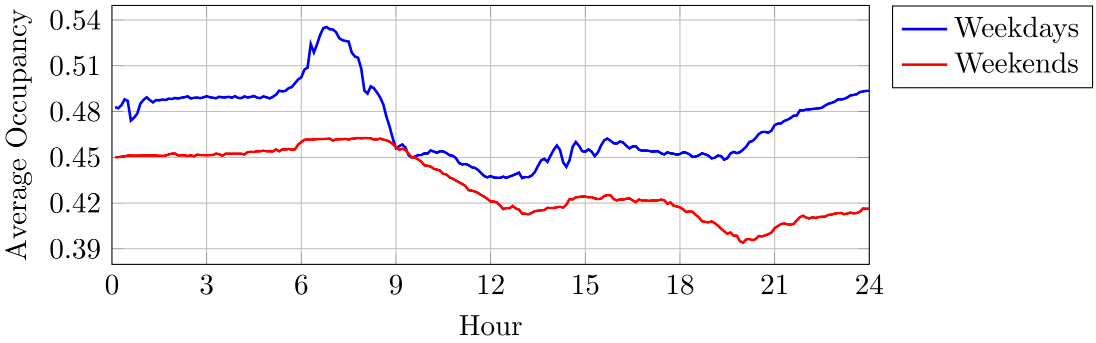


Figure 2.2: Girona occupancy series.

The second of our studied systems is Girona, a city in the northeast of Catalonia, Spain (Fig. 2.2), with 10 stations. The system is more heavily used on weekends than it is on weekdays. Usage is particularly low during weekday mornings, and particularly high on weekend evenings.

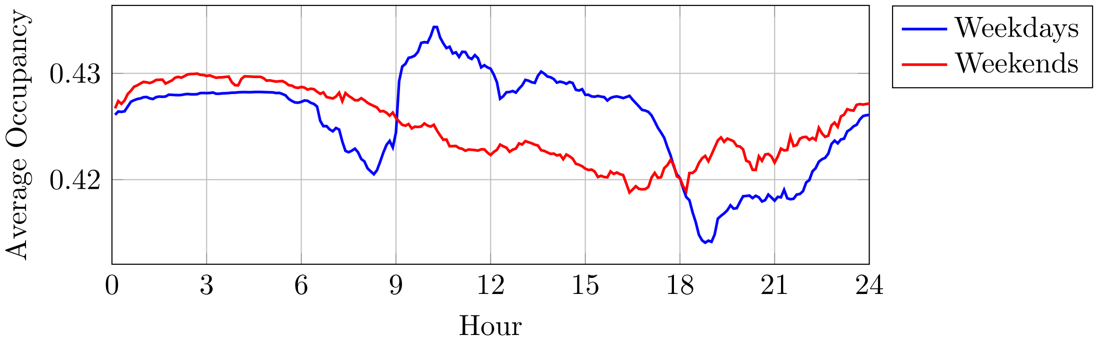


Figure 2.3: Taipei occupancy series.

The third is Taipei, capital of Taiwan (Fig. 2.3), also with 10 stations. The sharp usage spikes around 8:00AM and 7:00PM on weekdays reflect the work culture of Taiwan; these spikes are completely absent on weekends.

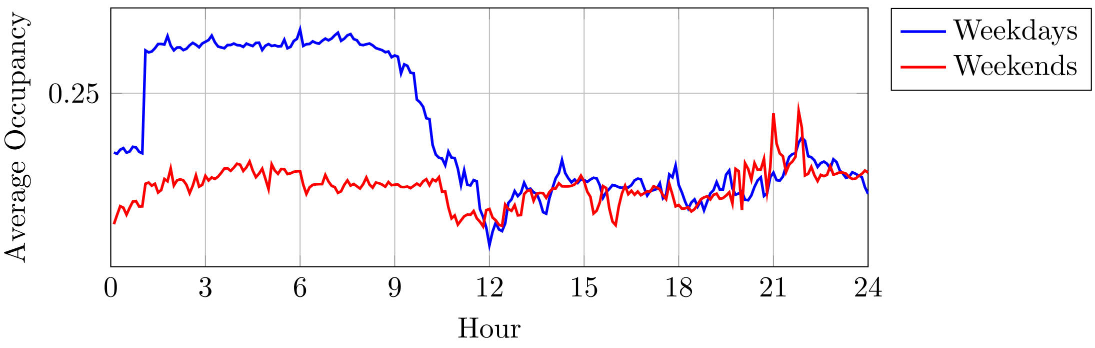


Figure 2.4: Siracusa occupancy series.

The fourth is Siracusa (Syracuse, Italy) (Fig. 2.4), with 18 stations. Morning usage us much higher on weekends than on weekdays. The odd jump in the weekday series at 1:00AM is most likely attributable to a redistribution scheme.

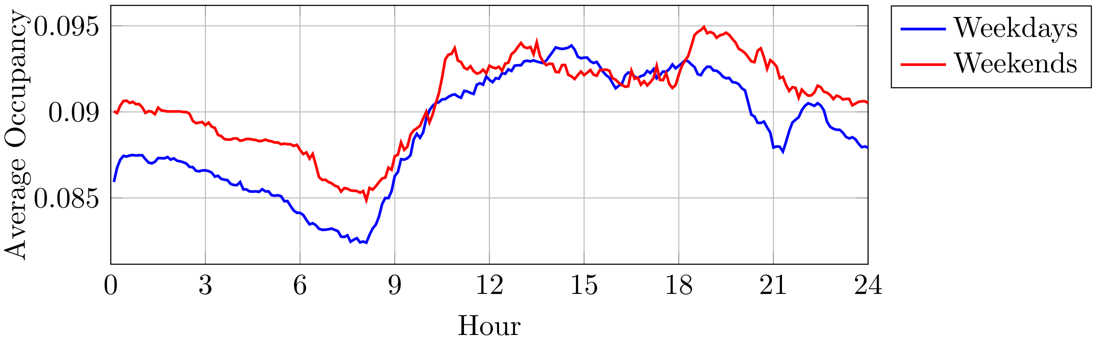


Figure 2.5: Rio de Janeiro occupancy series.

The fifth is Rio de Janeiro, Brazil (Fig. 2.5), with 22 stations. While we do observe hours-of-work related "peaks" in usage, it is important to note that the scale of the y-axis is very small in comparison to the other systems, making the lines essentially flat. The weekday and weekend patterns are almost identical. Rio's stations appear to spend most of their time at near-empty states. We confirm this by analysing the stations' occupancy distributions in §3.4.

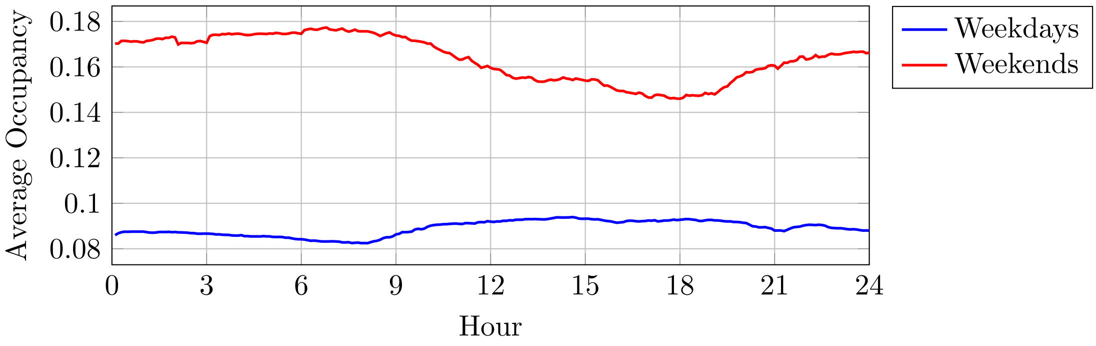


Figure 2.6: Rome occupancy series.

The sixth is Rome, capital of Italy, (Fig. 2.6), with 30 stations. The usage patterns are unusual and distinctively flat, with higher usage on weekdays.

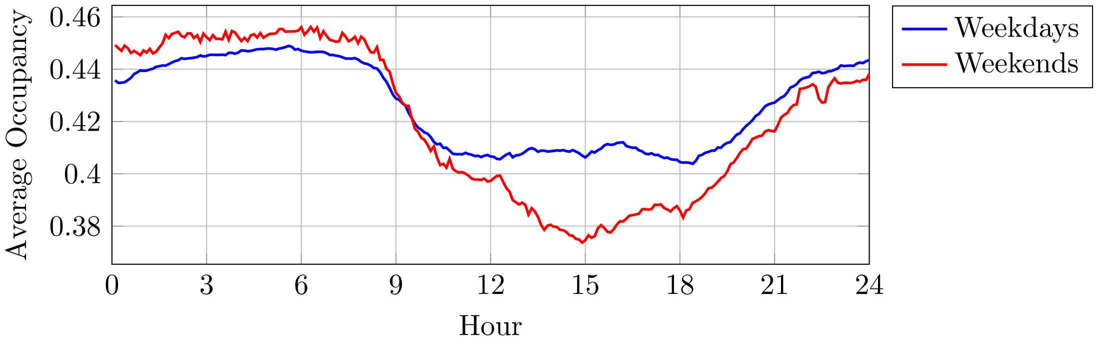


Figure 2.7: Miami occupancy series.

The seventh is Miami (Florida, U.S.A.) (Fig. 2.7), with 50 stations. The two series are more or less coincident except for a higher weekend afternoon usage.

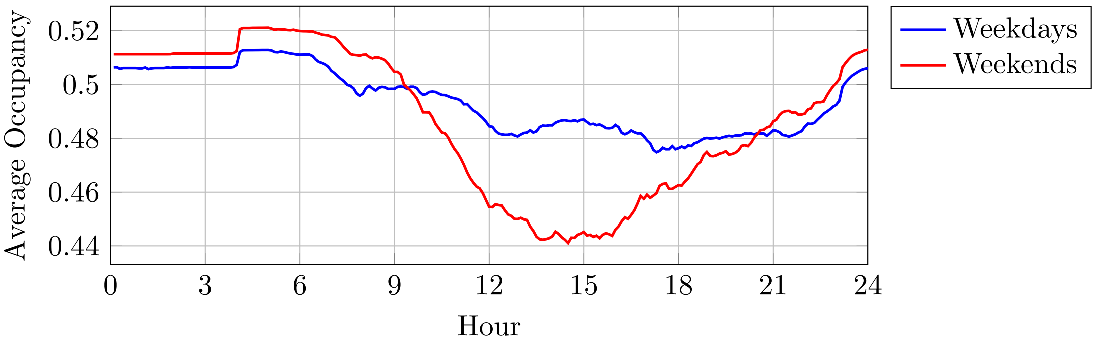


Figure 2.8: Denver occupancy series.

The eighth is Denver (Colorado, U.S.A) (Fig. 2.8), with 53 stations. Like Miami, the weekday and weekend series are nearly coincident, with the exception of a high 3:00pm peak in usage on weekends. Neither Miami nor Denver exhibit any increase in usage in the mornings, as many other systems do, indicating that the users of the American systems are not using the bikes to commute between their work and home environments.

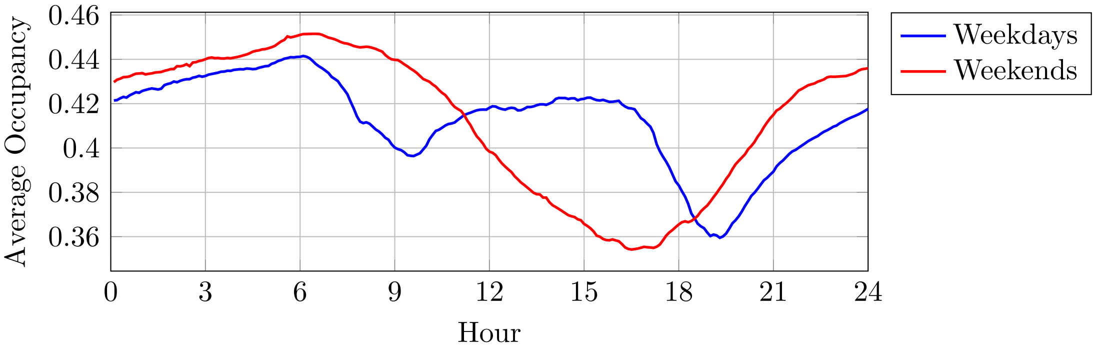


Figure 2.9: London occupancy series.

The ninth is London, capital of the United Kingdom (Fig. 2.9), with 390 stations. London exhibits a very clear difference in weekend and weekday usage. On weekdays, system usage is driven by the hours of work, causing spikes at around 9:00AM and 7:00PM. On weekends, usage is more casual, with a gentle increase in usage and a single peaking at around 5:00PM.

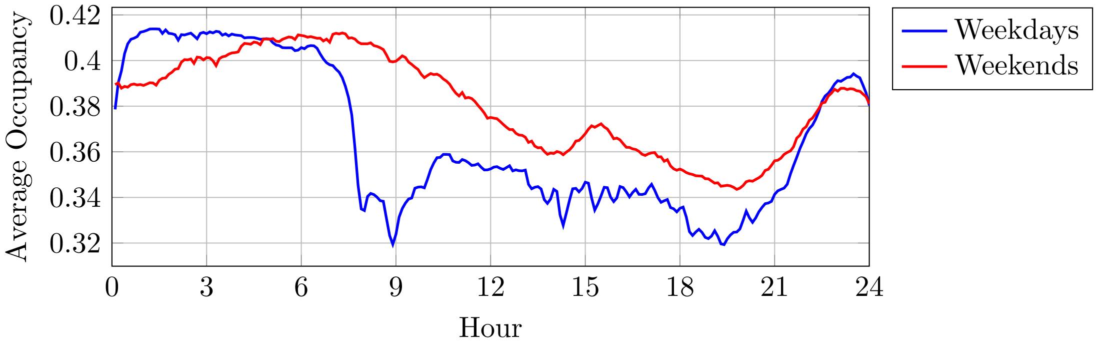


Figure 2.10: Barcelona occupancy series.

The tenth and final city is Barcelona, Spain (Fig. 2.10), with 409 stations. Barcelona's usage patterns also differ starkly between weekdays and weekends. On weekdays, we observe workday-driven peaks at 9:00AM and 7:00PM, and usage remains high throughout the workday. The system is less used on weekends, but a small interruption to the otherwise steady upwards trend in usage is observed at around 4:00PM, corresponding to the siesta culture of Spain.

General observations

We observe, with the exception of Rio de Janeiro, a clear dissimilarity between each system's weekend series and its weekday series. This observation the basis for our subsequent decision to keep them separate, and to focus on weekdays alone for our clustering and time series forecasting studies in Chapters 3 and 4.

Peaks in weekday usage corresponding to the start and end of typical hours of work were observed in the large European and Asian systems, clearly demonstrated in Taipei, London and Barcelona. The smaller systems (and Miami and Denver) did not demonstrate this, which suggests that casual use makes up the majority of their use cases.

Chapter 3

Identifying Naturally-Occurring Station Clusters

Given unlimited resources, each station would be given individual care by the administrative authority. Its usage would be analysed on a daily basis, in the context of street traffic, footfall, and nearby attractions and transport hubs. Custom decisions regarding its expansion, contraction, relocation, and status in the bicycle redistribution schemes are made on a per-station basis.

In practice, however, it is infeasible to devote this level of attention to individual stations, especially in larger systems consisting of hundreds of stations. Moreover, a highly station-specific administration methodology has little value to designers of new systems and generates little transferable administrative experience.

It would be more useful think of stations as belonging to behavioural categories. In this more realistic scenario, decisions may not be tailored to specific stations. However, if some stations really did behave much like each other, making decisions based on the activity of the group as a whole could achieve similar effects for a fraction of the administrative effort.

In this chapter, we investigate whether stations were organised into general groups according to their behaviour. We defined three notions of "behaviour" and developed unsupervised clustering methods for each:

1. **Daily flux in station occupancy**: the rise and fall in a station's occupancy over the course of a day.
2. **Station activity level**: the approximate number of daily borrow/return events at a station, expressed as a fraction of its capacity.
3. **Occupancy distribution**: the probability distribution of a station's observed occupancies. That is, the percentage of the time a particular station spends at a various levels of fullness.

Since we found that station usage patterns are markedly different on weekdays and weekends, we restricted the scope of this investigation to weekdays only.

We clustered all stations across the global dataset, not within each system separately. We discovered naturally-occurring behavioural classes for stations across all systems under investigation. In this chapter, we reveal that certain behavioural classes are system-independent, and that there is, in fact, significant transferable knowledge between stations and between systems. We first describe our general approach to the unsupervised clustering problem in §3.1. We then describe our specific methods and discuss our analytical results for each of the three notions of behaviour in §3.2, §3.3, and §3.4 respectively.

3.1 Clustering methodology

We employed hierarchical clustering, also known as dendrogram clustering [12], using an agglomerative strategy. In this bottom-up approach, each bicycle station (or rather, its vector representation) is initialised as a singleton cluster. In each iteration of the algorithm, the distance between every pair of clusters is computed, and the two clusters which have the smallest distance (highest similarity) between them are merged into a single cluster containing the stations from both. Thus in each iteration the total number of clusters decreases by one. This procedure can be repeated until only a single cluster remains. Our method for deciding when to stop iterating (i.e. determining the number of clusters) is described in §3.1.1.

The vector representation of the bicycle stations was different for each of our notions of station behaviour. For example, we used a 240-point averaged mean-normalised occupancy series as the vector representation of a station's "daily flux in station activity", whereas we used a 10-bin discrete probability distribution as the vector representation of a station's "occupancy distribution". Details of how we produced these representations follows in §3.2, §3.3, and §3.4. Consequently, for each of our three notions of station behaviour, we developed a separate metric for computing the distance between any two station vectors.

This "inter-station" distance metric alone does not suffice for computing the distance between any two clusters of stations. We also require a criterion for determining the distance between entire clusters, i.e. an "inter-cluster" distance metric. This is usually referred to as the *linkage criterion.* For instance, the linkage criterion might be that the maximum pairwise distance between any two stations belonging to either cluster is the distance between the two clusters[1].

For our linkage criterion, we used *centroid linkage*, also known as unweighted pair-group method using centroids (UPGMC) [13]. In this scheme, the vectors for all stations within each cluster are merged to produce a "centroid". This centroid vector represents a station possessing the "average" behaviour of that cluster of stations. Then, the distance between any two clusters is the station-pairwise distance between the centroids of the two clusters.

For each of our three notions of station behaviour, we therefore specify both a inter-station distance metric as well as a method for generating the centroid of a cluster of stations. We provide these specifications in detail as we encounter each notion of behaviour §3.2, §3.3, and §3.4 respectively.

