Machine Learning for Real-time Demand Forecasting

by

Runmin Xu

Bachelor of Science in Automation
Tsinghua University, Beijing, China, 2012

Submitted to the Department of Civil and Environmental Engineering and the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Master of Science in Transportation

and

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

© 2015 Massachusetts Institute of Technology. All rights reserved.

ARCHIVES

MASSACHUSETTS INSTITUTE OF TECHNOLOLGY

JUL 02 2015

LIBRARIES

Author ..... Signature redacted .....
Department of Civil and Environmental Engineering
Department of Electrical Engineering and Computer Science
May 21, 2015

Certified by ..... Signature redacted .....
Una-May O'Reilly
Principal Research Scientist of CSAIL
Thesis Supervisor

Certified by ..... Signature redacted .....
Marta C. Gonzalez
Assistant Professor of Civil and Environmental Engineering
Thesis Reader

Accepted by ..... Signature redacted .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Accepted by ..... Signature redacted .....
Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Departmental Committee for Graduate Student

Machine Learning for Real-time Demand Forecasting

by

Runmin Xu

Submitted to the Department of Civil and Environmental Engineering
and the Department of Electrical Engineering and Computer Science
on May 21, 2015, in partial fulfillment of the
requirements for the degrees of
Master of Science in Transportation
and
Master of Science in Electrical Engineering and Computer Science

Abstract

For a taxi company, the capability to forecast taxi demand distribution in advance provides valuable decision supports. This thesis studies real-time forecasting system of spatiotemporal taxi demand based on machine learning approaches. Traditional researches usually examine a couple of candidate models by setting up an evaluation metric and testing the overall forecasting performance of each model, finally the best model is selected. However, the best model might be changing from time to time, since the taxi demand patterns are sensitive to the dynamic factors such as date, time, weather, events and so on.

In this thesis, we first study range searching techniques and their applications to taxi data modeling as a foundation for further research. Then we discuss machine learning approaches to forecast taxi demand, in which the pros and cons of each proposed candidate model are analyzed. Beyond single models, we build a five-phase ensemble estimator that makes several single models work together in order to improve the forecasting accuracy. Finally, all the forecasting approaches are evaluated in a case study over rich taxi records of New York City. Experiments are conducted to simulate the operation of real-time forecasting system. Results prove that multi-model ensemble estimators do produce better forecasting performances than single models.

Thesis Supervisor: Una-May O'Reilly
Title: Principal Research Scientist of CSAIL

Thesis Reader: Marta C. Gonzalez
Title: Assistant Professor of Civil and Environmental Engineering

Acknowledgments

I would like to express my immerse gratitude to my advisors, colleagues, friends and families who contributed to this thesis. First of all, I would like to thank my thesis supervisor, Dr. Una-May O'Reilly, for her excellent guidance and unwavering support all the time. Her insightful advices greatly helped me during the research and writing of this thesis.

I am indebted to my academic advisor and administrators, Professor Marta Gonzalez. Marta is always patient, enthusiastic and caring for students. Her course, 1.204-Computer Modeling, gave me the first idea of this thesis. Besides valuable guidance on my academy and research, more importantly, her understanding and support during my hardest time at MIT helped me overcome the difficulties and go through the rewarding journey.

Thanks to numerous people who indirectly contributed to this thesis. They are my colleagues from the MST and EECS programs; my students from the classes I assisted in teaching; the departmental administrators, Kris Kipp, Janet Fisher and Kiley Clapper, as well as all my friends who have always been there.

My deepest gratitude and appreciation go to my families. I am lucky to have my parents, Longquan Xu and Meiyu Chen, who brought me up and taught me honesty, kindness and responsibility. I also want to thank my beautiful wife Yihan Xu for standing beside me throughout the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Revolution of taxi services

Taxi is an important transportation mode that provides useful and convenient services for passengers as a part of transportation system. Taxi service started in the end of 19th century and has become one of the indispensable travel modes nowadays. Almost every taxi company has long been exploring solutions to save the costs of daily operations such as fuel costs and depreciation rates, as well as to provide better service for customers. Obviously, the key is to optimize the demand-supply metric of drivers and passengers, in other words, to reduce the empty-loaded rates of a taxi and the time that a passenger spends on waiting for a taxi.

Benefited from the development of GPS techniques, the real-time vehicle location systems attracted attentions of large amounts of taxi companies and researchers in last decades. By collecting rich spatiotemporal information, well developed systems are able to provide strategy supports such as reasonable taxi dispatching and optimized route finding, with what the efficiency of taxi services can be significantly improved.

In recently years, traditional taxi industry is under revolution. Fast-growing start-tups like Uber and Lyft have dramatically changed the behaviors people take taxis. Instead of meeting by chance, the taxi drivers and passengers are able to make connections via smartphones, reducing the financial and time costs for both sides. With the huge amounts of taxi services data, researchers have more abilities to explore

the underlying nature of the taxi pattern, offering innovative solutions to drivers, passengers and taxi companies.

1.2 Problem statement and thesis motivations

As a taxi company or a taxi driver, the capability to estimate how many customers will need taxi services(and where are they) in advance is one of the most attractive *magic*.

Traditional researches usually examine a couple of mathematical models to forecast the demand, and compare them by setting up an evaluation criteria and looking at the performance of each model. Those performances are typically measured upon the entire datasets over a large-scale area. However, the taxi demand patterns of different locations can be much different because of the varies landscapes, population densities and so on.

The goal of this thesis is to propose a design for real-time adaptive system that forecast taxi demand around a specific location at hourly interval, which offers decision support for the taxi drivers as well as strategy support for taxi companies. More specifically, machine learning models are built, validated, tested over the taxi data in New York City in 2012 and 2013. We are going to analyze the accuracy of each model under different conditions, and propose ensemble methods to have them work together to provide better performances.

The objective of machine learning models in this thesis is to forecast the taxi demand in the future of specific locations, as shown in figure 1-1.

At the end of each hour, we forecast hourly taxi demand in the next 12 hours. A specific model is generated for each hour lead, which means 12 different models are built totally: 1-hour-lead model, 2-hour-lead model, all the way to 12-hour-lead model. Starting from using 12 hours lag for modeling, each model uses a one hour longer lag for each added hour of lead. For example, 12-hour lag for 1-hour-lead model, 13-hour lag for 2-hour-lead model and so on.

In addition to historical data providing spatiotemporal pickup info, additional

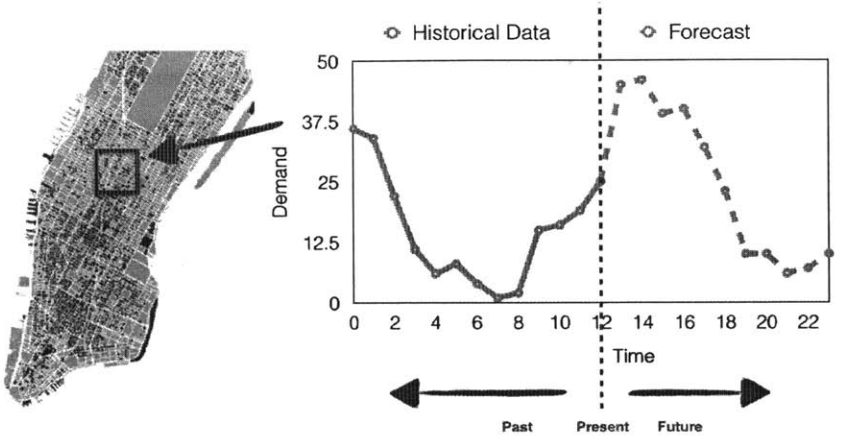

Figure 1-1: Objective of machine learning approaches

features are used for building the models. Calendar(e.g., day of the week, holidays), weather, demography, events and public transit data are used and analyzed in this thesis.

1.3 Thesis outline

This thesis is organized as follows:

Chapter 2 overviews the theory of time series and their applications to the taxi demand forecasting problem. Stochastic models and machine learning approaches are introduced in section 2.2 and section 2.3. Four multi-step forecasting strategies are reviewed and compared in section 2.4. Moreover, previous predictive models for taxi demand are described in section 2.5.

Chapter 3 introduces three data structures for range searching techniques. Range searching is used to preprocess data and extract subset of data within a given range, e.g., query taxi records around Time Square from Jan 1st, 2013 to Jan 7th, 2013. These techniques play an important role in a forecasting system when we frequently select target locations and time periods for taxi demand analysis. Kd-trees, range trees and layered range trees are illustrated and their efficiencies are compared in this chapter.

In Chapter 4, analysis of machine learning approaches to forecast taxi demand is provided. There are many variants of the basic ML models. In this chapter, we focus on the basic model and specifically their applications to the taxi demand forecasting problem. K-nearest neighbors, artificial neural networks, support vector machines, decision trees and random forests are discussed in this chapter.

In Chapter 5, ensemble estimator of distinct single ML models is proposed. We first propose three types of ensemble methods of multiple distinct models in section 5.1. Then we build a five-phase ensemble estimator in section 5.2, including single candidate model training, parameters optimization, ensemble construction, ensemble selection and forecasting.

Chapter 6 talks about a case study which the models are evaluated with the taxi records of New York City. This chapter contains data introduction in section 6.1, experiment design in section 6.2, performance measures in section 6.3 and key findings in section 6.4.

Finally, we would come to the conclusion in chapter 7.

Chapter 2

Literature Review

This chapter reviews the time series modeling in the aspects of stochastic models and machine learning approaches, as well as multi-step forecasting strategies which are practically useful for building a forecasting system. In section 2.1, the basics of time series analysis are reviewed because forecasting taxi demand is essentially a time series problem. In the past decades, stochastic models and machine learning approaches were considered to be two fundamental methodologies to solve the time series forecasting problems, as introduced in section 2.2 and 2.3.

2.1 Time series analysis

Time series is typically measured over successive times, representing as a sequence of data points [5]. The measurements taken during an event in a time series are arranged in a proper chronological order. Basically there are two types of time series: continuous and discrete. In a continuous time series observations are measured at every instance of time, whereas a discrete time series contains observations measured at discrete points of time. Usually in a discrete time series the consecutive observations are recorded at equally spaced time intervals such as hourly, daily, monthly or yearly time separations. In general, to do further analysis, the data being observed in a discrete time series is assumed to be as a continuous variable using the real number scale [15].

Time series analysis fits a time series into a proper model. The procedure of fitting a time series to a proper model is termed as Time Series Analysis. Practically, parameters of the model are estimated using the known data values, which comprises models that attempt to analyze and understand the nature of the series. These models are useful for future simulation and forecasting after beings validated. A time series in general is assumed to be affected by four main components: trend, cyclical, seasonal and irregular components [23]. These four components can be extracted and separated from the observed data. Considering the effects of these four components, in general, addictive and multiplicative models are used for a time series decomposition. Additive model is based on the assumption that the four components of a time series observation are independent of each other. However, in multiplicative model, the four components can affect the others meaning they are not necessarily independent.

2.2 Stochastic models

A stochastic model is a tool for estimating probability distributions of potential outcomes, the application of which initially started in physics and is now being applied in finance, engineering, social sciences, etc. The selection of a proper model is extremely important as it reflects the underlying structure of the series, and more importantly, the fitted model is useful for future forecasting.

There are two widely used linear time series models: Autoregressive (AR) and Moving Average (MA) models [4, 15]. The AR and MA models are widely analyzed and used so will not be introduced in details here. An ARMA(p, q) model is a combination of AR(p) and MA(q) models and is particularly suitable for univariate time series modeling [18]. However, the ARMA models, described above can only be used for stationary time series data. In practice, many time series show non-stationary behavior, such as those related to business and socio-economic as well as those contain trend and seasonal patterns [13, 9]. Thus from application point of view, ARMA models are inadequate to properly describe non-stationary time series, which are frequently encountered in practice. For this reason the ARIMA model [28]

is proposed, which generalizes ARMA model to include the case of non-stationarity as well [15].

Besides the typical models above, more specific and varied models are proposed in literature. For instance, the Autoregressive Fractionally Integrated Moving Average (ARFIMA) [10] model generalizes ARMA and ARIMA models. For seasonal time series forecasting, a variation of ARIMA, the Seasonal Autoregressive Integrated Moving Average (SARIMA)[13] model is used. ARIMA model and its different variations are also broadly known as the Box-Jenkins models for the reason that they are based on the famous Box-Jenkins principle[43].

Linear models have drawn much attention because of their relative simplicity in understanding as well as implementation. However, it is not negligible that many practical time series show non-linear patterns. For example, non-linear models are appropriate for predicting volatility changes in economic and financial time series, as mentioned by R. Parrelli in[33]. Considering these facts, various non-linear models have been suggested in literature, such as the Threshold Autoregressive (TAR) model [38], the Autoregressive Conditional Heteroskedasticity (ARCH) model and its variations like Generalized ARCH (GARCH) [32], the Non-linear Autoregressive (NAR) model [44] and so on.

2.3 Machine learning approaches

Machine learning models have been established as serious contenders to classical statistical models in the area of forecasting in the last decade. Subsequently, the concept is extended to other models, such as support vector machines, decision trees, and others, that are collectively called machine learning models [2]. Some of these models are developed based on the early statistics models [14]. Huge advances have been made in this field in the past years, both in the amount and variations of the models developed, as well as in the theoretical understanding of the models.

Machine learning approaches are widely used for building accurate forecasting models [31, 36]. Literature searches have found the previous works where the ma-

chine learning techniques are used in combinations with the econometrics models. The results from different techniques are integrated to obtain better forecasting accuracy [16]. Although many machine learning algorithms focus on solving classification problems, they also can be applied to regression problems, which is actually used in this thesis to forecast the taxi demand.

Artificial neural networks (ANN) approach has been suggested as a widely used technique for time series forecasting and it gained immense popularity in last few years. The basic objective of ANNs was to construct a model for simulating the intelligence of human brain into machine [19]. Although the development of ANN was mainly biologically motivated, but they have been applied in various areas, especially for classification and forecasting purposes [18]. Similar to the mechanics of a human brain, ANN tries to recognize essential patterns and regularities in the input data, learn from experience and then provide generalized results based on the previous knowledge. In the class of ANN, the most widely used ones in forecasting problems are multi-layer perceptrons, which use a single hidden layer feed forward network[42]. The model is characterized by a network of three layers connected by acyclic links: input, hidden and output layer.

A major breakthrough in the area of time series forecasting occurred with the development and improvement of support vector machines (SVM) [3, 8]. The initial aim of SVM was to solve pattern classification problems but afterwards they have been widely applied in many other fields such as function estimation, regression, and time series prediction problems [1]. The impressive characteristic of SVM is that it is intended for a better generalization of the training data. In SVM, instead of depending on whole data set, the solution usually only depends on a subset of the training data points, called the support vectors [25]. Furthermore, with the help of support vector kernels, the input points in SVM applications are usually mapped to a high dimensional feature space, which often generates good generalization outcomes. For this reason, the SVM methodology has become one of the well-known techniques in recent years, especially for time series forecasting problems. In addition, numbers of SVM forecasting models have been developed during the past few years. Two famous

time series forecasting models are Least-square SVM (LS-SVM) [34] and Dynamic Least-square SVM (LS- SVM) [1].

Ensemble learning is a more complicated approach that combines results from multiple learners and provide a *summarized* result. Ensemble learners are often used for classification [22] and regression [21] problems.

2.4 Multi-step forecasting

Multi-step forecasting is an important function of a demand forecasting system. Recursive, Direct and DirRec are three representative types of multi-step forecasting strategies.

Recursive strategy uses forecasted values of near future as inputs for the longer future forecasting. The function of Recursive is defined as:

$$\begin{aligned} \hat{y}_{t+1} &= f\left(y_t, y_{t-1}, \cdots, y_{t-d+1}\right) \\ \hat{y}_{t+2} &= f\left(\hat{y}_{t+1}, y_t, \cdots, y_{t-d+1}\right) \\ &\cdots \\ \hat{y}_{t+n} &= f\left(\hat{y}_{t+n-1}, \hat{y}_{t+n-2} \cdots, \hat{y}_{t-d+n}\right) \end{aligned} \tag{2.1}$$

where t is the current time, $\hat{y}_{t+1}$, $\hat{y}_{t+2}$, ..., $\hat{y}_{t+n}$ are the forecasted values, y_1, y_2, ..., y_t are the historical values, d is the dimension of inputs.

As shown in equation 2.1, the drawback of the recursive method is its sensitivity to the estimation errors. Errors are more and more used and accumulated when we get further forecasts in the future.

