Commuter classification and behavior clustering: Beijing use case

Selene Baez Santamaria
s.baezsantamaria@student.vu.nl

Abstract

Public transportation, centered on subway and bus networks, is an data-rich domain that can benefit from data mining and machine learning techniques. The classification of commuters versus non-commuters/occasional travelers can help government, transport management and operators to better target their policies in order to improve the transportation network in large cities. Furthermore, characterizing commuters by behavior clustering can bring deeper insight into their needs and routines as a whole. This project proposes the usage of ensemble models for classification and clustering of public transport users. For this purpose, transit card data will be used, available from the city of Beijing, China.

Contents

1 Introduction

1.1 Urban public transportation

Urban public transportation includes systems that are available for use by anyone in urban areas. Its facilities are commonly composed by buses, subway/metro lines, light rails, tramways, trains, taxis and others. As a network, they provide service for the majority of citizens in urban areas.[26]

Figure 1 shows the passsenger transport usage, as million passengers per kilometer, in several different countries according to the Organisation for Economic Cooperation and Development (OECD). From all OECD countries, the United States, China, Germany, France, Italy, and the United Kingdom contitute the six countries with the most passenger transport, according to their reported data from 2015 or later.[18]

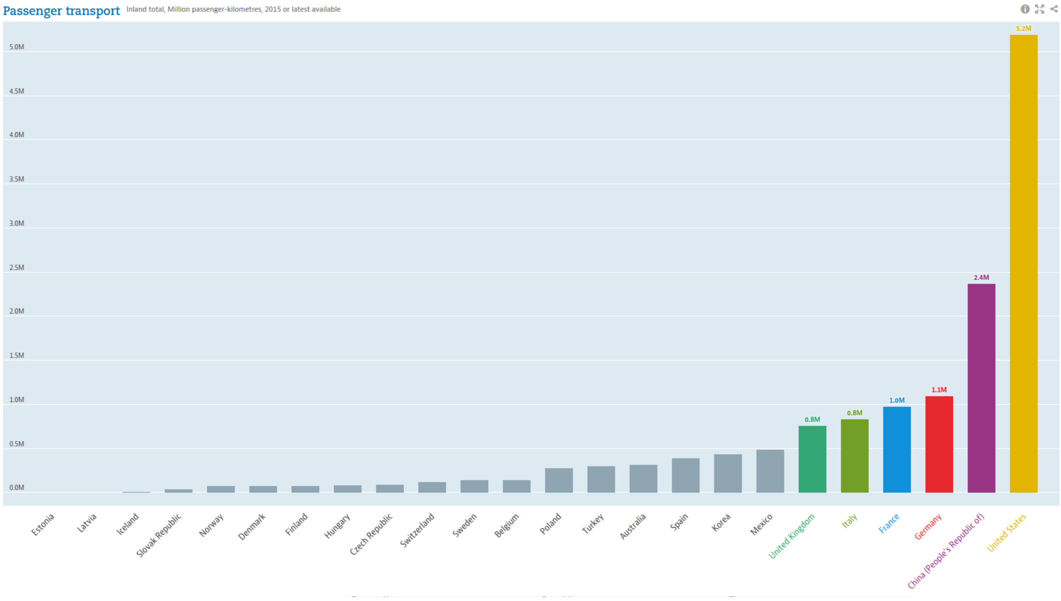


Figure 1: OECD countries and their passenger transportation data.

Furthermore, historical data in Figure 2 reveals the 15 years behavior for each of the aforementioned countries. Most of the countries show stability, with increase or decrease of less than 0.10 million passengers for European countries, and 0.5 million passengers for the United States. China, however, shows a trend with steep increase for most of the selected years. In fact, comparing to its less than 1.2 million passengers in 2000, China doubled its public transport usage to 2.4 million passengers in 2015.

Though it is a more sustainable alternative compared to private car usage, public transport usage has a significant environmental impact, affecting noise and air pollution. Diesel buses, which generally make up a major part of public buses, have large fuel consumption needs and contribute significantly to CO_2 emissions. Even eco-friendly alternatives such as hybrid diesel buses are sensitive to operating conditions, as their fuel consumption may increase by up tp 50% when the on-board air conditioning is on.[32]

Consequently, public transportation directly relates to energetic demand, since its facilities are mostly petroleum or electrical based. In terms of global energy consumption, passenger transportation accounts for about 25% of the total world energy consumption. Furthermore, the transportation sector consupmtion increases at an annual average rate of 1.4.% [10] This may bring further economical implications for countries with high public transportation demand.