3.1.1 Determining k, the number of clusters

The hierarchical clustering algorithm is not intrinsically capable of determining when to stop clustering. Rather, the choice of where to stop[2], and consequently the number of resultant clusters (popularly denoted k), is left to us. The choice of k is often guided by intuition, based on what level of clustering yields the most valuable analytical insight. Automating the choice of an optimal k is the subject

[1] Also known as "maximum" or "complete-linkage" clustering.

[2] Also referred to as "cutting the dendrogram", as the hierarchical clustering algorithm creates a tree, or dendrogram, of possible clusterings, and stopping the iteration "cuts" the tree.

of much research, and several complex heuristics have been proposed, such as the weighted gap statistic [14].

We use a simple heuristic based on incremental merge distances that approximates the L-method proposed by Salvador and Chan [15]. When initialised with n objects, the algorithm begins with n separate singleton clusters and ends with a single cluster containing n items, with k decreasing by one at each iteration. Thus there are $n - 1$ iterations, and at the end of the i^{th} iteration, $k = n - i$.

Recall that in the i^{th} iteration, the distance d_i between the two clusters which are selected to be merged is the *minimum* pairwise distance between any two cluster centroids in that iteration. We record d_i for $1 \leq i \leq n - 1$. This series is differenced to yield a series $\Delta_i = d_i - d_{i-1}$ for $2 \leq i \leq n - 1$. The series Δ_i is the incremental merge distance: how much *further* apart the two closest clusters in the i^{th} iteration are than the two closest clusters in the previous iteration.

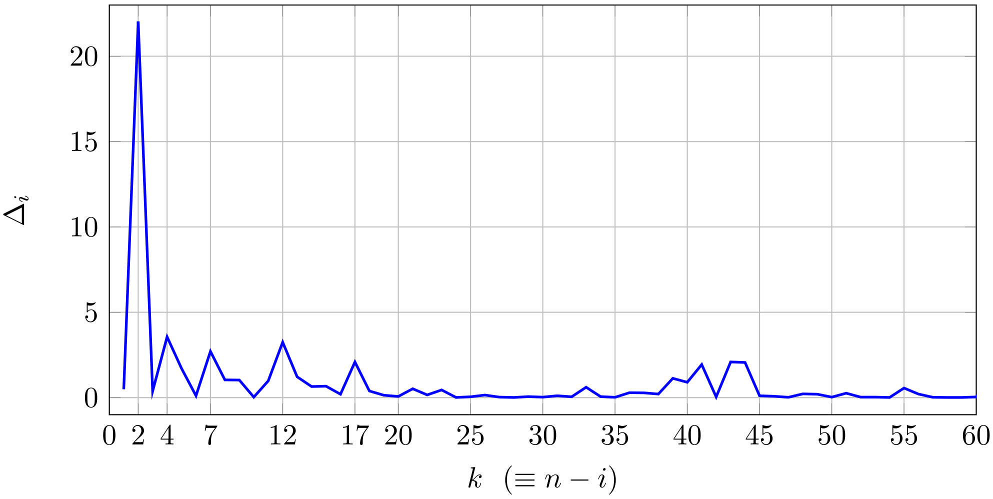


Figure 3.1: Incremental merge distances for final 60 iterations of clustering.

Consider Figure 3.1, which depicts Δ_i as a function of k for the final 60 iterations of one of our clustering studies. The spikes at k =2, 4, 7, 12 and 17 indicate unusual jumps in merge distances, suggesting that clusters which perhaps ought to stay separate are being lost. Specifically, Figure 3.1 suggests that some candidates for k are: 3, 5, 8, 13 and 18. We were able to decide on an optimal k for all our clustering cases using a combination of the Δ_i heuristic and manual tuning.

3.2 Daily flux in station occupancy

Our first notion of station behaviour relates to its occupancy pattern over the course of an average day. If we are able to detect naturally-occurring classes of this type of behaviour, the geospatial layout of the station types would be immediately useful for the planning of routes for bicycle redistribution schemes.

For instance, consider the situation where the entire East half of a system consistently loses bikes throughout the morning, and the West half gains bikes at that time, and in the evening this behaviour is reversed. Given this analysis, it is simple to conclude that redistribution vehicles should travel in predominantly Eastbound routes in the morning, and Westbound routes in the evening.

In this section we describe in detail a robust methodology for carrying out such analysis. We present the global behavioural clusters that emerge in our dataset.

3.2.1 Creating a station's vector representation

Recall that occupancy is defined as the fraction of a station's total slots currently occupied by bicycles. As with the preliminary analysis, we created a 240-point occupancy series for each station by dividing each 24-hour day into 240 6-minute bins and averaging observations within each bin. The averaged observations, tuples of the form (b_i, v_i), are then expressed as occupancies; $o_i = \frac{b_i}{b_i+v_i}$.

In the preliminary analysis of each system's global properties, we then aggregated this occupancy series by averaging it across all stations in a system. We do not do that here, as we want to retain a unique vector representing each station.

In a departure from the work of Lathia et al., which proceeded immediately to clustering on these series, we then normalised the series by subtracting its mean from each element; $o'_i = o_i - \frac{1}{240}\sum_{i=1}^{240} o_i$. We perform mean-normalisation because we want to cluster together stations with the same rise and fall in occupancy; those which have bicycles borrowed and returned at similar times of day, regardless of what mean that rise and fall rotates around.

To elaborate, consider the three stations in Table 3.1 with the following occupancy series, truncated so that only 5 points in the time series are shown. The occupancies are *not* mean-normalised.

Station	**Bin 1**	**Bin 2**	**Bin 3**	**Bin 4**	**Bin 5**
A	0.5	0.6	0.7	0.6	0.5
B	0.6	0.6	0.6	0.6	0.6
C	0.1	0.2	0.3	0.2	0.1

Table 3.1: Example truncated occupancy series for three stations.

A simple distance metric might compute the distance between two stations as the sum of the absolute differences between the occupancy values of individual bins. For instance, the distance between A and B would be $|0.5 - 0.6| + |0.6 - 0.6| + |0.7 - 0.6| + |0.6 - 0.6| + |0.5 - 0.6| = 0.3$. Similarly, the distance between A and C by this metric is 2.0. The clustering algorithm would therefore determine that A and B are much more similar to each other than A is to C, since the distance between them is much smaller.

However, closer inspection reveals that A and C are in fact more similar to each other in terms of their usage patterns. While B remains static, A and C exhibit occupancy rises of equal magnitude, peaking at bin 3, and then exhibit falls in occupancy of equal magnitude. Mean-normalisation makes this similarity transparent to our simple distance metric. The mean-normalised occupancy series for the same stations are presented in Table 3.2.

Station	**Bin 1**	**Bin 2**	**Bin 3**	**Bin 4**	**Bin 5**
A	-0.08	0.02	0.12	0.02	0.58
B	0.00	0.00	0.00	0.00	0.00
C	-0.08	0.02	0.12	0.02	0.58

Table 3.2: Example truncated mean-normalised occupancy series.

After mean-normalisation, it is much clearer that in fact A and C are identical. Their new distance as given by our simple distance metric is 0. The clustering algorithm will now prefer to group them together, which is the result we desired. While the distance metric we actually use (subsequently described in greater detail) is slightly more complex than the simple one used to illustrate our example, it is still subject to the same benefits of mean-normalisation.

Consequently, for each station, we are left with a 240-point series representing the station's average occupancy (with respect to its mean occupancy) in each 6-minute bin over the course of a 24-hour day. Since we are more concerned with the *changes* in levels of occupancy rather than the absolute values of occupancy, we refer to this as the occupancy *flux* series.

To calculate the centroid series c of a cluster of stations S, we simply average the series of all stations in the cluster:

$$c_i = \frac{\sum\limits_{s \in S} s_i}{|S|} \qquad 1 \leq i \leq 240 \tag{3.1}$$

In practice, whenever two clusters were merged, instead of recalculating this average over all stations in the new cluster, we calculated the centroid of the resultant cluster by performing a weighted average of the two constituent cluster centroids. The resultant centroid is numerically identical to the one produced using the formula above.

3.2.2 Dynamic time warping: a flexible distance metric

In §3.2.1 we described a simple distance metric for two occupancy flux series, namely $distance(s, s') = \sum_i |s_i - s'_i|$. In its current form, this does not suffice for our purposes. This is because slight temporal distortions in the series are unduly penalised. Consider the case where one series is identical to another, lagged by one step. That is, $s'_i = s_{i-1}$ for $i > 1$. Our simple distance metric penalises this difference heavily because it only compares corresponding elements in both series.

To account for this, we used a distance metric based on the dynamic time warping (DTW) algorithm [16]. This is a well-known technique for finding the optimal alignment of two temporal sequences. It allows for the insertion of arbitrary gaps in either of the two sequences to minimise the distance between them.

Since we wanted to allow for slight temporal distortions but not arbitrarily large ones, we constrained the algorithm using a 1-hour Sakoe-Chiba band [17] in a manner consistent with Froehlich et al., 2009. Therefore our final DTW implementation allows segments of the series to fall out of synchronisation by as much as one hour before incurring a heavy distance penalty. The dynamic time warping algorithm also uses an atomic distance metric[3] to compute the distance between any two occupancies. For this we used absolute difference: $cost(o, o') = |o - o'|$.

3.2.3 Naturally-occurring occupancy flux clusters

Using this distance metric, the hierarchical clustering methodology described in §3.1, and the heuristic for choosing the number of clusters described in §3.1.1, we produced 4 clusters (Fig. 3.2) from stations across all systems.

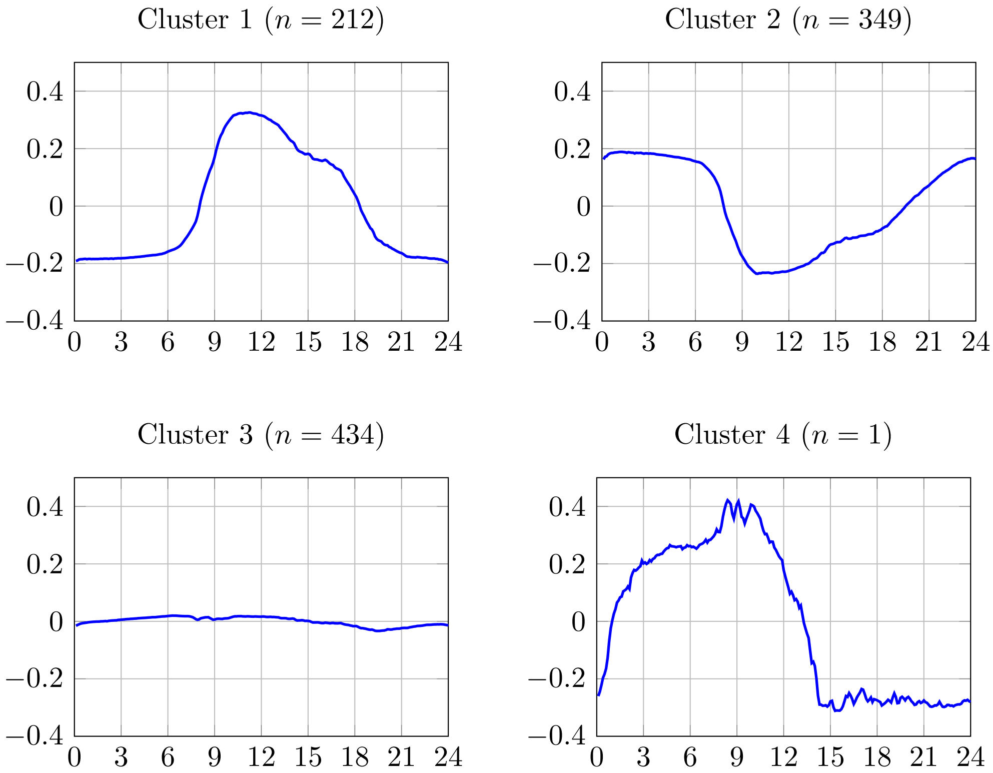


Figure 3.2: Cross-city mean-normalised occupancy cluster centroids at $k = 4$. On the x-axis: hour of day. On the y-axis: difference in occupancy from the mean.

[3] known as the cost function

There are 3 major clusters and one minor cluster. Of the 3 major clusters, the first, consisting of 212 stations, exhibits a sharp rise in mean-normalised occupancy starting at approximately 8:00AM. Occupancy peaks at around 11:00AM, then starts to decline until around 3:00PM, when the decline slows temporarily. The decline resumes at around 5:30PM and continues until approximately 9:00PM. Stations in this cluster start the day relatively empty, then are rapidly filled up in the morning, and slowly re-emptied over the course of the day. These stations can be considered “morning sink, daytime source” stations, as they act as bicycle sinks in the morning and as bicycle reservoirs during the day.

The second major cluster consists of 349 stations, and is an almost perfect inverse of the first cluster. Stations in this cluster start the day relatively full, then are rapidly emptied in the morning, and slowly re-filled over the course of the day. These stations can be considered “morning source, daytime sink” stations, as they act as bicycle reservoirs in the morning, and are a sink for bicycles over the course of the day.

The final major cluster consists of 434 stations. Stations in this cluster do not vary significantly in their levels of occupancy over the course of the day. This cluster does exhibit a slight increase in occupancy in the morning and a decrease in the late evening, but when compared to the magnitude of fluctuation in the other clusters, the mean-normalised occupancy of this cluster is essentially a flat line. These stations act neither as reservoirs nor sinks, but rather act equally as both. It is important to note that from a flat mean-normalised occupancy series, one cannot conclude that these stations are not very active, although that is one possible explanation. At most, one can conclude that the stations have roughly equal rates of bicycle inflow and outflow. Clustering on activity level requires a different treatment altogether, which we explore in §3.3.

We label the fourth cluster “minor” as it only contains a single station. The station in question lies in the heart of Barcelona and has a highly distinctive occupancy series. Judging by the fact that its occupancy starts climbing after midnight and peaks at 9:00AM, after which its occupancy rapidly drops, it is either close to a number of night-time attractions or it is being used as a depot for the redistribution scheme.

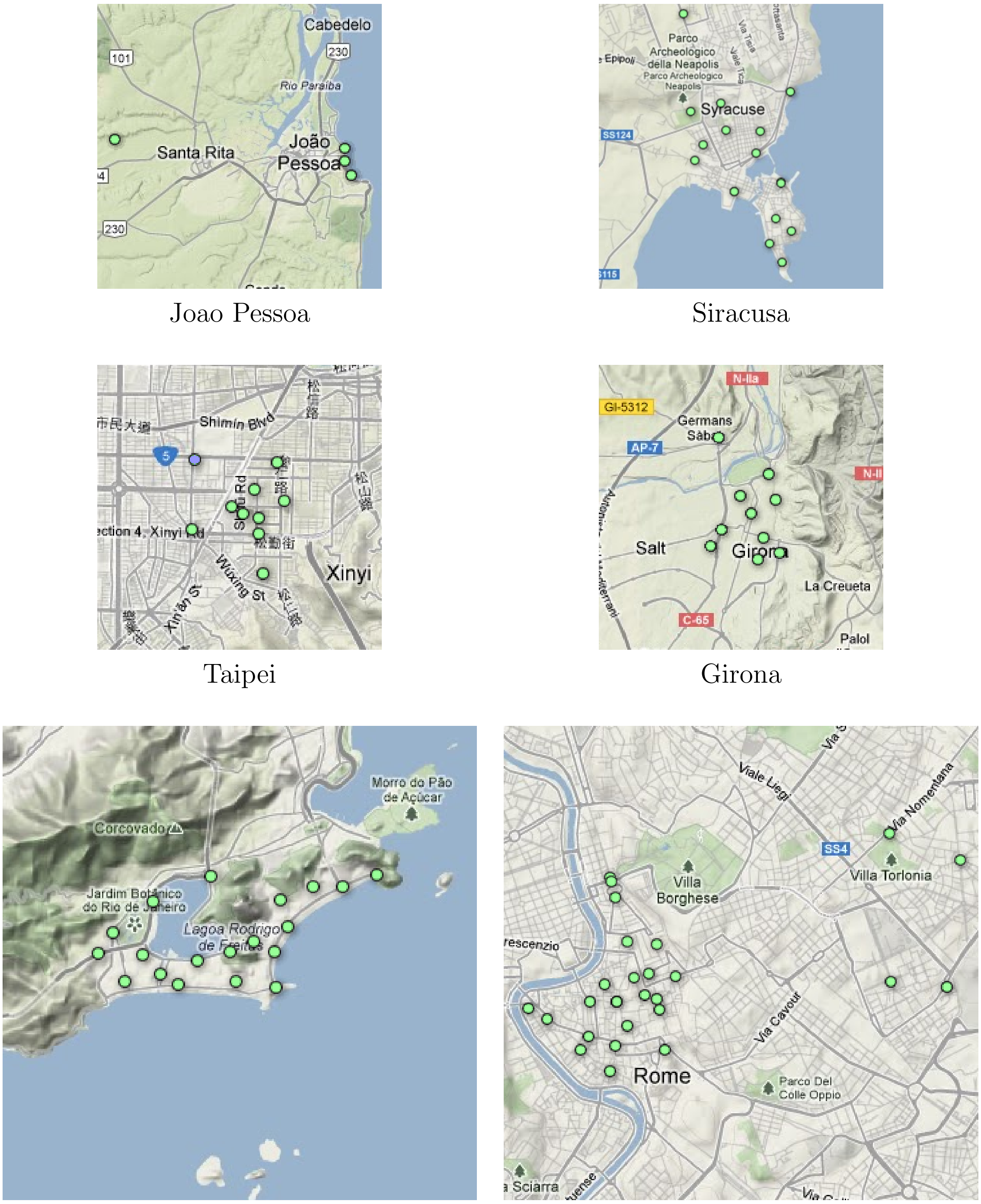
3.2.4 Mapping occupancy flux clusters

In this section we present map visualisations of the four flux clusters across our ten cities. We use a small red circular marker to denote a station belonging to cluster one, a blue marker for cluster two, a green marker for cluster three and a yellow marker for cluster four.

Figure 3.3 shows these clusters for the six smaller systems. We immediately note that these systems are homogeneous: all the stations in each system belong to the same cluster, namely the third cluster with the "flat" occupancy series. This implies that all stations in small systems behave similarly to each other. It is likely that because the supply of stations stations is so constrained in these systems, demand for bicycles and vacancies is distributed more evenly, leading to the flat occupancy line.