The Direct strategy is not prone to the problem of error accumulation, since it

does not use any predicted values to compute the forecasts:

$$\begin{aligned}
\hat{y}_{t+1} &= f_1\left(y_t, y_{t-1}, \cdots, y_{t-d_1+1}\right) \\
\hat{y}_{t+2} &= f_2\left(y_t, y_{t-1}, \cdots, y_{t-d_2+1}\right) \\
&\cdots \\
\hat{y}_{t+n} &= f_n\left(y_t, y_{t-1}, \cdots, y_{t-d_n+1}\right)
\end{aligned} \tag{2.2}$$

where t is the current time, $\hat{y}_{t+1}$, $\hat{y}_{t+2}$, ..., $\hat{y}_{t+n}$ are the forecasted values, y_1, y_2, ..., y_t are the historical values, f_1, f_2, ..., f_n are the models for each time step, d_1, d_2, ..., d_n are the dimension of inputs of each model.

Basically, Direct builds one model for each time step, and the models could have different input dimensions. Since it has to model the stochastic dependency between two distant series values, higher functional complexity is often required [38, 29]. Moreover, Direct strategy takes a large computational time since the number of models to learn is relevant to the size of the horizon.

The DirRec strategy [37] combines the principles and architectures underlying the Direct and the Recursive strategies. DirRec computes the forecasts with different models for every horizon. At each time step, it enlarges the set of inputs by adding variables with the forecasts of the previous step: as shown in equation 2.3.

$$\begin{aligned}
\hat{y}_{t+1} &= f_1\left(y_t, y_{t-1}, \cdots, y_{t-d_1+1}\right) \\
\hat{y}_{t+2} &= f_2\left(\underline{\hat{y}_{t+1}}, y_t, y_{t-1}, \cdots, y_{t-d_2+1}\right) \\
&\cdots \\
\hat{y}_{t+n} &= f_n\left(\underline{\hat{y}_{t+n-1}, \cdots, \hat{y}_{t+1}}, y_t, y_{t-1}, \cdots, y_{t-d_n+1}\right)
\end{aligned} \tag{2.3}$$

Table 2.1 summarizes the characteristics of the three strategies clearly. In this thesis, Direct strategy is selected to forecast taxi demand.

Table 2.1: Comparison of Recursive, Direct and DirRec

	Recursive	Direct	DirRec
Pros	Good for noise-free series	No error accumulation	Tradeoff between Recursive and Direct
Cons	Error accumulation	Independent assumption	Computational inefficiency
No. of Models	1	No. of steps	No. of steps
Computational Time	+	++	+++

2.5 Predictive model for taxi demand

In the last decade, GPS-location systems have attracted the attention of both researchers and companies due to the new type of information available. Specifically, the location-aware sensors and the information transmitted increases are tracking human behavior and they can be used collaboratively to reveal their mobility patterns. Trains [7], Buses [20] and Taxi Networks [26] are already successfully exploring these traces. Gonzalez et. al [12] uncovered the spatiotemporal regularity of human mobility, which were demonstrated in other activities such as electricity load [11] or freeway traffic flow [39].

The fuel cost has been decreasing the profit of both taxi companies and drivers. It causes an unbalanced relationship between passenger demand and the number of running taxis, as a result, it reduces the profits made by companies as well as the passenger satisfaction levels [35]. In recent years, Uber and Lyft become two popular taxi companies using location based services which significantly reduce the waiting time for both passengers and drivers. Wong presented a relevant mathematical model to express this need for equilibrium in distinct contexts [40]. An equilibrium fault may result in one of two scenarios: (1) excess of vacant vehicles and excessive competition; (2) larger waiting times for passengers and lower taxi reliability [30].

The taxi driver mobility intelligence is an important factor to maximize both profit and reliability within every possible scenario. Knowledge about where the services

will actually emerge can be an advantage for the drivers. The GPS historical data is one of the main variables of this topic because it can reveal underlying running mobility patterns. This kind of data represents a new opportunity to learn/predict relevant patterns while the network is operating in real-time.

Several researches have already explored this kind of data successfully with distinct applications like modeling the spatiotemporal structure of taxi services [27], smart driving [41] and building intelligent passenger-finding strategies [24]. Despite their useful insights, the majority of techniques reported are based on offline test, discarding some of the main advantages of the real-time signal. In other words, they do not provide any live information about the passenger location or the best place to pick up passengers in real time, while the GPS data is essentially a data stream. One of the recent advances on this topic was presented by Moreira Matias [30], where a discrete time series framework is proposed to forecast the service demand. This framework handles three distinct types of memory range: short term, mid term and long term [6, 17].

2.6 Summary

In this chapter, we have reviewed stochastic and machine learning models, especially for the fields of time series modeling and forecasting. Stochastic models have been widely used and optimized for the finance, engineering and social science problems. Machine learning approaches are considered as superior methods for the forecasting problems, in which neural networks and support vector machines are two classical ones especially for time-series observations. By applying these mathematical forecasting techniques into the real-world taxi industry, we found that the cost of fuel, waiting time of passengers and vacant taxis can all be significantly reduced if we are able to predict taxi demand accurately. However, beyond these promising researches, there are some remaining challenges. Obviously, the information of driver-passenger demand metric is always changing. What models or algorithms should we use to forecast the demand in real time? What if the chosen model doesn't work? How

to build a robust and reliable system that can always find or even build the most appropriate models? In this thesis, we are going to solve these problems.

Chapter 3

Range Searching

In this chapter, range searching algorithms for time-series taxi records are discussed. This is an useful tool to extract information we need from the data, such as records in specific time ranges and/or areas.

Orthogonal range searching problem is one of the fundamental and cutting-edge topics of computational geometry, especially when advanced data structures are considered. As figure 3-1 shows, a pair of x coordinates, a pair of y coordinates and a time range are given to query data. After getting the reported taxi records, our system counts the pick-up demand and uses it as a feature in modeling.

Figure 3-1: Example of range searching input

In this thesis, we implement data structures for orthogonal range searching for the spatiotemporal taxi data and analyze their practical efficiencies.

The range searching task in this thesis is defined as

$$data = query\,(Longitude, Latitude, Time) \tag{3.1}$$

where $Longitude$=[Lon_{lower}, Lon_{upper}] is the range of longitude, $Latitude$=[Lat_{lower}, Lat_{upper}] is the range of latitude, $Time$=[$Time_{lower}$, $Time_{upper}$] is the range of time, *data* is specifically the taxi records reported.

Three data structures for range searching are applied and analyzed in this chapter: Kd-trees, range trees and layered range trees.

3.1 Kd-trees

A Kd-tree is a space-partitioning data structure for organizing data in k-dimension spaces. Let's take the Figure 3-2 as an example to see how points are split to construct Kd-trees.

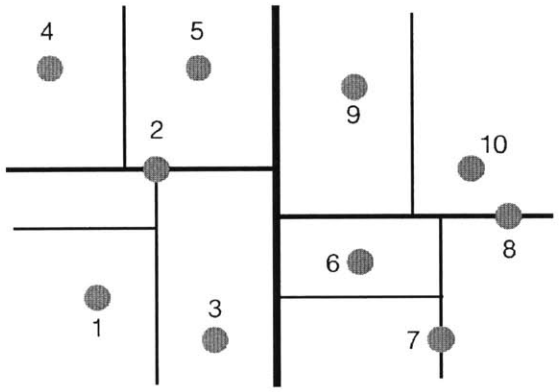

Figure 3-2: An Example of how points are split to construct Kd-trees

In short, points are split based the x-y coordinates to construct a Kd-trees.

1. Equally split the points by their x coordinates

2. For each 2nd-level subset, equally split the points by their y coordinates

3. For each 3rd-level subset, equally split the points by their x coordinates

4. Repeat step 2 and step 3 until each point has been assigned to a specific area

Median finding algorithms may be used to equally split the points. After all, the process leads to a balanced Kd-tree, as shown in Figure 3-3.

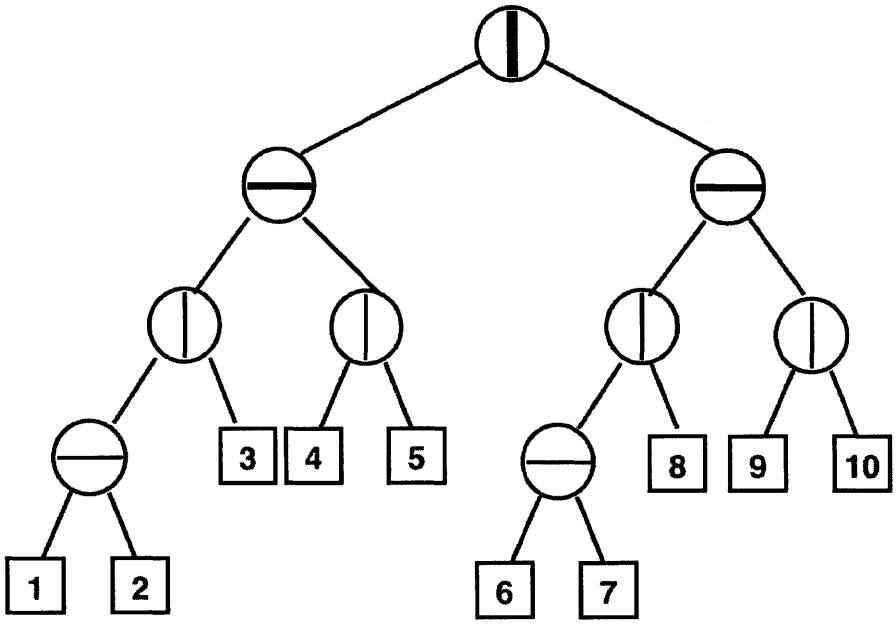

Figure 3-3: An example of constructed Kd-trees

The query process is operating just as Binary Search Tree(BST) until finding the contained area or reaching the leaf. Regarding the computational efficiency, Kd-trees needs $O(n)$ spaces and $O(n^{1-1/d}+k)$ query time, where d is the dimension and k is number of reported points. In this thesis, the task is to find taxi pickup records within specific time and location ranges, so that the data structure is a three-dimensional Kd-tree: the ranges of longitude, latitude and time. Therefore, the query efficiency of Kd-trees in this case is $O(n^{2/3}+k)$.

3.2 Range trees

A range tree is a tree data structure which enables points inside a given range to be reported efficiently. A range tree on a set of one-dimensional points is a balanced binary search tree on those points. Moreover, a range tree on a set of points in d dimensions is a recursively defined multi-level binary search tree.

Let's also take two-dimensional points (x, y) to explain the construction process of a range tree.

1. Construct a Binary Search Tree(BST) according to the x-coordinates. After this step, all the data points are at the leaves of this BST.

2. For every x-node, a sub-BST is constructed according to the y-coordinates of the leaves in the subtree below the x-node.

Figure 3-4 shows the idea of the range trees construction.

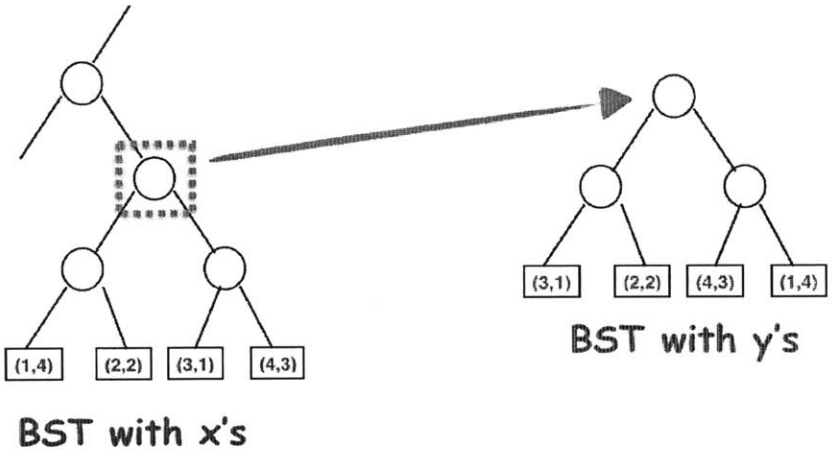

Figure 3-4: A simple example of Range Trees

To query data, we firstly go through the x-BST and then the y-BSTs from the selected x-node to find the points within the given range. Extending the two-dimensional points into d dimensions, the space complexity is $O(n \log^{d-1} n)$ to store the BSTs, and the query efficiency is $O(\log^d n + k)$, where k is number of reported points. In this thesis, three-dimensional range searching is the task so that the query efficiency of

range trees among taxi trips is $O(\log^3 n + k)$, which shows significant improvement compared with the Kd-trees before.

3.3 Layered range trees

The query efficiency of range trees can be further improved with fractional cascading.

- Instead of constructing a y-coordinates subtree for every x-node, a sorted array based on y-coordinates is used.
- Links are created between layers in order to narrow the range of y coordinates.
- As a result, 1 power of $\log n$ is eliminated regarding the query efficiency.

Figure 3-5 shows the general ideas of constructing layered range trees.

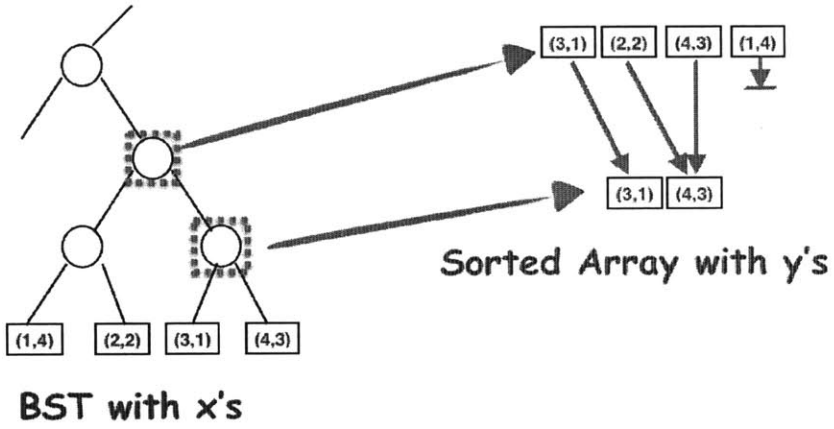

Figure 3-5: A simple example of layered range trees

If we extend the methods to d dimensions, we would see that layered range trees can store the data in $O(n \log^{d-1} n)$ space and query in $O(\log^{d-1} n + k)$ time, meaning $O(n \log^2 n)$ space and $O(\log^2 n + k)$ query time for the taxi data.

3.4 Comparative analysis

In theory, Kd-trees require least spaces $O(n)$ but largest query time $O(n^{1-1/d} + k)$, range trees and layered range trees require smaller query time but larger spaces

Table 3.1: Complexity analysis of data structures for Range Searching

Complexity	Kd-trees	Range trees	Layered range trees
Storage	$O(n)$	$O(n\log^{d-1} n)$	$O(n\log^{d-1} n)$
Preprocess	$O(n\log n)$	$O(n\log^{d-1} n)$	$O(n\log^{d-1} n)$
Query	$O(n^{1-1/d}+k)$	$O(\log^{d} n+k)$	$O(\log^{d-1} n+k)$

$O(n\log^{d-1} n)$ than Kd-trees, as Table 3.1 shows. Comparing range trees and layered range trees, the query efficiencies differ by $\log n$, which should be reflected in term of running time in experiments when the data size goes larger.

In order to understand the algorithms' efficiencies practically, experiments are conducted using the NYC taxi data. More details about the dataset will be introduced in chapter 6. The programing language of this data structure is C++. The test runs on Aliyun ECS server, with Ubuntu operating system, 16 cores and 64GB memory. Kd-trees is built independently. Large range trees are developed on top of range trees. We focus on testing the query time of the three data structures.

10,000 random queries are generated and the total querying time for each data structure is recorded in Table 3.2. There are limitations of the randomly generated ranges: the side length of the targeted area must be no more than 1000 meters, and the time range should be no more than 3 days. Longitude range from -74.279784 to -73.694440, latitude ranges from 40.486913 to 40.935435 and time ranges from Jan 1, 2012 to Dec 31,2013.

For each query, we do not only return the number of records inside a given range, but also return the contents of the qualified records.

As we can see from the tests results in Table 3.2, the query time of these three data structures are different, however, the differences are smaller than we expected based on the complexity comparison. Looking back to the complexity analysis, we observe that all of the three complexities are related to the number of points to report. So when there are a huge number of points reported, k will dominate the complexity, which makes no obvious difference among the query time of the three data structures.