We focus on London and Barcelona in Figure 3.5, where the heterogeneity of station behaviour is clearly visible. It seems that this diversification is only possible when there are enough stations that spatial demand for bicycles and vacancies is free to manifest without hitting supply ceilings. London's clusters look like concentric circles, with cluster one (morning sink) stations at the centre, surrounded by successive layers of cluster three and cluster two (morning source) stations. This reflects a morning surge of bicycles from outside the centre moving inwards, and a slow outwards flow over the course of the day. This suggests that bicycle redistribution vehicles should move outwards in the morning to counteract the morning surge, preventing depletion of the stations in the outermost cluster and the saturation of stations in the centre.

Similarly, Barcelona's morning sink stations run through the city centre and spread out along the coast, and the morning source stations are spread out over the rest of the city. This map corresponds well with Barcelona's elevation; it is in a hilly region and the placement of the "sink" stations corresponds to lower elevations, while the "source" stations are at higher elevations, a consequence of the natural tendency of users to prefer riding downhill rather than uphill. Barcelona's map visualisation suggests that redistribution vehicles should move outwards in the morning, and slowly back inwards during the day.



Joao Pessoa

Siracusa

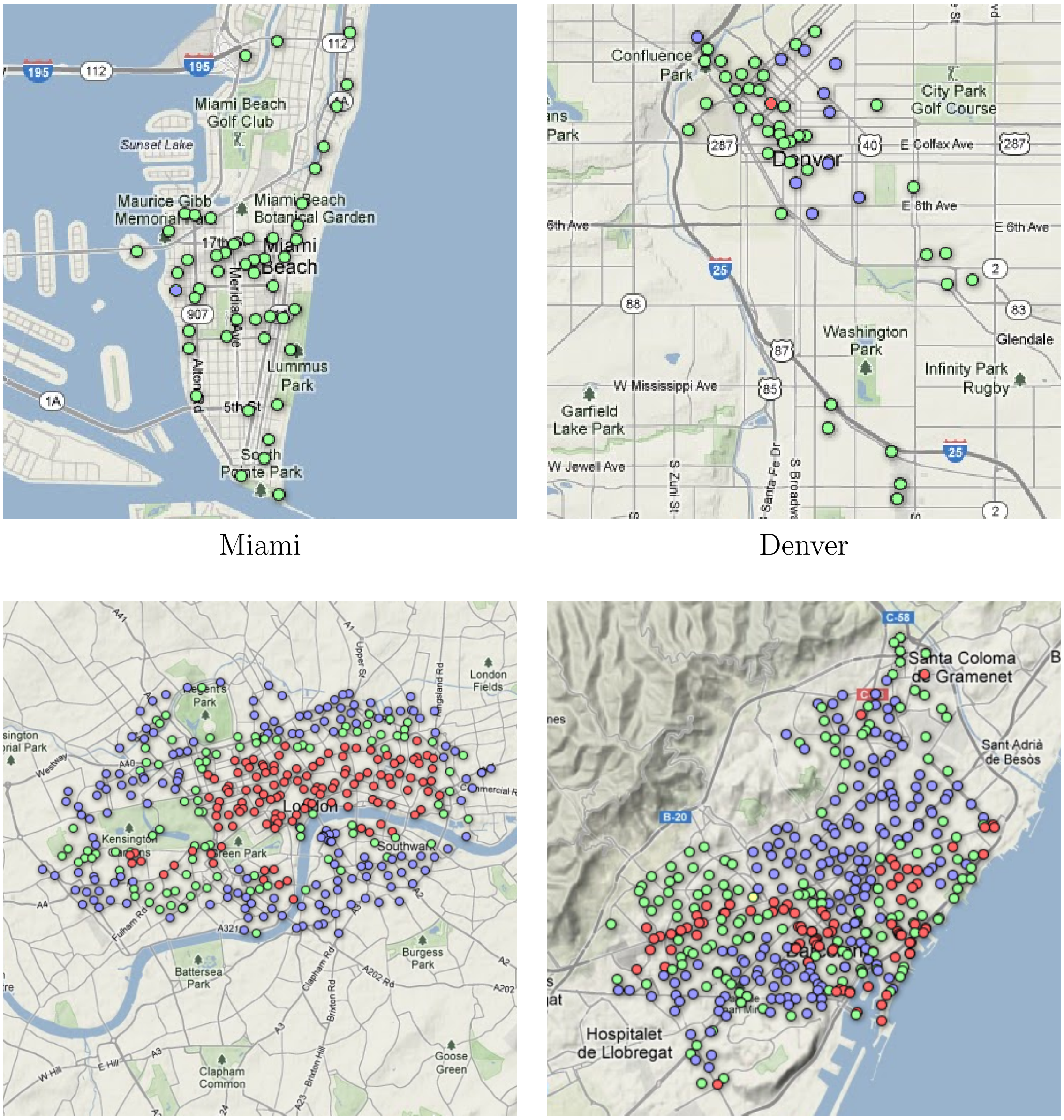
Taipei

Girona

Rio de Janeiro

Rome

Figure 3.3: Cross-city flux clustering: maps. Cluster 1 in red, cluster 2 in blue, cluster 3 in green, and cluster 4 in yellow.



Miami

Denver

London

Barcelona

Figure 3.4: Cross-city flux clustering: maps (continued). Cluster 1 in red, cluster 2 in blue, cluster 3 in green, and cluster 4 in yellow.

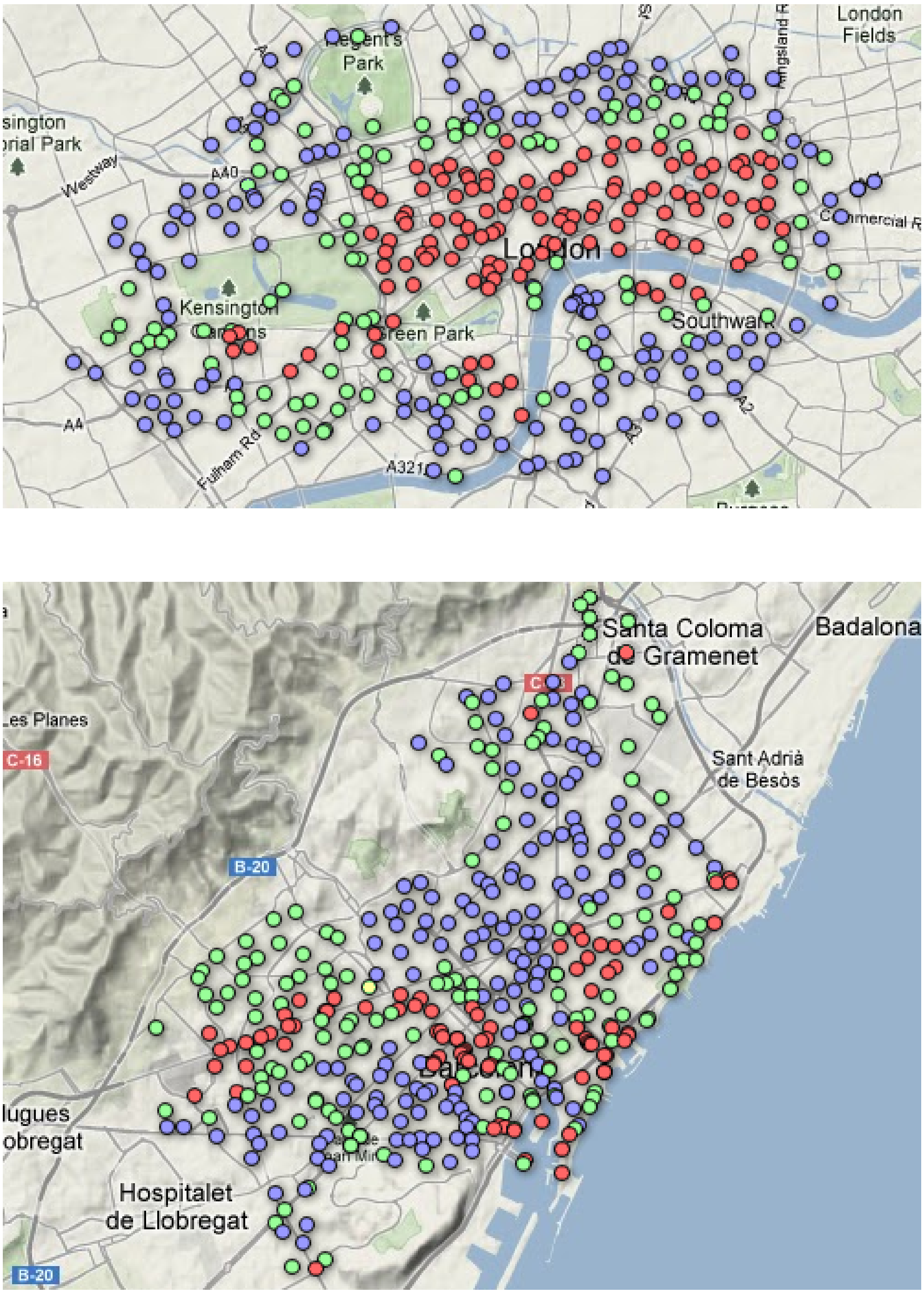


Figure 3.5: Cross-city flux clustering: a closer look at the maps of London (above) and Barcelona (below). Cluster 1 in red, cluster 2 in blue, cluster 3 in green, and cluster 4 in yellow.

3.3 Station activity level

Next, we performed clustering by how *active* the stations are in terms of their approximate daily borrow/return events. Detecting natural classes for this notion of behaviour would also be immediately useful. By mapping these classes, we can detect neighbourhoods of “stressed” stations; those which have unusually high levels of activity, suggesting their expansion or augmentation by the introduction of new stations in the vicinity to better balance the load. Conversely, we would also be able to detect neighbourhoods of underutilised stations, suggesting that closing down one or more stations in that vicinity would reduce operating costs without a large penalty on the system load or perceived quality of service.

In this section we describe a methodology for carrying out such analysis, and present the global behavioural clusters that emerge in our dataset.

3.3.1 Calculating a station’s activity level

Our measure for a station’s activity level is considerably simpler than the vector representation we used for its occupancy flux. We refer to the measure as a station’s “delta”. It is a single number generated by summing the absolute differences between every consecutive pair of points in its occupancy series o:

$$delta(o) = \sum_{1<i\leq 240} |o_i - o_{i-1}| \tag{3.2}$$

This number represents a notion of “churn”, or “turnover”. For example, if a station’s occupancy series delta is 3, then over the course of the day it has seen *approximately and no less than* 3 times as many borrow+return events as its capacity. Similarly, a station with an occupancy series delta of 0.5 sees approximately and no less than half its capacity in borrow+return events.

We cannot be exact because the series consists of averaged occupancy values within 6-minute windows, and so we only account for activity that occurs between one window and the next, losing any activity that has occurred over the course of the window. For example, consider this 2-minute sampled series of occupancies: [0.5, 0.6, 0.4, 0.5, 0.5, 0.5]. That is, at some timepoint t the sampled occupancy at a station was 0.5, at $t + 2$ minutes it was 0.6, *etc.* This series becomes the 6-minute averaged series [0.5, 0.5]. From the 6-minute series, it appears that there is no activity at the station from one sample to the next, but as is evident from the 2-minute series, this is not an accurate conclusion.

It is worth noting that even if we had used the samples at 2-minute intervals, some level of activity would still have been lost. If we observe b bikes at timepoint t, and b bikes at timepoint $t + 2$ minutes as well, it is indeterminable whether there was no activity at the station, or whether there was an equal number of borrowed and returned bicycles during the interval between the two observations. The only way to guarantee that no such activity loss is incurred is to continuously sample station statuses, which is clearly infeasible, or by having access to the rental logs of the transport authorities, which are currently not in the public domain. In practice, however, borrow/return events at most stations are sparse enough that the 6-minute series deltas are usable approximations for activity level.

To calculate the centroid $delta_c$ of a cluster of stations S, we simply averaged the deltas of all stations in the cluster:

$$delta_c = \frac{\sum_{s \in S} delta(s)}{|S|} \tag{3.3}$$

Absolute difference was used for inter-station distance.

By clustering stations on the basis of their occupancy series delta, we identified groups of stations which have similar levels of activity with respect to their capacity; those which have similar proportions of bicycles borrowed and returned over the course of the day.

3.3.2 Naturally-occurring delta clusters

Using the clustering methodology from §3.1 and the heuristic for choosing the number of clusters described in §3.1.1, we produced 6 clusters (Table 3.3) from stations across all systems.

Cluster	n	$delta_c$
1	824	0.224
2	31	0.884
3	134	0.512
4	5	1.470
5	1	3.323
6	1	3.047

Table 3.3: Series delta: final clusters.

There are 3 major clusters and 3 minor clusters. The first and largest cluster contains the majority of stations, 824. The value of $delta_c$ for the centroid of this cluster is 0.224, or approximately 20%. Stations in this cluster see approximately 20% of their total capacity in bike borrowing and returning events over the course of a typical day. So a bike station in this cluster with a capacity of 10 experiences approximately 2 daily borrow+return events on average. A station with a capacity of 40 experiences approximately 8 borrow+return events, etc.

The 824 stations in the first cluster, the vast majority under study, are arguably underutilised with respect to their capacity. This may be for reasons unique to each station: there may be a demand for the station to be a source, but it cannot act as one because it runs out of bikes too quickly; there may be a demand for it to be a sink, but it cannot be because it runs out of vacancies easily; or it may simply be too far away from anything of interest to be actually useful.

The second major cluster contains 31 stations and has a centroid $delta_c$ of 0.884, or ∼90%. By our measure, this is the most "active" of the three major clusters. A station in this cluster with a capacity of 10 sees around 9 daily borrow+return events. These stations are well-utilised with respect to their capacity.

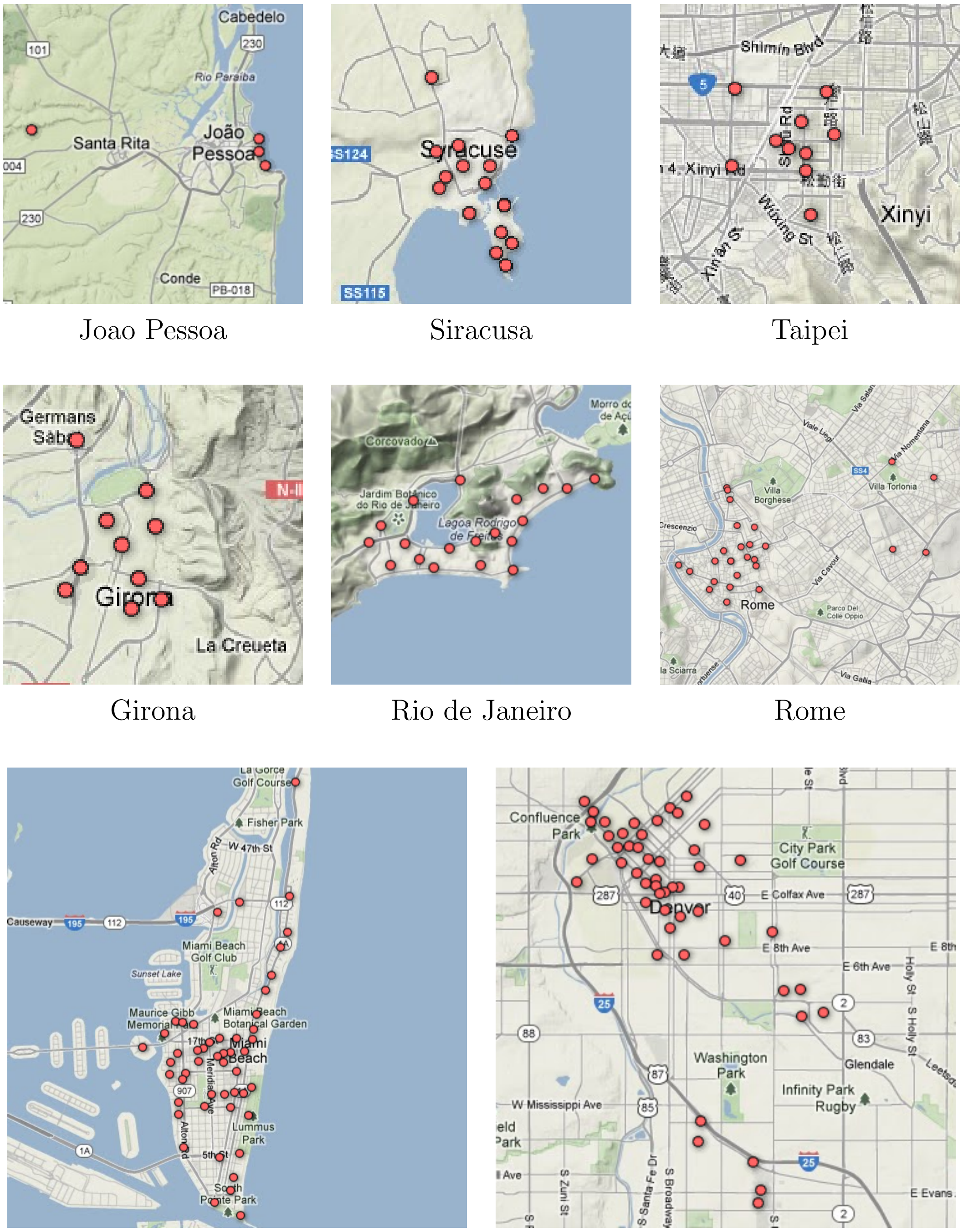
The third major cluster contains 134 stations and has a centroid $delta_c$ of 0.512, or approximately 50%. These stations can also be said to be well-utilised with respect to their capacity. While these stations are not as active as those in the second cluster, they are likely to cope better with sudden bursts in activity. A station that is almost perfectly utilised may become completely unused if a sudden activity spike causes a severe supply/demand mismatch.

The fourth, fifth and sixth clusters are minor clusters containing 5, 1 and 1 station(s) respectively. Their values of $delta_c$ for their respective centroids are 1.470, 3.323, and 3.047, or approximately 150%, 330% and 300%. A station in the sixth cluster with a capacity of 10 sees approximately 30 borrow+return events daily. These stations are clearly being very well utilised. However, the extreme levels of activity at these stations may be stressing the stations' facilities, leading to increased maintenance and repair costs. Stations with extreme levels of activity occur where demand is high throughout the day, and supply and demand are consistently well matched for *both* bikes as well as vacant slots. These stations may not necessarily cope well with sudden changes in activity pattern, suggesting that auxiliary stations should be built nearby to balance the load.

Our analysis shows that globally, the vast majority of stations belong to the first cluster, and are arguably underutilised (or oversized). A large number of stations belong to the second and third clusters, which appear to hit an activity level "sweet-spot". In future work, it would be useful to study the properties of stations in the well-utilised clusters so that attempts can be made to move a larger proportion of the world's stations into those clusters.

3.3.3 Mapping delta clusters

Map visualisations of the four flux clusters across our ten cities are presented in Figures 3.6 and 3.7. As with the flux series clustering, all of the small systems are homogeneous and are similar to each other. In fact, the only system that displays considerable heterogeneity is Barcelona, which is composed primarily of stations from cluster one and three.



Joao Pessoa

Siracusa

Taipei

Girona

Rio de Janeiro

Rome

Miami

Denver

Figure 3.6: Cross-city delta clustering: maps. Cluster 1 in red, cluster 2 in green, cluster 3 in blue, cluster 4 in brown, cluster 5 in grey, cluster 6 in white.

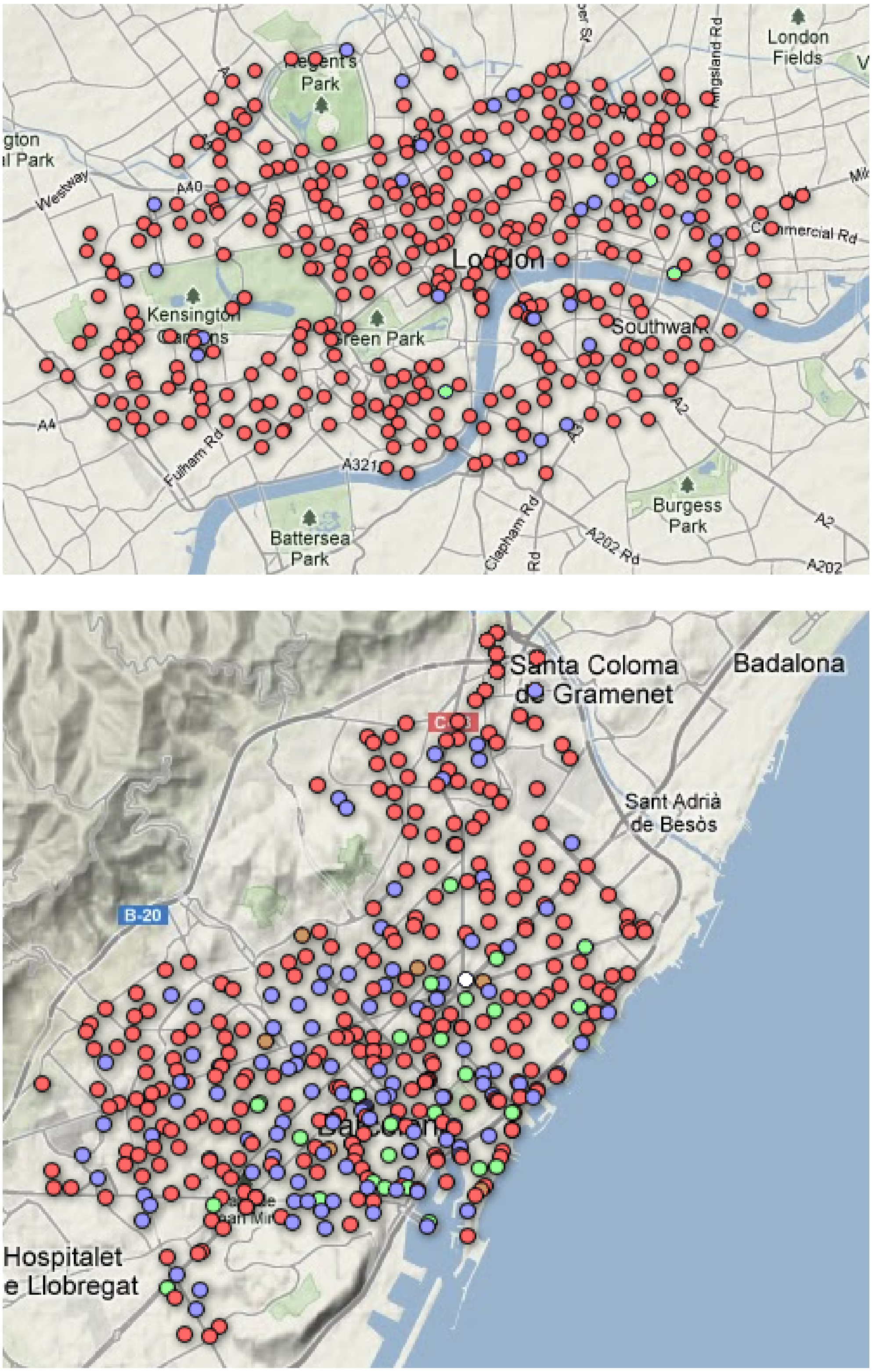


Figure 3.7: Cross-city delta clustering: London (above) and Barcelona (below). Cluster 1 in red, cluster 2 in green, cluster 3 in blue, cluster 4 in brown, cluster 5 in grey, cluster 6 in white.

3.4 Occupancy distribution

Finally, we performed clustering on the distribution of occupancies observed for each station. By clustering stations on the basis of their occupancy distributions, we attempted to discover those groups of stations which spend similar amounts of time at similar levels of fullness.

The detection of natural classes of this type of behaviour would be immediately useful for optimising station sizes as well as planning redistribution vehicle routes. For instance, stations that spend almost all their time completely full are candidates for expansion by the inclusion of new docking points. Conversely, stations that spend almost all their time completely empty are candidates for downsizing.

In this section we describe a robust methodology for carrying out such analysis. We present the global behavioural clusters that emerge from our dataset.

3.4.1 Calculating a station's occupancy distribution

For each station, we discretised the occupancy for each observation into deciles. We then counted the observations in each decile and expressed these counts as percentages of the total number of observations, producing a 10-bin histogram representing the occupancy distribution for that station.

To calculate the centroid distribution of a cluster of stations, we simply average the distributions of all stations in the cluster. As with the normalised occupancy series, in practice, we obtained the centroid of the cluster resulting from a merge by calculating a weighted average of the two constituent cluster centroids.

3.4.2 The Hellinger distribution distance metric

The station-pairwise distance between two distributions is calculated by the Hellinger distance [18]. For two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$, the Hellinger distance $H(P, Q)$ between them is defined as:

$$H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2} \tag{3.4}$$

This can be rewritten using the Bhattacharya coefficient [19] of P and Q:

$$H(P,Q) = \sqrt{1 - BC(P,Q)} \tag{3.5}$$

where

$$BC(P,Q) = \sum_{i=1}^{k} \sqrt{p_i q_i} \tag{3.6}$$

Hellinger distance expressed in terms of the Bhattacharya coefficient has previously been found effective for measuring the divergence between two discrete distributions [20]. It is important to note that while we use the Bhattacharya coefficient to calculate Hellinger distance, the resultant distance metric is quite different[4] from the *Bhattacharya distance* metric, $B(P,Q) = -ln(BC(P,Q))$.

3.4.3 Naturally-occurring occupancy distribution clusters

Using this distance metric, the hierarchical clustering methodology described in §3.1, and the heuristic for choosing the number of clusters described in §3.1.1, we produced 10 clusters from stations across all systems.

The occupancy distributions for the centroids of the five largest clusters are presented in Figure 3.8. The first large cluster contains the overwhelming majority of stations (734). The occupancy distribution of this cluster shows a low, similar probability of being in any level of occupancy. Thus most stations spend small and roughly equal amounts of time at many levels of fullness. This is not necessarily desirable, as a station should ideally spend the most time at a level of occupancy proportional to the demand for bikes and vacancies at that station.

[4] Most notably, the Bhattacharya distance does not obey the triangle inequality, whereas Hellinger distance does [21].

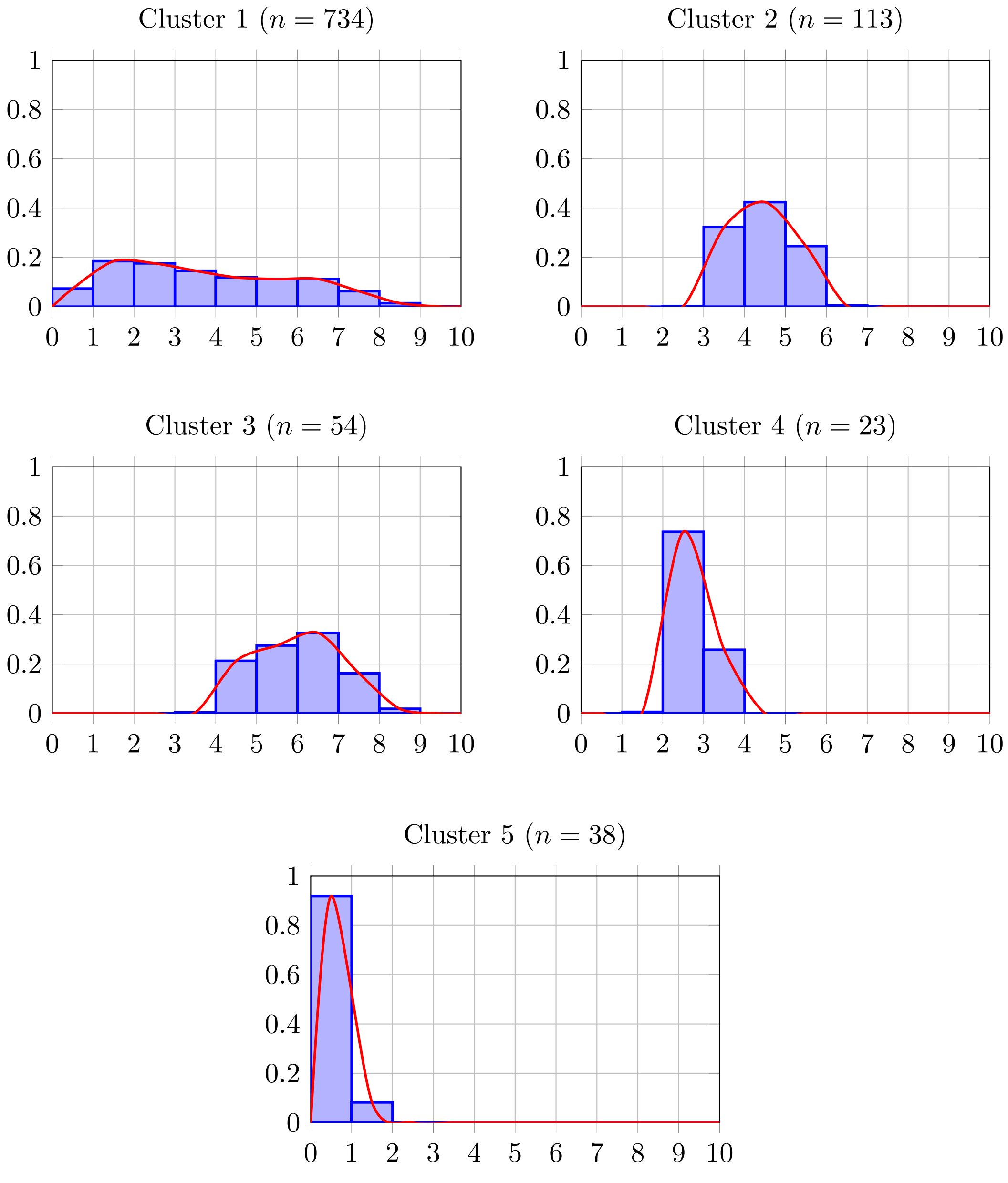


Figure 3.8: Cross-city distribution clustering: five largest centroids at $k = 10$. On the x-axes is the occupancy decile. On the y-axes is the probability of a station in that cluster being at that level of occupancy.

The second large cluster contains 113 stations. Stations in this cluster spend most of their time being 30-60% full. These stations might appear to be underutilised in comparison to those in the first cluster, but on the other hand they are also more resilient to sudden changes in demand, as they always have some bikes and vacancies available. Similarly, the 54 stations in the third large cluster are usually between 40% and 80% full, the 23 stations in cluster 4 are between 20% and 40% full, and are also likely to be resilient to sudden changes in demand.

The final large cluster contains 38 stations, and spends most of its time at 0-20% occupancy. These stations are usually near-empty, suggesting that they are more used as sources of bicycles rather than sinks, and would benefit from being reprioritised in the redistribution schemes so that there are more bikes present.

The occupancy distributions for the centroids of the five smaller clusters are presented in Figure 3.10. These clusters have low membership but exhibit quite interesting behaviour. The occupancy distributions in each of these minor clusters are very narrow. That is to say, bikes in these clusters seem to spend most of their time at a very specific level of occupancy. For instance, stations in cluster 6 spend the overwhelming majority of their time between 10% and 20% full. Stations in cluster 10 spend almost all their time between 40% and 50% full. This occurs when the demand for bikes and vacancies at a station are equal, resulting in similar rates of borrowing and returning events. As a consequence, the occupancy level of the station remains relatively stable.

Stations in cluster 7 suffer from the opposite problem to those in cluster 5; they are *too* full, spending most of their time at 80-100% occupancy. They would also benefit from being reprioritised in bicycle redistribution schemes so that bicycles are periodically removed to address the demand for vacancies.

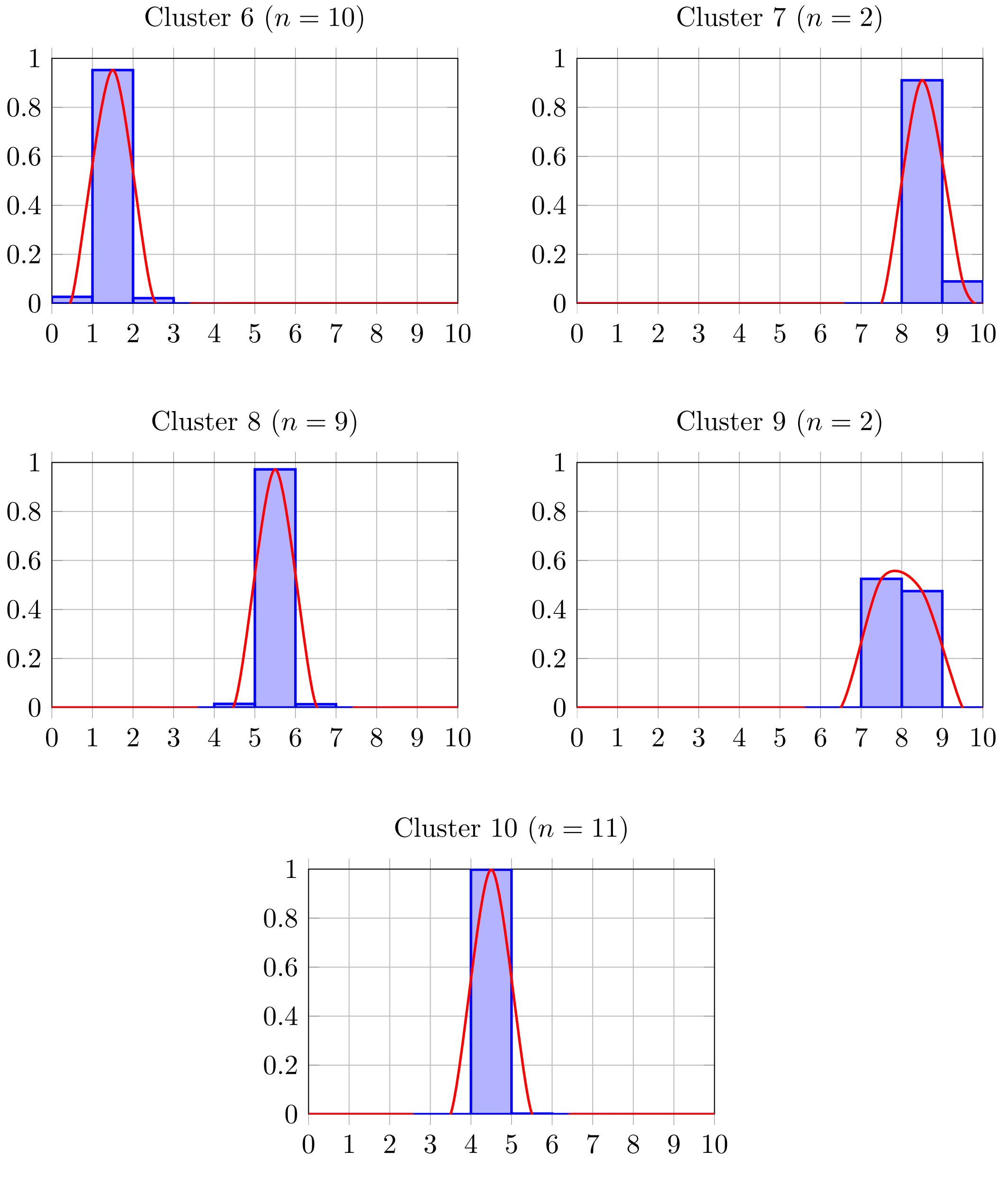


Figure 3.9: Cross-city distribution clustering: five smaller centroids at $k = 10$. On the x-axes is the occupancy decile. On the y-axes is the probability of a station in that cluster being at that level of occupancy.

3.4.4 Mapping occupancy distribution clusters

In this section we present map visualisations of the ten distribution clusters across our ten cities. We use the colour-coded markers as listed in Table 3.4.