Table 3.2: Query time(ms) of Kd-trees, Range Trees and Layered Range Trees

n points	Kd-trees	Range trees	Layered range trees
16	2.93	2.95	2.84
64	5.12	5.76	4.39
256	10.60	10.58	6.17
1k	15.68	15.07	7.81
1M	84.28	79.86	42.81
16M	294.51	285.74	281.48

This raises another question: when will we see a large amount of reported points? There are two typical situations

1. The queried range is comparatively large compared to the whole dataset

2. Density in the queried range area is relatively large

Therefore, the selection of data structures to preprocess the taxi data really depends on the specific requirements. For example, if fast query is the first priority and we do have enough storage spaces available, layer range trees is probably the best choice. On the other hand, if the storage space is restricted or there seems to be many records to report, Kd-tree will be more suitable. Furthermore, the decision of data structure selection is also affected by the computational power as well as the query frequency.

This chapter provides data structures as a tool to preprocess data, which can be used to find targeted data efficiently. This technique is more valuable from an application point of view to build a practical system. While in this thesis, we use the range searching just once in order get the data for further research. In next chapter, machine learning approaches to forecast taxi demand will be introduced.

Chapter 4

Machine Learning Approaches

Machine learning approaches are widely used for building accurate forecasting models. Section 4.1 describes the features chosen to model taxi demand patterns. A couple of machine learning models were discussed in section 4.2. Each considered model has several variations. In this thesis, we do not dig into the details and variations of each model, instead we focus on the basic version of each models and discuss its pros and cons applying to taxi demand forecasting problem. At the end, parameters optimization are introduced in section 4.3.

4.1 Features for taxi demand forecasting

To forecast taxi demand in the future, many features can be considered to build a good model. According to intuition and knowledge of transportation demand modeling, we identify several factors that are very likely to affect taxi demand, as summarized in Figure 4.1.

Trip records, calendar, weather, demography, event and public transit are six categories of features likely to be useful to build the forecasting model.

- **Trip Records** : Real-time taxi trip data that contains the time and location of every pick-up. It is the key to build time-series forecasting model. The taxi pick-up demand is aggregated hourly and the feature is defined as (x_t, x_{t-1}, ...,

Figure 4-1: Features for taxi taxi demand forecasting

x_{t-d+1}), where x_t is the number of taxi pick-ups in hour t and d hours' data is used to predict the demand at hour $t+1$

- **Calendar** : Time information consists of a few variables - month of the year, day of the week, time of the day, holiday or not, etc. Apparently, taxi demand patterns vary from time to time, therefore, *calendar* is a must-have feature. One example of calendar feature is (Jan, Mon, $6am$, $holiday_{yes}$)

- **Weather** : Weather information including temperature, wind speed, humidity and weather type(e.g., cloudy, light rain, heavy snow). One great benefit of weather information is that we are able to $forecast$ the extreme cases such as snowstorm which has huge impact to the taxi demand pattern. It seems promising in fitting the models, however, one problem arises that the accuracy of the model is inevitably affected by the accuracy of the weather forecasting. Tradeoff has to be considered for the weather features. In this study, hourly weather features of last 6 hours and the forecasting of future 6 hours are used for modeling.

- **Demography** : Statistical study of human populations such as population density, income/age distributions, level of education, etc. It is easy to understand that, for the long term, population density and income distributions are

correlated to taxi demand . One shortcoming is that the demography of a place changes slowly compared with the traffic dynamics, so that in some cases it may not be able to provide valuable inference besides historical taxi data.

- **Events** : Events such as parade, popular sport game and big concert can sharply change the taxi demand. In fact, experienced taxi drivers are quite sensitive to those events. They will be standing by the crowded places in advance so that they would have more chances to *meet* customers. In this thesis, events are manually obtained from the historical news and simply defined as a boolean variable, $event = 0$ or 1, where $event$=1 means there are events that may affect the taxi demand.

- **Public Transit** : The schedule and recorded demand of public transit. Relationship between different transportation modes plays an important role in demand forecasting. We define this feature as $(p_t, p_{t-1}, ..., p_{t-d+1})$, where p_t is demand of public transit in hour t, d is the feature dimension and in particular d=6 in this thesis. Transit data are particularly useful for locations that have subway stations nearby. In reality, some of the transit data is missing in this study so that the importance of transit data might be under estimated.

Summary of features are provided in Table 4.1. Feature class for collaborative modeling describes the class that each feature belongs to in order to build multi-model ensembles. T stands for trend information, S is seasonal information, H stands for historical data and E represents exogenous inputs. More details about collaborative modeling is discussed in "section 5.1.3'

4.2 Machine learning algorithms

4.2.1 K nearest neighbors

K nearest neighbors(KNN) regression is a non-parametric model that makes predictions according to the outputs of the K nearest neighbors in the dataset. Specifically,

Table 4.1: Summary of features

Feature	Data type	Explanation	feature class for collaborative modeling
hourly pick-ups	integer array	$(x_t, x_{t-1}, ..., x_{t-d+1})$ d=12,13...23	H, T, S
month	integer	1: Jan, 2: Feb, ..., 12: Dec	E
day of week	integer	1: Mon, 2: Tue, ..., 7: Sun	E
hour of day	integer	0: 0am, 2: 1am, ..., 23: 11pm	E
temperature	float array	historical: $(t_t, ..., t_{t-5})$ forecasted: $(t_{t+1}, ..., t_{t+6})$	E, T, S
wind speed	float array	historical: $(w_t, ..., w_{t-5})$ forecasted: $(w_{t+1}, ..., w_{t+6})$	E, T, S
humidity	percent array	historical: $(h_t, ..., h_{t-5})$ forecasted: $(h_{t+1}, ..., h_{t+6})$	E, T, S
weather type	integer array	historical: $(wt_t, ..., wt_{t-5})$ forecasted: $(wt_{t+1}, ..., wt_{t+6})$ 1:sunny, 2:light rain, 3:heavy rain,4:light snow, 5:heavy snow, 6:others	E, T, S
monthly population density	integer	e.g., 1254 means $1254/mile^2$	E
monthly average income	float	e.g., 3231.5 means \$3231.5/mont	E
current event	boolean	0: no event, 1: events happen	E
demand of public transit	integer array	$(p_t, p_{t-1}, ..., p_{t-5})$	E, T, S

given a data point: we will

1. Compute the distance(typically the Euclidean distance) between that point and all the other points in the training set.

2. Select the closest K training data points.

3. Compute average or weighted average of the target output values of these K points, which is the final predicted result.

The simple version of KNN is easy to implement by computing the distances to all stored examples. The critical setting of KNN is the parameter K, which should be selected carefully. It is easy to see that a large K will help to build a model with lower variance but higher bias. By contrast, a small K will lead to higher variance but lower bias.

In the field of taxi demand forecasting, KNN is interpreted to find the nearest patterns in the history with the *current* pattern. Those neighbors work as references to make predictions for the future standing at *current* time. Figure 4-2 is a simple example of how KNN works for time series cases. Suppose we stand at the end of series B and try to make prediction. By KNN, a similar series, called series A, is identified as the nearest neighbor of series B, so that the historical point can be a good reference to make prediction.

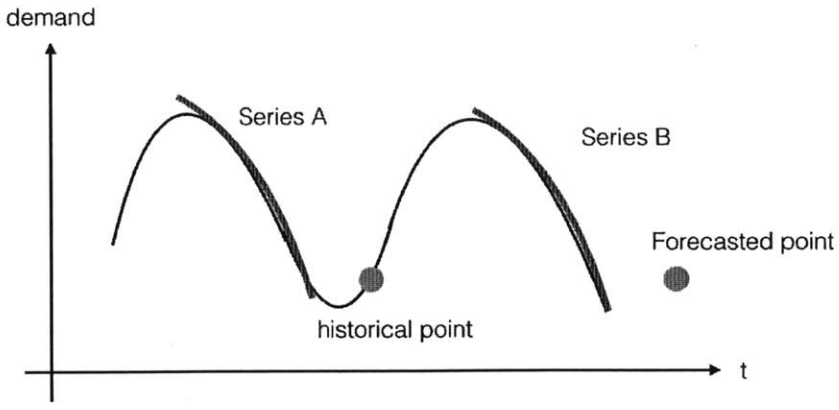

Figure 4-2: An example of KNN for time series

KNN is especially useful for special situations which have already happened before and the records are included in the training set, e.g. snow storm, sport game, etc. One disadvantage of KNN is that it is computationally inefficient for large training sets. Using an appropriate nearest neighbor search algorithm will reduce the number of distance evaluations actually performed. Many neighbor search algorithms have been proposed in last decades, but it is not the topic this thesis is focusing on. The KNN package we use in this thesis is *sklearn.neighbors*.

4.2.2 Artificial neural networks

Artificial neural networks (ANN) approach has become a widely used technique for time series forecasting. Basically ANN contains fully or partially connected neurons, which are single processing units in a neural network. Connections are made between the neurons and weights are assigned for all the connections. From the architecture point of view, each neuron consists of inputs, an activation function and outputs. The simple calculation taking place in a single neuron is

$$y = f\left(b_0 + \sum_i w_i x_i\right) \tag{4.1}$$

where y represents the output, x is the input vector, w is the weight vector, b_0 is the bias and $f(b_0, w, x)$ is the activation function, which performs a transformation on the calculated results.

However, a single neuron works appropriately only for inputs that are linearly separable. To work for nonlinearity, more than one neuron is needed in a neural network. Neural networks can have multiple layers, where each of the layers consists of one or more neurons. The neurons from one layer are connected to the adjacent layer neurons. Typically, a multilayer neural network contains an input layer, one or more hidden layers, and an output layer, as shown in Figure 4-3.

The connected neurons in a multilayer artificial neural network are able to perform as effective nonlinear models, in which the weights of the connections between neurons can be learned by appropriate training algorithms. ANN is good at capturing

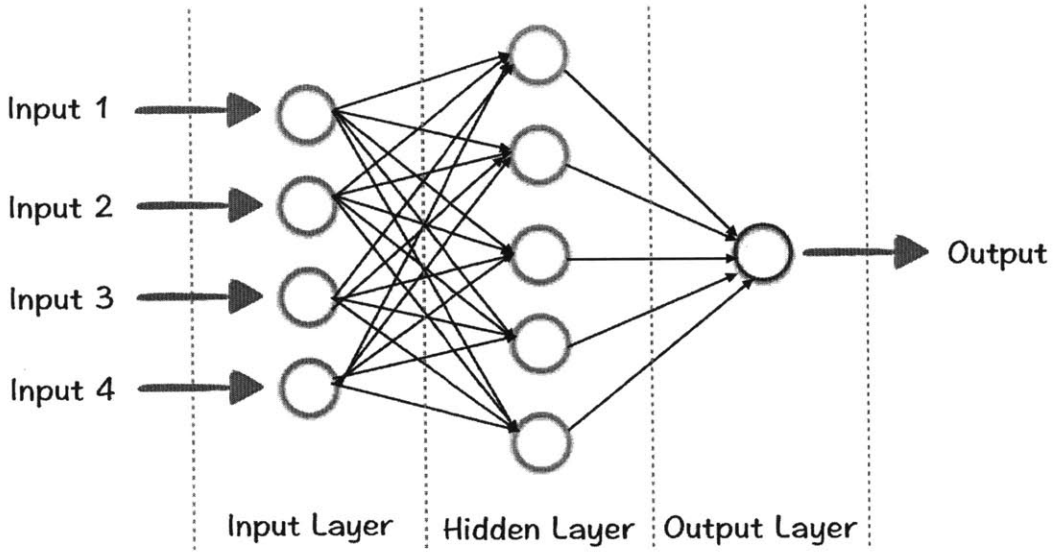

Figure 4-3: A multilayer feed-forward artifical neural network

associations or discovering regularities within a set of patterns. It has been widely used for demand forecasting which is essentially a time-series forecasting problem. In taxi demand forecasting, ANN works well for the seasonal pattern. Obviously, the seasonal pattern exists in the taxi demand series since the demand pattern is influenced by seasonal factors such as the quarter of the year, the month, day of the week, etc. In this thesis, the ANN package we use is *PyBrain*.

4.2.3 Support vector machines

Support vector machines(SVM) is suggested as a useful technique based on using a high-dimensional feature space and penalizing the ensuing complexity with a error function.

Even though SVM is usually used to solve classification problems, it can be applied to regression as well. Considering a linear model first for illustration, the prediction is given by

$$f(x) = w^T x + b_0 \tag{4.2}$$

where w is the weight vector, b_0 is the bias and x is the input vector.

The objective is to minimize the error function given by

$$J = \frac{1}{2}\|w\|^2 + C\sum_{m=1}^{M} Loss(y_m, f(x_m)) \tag{4.3}$$

where w is the weight vector, x_m is the mth training input, y_m is the target output and $Loss(y_m, f(x_m))$ is the loss function.

Support vector regression(SVR) has two significant advantages:

1. The model produced by SVR depends only on a subset of the training data, because the loss function ignores any training data close to the model prediction. Therefore, SVR is suggested to produce a better generalization results.

2. With the help of support vector kernels, the inputs of SVR are usually mapped to a high dimensional feature space. Figure 4-4 is an example of how different kernels perform.

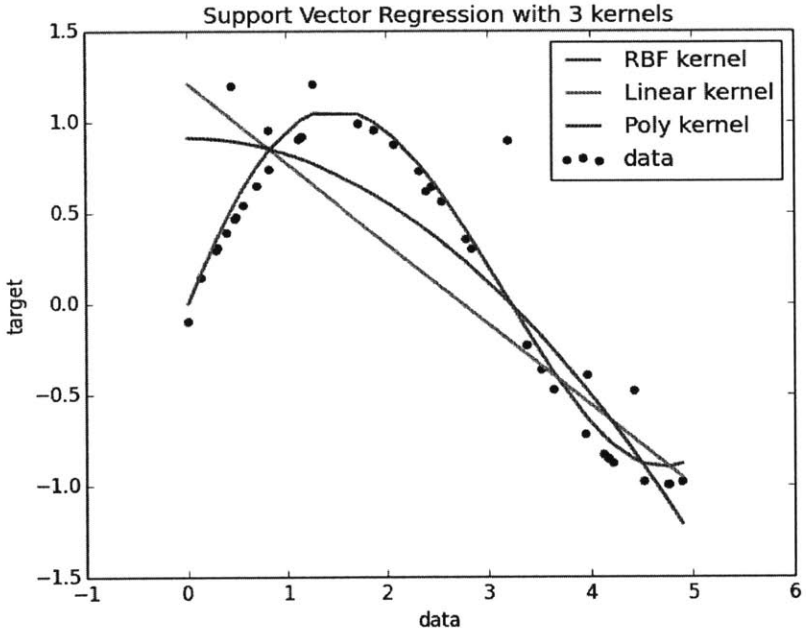

Figure 4-4: An example of SVR with different kernels

More details about SVR was introduced in the literature review, in this section we focus on its application to taxi demand forecasting. There is much *noise* existing in the taxi demand series resulted by events such as irregular working schedule of companies, big concerts, parade, etc, which may cause overfitting for models. Due to the characteristics of better generalization, SVR is expected to have a good forecasting performance in these cases. In this thesis, we select *sklearn.svm* to be the SVM package.

4.2.4 Decision trees

The goal of decision trees(DT) is to make predictions by learning simple decision rules inferred from the data features. It is a non-parametric supervised learning algorithm applied to both classification and regression. Learning from data features, DT is able to generate a set of if-then-else decision rules. Decision Trees can also be applied to regression problems. The depth of the tree is a good measure of the complexity of the model. Generally the deeper the tree, the more complex the decision rules are. Figure 4-5 shows the different effect of decision tree regression with $max\ depth = 3$ and $max\ depth = 6$.

Decision trees are very simple to interpret, with that we are able to understand the decision rules, for instance, the relationship between certain features and the outcomes. Important insights can be extracted by describing a situation and its inference for the results. In this thesis, the package of decision trees we use is $sklearn.tree.DecisionTreeRegressor$.

One significant advantage of DT is that it works well even for the dataset that contains missing features. Applying to taxi demand forecasting, there are many missing features such as events or transits data, which requires the model to be able to work effectively with a subset of features. DT is a good fit in this case. One disadvantage of DT is that the model tends to be overfitting easily so the forecasted results might be wrongly affected by abnormal cases, but this problem can be solved by ensemble methods.