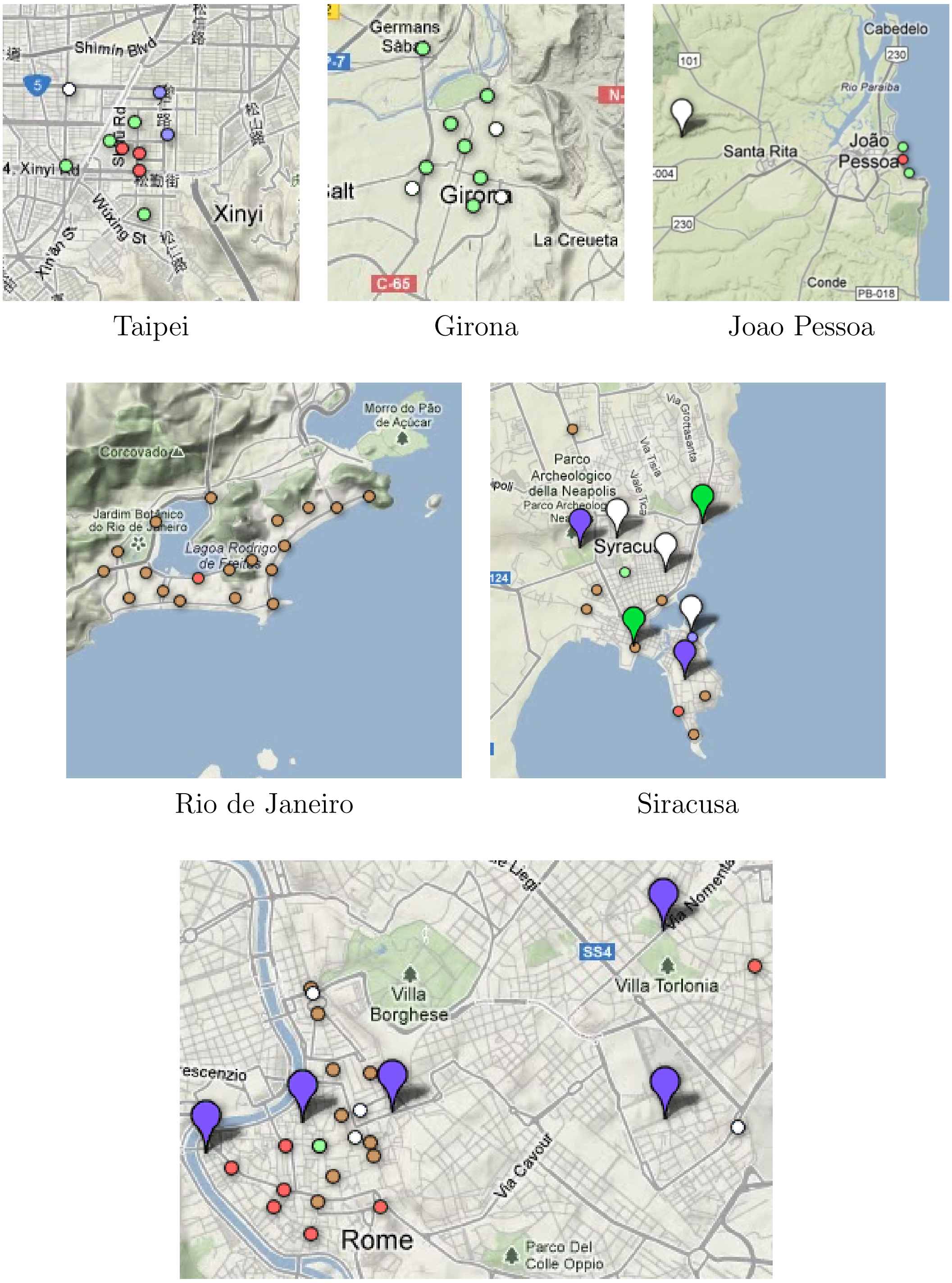
Cluster	**Marker**
1	Small white
2	Small green
3	Small blue
4	Small red
5	Small brown
6	Large purple
7	Large orange
8	Large green
9	Large red
10	Large white

Table 3.4: Distribution cluster map legend.

The maps of our six smaller systems are presented in Figure 3.10, and for Miami and Denver in Figure 3.11. Our first observation is that the diversification behaviour (heterogeneity) of systems is the complete inverse of our flux clustering study in §3.2. That is to say, the smaller systems display greater heterogeneity in types of station, whereas the larger systems are more homogeneous.

Rio de Janeiro is composed primarily of cluster five stations; those which spend almost all their time in 0-10% occupancy. It seems that there are very few bicycles in circulation in this system, or that station capacities are much larger than necessary. Girona is composed primarily of cluster two and cluster one stations, which have well-distributed occupancies, suggesting that station sizes and bike circulation are well-planned. Similarly, Miami and Denver also primarily contain stations from clusters with occupancy distributions that lie far from the extremes.

The maps for London and Barcelona are presented in Figure 3.12. Both these cities display great homogeneity, with most of the stations belonging to cluster one, suggesting that most station sizes are within their ideal bounds.



Taipei

Girona

Joao Pessoa

Rio de Janeiro

Siracusa

Rome

Figure 3.10: Cross-city distribution clustering maps. Small white: cluster 1. Small green: cluster 2. Small blue: cluster 3. Small red: cluster 4. Small brown: cluster 5. Large purple: cluster 6. Large orange: cluster 7. Large green: cluster 8. Large red: cluster 9. Large white: cluster 10.

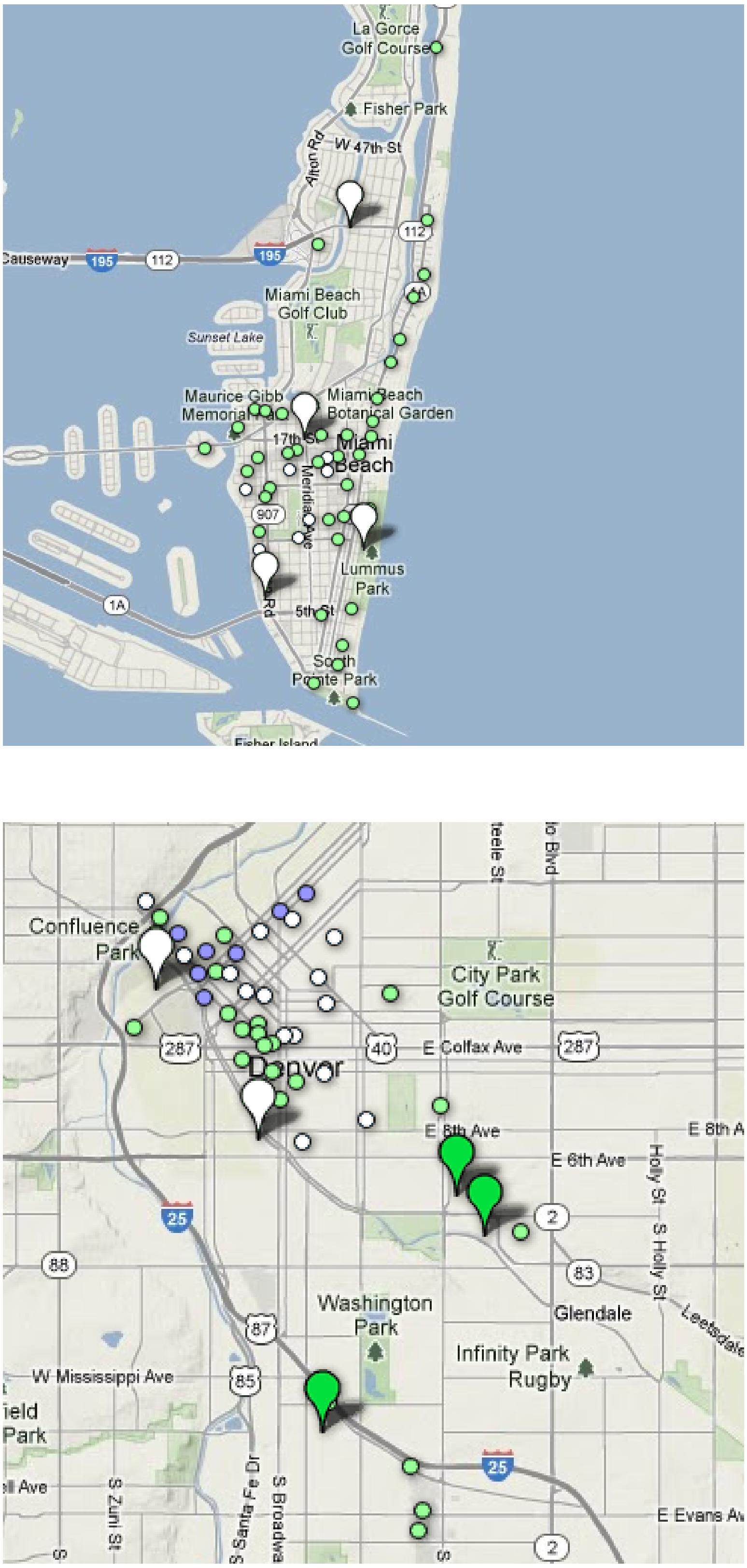


Figure 3.11: Distribution clustering: Miami (above) and Denver (below).

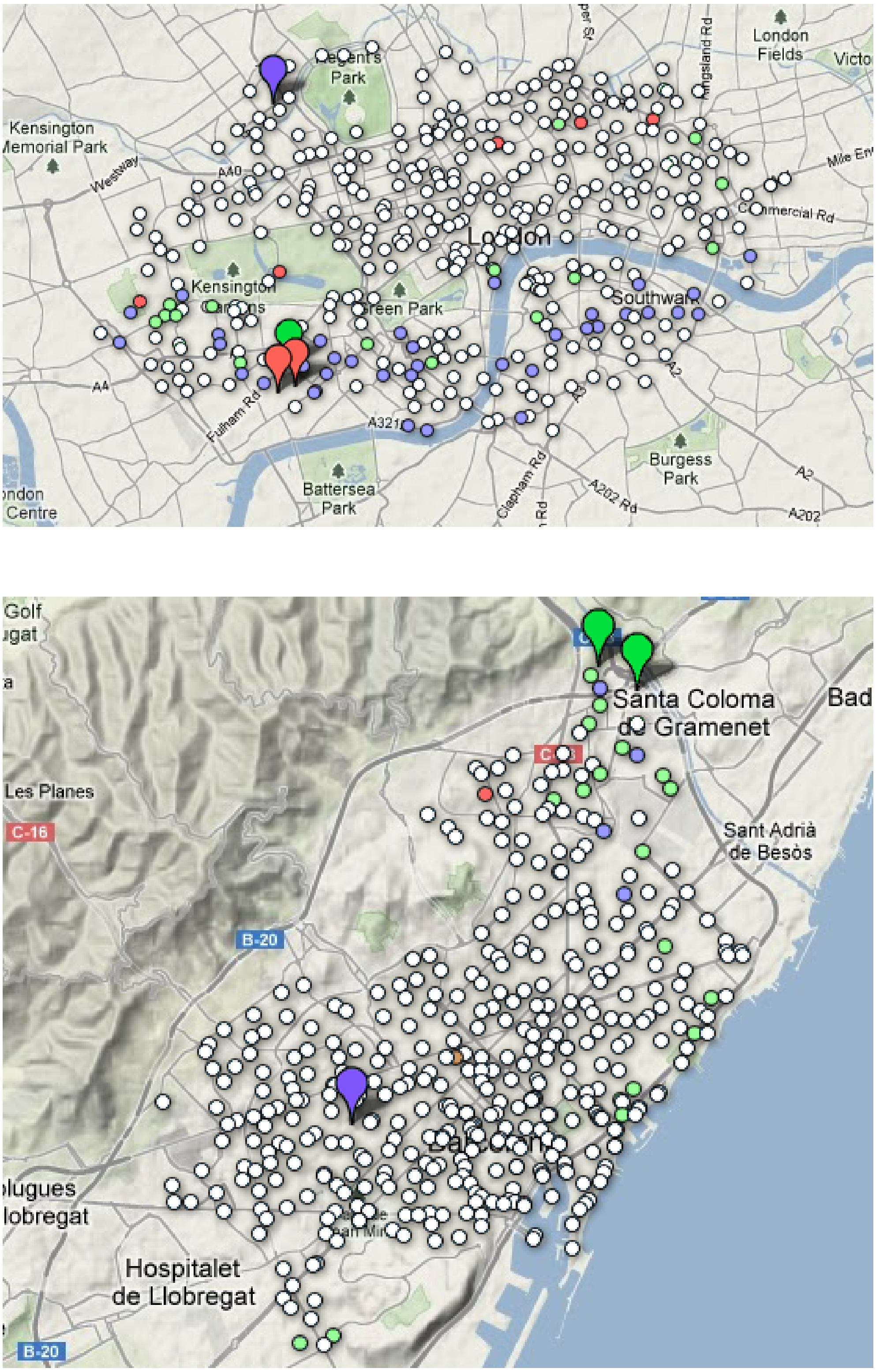


Figure 3.12: Distribution clustering: London (above) and Barcelona (below).

Chapter 4

Forecasting Utilisation

The behavioural clustering analysis in Chapter 3 was conducted on the basis of a temporally static representation of a bicycle station. This would help designers and administrators make decisions to improve various aspects of the system on a long-term basis, such as the addition or removal of bike stations. If stations were instead represented as functions of sliding windows of their activity history, it might be possible to discover new behavioural clusters that appear and disappear over time. This was beyond the scope of the project, but is worthy of future study.

Moreover, a short-term forecast of the utilisation of a station would allow for dynamic, adaptive decision making. For instance, a large event organised in a remote area of the city may cause an unusual load in the stations in the vicinity of the venue. If that sudden heavy load was predicted a half-hour ahead, bike sharing administrators would be able to trigger a preemptive early redistribution to ensure quality of service. The forecast would also be immediately useful for users of the system, who would be able to choose their origin and destination stations on the basis of the future availability of bicycles and docking points.

In this chapter, we describe our investigation into predicting (or forecasting) the number of bicycles at a given station at fixed intervals in the future, given *only* the known history of activity at that station. We did not attempt to use the activity at other stations or external datasets such as weather or topography to improve this forecast as it was out of scope, but this is suitable for further research.

4.1 Problem formulation and testbed

The problem we tackled is simply that of modelling the number of bicycles at a station as a univariate time series. Consistent with the clustering study, we focused exclusively on weekdays. For each station, we considered 2 separate series: the series of observations sampled every 2-minutes, and the same series of observations as averaged, 6-minute samples. Unlike in the clustering analysis, the series were not averaged across all days. Thus, for example, a 2 day series at 2-minute samples contains 1440 observations. The same 2 days as a 6-minute averaged series contains 480 observations.

For both series, forecasts were made at four horizons: 6, 12, 24 and 48 minutes ahead. This geometric progression of horizons was chosen on the basis that each is a reasonable window ahead for either the user or the system administrator to be planning actions. Given the history of a station's observations as 2-minute samples, intervals of 6, 12, 24 and 48 minutes correspond to 3, 6, 12 and 24 samples respectively. Similarly, given a station's history as a 6-minute averaged series, the same intervals correspond to 1, 2, 4 and 8 samples respectively.

To test the predictive models, we used weekday data from weeks 11-19 (9 weeks) of our dataset. In 9 weeks, there are 10800 ($9 \times 5 \times 240$) 6-minute intervals. Excluding the first and final days, there are 10320 6-minute intervals. We chose 120 equispaced points at the edges of these intervals to be the points in time from which to run forecasts. That is, we simulated the situation where each of these points was the "present" state of the station, and allowed the predictive models to speculate on the values of (b, v) at each forecast horizon. We then compared the predictions against the actual values to evaluate them.

It was not an arbitrary decision to use 120 equispaced points. To evenly space 120 points across 10320 6-minute intervals, the points must be 86 intervals apart from each other. The lowest common multiple of 86 and 240, which is the number of 6-minute intervals in a 24-hour day, is 10320: the total number of points in our 9-week span of test data. Consequently, the 120 timepoints we have chosen do not repeat with respect to their position in the day. That is, at most one timepoint is at 00:00, at most one is at 00:12, and so on. Each minute during the day is represented by at most one time point in our test datasets.

At each of these 120 timepoints, we made 4 predictions for 2 series per station. Since we are forecasting at 4 horizons for 2 series at 120 timepoints for each station, we make a total of 960 predictions per station.

4.2 Prediction evaluation metrics

We forecast at 4 horizons for 2 series at 120 timepoints per station, making 960 predictions per station as described in §4.1. In this section we describe the methodology used to evaluate these predictions.

As mentioned in Chapter 1, all previous work has only considered a single bicycle sharing system, and within any one system the spread of station capacities is very narrow. Consequently, mean error in predicted number of bicycles has sufficed as a measure of predictor performance.

However, the stations across the 10 cities we are considering have a very diverse range of capacities. In this case, the mean error in predicted number of bicycles is not a reliable value on which to compare the predictability of different stations. For example, a mean error of 2 bicycles in a station with a capacity of 4 bicycles is indicative of poor prediction quality, but the same mean error of 2 bicycles in a station with a capacity of 30 indicates arguably better predictions.

To account for this discrepancy in station sizes, we used error metrics based on occupancy instead. For a series of n predicted observations of the form (b_i^P, v_i^P), and the corresponding series of ground truths (b_i^T, v_i^T), where $1 \leq i \leq n$, we calculate error metrics as follows:

1. Mean absolute error (MAE) in predicted occupancy:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{b_i^P}{b_i^P + v_i^P} - \frac{b_i^T}{b_i^T + v_i^T} \right| \tag{4.1}$$

   This metric simply presents the average absolute error across all predictions.

2. Root-mean-square error (RMSE) in predicted occupancy:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{b_i^P}{b_i^P + v_i^P} - \frac{b_i^T}{b_i^T + v_i^T}\right)^2} \tag{4.2}$$

   This averages the squared errors across all the predictions, and then presents its square root. As the errors are squared before they are averaged, larger errors automatically incur higher penalties. Taken together, MAE and RMSE can often help detect the presence of a small number of large errors.

3. Corrected sample standard deviation (SD) of error in predicted occupancy:

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left|\left(\frac{b_i^P}{b_i^P + v_i^P} - \frac{b_i^T}{b_i^T + v_i^T}\right) - MAE\right|} \tag{4.3}$$

   Strictly speaking, this is not a measure of predictive accuracy. Rather, it is a measure of predictive *precision*; models with low standard deviations in error often just need to be offset in order to start producing better predictions, whereas models with high standard deviations in error are unlikely to produce better predictions with simple linear corrective measures.

All three metrics were used to tune the parameters of our models. We only present RMSE performance towards the end of this chapter, for the sake of brevity and to keep the discussion focused.

4.3 Predictive models

We built four kinds of models. The study conducted by Froehlich et al., 2009 experimented with linear regression and Bayesian networks. We expand upon this using multilayer perceptrons and decision tree ensembles. Unfortunately, it would be meaningless to compare our results with theirs because we use entirely different datasets and entirely different evaluation strategies.

4.3.1 Random and static models

Our baseline was a random model. For all horizons, this simply predicts a random value drawn from the discrete uniform distribution ranging from 0 to the station capacity. This serves as the benchmark against which other models are compared.

The second predictive model was a static model. This assumes that the time series is horizontal, and predicts that the currently observed number of bikes will persist for all horizons. That is, if there are currently b bikes at the station, the static model will predict b bikes at all points in the future. As subsequently demonstrated, this is not as unreasonable as it may first appear.

4.3.2 Multilayer perceptron

The third kind of model was a multilayer perceptron[1]. The multilayer perceptron has previously been shown to be effective for univariate time-series forecasting [22, 23]. We did not train a single network with output dimensions for each forecast horizon. Instead, we trained 8 separate networks with a single output dimension: one for each forecast horizon, for the 2-minute sampled series and the 6-minute averaged series.

We did not train the multilayer perceptrons to directly predict the number of bicycles. Rather, they predict *occupancy* based on previous values of occupancy. Neural network regression is more effective when the range of the input and output dimensions correspond well to the sensitive range of the activation functions [24]. Occupancies fall in the range [0,1], corresponding better to the the sensitive range of our chosen activation functions (linear and sigmoid)[2].