Ensemble methods is designed to combine the predicted results of a set of base

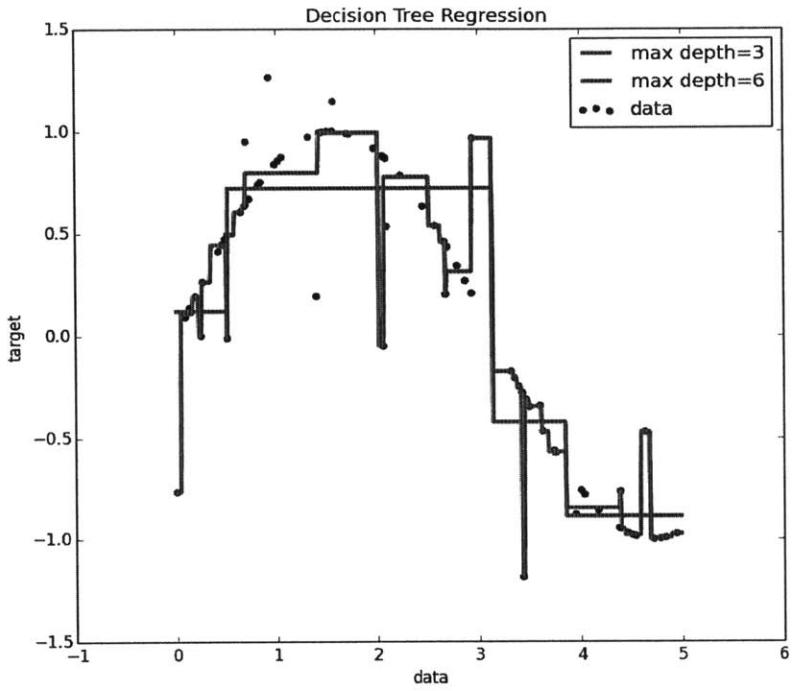

Figure 4-5: Decision Tree Regression with diffrent maximum depths

estimators, in order to provide better generalization and robustness compared with a single estimator. Bagging methods build a set of estimators on random subsets of the original training set, and their individual predictions are aggregated into a final prediction. On average, the combined estimator workers better than single one. Because of the randomization in the ensemble construction procedure, bagging methods are good at reduce the variance of a base estimator. Since bagging methods are able to prevent overfitting, strong and complex models such as fully developed decision trees are suitable to be the base estimator. Random forests are good examples of bagging methods. Although the bias of the forest is usually slightly larger than the basic non-random tree, its variance usually decreases more than compensating for the increase in bias, therefore coming up an overall better model.

Table 4.2: Hyperparameters to be optimized for ML algorithms

ML algorithms	Hyperparameters to be optimized
K Nearest Neighbors	K, distance metric
Artificial Neural Network	hidden layer size, max epochs
Support Vector Machines	kernel, C
Decision Trees	max depth, max features, min samples split
Random Forests	n estimators, max depth, max features

4.3 Hyperparameters selection

Each of the machine learning algorithms introduced above has a set of parameters. Some parameters such as coefficients are directly learnt from training. Other parameters such as C and *kernel* of SVM have to be set before fitting the model. Those parameters are referred to as hyperparameters. In this section, we focus on hyperparameters selection for the purpose of optimizing the algorithms' performance. Table 4.2 summarizes the critical parameters need to be optimized for machine learning algorithms discussed in this chapter.

Hyperparameters are typically selected in the validation process. Cross validation is considered as a good method to find the optimal parameters. However, since taxi demand record is a time series data which is not suitable to use cross validation. We simply split the data into training and validation sets, where the validation set is used for parameters selection.

Grid search and random search are two approaches to select parameters. Grid search is simply an exhaustive searching through a manually specified hyperparameter space (all the combinations of given hyperparameter options). One disadvantage of grid search is the curse of dimensionality, which leads to high requirement of computational power. Luckily, the evaluations of parameter settings are usually independent of each other so that the process can be parallelized. Under computational or time constraints, random search is likely to be a more suitable approach. Random search samples parameter settings a fixed number of times by specific random distributions, which has been found to be much more effective in high-dimensional spaces.

By selecting the best performed hyperparameters for each model, we can have a better understanding and fair evaluation when we are trying to choose among several candidate models, which is a key component of the five-phase ensemble estimator proposed in chapter 5.

4.4 Summary

In this chapter, we first proposed six-category features that have impact on taxi demand. We talked about machine learning algorithms that suitable to be applied to taxi demand forecasting problems, including single models(KNN, SVM, ANN, DT) and ensemble methods. We focused on the basic version of each model and analyze their potential pros and cons in taxi demand modeling and forecasting. Then we discussed the hyperparameters selection as a further improvement of each model.

However, the taxi demand patterns vary a lot from time to time. It is very hard to say any specific models work best in all scenarios. In next chapter, a five-phase ensemble estimator will be proposed. It is an adaptive forecasting system that not only keeps finding the most appropriate model, but also combines single models to build a better estimator.

Chapter 5

Ensemble Estimator

As discussed in chapter 4, every single machine learning model has its pros and cons in taxi demand forecasting. Moreover, every single model might be appropriate for certain scenarios. For instance, decision trees are good at capturing the relationship between exogenous inputs and targeted value.

Due to the variety of taxi demand patterns, even selecting algorithms from the candidates with most accurate predictive performance is far from optimal. In this chapter, we propose methodologies to construct an ensemble estimator that takes advantages of multiple distinct algorithms in order to produce better results. Three multi-model ensemble methods are introduced first, followed by the five-phase ensemble estimator construction and some of its variants for practical applications.

5.1 Multi-model ensemble

Differ from well-known ensembles such as random forests, which combines several same-type base models into one stronger model, we focus on the ensemble of different base models, for example, a combination of decision trees and SVM. Select-best, weighted combination and collaborative modeling are three methods to construct multi-model ensembles.

5.1.1 Select-best

Select-best approach is easy to interpret. Basically a part of the whole dataset is split for model training, then the trained models are evaluated with certain criterion in order to select one with best performance. More details about performance measures are introduced in Section 6.3.

There is one important thing should be mentioned here. In many situations, models are validated via cross-validation to avoid the distribution bias of dataset, however, cross-validation is not suitable for the taxi demand forecasting because it is a time-series problem and the order of data matters a lot. For example, it makes no sense to use previous data to validate a model trained by future data. In practice, recent data points in the time series are suitable for model validation and selection.

5.1.2 Weighted combination

Weighted combination is an ideal method to produce a more robust estimator then select-best method, by reducing variance among the candidate models. After doing weighted combination, some extreme errors can be effectively avoided which is a critical requirement of practical forecasting systems.

Firstly, individual models are trained and validated with two different datasets, typically two consecutive time series points with the former one for training and the later one for validation. Through validation, each individual model obtains optimized parameters, as discussed in Section 4.3. After all, results estimated by those optimized single models are weighted combined together to get the final estimated value,

$$\hat{Y} = c + \sum_{m=1}^{M} w_m \hat{Y}_m \tag{5.1}$$

where $\hat{Y}$ is the final result, M is the number of candidate models, w_m is the weight of mth model, $\hat{Y}_m$ is the estimated result of mth model. Figure 5-1 shows the basic flow of weighted combination method.

Two types of weights setting methods are implemented in this thesis.

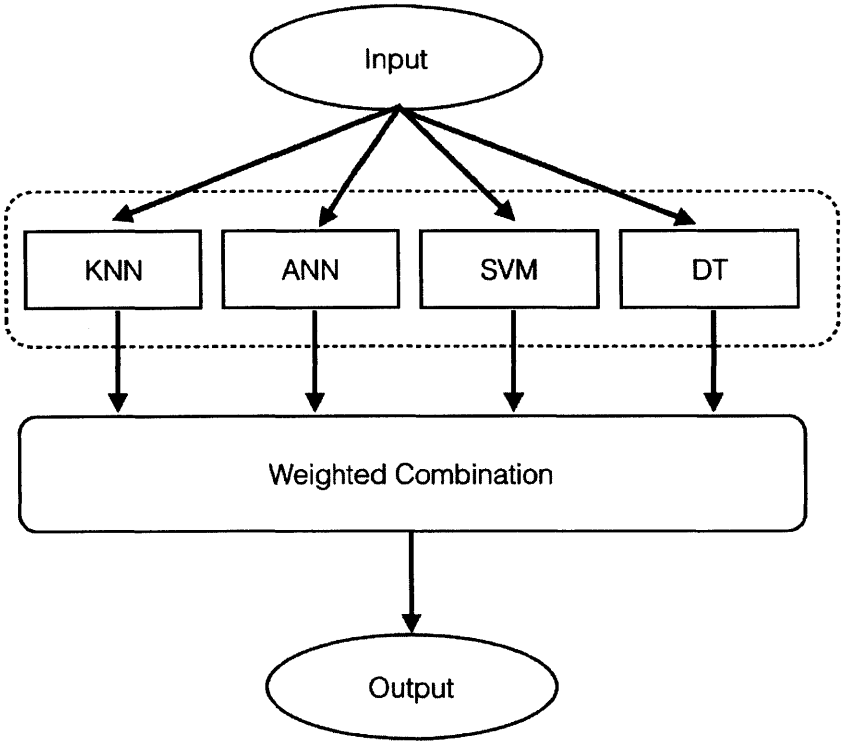

Figure 5-1: Working flow of weighted combination method

1. Simply transform the training/validation error of each model into its weight, with the principle that better models have greater weights. For example,

$$w_m = \frac{1}{1 + e^{|\epsilon_m|}} \tag{5.2}$$

is a simple method to determine the weight of mth model, where ϵ_m is the estimation error of mth model. Usually normalization is needed for combination.

2. Select a new subset of data besides training and validation, and do regression in which the targets are the real values and features are the estimated values from single models.

In this thesis, the second methods, regression is used to build weighted combination.

Table 5.1: Selected model for each feature subset in target locations

Location	Trend information	Seasonal information	Historical data	Exogenous inputs
Metropolitan Museum	KNN	KNN	ANN	DT
Time Square	DT	SVM	SVM	DT
Grand Central Terminal	ANN	ANN	SVM	SVM
Public Library	KNN	ANN	SVM	DT
Empire State Building	KNN	SVM	SVM	DT
Union Square	KNN	ANN	DT	SVM
New York University	KNN	ANN	ANN	DT
City Hall	SVM	ANN	ANN	DT
Wall Street	ANN	ANN	DT	DT
JFK Airport	ANN	SVM	SVM	DT
Choice	KNN	ANN	SVM	DT

5.1.3 Collaborative modeling

The idea of collaborative modeling is to separate the forecasting task into multiple subtasks to be taken care of by different models. The whole feature space is split into several feature subsets.

A specific group of candidate models are examined against different subsets. Using the select-best approach, the most appropriate model among the candidate models in a group is chosen for each subset. In order to decide which model fits which subset, we select 10 target locations at New York City. For each location, we decompose the taxi demand series to get the trend and seasonal information. Then we test machine learning models on each of the feature subsets, including trend information, seasonal information, historical data and exogenous inputs. Table 5.1 shows a test in which candidate models are examined and the best model is selected for each feature subset in each target location. More details about the 10 target locations are introduced in section 6.2.2. In this test, the best model is defined the model being selected most times.

As shown in Table 5.1, KNN is suitable for trend information; ANN is the best choice for seasonal information; SVM works well for the historical pick-up data; and

DT has significant advantages over any other models for the exogenous inputs. Among the four subsets, trend information is the most simple one obtained by moving averages, KNN works well to find the similar trend from the smoothed time series. The exogenous inputs including time information, weather, events and public transit are good indicators. Decision trees are good at finding the rules that how the indicators affect taxi demand. Further researches can be directed to analyze the the spatiotemporal characters of specific models across specific locations and times. In these thesis, we select the overall best models for each feature subset.

After selecting the best models, estimated results from these models are weighted combined together to have a final output. Figure 5-2 illustrates the idea of collaborative modeling.

In this study, we use R packages to decompose the data. Trend information is obtained by smooth function $R.TTR.SMA()$, and seasonal information is obtained by $R.decompose()$ function.

5.2 Five-phase ensemble estimator

In this section, a five-phase procedure is proposed to build a multi-model ensemble estimator, including single model training, parameters optimization, ensemble construction, ensemble methods selection and forecasting.

- **Phase one: Single model training.**

  Individual model training is the first phase to build the ensemble estimator. In this phase, single models are trained separately as introduced previously. Two key factors are carefully considered here: the size of training data and the frequency of single model training. The answers depend, generally the larger the training data, the lower the training frequency should be, due to the constraint of computational power. However, sufficient training data is usually necessary as there should be enough scenarios stored in the dataset. For example, rich information such as seasonal trend and impacts of weather,

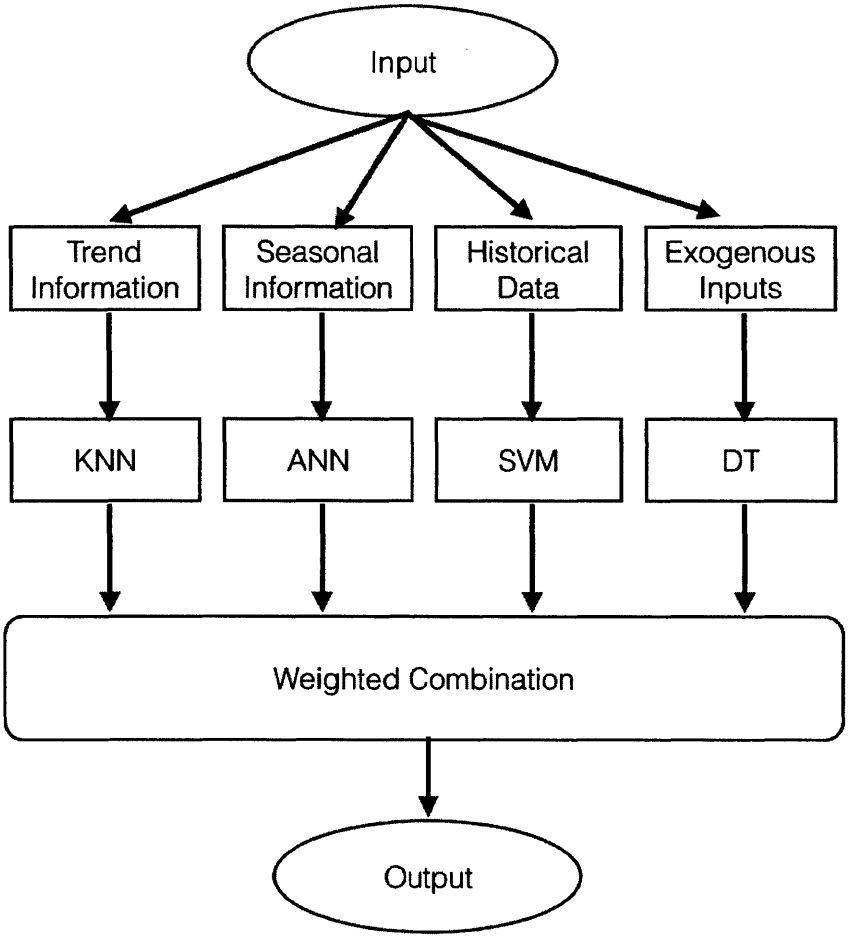

Figure 5-2: Basic working flow of collaborative modeling

would effectively help the model recognize and understand current status, so as to forecast accurately. In this thesis, historical data up to one-year lag is used for model training.

Furthermore, large size of training data not only affects the computational efficiency of training, but also slows down the validation and forecasting of some models. For instance, KNN is an instance based learning model. If no advanced neighbor searching algorithm is introduced, KNN will go through the entire dataset in order to select neighbors to make a prediction. In this thesis, models are trained once per day and updated once per hour.The reason is that training takes much more time and computational power than updating. Up-

dating steps only involve model validation and parameters selection, without re-training the models.

- **Phase two: Parameters optimization.**

  After training all the candidate models, we come to the second phase - parameters optimization. As discussed in Section 4.3, hyperparameters that are not directly learnt within estimators can be set by searching a parameter space for the best validation. A new subset of data is used in this phase. Grid search and random search are the two methods to find the optimal hyperparameters for each model. Tradeoff between accuracy requirements and computational efficiency are considered to choose the appropriate one.

  Sometimes the phase one and phase two can be combined together so that single models can be processed in parallel. After this phase, optimized single models are produced.