[1] A specific kind of feedforward artificial neural network with multiple layers of interconnected perceptrons, hence the name.

[2] Furthermore, this so-called "feature scaling" generally improves the speed of convergence for the gradient descent used internally by the backpropagation algorithm classically employed to train neural networks.

Determining input vector length through autocorrelation study

For both the 2-minute and 6-minute series, we decided to use the 10 most recent observations at the station as input features. Consequently, the predictors using the 2-minute series extrapolate on 20 minutes of recent history, and when using the 6-minute averaged series extrapolate on 60 minutes of recent history.

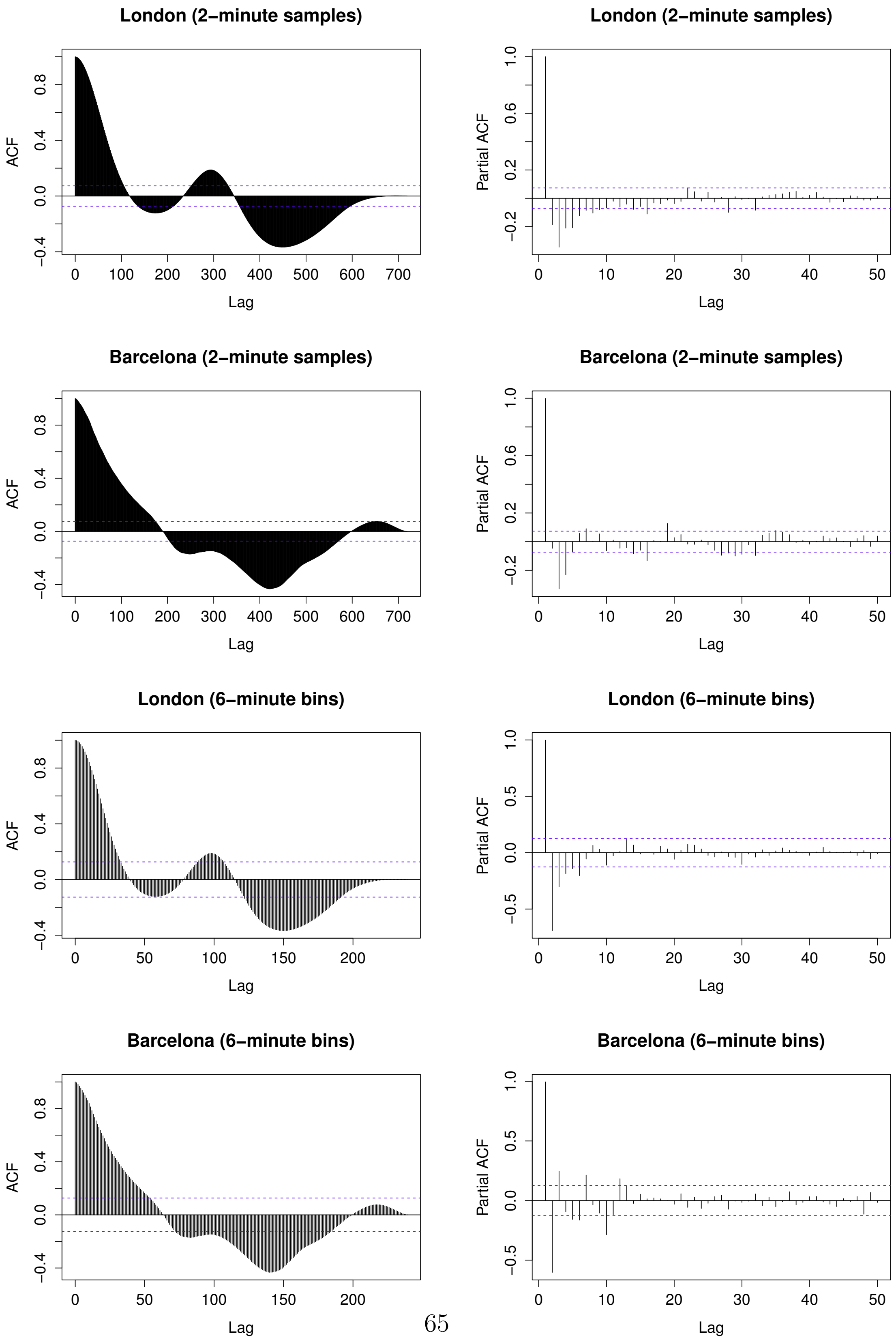
The decision to use a sliding window of only 10 samples is supported by the partial autocorrelation functions of the cities' averaged time series. For each bike sharing system, all observations across all stations at each timestamp during the day are averaged to yield an averaged pictures of a system's daily occupancy as in §2.4. Given measurements of occupancy $o_1, o_2, ..., o_n$ at timepoints $1, 2, ..., n$ respectively, the autocorrelation at lag k (ACF_k) is given by:

$$ACF_k = \frac{\sum_{i=1}^{n-k}(o_i - \bar{o})(o_{i+k} - \bar{o})}{\sum_{i=1}^{n}(o_i - \bar{o})^2} \tag{4.4}$$

where $\bar{o}$ is the mean of the series. The autocorrelation is simply the ordinary Pearson product-moment correlation of a time series with itself at a specified lag. The partial autocorrelation at lag k ($PACF_k$) is the autocorrelation that is not accounted for by autocorrelations at shorter lags. To calculate $PACF_k$, o_i is first regressed against $o_{i-1}, o_{i-2}, ..., o_{i-(k-1)}$. Then the correlation of the residuals of this regression with o_{i-k} is calculated. This is the autocorrelation which remains at lag k after the effects of shorter lags $(1, 2, ..., k-1)$ have been removed by regression. An efficient method for the calculation of $PACF_k$ is given by the Durbin-Levinson algorithm [25, 26], the discussion of which is out of scope.

Observation of the partial autocorrelation plots of these time series reveals, in every system, a significant partial autocorrelation at lag 1, and no other significant partial autocorrelations. As examples, plots of autocorrelation and partial autocorrelation at various lags for the 2-minute and 6-minute occupancy series for London and Barcelona are presented in Figure 4.1. It is clear, based on these observations, that most of the useful predictive information in these time series lies within a short sliding window, which we set at 10 samples.

Figure 4.1: Plots of autocorrelation and truncated plots of partial ACFs for London's and Barcelona's averaged 2-minute and 6-minute occupancy series. Significance thresholds are indicated by the dashed horizontal lines.



Structure of resultant networks

Our neural networks, therefore, had 10 input dimensions (neurons) and a single output neuron. We experimented with the following parameters: activation function of input layer (linear or sigmoid), number of hidden layers (0 or 1), number of neurons in the hidden layer (0, 3, or 10), activation function of hidden layer (linear or sigmoid), and activation function of output layer (linear, sigmoid, or softmax). We did not investigate additional hidden layers or activation functions as it was beyond scope, but it would be an interesting subject for future study.

The combination of parameters which consistently yielded lowest training error for a subset of our testbed was as follows:

- an input layer of 10 neurons, each with a linear (or identity) activation function: $\phi_I(x) = x$
- a single hidden layer of 3 neurons, each with a sigmoid (logistic) activation function: $\phi_S(x) = \frac{1}{1+e^{-x}}$
- an output layer of 1 neuron with a linear activation function.

The model described by this network is visualised in Figure 4.2.

Improving training speed through resilient backpropagation

Training a neural network such as the one in Figure 4.2 involves weighting the edges to minimise the prediction errors of the model over a training set. With 8 neural networks per station, we trained $\sim$8000 neural networks in total. This would have been extremely slow using classic backpropagation. We employed the *resilient backpropagation* heuristic [27, 28], also known as RPROP, a first-order optimisation algorithm that uses just the sign of the derivative during gradient descent instead of its magnitude. Resilient backpropagation is one of the fastest known learning techniques for neural networks, along with the cascade correlation [29] and Levenberg-Marquardt [30, 31] techniques. We trained each network for 20 epochs or until convergence, whichever completed earlier.

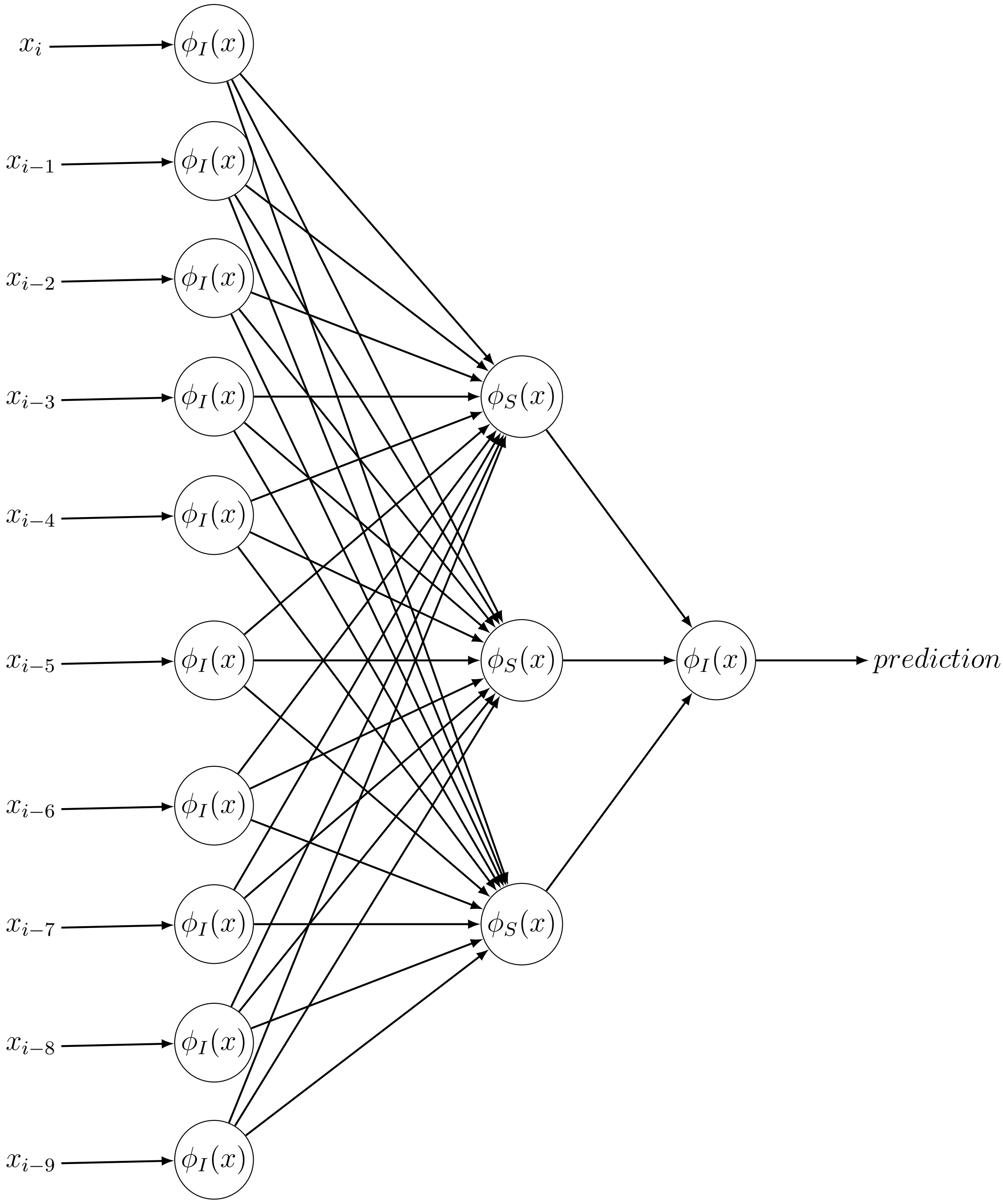


Figure 4.2: A visualisation of the fully-connected multilayer perceptron used to forecast station occupancy. ϕ_I is the identity function, also known as the linear activation function. ϕ_S is a sigmoid function, specifically, the logistic function.

4.3.3 Decision tree ensemble

Our fourth predictive model was an ensemble of decision trees with random feature selection and bootstrap aggregation, known colloquially as a random forest. As with the multilayer perceptron, we trained 8 separate models per station, corresponding to 4 horizons across 2 time series.

While our neural networks were trained exhaustively on all examples from our training set (§4.4), each tree in the ensemble was built from a random subsample drawn with replacement from the training set, that is to say, a bootstrap sample.

We tested the performance of ensembles containing 1, 5, 10, 25, 50, and 100 decision trees. Increasing the number of trees beyond 10 resulted in an extremely small improvement in training error, at the cost of a heavy penalty in the time taken to train the forest, so we set the number of trees in each ensemble to 10.

Each tree was built using a modified version of Quinlan's C4.5 decision tree learning algorithm [32]. Normally, C4.5 splits nodes based on the feature (or attribute) of the data that most effectively splits the training examples into subsets enriched in particular classes. Formally, this is the normalised information gain criterion, also known as Kullback-Leibler divergence [33]. Instead of choosing a feature to split out of all available features, we chose a feature out of a random subsample of the feature space. This randomness results in a slight increase in the bias of the ensemble[3]. However, ensemble averaging lowers the variance enough to compensate for the increase in bias. This yields a better model overall in comparison to a single tree.

The original publication [34] prescribes a voting mechanism to combine the predictions from individual trees. While this approach is sensible for classification-type problems, it is does not yield good results for regression problems. Our implementation combined predictors by averaging their probabilistic prediction, instead of letting each tree vote for a single value. Consequently, the predicted regression output of an input vector is computed as the mean predicted regression outputs of all the trees in the ensemble.

[3] with respect to the bias of a single non-random tree

4.4 Training data

Given the high volatility of the stations' time series, as evidenced by the partial autocorrelation function plots, it does not seem useful to train the models on vast portions of each station's distant history. Furthermore, as we trained nearly 16,000 models in total (8 neural networks and 8 random forests each for 996 stations), using large amounts of training data would require impractical computation time. Consequently, we decided to use a single week of training data. Since the test data began from week 11 onwards, we used week 10.

This one week has a 2-minute sample series 3600 (=720×5) samples long. We generated training examples from this series with an incremental 10-sample sliding window, yielding a maximum of 3566 (=3600-24-10) 10-feature training examples which have data points up to 24 samples ahead.

Similarly, when the series is converted to a 6-minute averaged series, it is 1200 (=240×5) samples long. We generated training examples from this series with an incremental 10-sample sliding window, yielding a maximum of 1182 (=1200-8-10) 10-feature training examples which have data points up to 8 samples ahead.

Therefore, each multilayer perceptron and decision tree ensemble was trained on a maximum of 3566 training examples in the 2-minute series, and a maximum of 1182 training examples in the 6-minute averaged series.

4.5 Predictor performance by horizon

The prediction error of models at various horizons, averaged over all stations for each system, are presented for the 2-minute series in Figure 4.3, and for the 6-minute series in Figure 4.4. We only present root-mean-square error for brevity.

All our models perform better than the random benchmark (dashed black), which has an error between 20% and 30%. The decision tree ensemble (blue) consistently outperforms the multilayer perceptron (red), and the static model (solid black) performs best of all.

In the smaller systems, with the exception of Girona, the prediction error generally remains stable as the forecast horizon is increased. That is, the models are approximately as good at predicting occupancy levels 6 minutes in the future as they are at predicting occupancy up to 48 minutes ahead.

This is not true of the larger systems, where predictor performance deteriorates as the prediction window increases. This suggests that the borrow/return patterns for stations in the smaller system are more consistent, and that the larger the system gets, the more stochastic this pattern appears, at least from the perspective of a univariate time series.

For the larger systems, prediction errors for the 2-minute series are almost identical to those of the 6-minute series. However, for the smaller systems, a variety of differences is observed. In general, the performance of the static model is unaffected, and the performance of the decision tree ensemble and multilayer perceptrons are worse in the 6-minute series.

One explanation for this result is that patterns in a station's occupancy are straightforward in the 2-minute series, but have a periodicity that is lost when averaging into 6-minute bins. For instance, consider the 2-minute occupancy series [0.0, 0.1, 0.2, 0.3, 0.4, 0.3, 0.2, 0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.3, 0.2, ...]. This series has a clearly observable triangular wave pattern. Taking the average of 3 elements at a time, we produce the 6-minute series [0.1, 0.33, 0.1, 0.2, 0.3, ...]. The pattern is completely destroyed as the periodicity of the original wave is not a harmonic of our averaging window, making the series harder to forecast.

This explanation supports the idea that larger systems are more stochastic than smaller ones. The performance of the predictors was unaffected by averaging for the larger systems, but negatively impacted by averaging for the smaller systems, suggesting that the models had been exploiting regular patterns in the smaller systems which were then being thrown off by averaging.

The simple static model performs better than the two sophisticated models, with an error of 0-10% in most cases. We believe that this has a few reasons:

1. The vast majority of stations have very sparse activity. This is supported by our activity clusters in §3.3, which showed that the largest cluster of stations only had a 20% occupancy turnover during a typical day. Under these circumstances, predicting no change is actually a good strategy because in an underutilised station, if there are b bikes at a given point in time, it is very likely that in 6 or 12 minutes there will still be b bikes.

2. Even the bike stations with moderate levels of activity usually experience the bulk their activity concentrated at certain times of day. For instance, our preliminary analysis in §2.4 showed that, with the exception of the American systems, spikes in usage occur corresponding to hours of work; people presumably use the bikes to commute to and from their workplace. The rest of the day, the stations are much less active. Our evaluation testbed gave equal representation to all times of day by design (§4.1), and so the good predictions made by the static model in off-peak hours would have compensated for its poor performance during peak hours.

3. Finally, the series themselves are only subject to incremental changes in value. That is, except for occasional large changes in occupancy due to redistribution vehicles, a single borrow/return event has only a very minor effect on the value of occupancy. So even if the static predictor gets the number of bikes wrong, it is likely to be only off by one or two.

The static model yields the most accurate predictions. Unfortunately, this result is not useful for the simulation, emergent cluster discovery, and preemptive redistribution purposes described in the introduction to this chapter. It is plausible that the multilayer perceptron and decision tree ensemble models have enough adaptability that they would be much better and more useful than the static model in certain stations during certain periods of the day. We did not investigate this, but it is the logical next step for future work.