- **Phase three: Ensemble construction.**

  Ensemble construction is the critical phase in building the estimator. In this phase, optimized single models are prepared in advance and the entire set of single models are examined to construct the ensemble. As discussed in Section 5.1, three methods can be used to build the multi-model ensemble: select-best, weighted combination and collaborative modeling, producing three ensembles as a result. In particular, collaborative modeling involves extra work in phase one and two since data decomposition is required in order to assign models to the suitable subsets.

- **Phase four: Ensemble selection.**

  In this phase, the best ensemble method is selected among three candidate methods. The implementation of this phase is pretty much the same as single model selection, which is based on a certain performance measure.

- **Phase five: Forecasting.**

Phase five is to forecast using the selected ensemble from Phase four. The only thing to mention about this phase is how far future the ensemble is going to forecast. In this thesis, the default setting of longest future to forecast is 12 hours.

In the five-phase process, phase 1 uses historical data up to one-year data, phase 2-4 use one-week data for each, and phase 5 forecasts the hourly demand of next 12 hours. More specially, the data for the five phases is in consecutive series, as shown in figure 5-3.

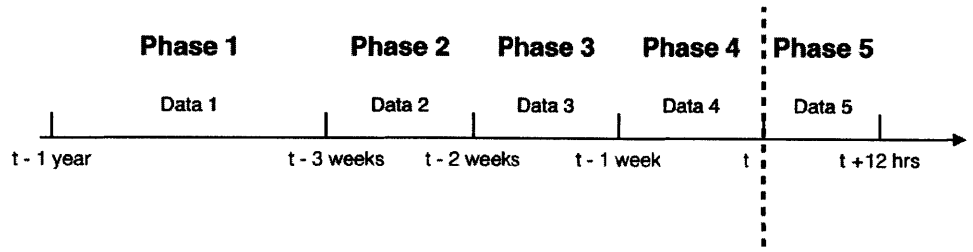

Figure 5-3: Example of data being used for each phase

In figure 5-3, *Data* 1-5 represent the dataset used in each phase. Each sample in *Data* 1-5 belongs to an hour, which contains the taxi demand value and all the features used for modeling at that hour. Let's take a simple example to see how this system works. Suppose t = July 22, 2013 0am.

1. **Prepare data for each phase**

   Data 1: data from July 22, 2012 0am to June 30, 2013 11pm

   Data 2: data from July 1, 2013 0am to July 7, 2013 11pm

   Data 3: data from July 8, 2013 0am to July 14, 2013 11pm

   Data 4: data from July 15, 2013 0am to July 21, 2013 11pm

   Data 5: data from July 21, 2013 0am to July 21, 2013 12am

2. **Single model training**

   Dataset: *Data* 1

   Procedure: train each single model on *Data* 1

Output: trained models M_k, where k=KNN, SVM, ANN, DT.

Note that M_k is a set of models with different hyperparameters, for example, SVM(kernel = rbf, C = 1), SVM(kernel =linear, C=10) and so on.

3. **Select the best hyperparameter setting of each model based on performance measures**

   Dataset: *Data* 2

   Input: M_k

   Output: optimized single models M_k^{opt}

4. **Construct ensembles using**

   Dataset: *Data* 3

   Input: M_k^{opt}

   Output: E_{sb}, E_{wc}, E_{cm}: three ensembles constructed by select-best, weighted combination and collaborative modeling.

5. **Select the best ensemble based on performance measures**

   Dataset: *Data* 4

   Input: E_{sb}, E_{wc}, E_{cm}

   Output: E^{opt}

6. **Forecast**

   Dataset: *Data* 5

   Input: E^{opt}

   Procedure: forecast taxi demand on *Data* 5 and measure the performance

There are some variants of the five-phase procedure.

First, when preparing five data subsets, rather than selecting five consecutive time series, we can alternatives select five periods that have something in common but are non-consecutive. For example, if the task is to forecast taxi demand in the afternoon of Wednesday, we may have the Wednesday data of the past a few weeks to be the subsets for building the five-phase ensemble estimator. These data subsets are likely to share some common characteristics of the taxi demand patterns on Wednesday and

they may provide better performance. However, one disadvantage of non-consecutive time series is that the estimator is slow-response to the special cases when they occur. Alert system might be necessary in this case.

Second, from the engineering point of view, getting the optimal forecasting is not always necessary considering the tradeoff between accuracy and computational cost. Setting up an error bar is a very good method to ensure the system is working properly without wasting too much computational power. For example, models can be temporarily deactivated if it performs poorly for a period of time. If the whole system keeps working well, the ignored models remain *deactivated.* However, when the error bar is reached, all models will be *activated* in order to make greatest efforts on forecasting.

5.3 Summary

Through the five-phase ensemble building procedure, we found the optimal parameters for every single model, constructed ensembles with the optimized models, selected the most appropriate ensembles and used the final ensemble estimator to make demand forecasting. Ideally, the estimator can always find the optimal model at every time step because of its sensitiveness to the pattern changing. Also, weighted combination and collaborative modeling significantly reduce the variance of forecasted results and avoid big errors which may mislead taxi drivers and companies. Furthermore, for locations with different characteristics such as population density and democracy, their taxi demand pattern varies a lot so that specific forecasting models should be applied to each case, which can be well solved by the ensemble estimator.

To evaluate how ensemble estimator performs for forecasting taxi demand, we will conduct a case study on New York City taxi records in the next chapter.

Chapter 6

Case Study

In previous chapters, a couple of single machine learning models and multi-model ensemble methods have been proposed. How do they perform to solve real-world problems? This chapter introduces a case study that all models are tested and analyzed upon New York City(NYC) taxi dataset. NYC has remarkably rich history of taxi industry. A huge dataset consisting of two-year NYC taxi trips is used for training, validating and testing the models. In addition, we will discuss the performance measures in section 6.3 that used for model evaluation as well as parameters selection.

6.1 Dataset

For several years, taxis in New York City have had GPS receivers to record information. The data I use in this case study is the taxi trip records in 2012 and 2013 covering all the five boroughs of NYC: Manhattan, Bronx, Brooklyn Queens and Staten Island, as Figure 6-1 shows. This dataset is obtained through a FOIA request(http://www.andresmh.com/nyctaxitrips/).

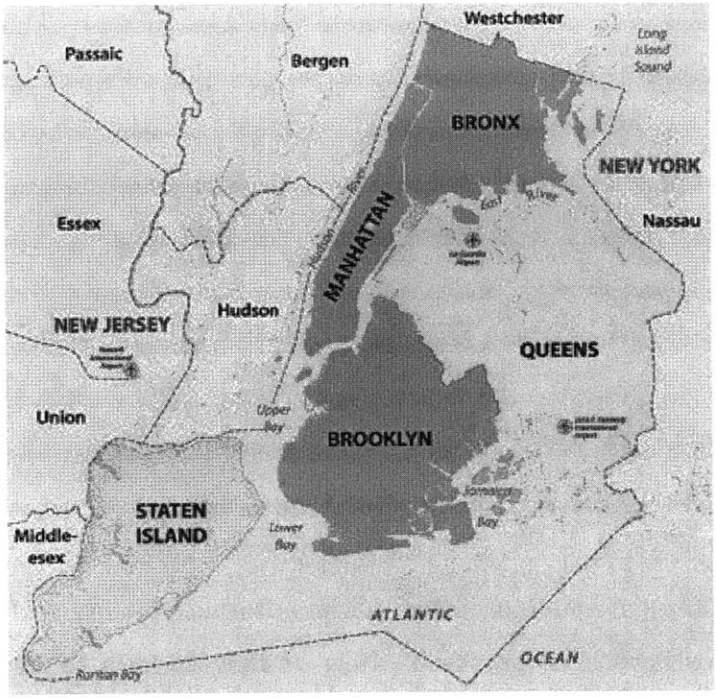

Figure 6-1: Areas that the NYC taxi dataset covers

6.1.1 Data content

The dataset contains approximate 168 million taxi trips/year. A single trip record has valuable information such as pick-up and drop-off locations, trip distance and fare, etc. The information provides us great help to understand the spatiotemporal taxi pattern. Detailed explanations of the data are as follows:

- **medallion**: a permit to operate a yellow taxi cab in New York City, which is effectively a (randomly assigned) car ID.
- **hack license**: a license to drive the vehicle, which is effectively a (randomly assigned) driver ID.
- **vender id**: e.g., Verifone Transportation Systems (VTS), or Mobile Knowledge Systems Inc (CMT)
- **rate code**: taximeter rate.
- **pickup datetime**: start time of the trip, e.g., 2012-01-06 21:51:00.

- **dropoff datetime**: end time of the trip, e.g., 2012-01-06 21:55:00.
- **passenger count**: number of passengers on the trip, default value is 1.
- **trip time in secs**: trip time measured by the taximeter in seconds.
- **trip distance**: trip distance measured by the taximeter in miles.
- **pick-up longitude and latitude**: GPS coordinates at the start of the trip, e.g., (-73.980003, 40.780548)
- **drop-off longitude and latitude**: GPS coordinates at the end of the trip, e.g., (-73.974693, 40.790123)

Particularly in this case study, we focus on demand modeling, so that the four variables - **pickup datetime**, **dropoff datetime**, **pick-up longitude and latitude** and **drop-off longitude and latitude** are used.

6.1.2 Data characteristics

Since demand forecasting is the topic of this thesis, we mainly focus on the time and location of each taxi trip. We extract the data within specific time and location ranges using the 3D range searching techniques as discussed in chapter 2. In order to understand more about the NYC taxi pattern, the data is aggregated and examined, showing some statistics and insights.

First of all, by looking at taxi demand distribution over a week, we identify the general weekly pattern of taxi demand. Figure 6-2 shows the hourly taxi pick-up demand from Monday to Sunday in a typical week in NYC. Clearly, the patterns show significant similarities among weekdays from Monday to Friday with small gaps on the daily peak. However, the demand pattern in the weekend reflects many differences to the weekday.

Second, taxi demand distribution over the whole NYC area is unbalanced. Manhattan attracts 90.3% taxi pick-ups and the other four boroughs attract the remaining 9.7%. Details about the taxi demand distribution is shown in Table 6.1. Moreover,

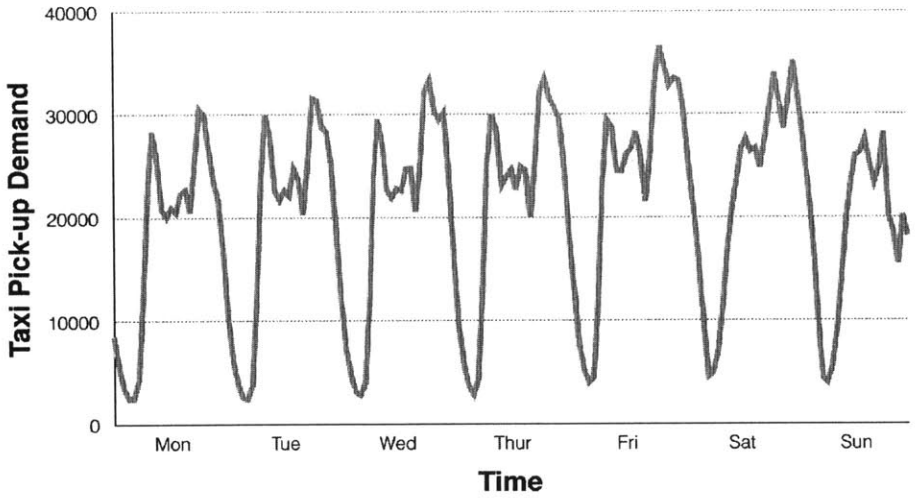

Figure 6-2: Taxi pick-up demand distribution over a week in NYC

Table 6.1: Taxi demand distributions over NYC area

Borough	% of taxi demand(pick-ups)
Manhattan	90.3%
Queens	5.0%
Brooklyn	3.1%
Bronx	0.9%
Staten Island	0.7%

in Queens' 5.0% share, JFK airport accounts for 3.5% which is more than half of the taxi demand in Queens and is greater than the total taxi demand in Brooklyn. Therefore, the taxi demand in JFK is airport is analyzed in this case study.

6.2 Experiment design

6.2.1 Objectives

The objective of this experiment is to evaluate and analyze the performance of each model over taxi pick-up demand forecasting problem, including the single models as well as the ensembles.

In addition to a overall evaluation, various locations and time ranges are chosen to examine how each model performs under different situations. Every model is assigned to forecast the pick-up demands of 10 locations in the future 12 hours, in which the the data was aggregated by one hour.

6.2.2 Target locations selection

In order to evaluate the models comprehensively, we need to analyze their performances with different taxi demand patterns. There are many locations in New York City having very different characteristics resulted by the population density, democracy, geographic environment and so on. For instance, taxi demand patterns around airport should be highly related to the flight schedules: high pick-up after flight arrivals and high drop-off before flight departures. Specific models might be appropriate to describe specific cases, which brought challenges to the demand forecasting system.

Ten locations in NYC are selected as target places for the experiment: Metropolitan Museum of Art, Time Square, Grand Central Terminal, Public Library, Empire State Building, Union Square, New York University, City Hall, Wall Street and JFK Airport.

These locations represent various characteristics: Grand Central Terminal and JFK Airport are typical transportation centers; Metropolitan Museum of Art, Empire State Building and Union Square are famous attractions; Time Square and Wall Street are always crowded; City Hall represents government institute while New York University represents schools. Their GPS coordinates are shown in Table 6.2.

More specifically, we have a center point for each location. The center point is defined as (longitude, latitude) and the area is default to be a square with 200 meters on a side. Especially, JFK Airport has a longer square side of 2000 meters because of its landscape. With the GPS range of the target locations, we are able to use range searching techniques to extract the taxi data within these areas.

The taxi demand pattern of each location varies a lot, resulted by underlying factors such as working mode, demography, landscape, public transits and so on. For example, as a typical weekly demand pattern in Figure 6-3 shows, the taxi pick-up de-

Table 6.2: Target locations for the experiments

Index	Location	(longitude, latitude)
1	The Metropolitan Museum of Art	(-73.963196, 40.779625)
2	Time Square	(-73.984739, 40.762564)
3	Grand Central Terminal	(-73.977064, 40.753197)
4	Public Library	(-73.982245, 40.753182)
5	Empire State Building	(-73.985728, 40.748639)
6	Union Square	(-73.990460, 40.735939)
7	New York University	(-73.996688, 40.730212)
8	City Hall	(-74.006134, 40.713233)
9	Wall Street	(-74.008840, 40.706216)
10	JFK Airport	(-73.784865, 40.645876)

mand of Wall Street significantly decreases on weekends compared with the weekdays because of the holiday factors. By contrast, weekend demand of the Metropolitan Museum of Art increases. The possible reason is that people usually visit museums on holidays. Moreover, there are two daily peaks with slight differences clearly shown in the Wall Street: one morning peak around 8am and one evening peak around 8-10pm. However, in the Metropolitan Museum of Art, typical one daily peak occurs between 2-4pm when visitors finish journey and take taxis to leave.

Another comparison is between Time Square and JFK Airport, as shown in Figure 6-4. Time Square is always crowded that we don't see significant demand decrease in the weekend. The daily peak of Time Square usually occurs in the evening which makes sense since people like to have fun at that time. For JFK Airport, we see three local peaks in a day representing the peak hours in the morning, afternoon and evening. The taxi demand pattern around JFK Airport is highly determined by the flight schedule. The weekly peak usually comes on Sunday resulted by the returned flights for people who come back for the Monday job.

6.2.3 Procedures

Experiment settings are as follows:

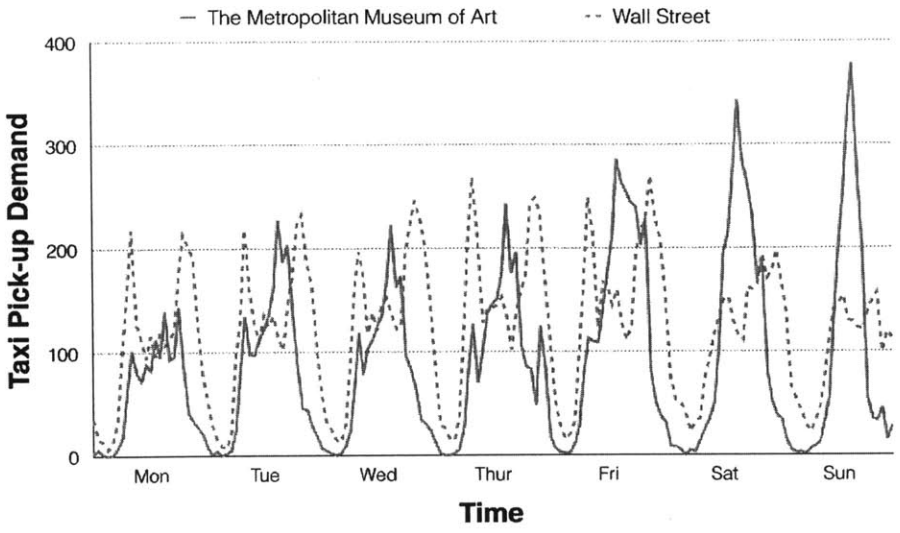

Figure 6-3: Typical taxi demand pattern of the Metropolitan Museum of Art and Wall Street

- This experiment is conducted on the Aliyun ECS server, with Ubuntu operating system, 16 cores and 64GB memory.
- The first-year(2012) data is used as training samples for every model, so that the forecasting experiment starts at the beginning of 2013.
- Taxi data is aggregated by hour. At the end of each hour, the task is to forecast the taxi pick-up demand in the next 12 hours.
- 10 target locations are modeled separately and in parallel.
- Considering the computational efficiency, models are trained once per day instead of once per hour.
- The models being tested are: K nearest neighbors, artificial neural networks, support vector machines, decision trees, random forests and multi-model ensemble.

Based on the above settings, let's take go through a few steps of the experiment:

1. At 2013-01-01 0am, process the aggregated historical data in the 10 target locations as training data.

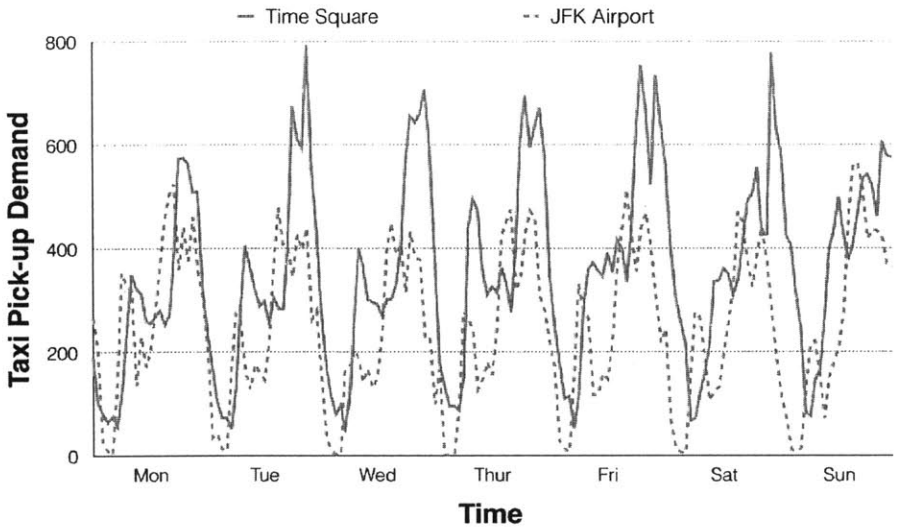

Figure 6-4: Typical taxi demand pattern of the Time Square and JKF airport

2. Use the models to forecast the taxi pick-up demands in the next 12 hours in each of the target locations, so that 6 *models* * 12 *hours* * 10 *locations* = 720 *values* will be produced every hour.

3. From 1am to 11pm, hourly repeat step 2 to produce the forecasted values.

4. At the beginning of a new day (0am), go back to step 1 and repeat the training and forecasting process.

5. Once forecasting process ends, all the forecasted values are compared with the real values to measure the model performances.

To summarize, there are totally 8748 time steps in 2013 to make predictions for the future taxi demands, calculated by 365 days * 24 hours - last 12 hours. At each step, 840 forecasted values are produced for performance measures.

Moreover, due to statistical characteristics of artificial neural networks and random forests, the test runs 5 times and the average results are generated.

6.3 Performance measures

Performance measure is a critical to model evaluation and selection. It verifies how accurate the fitted models perform in forecasting. As discussed in chapter 4 and 5, performance measures are frequently used to select the best parameters and the most appropriate ensemble method. In this thesis, we focused on three types of performance measures: Mean Absolute Error(MAE), Mean Absolute Percentage Error(MAPE) and Root Mean Squared Error(RMSE). Each type has specific properties that can be good references to evaluate and select models.

Mean Absolute Error(MAE) is defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \|\hat{y}_i - y_i\| \tag{6.1}$$

where n is the number of points, $\hat{y}_i$ is the forecasted value and y_i is the real value. It measures the average absolute deviation of forecasted values from real ones, showing the magnitude of overall error. Obviously, for a good forecast, the obtained MAE should be as small as possible. Also, MAE depends on the scale of measurement and data transformations.

Mean Absolute Percentage Error (MAPE) is defined as

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \|\frac{\hat{y}_i - y_i}{y_i}\| \tag{6.2}$$

where n is the number of points, $\hat{y}_i$ is the forecasted value and y_i is the real value. This measure represents the percentage of average absolute error occurred. Unlike MAE, MAPE is independent of the scale of measurement, but affected by data transformation. MAPE is very meaningful in measuring taxi demand forecasting problem, since the scale of demand varies from time to time and target locations have different taxi demand patterns.

Root Mean Squared Error (RMSE) is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y)^2}{n}} \tag{6.3}$$

where n is the number of points, $\hat{y}_i$ is the forecasted value and y_i is the real value. It is a measure of average squared root deviation of forecasted values. It emphasizes the fact that the total forecast error is in fact much affected by large individual errors. Although RMSE is a good measure of overall forecast error, but it is not as intuitive and easily interpretable as MAE and MAPE.

In order to have a clear understanding and evaluation of how the models actually perform, we set up a baseline which is a very simple method to make predictions. The baseline forecasting can be regarded as a reference point to see how much better(or even worse) the machine learning models work than normal strategy. The baseline we use is moving the average of demand at the same time of previous weeks, as defined:

$$\hat{D}_t = \frac{1}{N}\sum_{i=1}^{N} D_{t-i*7days} \tag{6.4}$$

where $\hat{D}_t$ is the forecasted demand at time t, $D_{t-i*7days}$ is the actual demand at the same time of ith week before, N is the number of weeks to make average. In this study, we use $N = 4$ to generate the baseline.

6.4 Key findings

6.4.1 Overall model performances

In order to have a general evaluation of each model's performance, we first measure the overall MAE, MAPE, RMSE of every model regardless of locations and times. Since the forecasted values are generated at each time step for the same locations and same future time steps, we get exactly the 1,049,760 forecasted results from each model. Their performance measures are summarized in Table 6.3, in which BS is the baseline, KNN is K nearest neighbors, ANN is artificial neural networks, SVM is

Table 6.3: Ovarall performance measures of each model

Model	MAE	MAPE(%)	RMSE
BS	44.5	18.5	60.0
KNN	37.8	15.2	51.5
ANN	34.8	14.2	46.2
SVM	34.1	14.3	44.2
DT	33.5	13.9	44.3
RF	30.0	12.7	40.2
ME	**25.3**	**10.6**	**33.1**

support vector machines, DT is decision trees, RF is random forests and ME is the multi-model ensemble.

6.4.2 Location-based model comparison

As discussed in section 6.2, taxi demand patterns in different locations vary a lot resulted by demography, landscape, etc. Evaluate model performances in different locations is meaningful that we will see whether the adaptive forecasting system is really necessary or not. In this experiment, we have 10 target locations. For each location, we measure the MAE, MAPE and RMSE of each model, as Table 6.4, 6.5 and 6.6 show. We also rank the model performances at each location, the the ranks of a model are summed up. For example, in Table 6.4, the numbers of KNN in Metropolitan Museum is 8.1(2), meaning its MAE equals 8.1 and the it is the 2nd best model for this location.

Even though the test has been run for 5 times, result of each run shows very little deviations. As we mentioned, for each model at each location, there are 8748 time steps in each 12 hours ahead are forecasted. This means the results are actually the average of 104976 numbers. Table 6.7 shows the standard deviations of MAPE of ANN and RF among the 5 runs.

As we can see from the tables, despite of the different characteristics of MAE, MAPE and RMSE, the results of these three measures are almost consistent. Clearly,

Table 6.4: MAE of models in target locations

Location	BS	KNN	ANN	SVM	DT	RF	ME
Metropolitan Museum	12.3(7)	8.1(2)	10.3(5)	11.0(6)	9.2(4)	8.6(3)	**6.7**(1)
Time Square	92.0(7)	79.6(6)	74.8(5)	71.5(4)	62.2(3)	56.0(2)	**46.1**(1)
Grand Central Terminal	85.6(7)	82.2(6)	64.2(2)	65.4(3)	72. 5(5)	66.2(4)	**55.1**(1)
Public Library	37.9(7)	25.9(2)	29.2(4)	29.7(5)	31.6(6)	28.1(3)	**22.8**(1)
Empire State Building	50.6(7)	38.2(4)	41.4(6)	34.2(2)	39.6(5)	36.9(3)	**30.2**(1)
Union Square	68.8(7)	58.4(6)	53.2(4)	55.6(5)	49.2(3)	42.0(2)	**37.2**(1)
New York University	10.9(7)	8.3(5)	6.5(2)	10.1(6)	8.2(4)	8.1(3)	**6.4**(1)
City Hall	10.5(7)	8.5(6)	6.8(2)	8.1(5)	7.4(4)	7.0(3)	**6.7**(1)
Wall Street	14.1(7)	13.2(6)	10.8(4)	11.1(5)	10.3(3)	9.8(2)	**8.8**(1)
JFK Airport	62.5(7)	55.2(6)	52.8(5)	44.2(3)	44.5(4)	40.1(2)	**33.2**(1)
Rank Sum	(70)	(49)	(39)	(44)	(41)	(27)	(10)

Table 6.5: MAPE(%) of models in target locations

Location	BS	KNN	ANN	SVM	DT	RF	ME
Metropolitan Museum	15.2(7)	9.8(2)	13.5(5)	14.0(6)	11.2(4)	10.5(3)	**8.1**(1)
Time Square	28.3(7)	24.5(6)	23.3(5)	21.5(4)	18.8(3)	17.5(2)	**14.3**(1)
Grand Central Terminal	21.2(7)	20.2(6)	15.6(2)	15.9(3)	17.7(5)	16.1(4)	**13.5**(1)
Public Library	17.9(7)	12.5(2)	13.4(3)	13.9(5)	14.9(6)	13.5(4)	**10.5**(1)
Empire State Building	18.2(7)	13.8(4)	15.0(6)	12.3(2)	14.5(5)	13.3(3)	**11.1**(1)
Union Square	23.2(7)	19.6(6)	17.5(4)	18.4(5)	16.9(3)	14.5(2)	**12.7**(1)
New York University	14.7(7)	11.2(4)	9.3(2)	13.4(6)	11.4(5)	10.9(3)	**8.7**(1)
City Hall	12.4(7)	10.3(6)	8.1(2)	9.3(5)	8.7(4)	8.2(3)	**7.6**(1)
Wall Street	12.8(7)	11.0(6)	8.7(2)	9.9(5)	9.3(4)	9.1(3)	**8.1**(1)
JFK Airport	21.2(7)	18.5(6)	17.5(5)	14.8(3)	15.1(4)	13.6(2)	**11.3**(1)
Rank Sum	(70)	(48)	(36)	(44)	(43)	(29)	(10)

Table 6.6: RMSE of models in target locations

Location	BS	KNN	ANN	SVM	DT	RF	ME
Metropolitan Museum	12.9(7)	9.1(2)	11.8(5)	12.0(6)	10.3(4)	9.5(3)	**7.4**(1)
Time Square	105.5(7)	89.7(6)	82.0(5)	75.4(4)	68.9(3)	61.7(2)	**48.4**(1)
Grand Central Terminal	95.1(7)	89.9(6)	65.5(2)	71.6(3)	77.9(5)	74.0(4)	**60.2**(1)
Public Library	41.6(7)	28.7(2)	32.1(4)	32.6(5)	34.2(6)	30.6(3)	**26.1**(1)
Empire State Building	53.1(7)	40.6(4)	46.4(6)	37.5(2)	43.4(5)	39.3(3)	**32.6**(1)
Union Square	77.0(7)	63.9(6)	57.9(4)	58.6(5)	54.5(3)	44.4(2)	**40.6**(1)
New York University	11.8(7)	9.1(5)	7.2(2)	10.7(6)	9.0(4)	8.9(3)	**6.7**(1)
City Hall	11.7(7)	9.5(6)	7.3(2)	9.2(5)	8.4(4)	7.8(3)	**7.2**(1)
Wall Street	16.1(7)	14.1(6)	12.4(4)	12.5(5)	11.0(3)	10.9(2)	**9.7**(1)
JFK Airport	68.0(7)	58.6(6)	58.2(5)	47.8(3)	49.2(4)	45.4(2)	**36.4**(1)
Rank Sum	(70)	(49)	(39)	(44)	(41)	(27)	(10)

Table 6.7: Standard deviations of the MAPE(%)

Location	ANN	RF
Metropolitan Museum	0.019	0.024
Time Square	0.009	0.012
Grand Central Terminal	0.008	0.013
Public Library	0.021	0.022
Empire State Building	0.020	0.016
Union Square	0.012	0.014
New York University	0.018	0.022
City Hall	0.023	0.021
Wall Street	0.025	0.018
JFK Airport	0.014	0.016

in all the 10 target locations, all the machine learning models perform better than the baseline. Meanwhile, multi-model ensembles have significant better performances over other models(approximately 3-5% decrease of MAPE). This result strongly proves that the multi-model ensemble estimator improves the accuracy of taxi demand forecasting.

Among KNN, ANN, SVM and DT, there is no *optimal* single model that dominates other in all cases, which also supports the advantages of combining single models together. Moreover, we have some findings regarding the single models:

- In most cases random forests perform better than decision trees. The possible reason is that it is easier for decision trees to be overfitting. Random forests are preferred since it provides better generalization.

- Artificial neural networks have good performance in Grand Central Terminal, New York University and City Hall. One common characteristic among the three locations is that the activities around there are highly determined by stable calendars (public transportation, school and working). As a result, the taxi demands in these three locations have useful seasonal information, which can be well captured by ANN.

- K nearest neighbors work well for Metropolitan Museum and Public Library, where the schedules are usually fixed. In these locations, we see the strong similarities among the taxi demand pattern from past to current periods. Therefore, KNN is usually able to find the near neighbors that happened before.

- Random forests perform significantly better than other single models in Time Square, Union Square and JFK Airport. One potential reason is that the taxi demand in theses locations fluctuates wildly and is more sensitive to exogenous inputs than other locations, in which the random forests are well fitted.

Table 6.8: MAPE of time-based forecasting performances

Model	Short-term	Mid-term	Long-term
BS	18.5	18.5	18.5
KNN	12.7	16.2	16.7
ANN	12.5	14.3	15.8
SVM	12.1	14.0	16.8
DT	11.6	14.8	15.3
RF	10.1	13.1	14.9
ME	**8.6**	**11.1**	**12.1**

6.4.3 Time-based model comparison

In addition to location, the number of time steps ahead for forecasting is another factor that affects the model performances. A model good at short-term forecasting may not work well for mid-term or long-term forecasting and vice versa. In this experiment, we measure the forecasting accuracy of each model over different time steps ahead to forecast, as shown in Table 6.8. Here short-term is defined as 1-4 hours ahead; mid-term is 5-8 hours ahead and long-term is 9-12 hours ahead respectively.

From the time point of view, multi-model ensemble also has better performances than any other single models for the short-term, mid-term and long-term future. In general, all models have better forecasting results in short-term future than mid-term and long-term futures, which makes sense intuitively. However, as shown in Table 6.8, the increase of MAPE from mid-term to long-term forecasting is greater than the increase from short-term to mid-term. One explanation is the cycling pattern of taxi demand that makes the accuracy differences not significant among the forecasting for relatively long future.