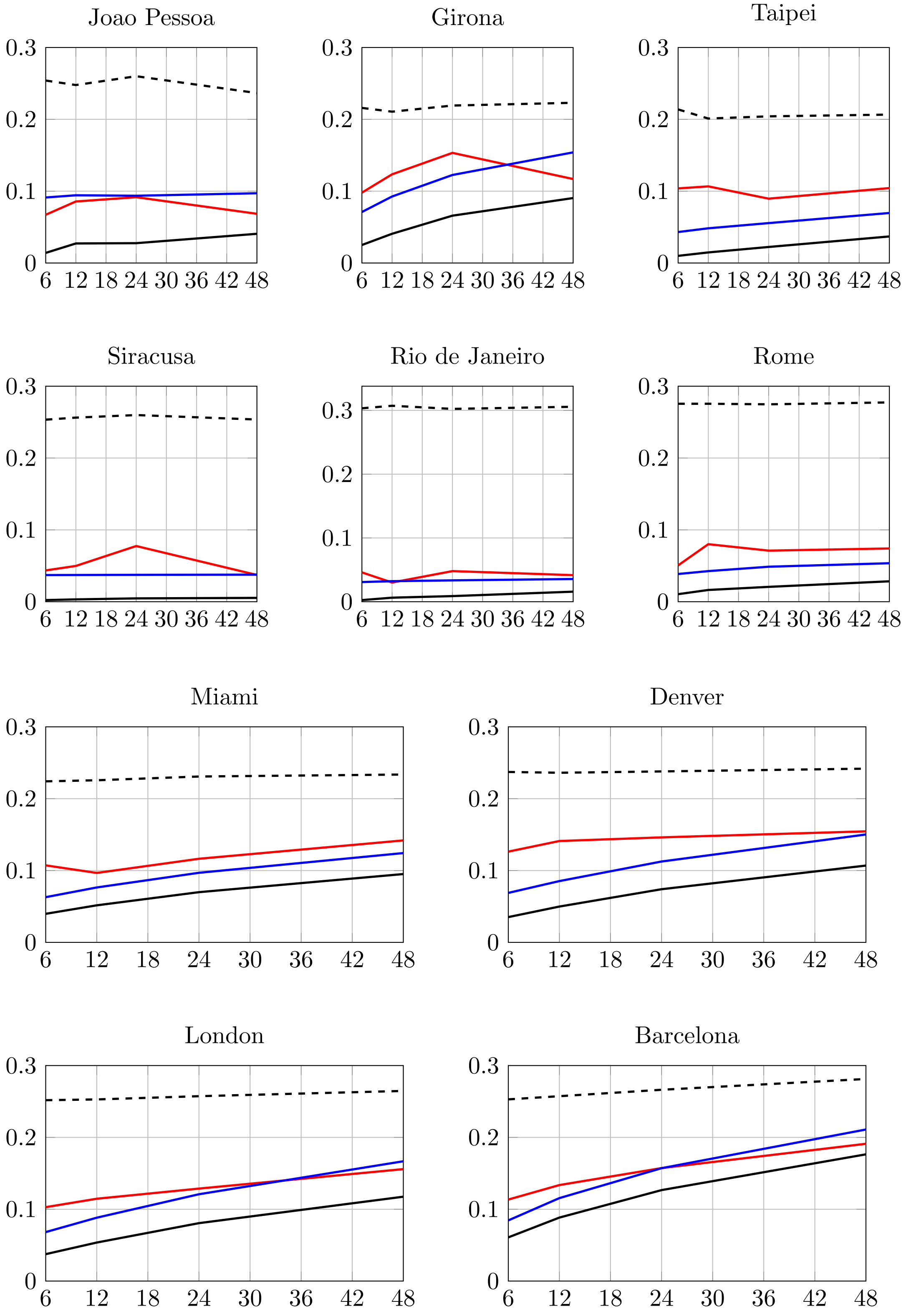


Figure 4.3: Predictor performance comparison for 2-minute series. Averaged root-mean-square error in predicted occupancy on the y-axis. Forecast horizon, in minutes, on the x-axis. Random model in dashed black, static model in black, multilayer perceptron in red and decision tree ensemble in blue.

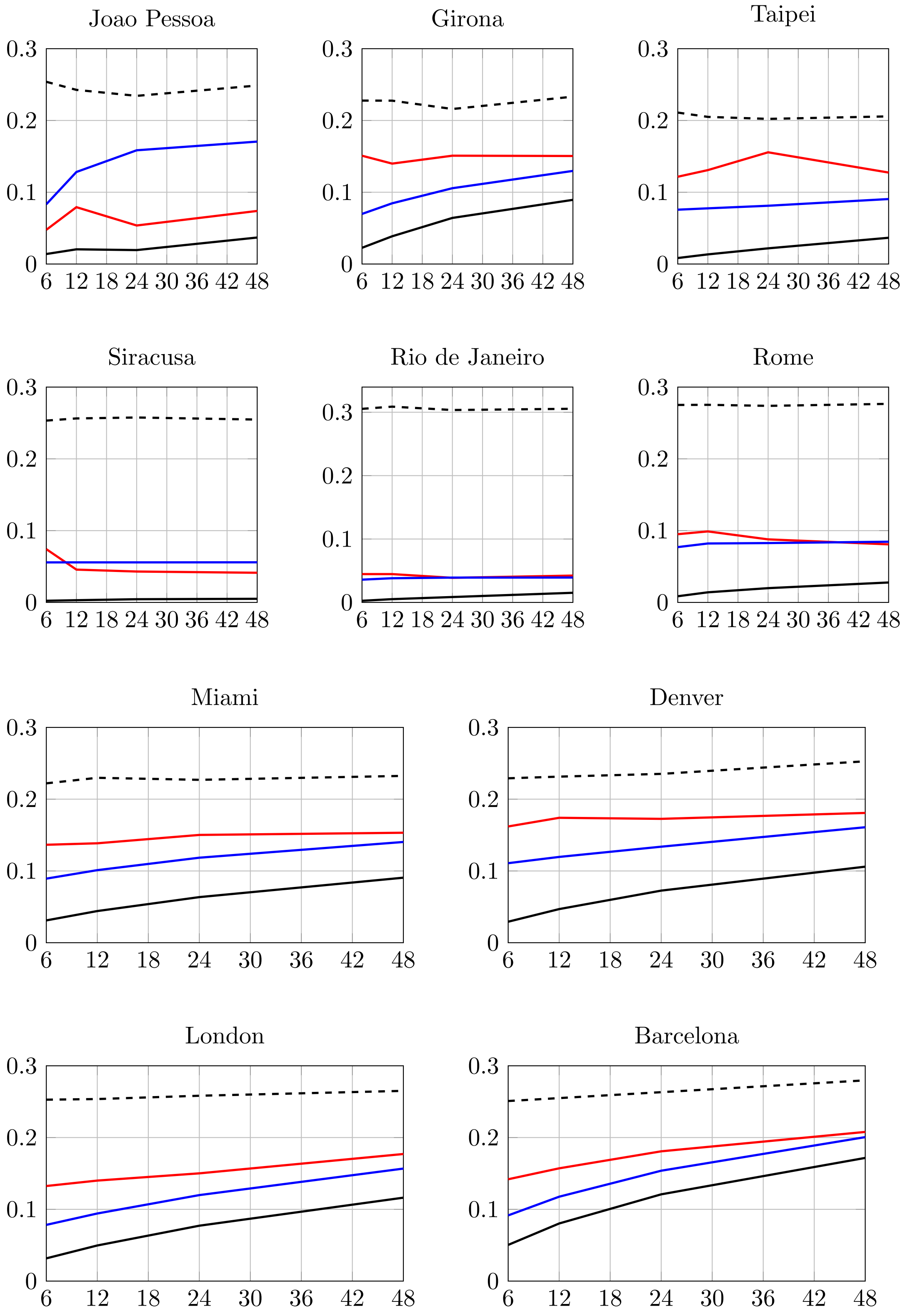


Figure 4.4: Predictor performance comparison for 6-minute series. Averaged root-mean-square error in predicted occupancy on the y-axis. Forecast horizon, in minutes, on the x-axis. Random model in dashed black, static model in black, multilayer perceptron in red and decision tree ensemble in blue.

4.6 The effect of proximity to training data

We chose one week of training data (week 10 of the dataset) to train our multilayer perceptron and decision tree ensemble models. We evaluated the models on 10 weeks of test data (weeks 11-19 of the dataset). Consequently, as we progress through the test set, the testing point gets temporally further and further away from the training data. In future work, it would be useful to train a separate model on a window of recent data for each evaluation point, so that the proximity to the training data remains constant. We did not pursue this avenue in our study since we were already training nearly 16,000 models.

We did however explore the effect that the proximity to training data had on our predictions. One might expect that as the training data becomes more and more outdated, it becomes less and less relevant and results in poorer predictions. Our results show that this is true for the small systems, such as that of Siracusa, as shown in Figures 4.7 and 4.8. However, for larger systems, such as that of London, there is no observable deterioration of performance, as shown in Figures 4.5 and 4.6. Each plot serially presents the error in our models' predictions for each of the 120 testing points. Each point is approximately 516 minutes later than the previous point in the series. The rightmost point of the plot is temporally furthest from the training data, and the leftmost point is temporally nearest.

Our observations are once again attributable to the idea that station behaviour in large systems is more stochastic than in smaller systems. Consequently, greater proximity to the training data yields better performance in the smaller systems, but is unimportant for the larger systems.

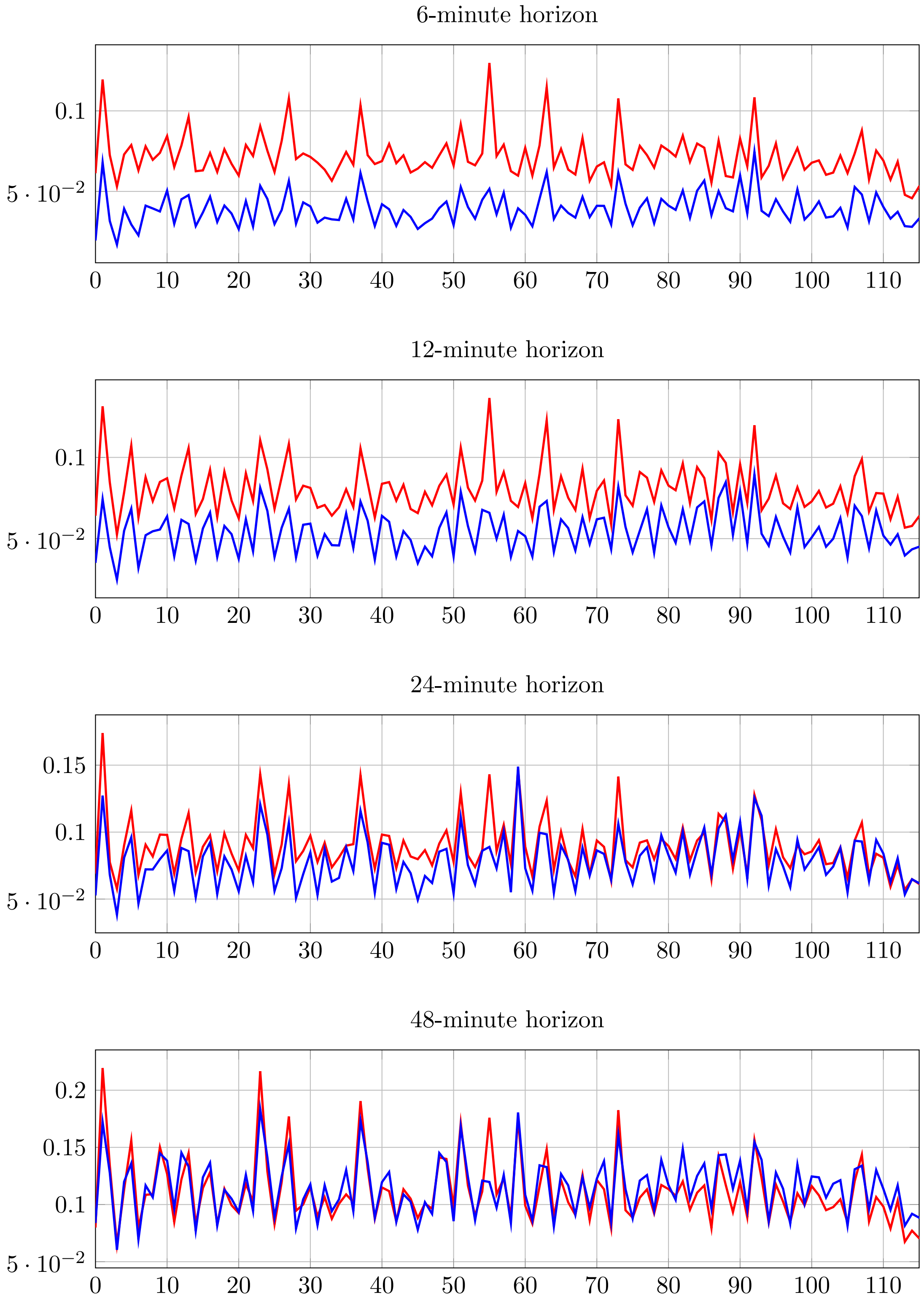


Figure 4.5: Predictor performance over time for London's 2-minute series. Averaged absolute error in predicted occupancy on the y-axis. 120 testing points, equispaced within a 9-week testing interval, on the x-axis. Multilayer perceptron in red and decision tree ensemble in blue.

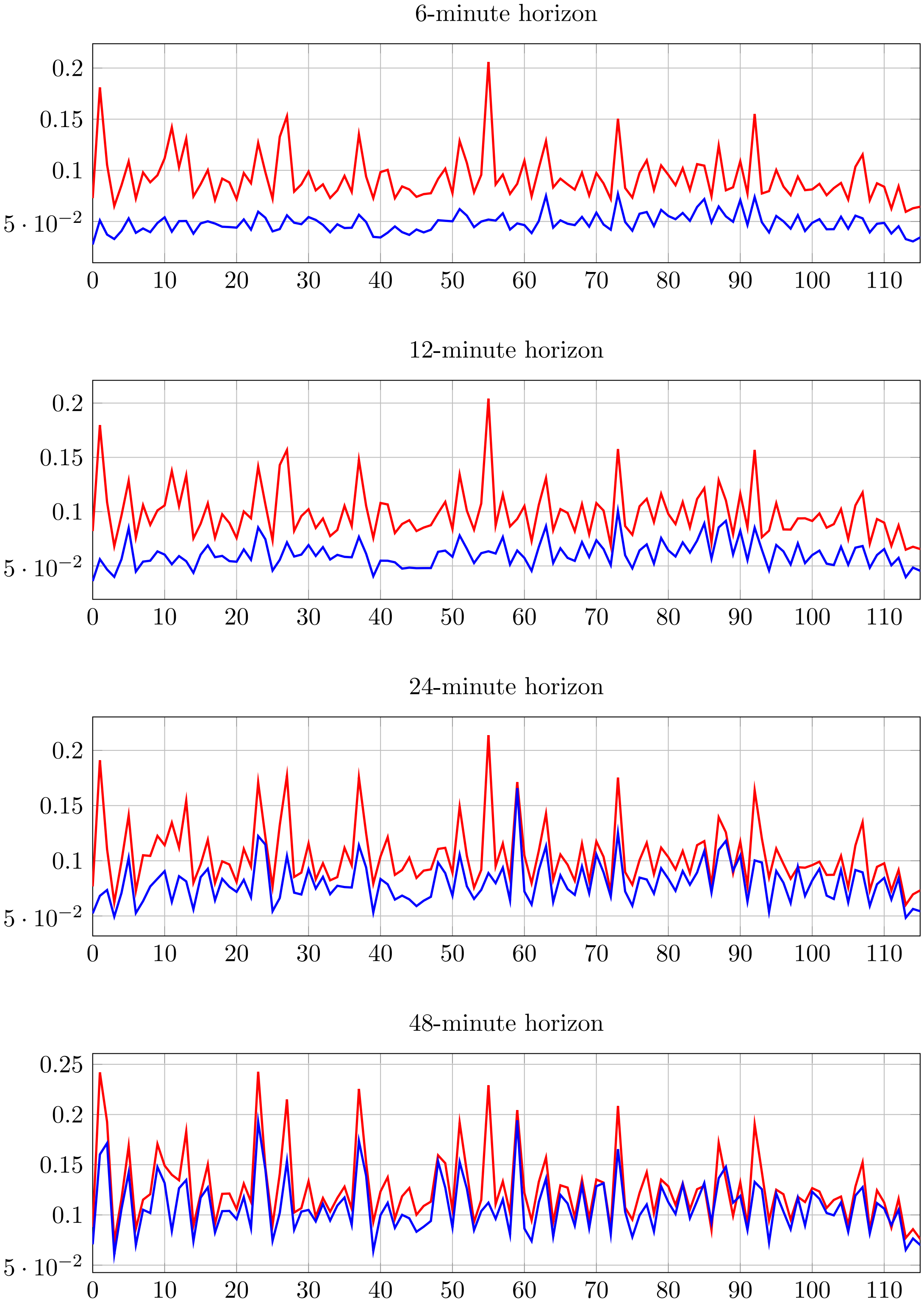


Figure 4.6: Predictor performance over time for London's 6-minute series. Averaged absolute error in predicted occupancy on the y-axis. 120 testing points, equispaced within a 9-week testing interval, on the x-axis. Multilayer perceptron in red and decision tree ensemble in blue.