Except for ME and RF, the short-term forecasting of KNN, ANN, SVM and DT have similar performances. However, from short-term to mid-term forecasting, the MAPE of KNN increases sharply from 12.7% to 16.2%. This makes sense since the similarity between near neighbors and current period doesn't represent the similarity for their mid-term and long-term future. It is also the reason that makes the overall

Table 6.9: Selected times of three ensemble methods under three performance measures

Model	MAE	MAPE	RMSE	Total
SB	403,108(38.4%)	432,501(41.2%)	328574(31.3%)	1,164,183(37.0%)
WC	373,714(35.6%)	412,555(39.3%)	422003(40.2%)	1,208,272(38.4%)
CM	272,938(26.0%)	204,703(19.5%)	299181(28.5%)	776,822(24.6%)
Total	1,049,760	1,049,760	1,049,760	3,149,280

performance of KNN worse than other models.

6.4.4 Additional findings

Besides the performance measures mentioned above, we have a few additional findings which are useful when considering taxi demand forecasting.

One finding is about ensemble methods. In this thesis, three ensemble methods have been proposed: select-best, weighted combination and collaborative modeling. Table 6.9 shows the times being selected of each methods under three measures. SB stands for select-best, WC is weighted combination, CM is collaborative modeling.

To compare select-best and weighted combination, we need to consider the trade-off between bias and variance. In general, select-best leads to lower bias but higher variance than weighted combination. If the performance measure is sensitive to large errors, e.g., RMSE, weighted combination works slightly than select-best since it provides more robust forecasting. If MAE or MAPE is used for evaluation, there is no significant performance difference between these two methods. Regarding collaborative modeling, the effects really depend on the specific situations. For the locations or periods that the data can be well decomposed, this method works better than select-best and weighted combination, since single models are assigned to their suitable subtasks and their results are well combined. However, in cases that there are missing data for some features, collaborative is not able to produce satisfied results.

The second finding is the dynamic features importance. In normal cases, historical data is the most important. The data is complete and rich that trends can be

easily inferred from the historical pattern. However, in special cases, the exogenous inputs also plays important roles. For instance, under extreme weather days such as blizzard, the weather features including weather forecasting would affect the taxi demand significantly. Figure 6-5 shows the low taxi demand when super storm Sandy came on Monday. As we can see, its impact lasted for the whole week until Sunday. Also, big events like parade and sport games also highly determine the taxi pattern when they occur. The dynamic features importance also supports the idea of adaptive forecasting system which provides faster response to the pattern changing than fixed models.

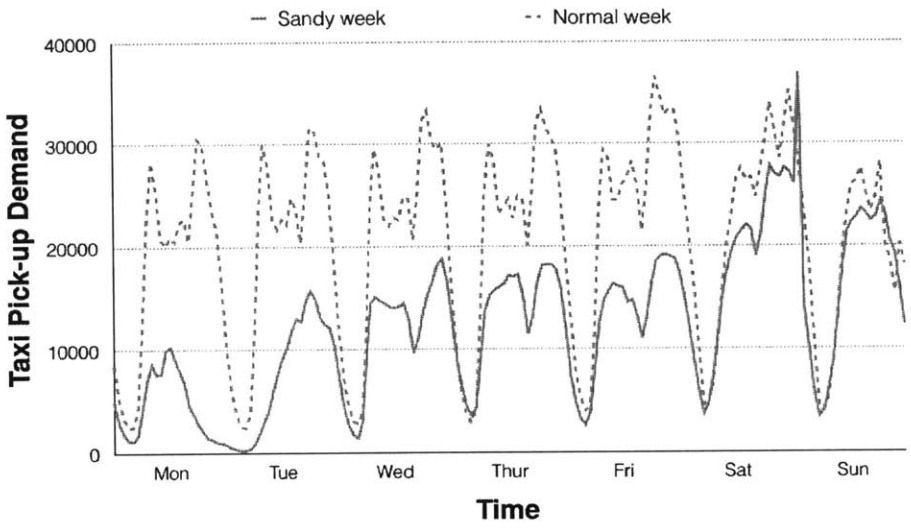

Figure 6-5: Taxi demand pattern of Sandy week vs. normal week

6.5 Summary

In this chapter, we conducted a case study to evaluate the models discussed in chapter 4 and chapter 5. NYC taxi records were used to measure the model performance, in which 10 target locations representing difference typical taxi demand patterns were selected. Three performance measures(MAE, MAPE and RMSE) were used in the experiments to evaluate each model. Results prove the multi-model ensemble's capability to provide better performances over other single models. Also, different

performances of single models in difference locations and time were also analyzed in order to understand the advantages of adaptive forecasting system.

Chapter 7

Thesis Summary

7.1 Thesis contribution

This thesis proposes a real-time forecasting system for taxi demand based on machine learning approaches. Rather than using single models and focusing on overall forecasting performance, this thesis proposes ensemble methods to combine different single models and have them *collaborate* to build a comprehensive and robust model.

Range searching is an important technique to query data within specific ranges, which is useful to preprocess multi-dimensional data. This thesis implements three data structure for orthogonal range searching for the taxi data. They are Kd-trees, range trees and layered range trees. Theoretical efficiencies and practical considerations of these three data structures are discussed. The selection of data structure is determined by the computational power and query frequency. Layer range trees is the optimal choice when fast query is the first priority, while Kd-trees is more appropriate if we only have restricted storage space. Moreover, when there are lots of records to report compared with the total amounts, the query efficiency is dominated by the number of records for either data structure, so that storage complexity should be more considered.

This thesis discusses machine learning approaches to taxi demand forecasting. Six factor categories that may affect taxi demand are analyzed. Single ML models such as K nearest neighbors, artificial neural networks, support vector machines and decision

trees are discussed. Instead of digging into details, we focused on the advantages and disadvantages of each model when facing taxi demand forecasting problem. Traditional ensemble methods including bagging and boosting families are compared. Their limitations draw my attention and motivated me to build an multi-model ensemble estimator that combines single models to provide better forecasting.

A multi-model ensemble estimator consists of five phases: single model training, parameter optimization, ensemble construction, ensemble selection and forecasting. This ensemble estimator is more sensitive to the pattern changing than using single models, which ideally will always find the most appropriate model in real time or construct a new complex model as a combination of several models. In facing taxi problems, big forecasting errors will produce negative impact by misleading the taxi drivers and companies. The weighted combination proposed in this thesis is designed to reduce the variance of forecasted results so as to avoid big errors.

Finally, we evaluate the multi-model ensemble estimator and single modeling through a case study. NYC taxi records are used to test the performance of each model. Rather than measuring models' performances over the whole dataset, we select 10 target locations that each represents specific characteristics and test the models in each location. Results show that no single model has dominant advantages over others. Instead, certain models are appropriate for certain cases. Only the multi-model ensemble estimator preforms significant better than any single models, since it is able to combine the advantages of several single models, provide faster response to the pattern changing, as well as reduce variances to effectively avoid big errors.

7.2 Future research direction

- **Intelligent hyperparameters optimization**

  As stated in the case study, hyperparameters are optimized in each time step. Grid search and random search are two general approaches to find the optimal results. For now optimization among time steps are totally independent. Since each set of hyperparameters is tested and the results are generated fre-

quently, intelligent hyperparameters optimization can be analyzed to improve the computational efficiency as well as optimization effect. For example, hyperparameters perform poorly in several time steps can be removed from the candidate sets.

- **Taxi dispatch system**

  The goal of this thesis is to build a real-time estimator for the taxi demand forecasting, which assists and supports taxi operations. Taxi companies can dispatch taxis according the forecasted taxi demand in the future. A meaningful direction is to build a system to provide decision support for taxi dispatches on the basis of the demand estimator.

- **Incorporate more models**

  The main work of this thesis is to validate the idea of combining single models for the taxi demand forecasting problem. So we only focus on the basic version of the machine learning models. Future research can incorporate the rich variants of those models.

- **Optimization of subsets for building multi-model ensemble estimator**

  The five-phase ensemble estimator contains single model training, parameter optimization, ensemble construction, ensemble selection and forecasting. Independent subsets of data are used for each of the five phases. Further analysis on how to optimize the subset selection for each phase is helpful to improve the performance.

Appendix A

Examples of holiday impacts on taxi demand

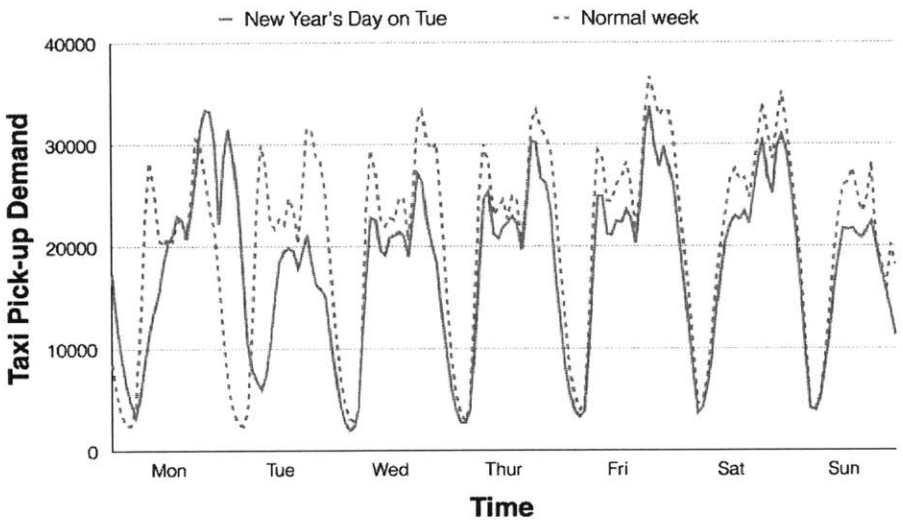

Figure A-1: Taxi demand in the week when 2013 New Year's Day came on Tuesday

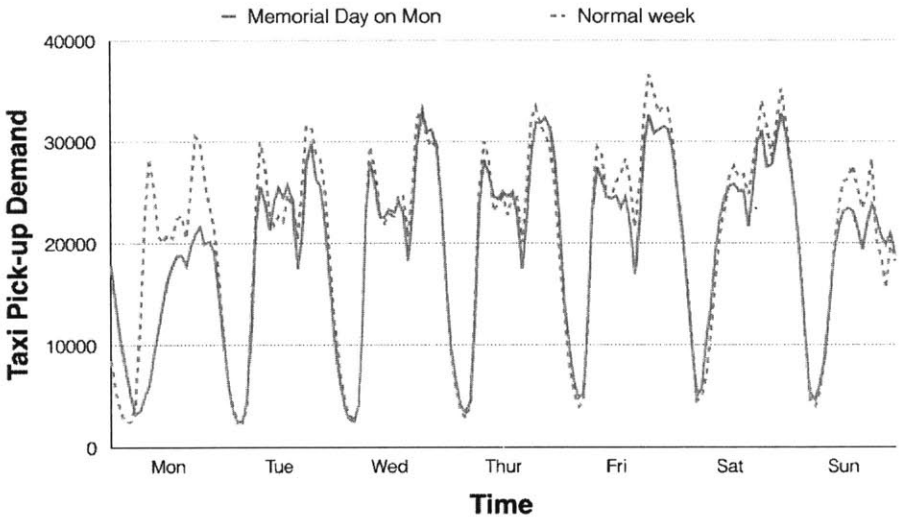

Figure A-2: Taxi demand in the week when 2013 Memorial Day came on Monday

Appendix B

Typical taxi demand pattern in target locations

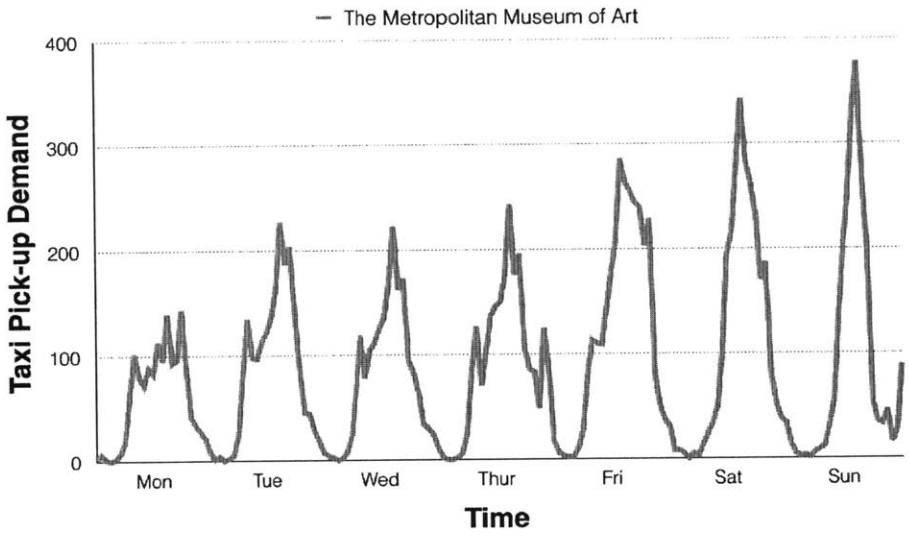

Figure B-1: Typical weely taxi demand in the Metropolitan Museum of Art

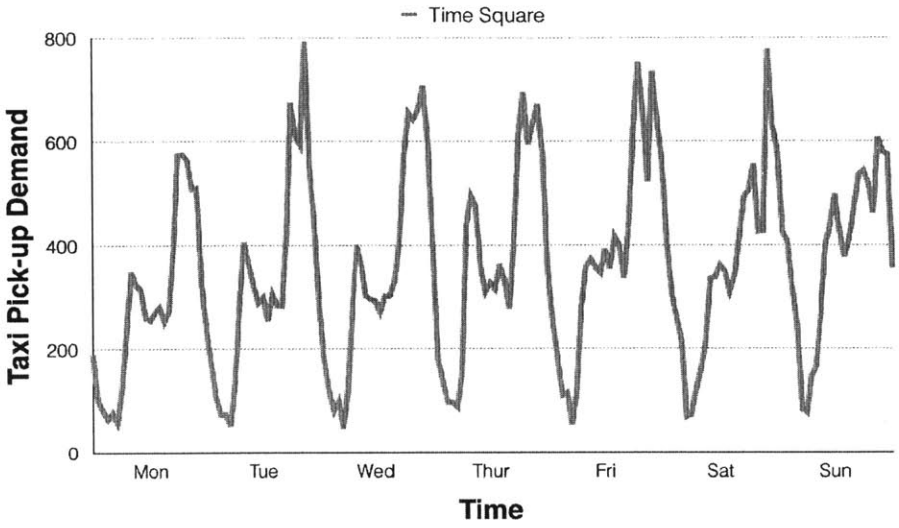

Figure B-2: Typical weely taxi demand in the Time Square

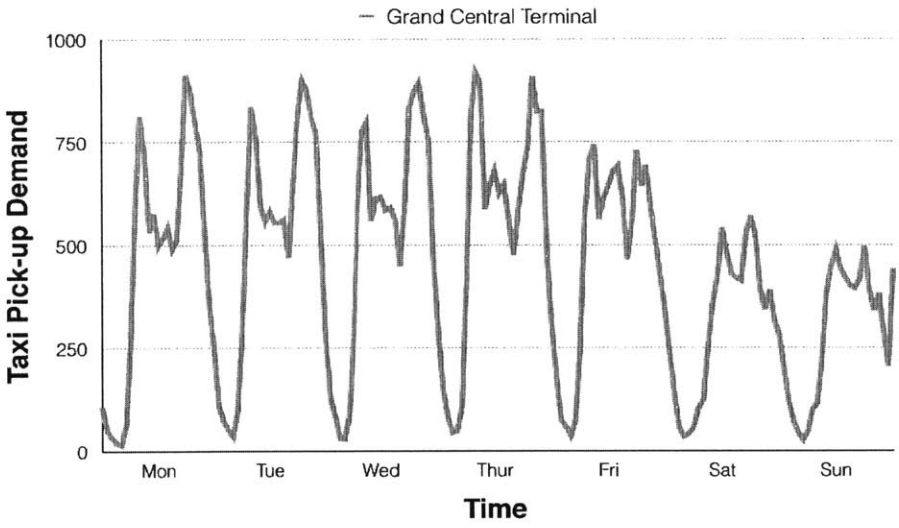

Figure B-3: Typical weely taxi demand in the Grand Central Terminal

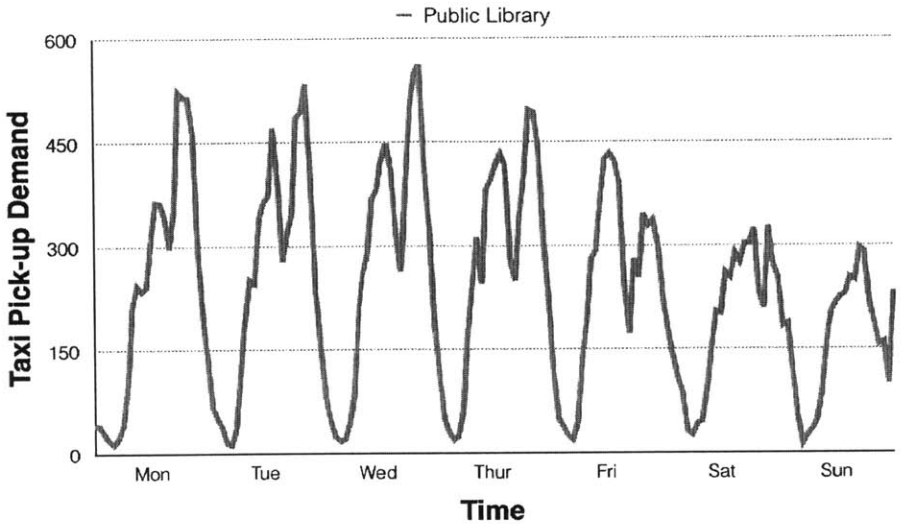

Figure B-4: Typical weely taxi demand in the Public Library

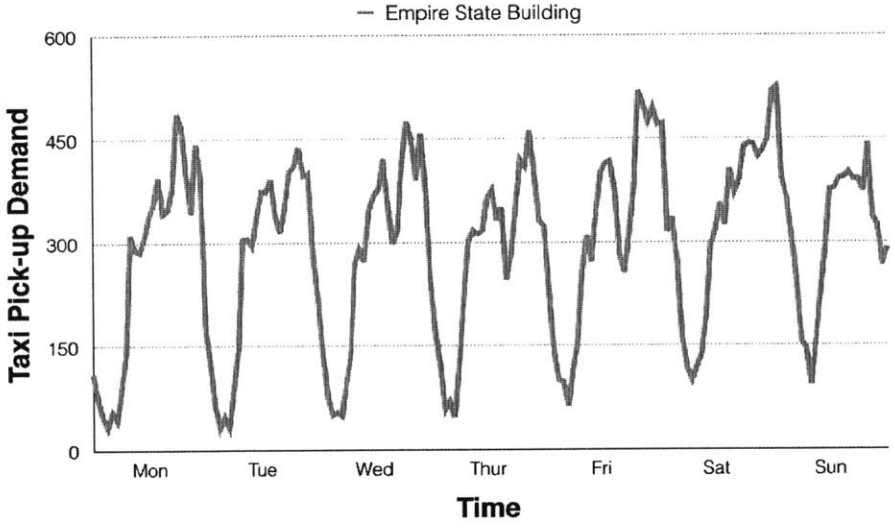

Figure B-5: Typical weely taxi demand in the Empire State Building

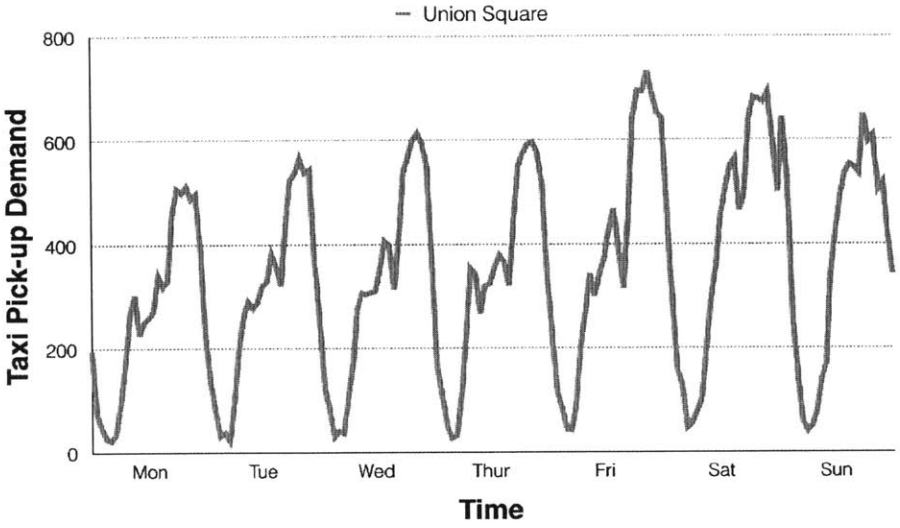

Figure B-6: Typical weely taxi demand in the Union Square

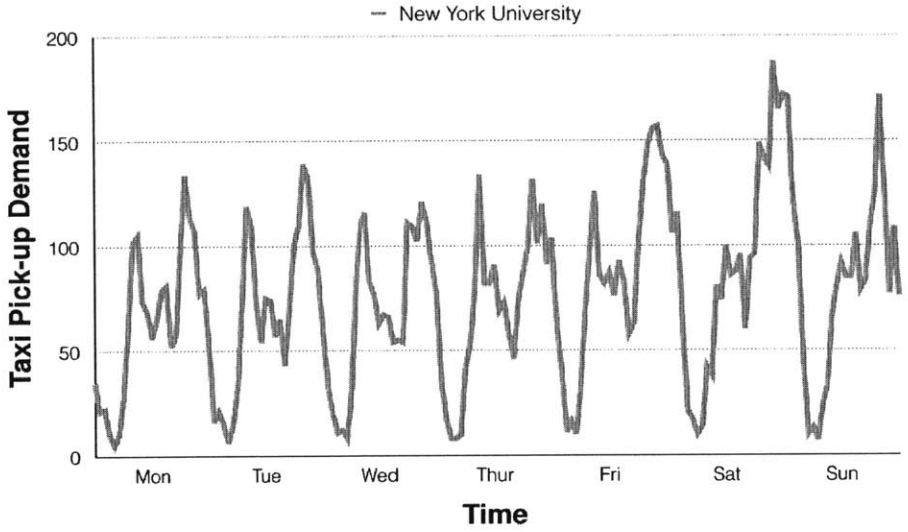

Figure B-7: Typical weely taxi demand in the New York University

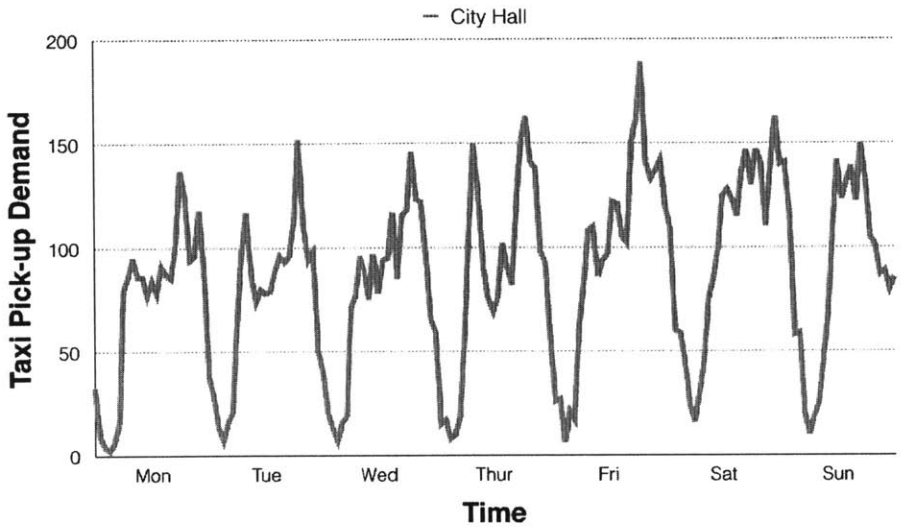

Figure B-8: Typical weely taxi demand in the City Hall

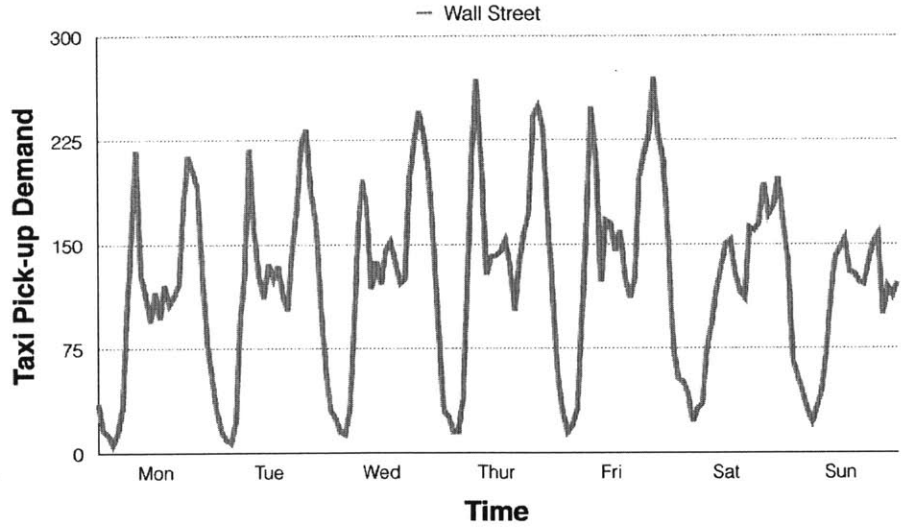

Figure B-9: Typical weely taxi demand in the Wall Street

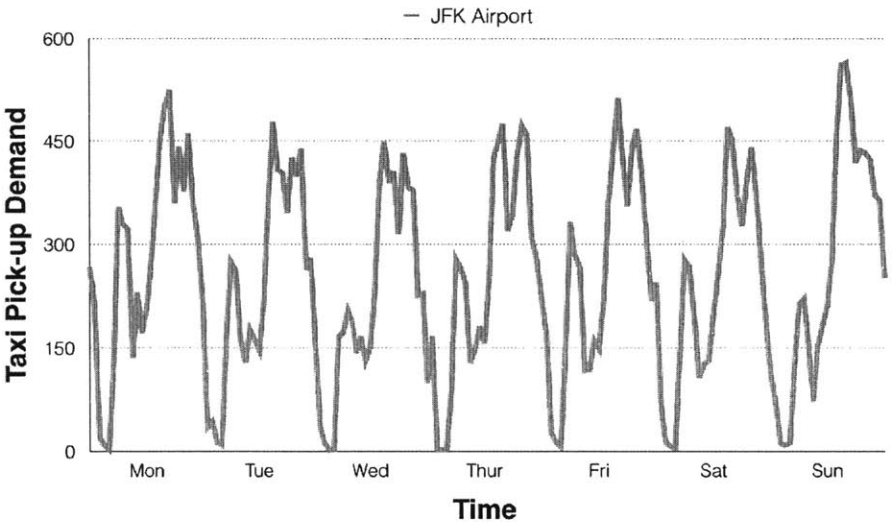

Figure B-10: Typical weely taxi demand in the JFK Airport

Bibliography

[1] A. Al-Smadi and D. M. Wilkes. On estimating arma model orders. *IEEE International Symposium on Circuits and Systems*, 1996.

[2] E. Alpaydin. *Introduction to Machine Learning.* MIT Press, 2004.

[3] A. Azadeh and Z.S. Faiz. A meta-heuristic framework for forecasting household electricity consumption. *Applied Soft Computing*, 2001.

[4] CG.E.P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control.* San Francisco, CA, 1970.

[5] John H. Cochrane. *Time Series for Macroeconomics and Finance.* 1997.

[6] J. Cryer and K. Chan. *Time Series Analysis with Applications.* R. Springer, 2008.

[7] B. Cule, B. Goethals, S. Tassenoy, and S. Verboven. Mining train delays. *Advances in Intelligent Data Analysis X, ser. LNCS vol. 7014*, pages 113–124, 2011.

[8] A.C. de Pina and G. Zaverucha. Combining attributes to improve the performance of naive bayes for regression. *IEEE World Congress on Computational Intelligence*, 2008.

[9] J. Faraway and C. Chatfield. Time series forecasting with neural networks: a comparative study using the airline data. *Applied Statistics*, 1998.

[10] J.W. Galbraith and V. Zinde-Walsh. Autoregression-based estimators for arfima models. *CIRANO Working Papers*, 2001.

[11] J. Gama and P. Rodrigues. Stream-based electricity load forecast. *Knowledge Discovery in Databases: PKDD*, pages 446–453, 2007.

[12] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, pages 779–782, 2008.

[13] C. Hamzacebi. Improving artificial neural netoworks' performance in seasonal time series forecasting. *Information Sciences*, 2008.

[14] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning.

[15] K.W. Hipel and A.I. McLeod. *Time Series Modelling of Water Resources and Environmental Systems.* 1994.

[16] Y. Hou. Forecast on consumption gap between cities and countries in china based on arma model. *3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2010.

[17] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, pages 207–216, 2006.

[18] R.O. Otieno J.M. Kihoro and C. Wafula. Seasonal time series forecasting: A comparative study of arima and ann models. *African Journal of Science and Technology (AJST) Science and Engineering Series*, 2004.

[19] J. Kamruzzaman, R. Begg, and R. Sarker. *Artificial Neural Networks in Finance and Manufacturing.* Idea Group Publishing, 2006.

[20] D.M. Kline. Methods for multi-step time series forecasting with neural networks. *Information Science Publishing*, pages 226–250, 2004.

[21] L.I. Kuncheva and J.J. RodrìÀguez. Combining attributes to improve the performance of naive bayes for regression. *IEEE World Congress on Computational Intelligence*, 2008.

[22] L.I. Kuncheva and J.J. RodrìÀguez. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 2012.

[23] C.F. Lee, J.C. Lee, and A.C. Lee. *Statistics for Business and Financial Economics.* World Scientific Publishing Co. Pte. Ltd, 1999.

[24] B. Li, D. Zhang, L. Sun, C. Chen, G. Qi S. Li, and Q. Yang. Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 63–68, 2011.

[25] X. Li, L. Ding, M. Shao, G. Xu, and J. Li. A novel air-conditioning load prediction based on arima and bpnn model. *Asia-Pacific Conference on Information Processing*, 2009.

[26] X. Li, G. Pan, Z. Wu, G. Qi, S. Li, D. Zhang, W. Zhang, and Z. Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science in China*, pages 111–121, 2012.

[27] L. Liu, C. Andris, A. Biderman, and C. Ratti. Uncovering taxi drivers mobility intelligence through his trace. *IEEE Pervasive Computing 160*, pages 1–17, 2009.

[28] R. Lombardo and J. Flaherty. Modelling private new housing starts in australia. *Pacific-Rim Real Estate Society Conference*, 2000.

[29] Z. Bai M. Guo and H.Z. An. Multi-step prediction for nonlinear autoregressive mod- els based on empirical distributions. *Statistica Sinica*, 1999.

[30] L. Moreira-Matias, J. Gama, M. Ferreira, and L. Damas. A predictive model for the passenger demand on a taxi network. *15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1014–1019, 2012.

[31] G. Ogcu, O.F. Demirel, and S. Zaim. Forecasting electricity consumption with neural networks and support vector regression. *Procedia - Social and Behavioral Sciences*, 2012.

[32] H. Park. Forecasting three-month treasury bills using arima and garch models. 1999.

[33] R. Parrelli. Introduction to arch and garch models. 2001.

[34] D. K. Ranaweera, G. G. Karady, and R. G. Farmer. Economic impact analysis of load forecasting. *IEEE Transactions on Power Systems*, 1997.

[35] B. Schaller. Entry controls in taxi regulation: Implications of us and canadian experience for taxi regulation and deregulation. *Transport Policy 14(6)*, pages 490–506, 2007.

[36] W. Shuai, T. Ling, and Y. Lean. Sd-lssvr-based decomposition-and-ensemble methodology with application to hydropower consumption forecastingn. *Fourth International Joint Conference on Computational Sciences and Optimization (CSO)*, 2011.

[37] A. Sorjamaa and A. Lendasse. Time series prediction using dirrec strategy. *European Symposium on Artificial Neural Networks*, 2006.

[38] H. Tong. *Threshold Models in Non-Linear Time Series Analysis.* Springer-Verlag, 1983.

[39] B. Williams and L. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, pages 664–672, 2003.

[40] K. Wong, S. Wong, M. Bell, and H. Yang. Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing markov chain approach. *Journal of Advanced Transportation 39(1)*, pages 81–104, 2005.

[41] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM*, pages 99–108, 2010.

[42] G. Zhang, B.E. Patuwo, and M.Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 1998.

[43] G.P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 2003.

[44] G.P. Zhang. A neural network ensemble method with jittered training data for time series forecasting. *Information Sciences*, 2007.