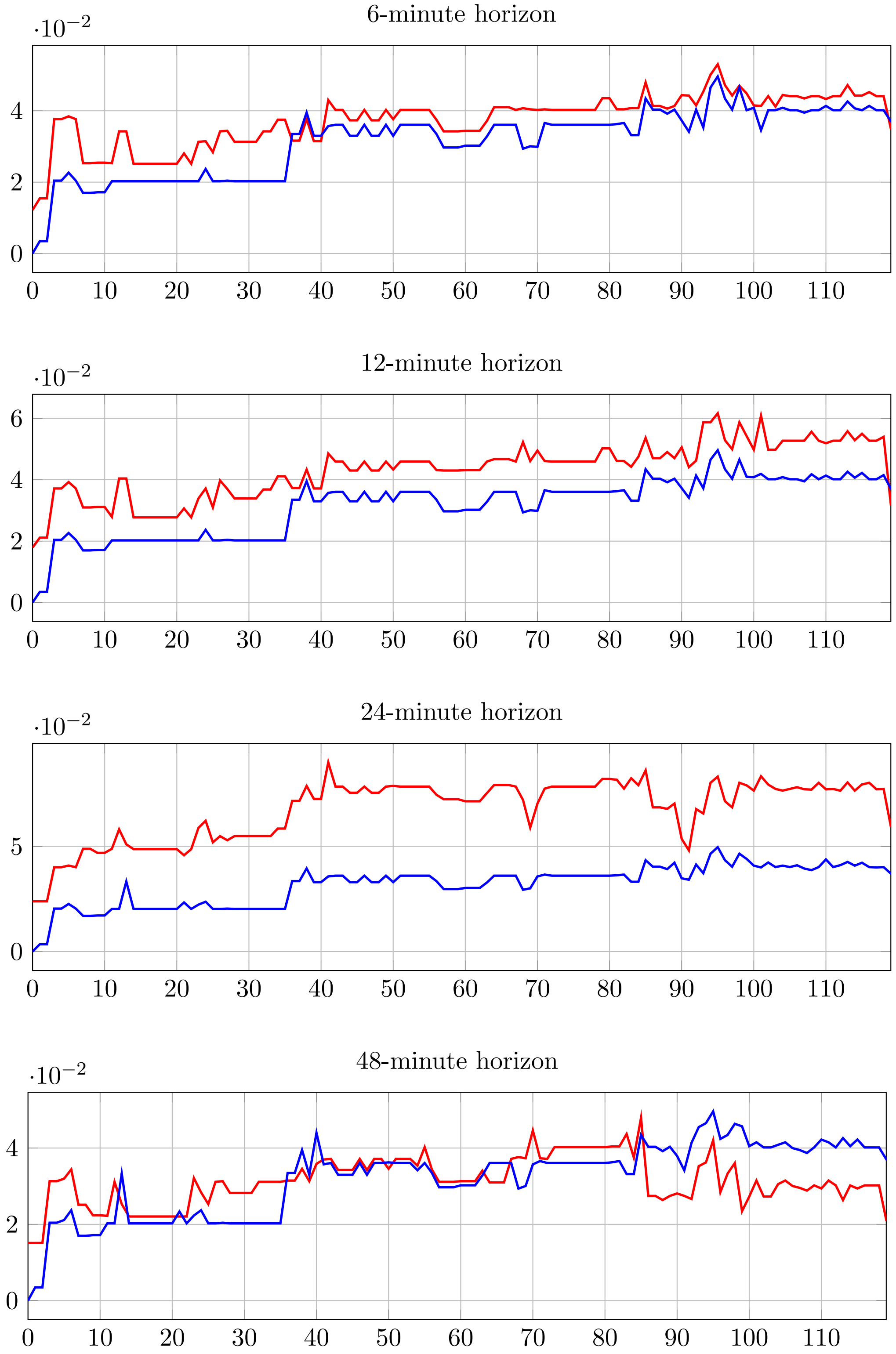


Figure 4.7: Predictor performance over time for Siracusa's 2-minute series. Averaged absolute error in predicted occupancy on the y-axis. 120 testing points, equispaced within a 9-week testing interval, on the x-axis. Multilayer perceptron in red and decision tree ensemble in blue.

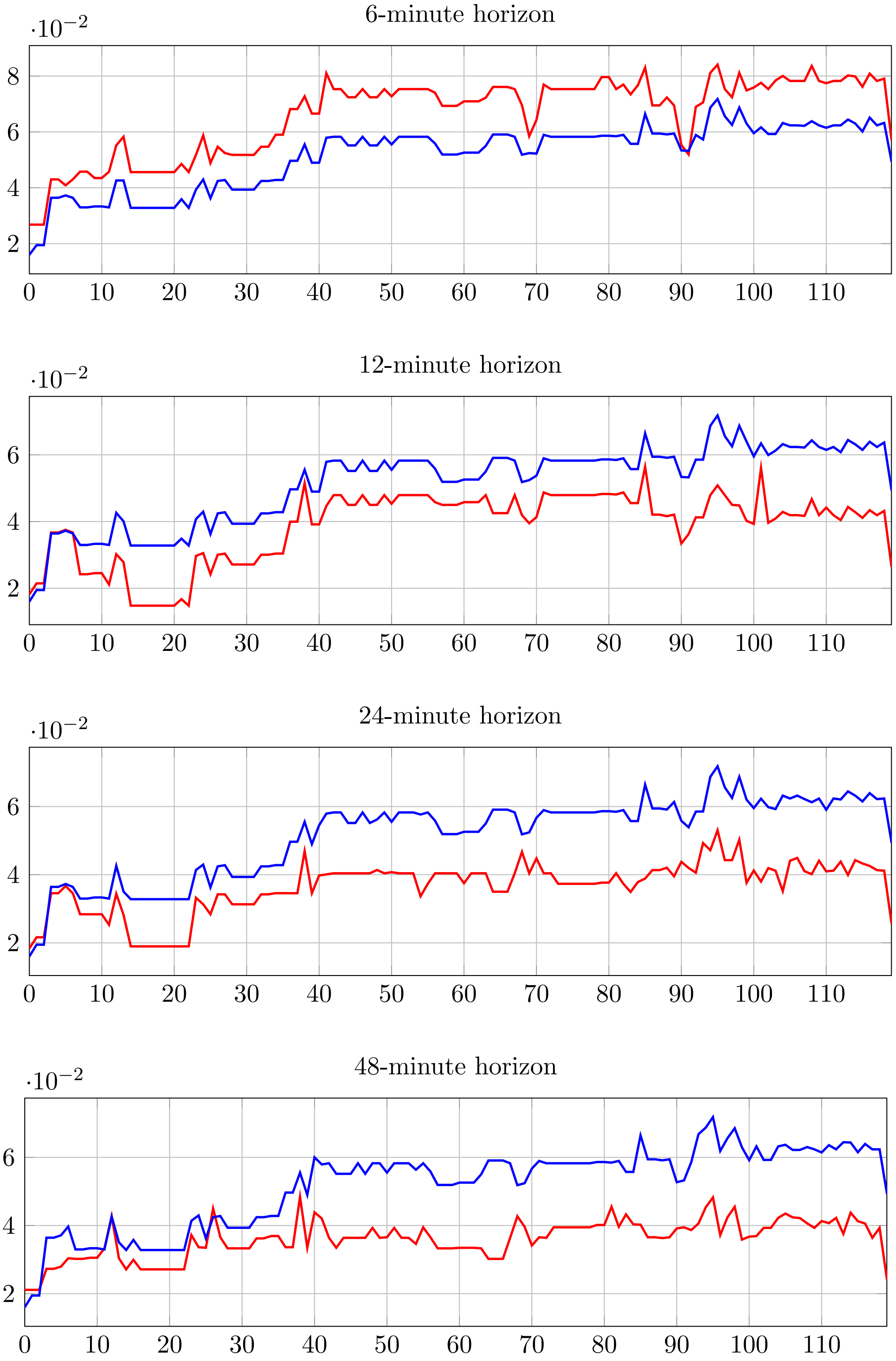


Figure 4.8: Predictor performance over time for Siracusa's 6-minute series. Averaged absolute error in predicted occupancy on the y-axis. 120 testing points, equispaced within a 9-week testing interval, on the x-axis. Multilayer perceptron in red and decision tree ensemble in blue.

Chapter 5

Conclusions & Future Work

We present an overview of our activities and main analytical results in §5.1 and §5.2 respectively. We discuss limitations and present some directions for future work in §5.3. Finally, our original scientific contributions are highlighted in §5.4.

5.1 Summary of Activities

We implemented a preprocessing pipeline to clean the observation data and ensure that data from different systems were transformed appropriately to enable cross-city comparisons (§2.3). We conducted an analysis of the system-wide occupancy series, looking at the average daily behaviour of stations at each system (§2.4).

We built upon previous work to develop a general methodology to detect naturally-occurring behavioural clusters of stations, and developed a simple, general heuristic for determining the number of clusters (§3.1). We developed three novel notions of station behaviour, applied our clustering methodology with appropriate metrics for each, and analysed the results (§3.2, §3.3, §3.4).

We framed the problem of predicting the future behaviour of a bike station as a univariate time series forecasting problem and developed an evaluation framework for models (§4.1, §4.2). We developed four kinds of predictive models, choosing parameters optimised against our evaluation testbed and based on auxiliary studies (§4.3). We trained our models and evaluated their performance based on the forecast horizon as well as their proximity to the training data (§4.4, §4.5, §4.6).

5.2 Summary of Analytical Results

In general, the usage of bike sharing systems varies greatly between weekdays and weekends. Work commuters have an observable presence on weekdays in the larger systems, but not in the smaller systems or American systems, suggesting a greater proportion of casual use (§2.4).

In terms of the rise and fall of their occupancy over the course of a typical day, bike stations around the world generally fall into one of three simple categories: morning-source-daytime-sink, morning-sink-daytime-source, and "flat" (§3.2).

With respect to borrow/return activity levels, stations generally belong into one of three major categories: 20% turnover, 90% turnover, and 50% turnover. The greatest proportion of stations fall into the first category, suggesting that stations are either systematically underutilised or oversized (§3.3).

Stations do not fall into neat categories according to their occupancy distribution. However the majority of stations exhibit a rough uniform distribution, with a slight tendency towards being emptier (§3.4).

By our first two notions of behaviour, heterogeneity in station behaviour is only present in the largest of systems. Not only do stations in smaller systems behave like most other stations within the same system, but all of the smaller systems appear to behave similar to each other, suggesting significant transferable knowledge between them. By our third notion of behaviour, this phenomenon is reversed.

The time series for the vast majority of bike stations, when viewed in sliding windows of up to 48 minutes, appear largely static. Under these circumstances, multilayer perceptron and decision tree ensemble predictors outperform a random benchmark, but the best results are obtained using a simple static model (§4.5).

Station behaviour in large systems is more stochastic, or at least more variable over time, than station behaviour in small systems. Under these circumstances, the performance of those models which are trained on a fixed training set does not deteriorate with decreasing proximity of the test example from the training set in large systems, but does deteriorate in small systems (§4.6).

5.3 Limitations and Future Work

A preliminary analysis of the maps in Chapter 3 provided evidence for the idea that heterogeneity in station behaviour is a function of the size of the system. In future work, it would be useful to quantitatively substantiate this claim. For example, it might be possible to profile systems based on the distribution of the clusters among their stations.

We only conducted our clustering study on highly aggregated pictures of station behaviour that spanned the entire length of the period under study. It would be interesting in future work to perform clustering over shorter spans of time, perhaps weekly sliding windows. This would allow us to see whether stations change clusters over time, and whether there are any smaller, ephemeral clusters that emerge and disappear during periods of unusual activity.

We analysed our time series data purely within its own context. In future work, it would be beneficial to consider other datasets in parallel. For example:

- A study of the concurrent weather data would reveal how adverse or favourable conditions impact system usage.
- A study of the concurrent usage of other transport systems (e.g., subways or public buses) would reveal how transport systems interact and the degree to which they complement each other.
- Investigating topography, major venues, footfall, and other aspects of the area might explain unusual or unique station behaviour. In particular, we uncovered in §3.4 that some stations were exceptionally well-utilised with respect to their capacity. These datasets might explain their success and suggest how less-utilised stations might be moved into this cluster.

The structure of our multilayer perceptron (§4.3) was determined manually, and we did not explore the usage of multiple hidden layers. In future work it might be useful to automate the construction of the network through a technique such as cascade correlation [29].

The RMSE performance of our predictors on each station yields a measure of the "predictability" of the station. Just as it might be possible to profile systems based on the distribution of behavioural clusters, it may also be possible to profile systems based on the distribution of their stations' predictabilities.

Our static predictor performed best when the results were averaged across all stations in a system at every test point in the dataset §4.5. As we speculated, this is likely due to the fact that the majority of stations have sparse activity, and even stations with moderate levels of activity usually experience the bulk of their activity concentrated at certain times of day. In future work, it would be useful to identify only those stations where activity is not sparse, and then focus on those times of day where the bulk of their activity is concentrated. This would yield models that were much more useful for the simulation/prediction purposes originally envisioned.

Finally, due to time constraints, we were unable to maintain constant temporal distance between the training and test data. Doing so would require separate training datasets and models for each evaluation point, and orders of magnitude greater computational time. Nonetheless, the resultant deterioration of the predictors' performance afforded us some insights into the relative variability of small vs large systems (§4.6). It would be useful in the future to train models on separate training sets for each evaluation point (or at least smaller groups of evaluation points), maintaining a constant temporal distance between the test data and the training data, which would potentially improve the performance of our multilayer perceptron and decision tree ensemble models in small systems.

5.4 Summary of Contributions

Previous research has found that the usage of bike sharing systems varies between weekdays and weekends in the systems of London and Barcelona. We provided evidence that this may be true of bike sharing systems in general (§2.4).

We built on previous work on clustering by occupancy series by adding mean normalisation, allowing us to get a clearer picture of the global categories into which bicycle stations fall (§3.2). This methodology can be used to inform redistribution vehicle routes and times across cities. In fact, both of the non-"flat" clusters exhibit sharp changes in their occupancy levels between 9:00AM and

12:00PM. This suggests that a minimal redistribution scheme, requiring only a single daily redistribution occurring within this time window, would still be very effective. Maps of these clusters can be used to determine rough directions for redistribution vehicle routes, as we have demonstrated in Figures 3.3 and 3.4.

Our methodology for clustering on the borrow/return activity levels of stations is useful for deciding whether to add or remove stations (§3.3). Physical groups of low-activity stations suggest the removal of one or more stations within those groups. Similarly, lone high-activity stations suggest the addition of one or more auxiliary stations to reduce the load and improve resistance to sudden spikes.

Our methodology for clustering on occupancy distribution (§3.4) is a novel addition to the study of bike sharing systems, and allows for the direct assessment of the demand for bicycles or vacancies at a particular station. This allows system administrators to adjust the capacity of the station and to adjust the number of bicycles added/removed during each pass of a redistribution vehicle so that demand is unlikely to outstrip supply.

While our more sophisticated models were not able to outperform the simple static model (§4.5), the multilayer perceptron did nonetheless achieve less than 10% root-mean-square error in most cases. Previous work on the use of neural networks for time-series forecasting is scarce; to this body of work we add our study. We introduce the use of the partial autocorrelation function, traditionally used to select the parameters for ARIMA[1] models, as a heuristic for setting the length of the input vector for a multilayer perceptron.

Finally, throughout our clustering and forecasting analyses, we found evidence to support the new idea that heterogeneity and variability in station behaviour are functions of the size of the system: the larger the system, the greater the spread of station types; and the larger the system, the greater the variability of an individual station's behaviour over time.

[1] AutoRegressive Integrated Moving Average

Bibliography

[1] New Figures Show Barclays Cycle Hire is a Convenient, Fun and Fast Way to Discover London. `http://www.tfl.gov.uk/corporate/media/newscentre/archive/22484.aspx` (published January 2012).

[2] Pedal Power: the Cycle Hire Scheme and Cycle Superhighways. *Greater London Authority*, November 2010. available at `http://www.london.gov.uk/sites/default/files/FINAL%20REPORT.pdf` (as of March 19, 2015).

[3] Obis Project. `http://www.obisproject.com` (as of March 19, 2015).

[4] Jon Froehlich, Joachim Neumann, and Nuria Oliver. Sensing and Predicting the Pulse of the City through Shared Bicycling. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1420–1426. Morgan Kaufmann Publishers Inc., 2009.

[5] Neal Lathia, Saniul Ahmed, and Licia Capra. Measuring the Impact of Opening the London Shared Bicycle Scheme to Casual Users. *Elsevier Transportation Research – Part C (Emerging Technologies)*, 22:88, 2012.

[6] Marcel C Guenther and Jeremy T Bradley. Journey Data Based Arrival Forecasting for Bicycle Hire Schemes – Preprint. 2013.

[7] Andry Randriamanamihaga, Etienne Côme, Latifa Oukhellou, and Gérard Govaert. Clustering the Vélib' Origin-Destinations Flows by Means of Poisson Mixture Models – Draft. 2013.

[8] Jia Shu, Mabel C Chou, Qizhang Liu, Chung-Piaw Teo, and I-Lin Wang. Models for Effective Deployment and Redistribution of Bicycles within Public Bicycle-Sharing Systems. *Submitted to Operations Research*, 2011.

[9] Jyh-Horng Lin and Ti-Chieh Chou. A Geo-Aware and VRP-Based Public Bicycle Redistribution System. *International Journal of Vehicular Technology*, 2012.

[10] CC Robusto. The Cosine-Haversine Formula. *The American Mathematical Monthly*, 64(1):38–40, 1957.

[11] In *Global Positioning System Standard Positioning Service Performance Standard (4th Edition)*. Department of Defense, United States of America, September 2008.

[12] Richard O Duda, Peter E Hart, and David G Stork. *Pattern Classification and Scene Analysis: Part 1, Pattern Classification*. Wiley, 2000.

[13] Louis Legendre and L Legrendre. *Numerical Ecology: Developments in Environmental Modelling*. Elsevier Science & Technology, 1998.

[14] Mingjin Yan and Keying Ye. Determining the Number of Clusters Using the Weighted Gap Statistic. *Biometrics*, 63(4):1031–1037, 2007.

[15] Stan Salvador and Philip Chan. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. In *16th IEEE International Conference on cTools with Artificial Intelligence (ICTAI 2004)*, pages 576–584. IEEE, 2004.

[16] Donald Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD Workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[17] Hiroaki Sakoe and Seibi Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[18] Ernst Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271, 1909.

[19] Anil Bhattacharyya. On a Measure of Divergence Between Two Statistical Populations Defined by their Probability Distributions. *Bulletin of Calcutta Mathematical Society*, 35(99-109):4, 1943.

[20] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-Time Tracking of Non-Rigid Objects Using Mean Shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149. IEEE, 2000.

[21] Thomas Kailath. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *Communication Technology, IEEE Transactions on*, 15(1):52–60, 1967.

[22] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[23] Sven F Crone, Michèle Hibon, and Konstantinos Nikolopoulos. Advances in Forecasting with Neural Networks? Empirical Evidence from the NN3 Competition on Time Series Prediction. *International Journal of Forecasting*, 27(3):635–660, 2011.

[24] Brian D Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 2008.

[25] N Levinson. The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction. *Journal of Mathematical Physics*, 25:261–278, 1947.

[26] James Durbin. The Fitting of Time-Series Models. *Revue de l'Institut International de Statistique*, pages 233–244, 1960.

[27] Martin Riedmiller and Heinrich Braun. RPROP – A Fast Adaptive Learning Algorithm, Technical Report. In *Proceedings of ISCIS VII (To Appear)*. Universität Karlsruhe, 1992.

[28] Martin Riedmiller and Heinrich Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.

[29] Scott E Fahlman and Christian Lebiere. The Cascade-Correlation Learning Architecture. In *Advances in Neural Information Processing Systems 2*, 1990.

[30] Kenneth Levenberg. A Method for the Solution of Certain Problems in Least Squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[31] Donald W Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[32] John Ross Quinlan. *C4. 5: Programs for Machine Learning*, volume 1. Morgan Kaufmann, 1993.

[33] Solomon Kullback and Richard A Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[34] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.