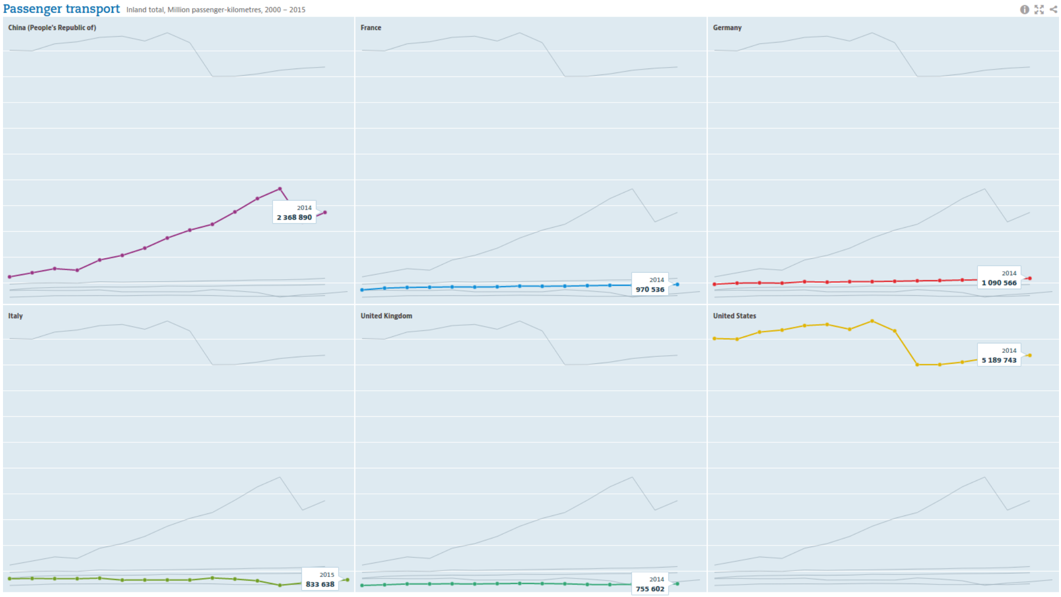


Figure 2: Historical data for the top six countries with most passenger transport usage. China (top-left image) has the steepest increase overall.

1.1.1 Who are the commuters?

A major proportion of public transport users is represented by commuters. These are regular users of public transit, with consistent spatiotemporal patterns in their travels. Driven by a routine, commuters travel back and forth from specific places, for example, from their home to work, school, or other similar locations.

As commuters are frequent users of public transit, the conditions of the public network directly influence their personal well being and generally impact their quality of life. Intuitively, if the commuting experience is unpleasant, daily travel can bring distress to commuters and/or even repel them from using the public transport at all. Several studies have looked into public transit evaluation from different perspectives, including commuters' needs [14]. The most common aspects of it include: travel time, average speed, delays, accessibility, service coverage, crowded level, facilities quality, and fare rate. Weng et al [28] identified five indexes (Convenience, Rapid, Reliability and Comfort) that summarize commuters priorities when choosing to travel by public transport.

From both of the above, the large presence of commuters and their known needs and preferences, it follows that identifying commuters and addressing their needs can help in creating a sustainable public transportation network. Public transit stakeholders should be able to understand the commuters' demands and its dynamics, consequently bringing long term planning and policies for improving the overall commuting experience.

1.2 The city of Beijing

The city of Beijing presents a special case of urbanization and rapid industrialization. This is reflected in a sudden population growth of 20% per decade since 1960, with the largest increase of 44% in the last ten years. The latest official census in 2010 reported the urban agglomeration of Beijing (including Beijing itself and its adjacent suburban areas) having a population of 19,612,368 people. The UN World Urbanization Prospects estimates the 2017 population at over 22 million inhabitants. [23]

As a result of the population explosion, many environmental and social resources are under pressure. From the environmental side, one of the most notable issues is related to air pollution, due to the significantly high pollutant emissions in the city [31]. Similarly, the city's downstream river pollution is serious, with most regions of the Yellow river being unable to comply with the lowest water quality standards. [27]

On the aspect of social resources, one of the main complications is mobility. In Beijing, public transport is the dominant mode of transportation, accounted for 44.0% of all trips compared to 32.6% attributed to private cars [14]. In 2008, the total ridership was 6.5 billion travels. Though the network is continually expanding, it is a fact that public transport is overcrowded, constantly reaching over 100% capacity [19].

Beijing public transport is composed of buses, subway and bicycles. The three types can be accessed by using a single smart card.

Bus: In 2015, there were 876 bus lines with 23,287 buses in operation. The bus network is the most extensive mode of transportation, expanding over 20,186 km. It observes an average daily traffic volume of 10.98 million passengers, with the highest daily volume reaching 13.07 million on one day. [4]

Subway: The Beijing subway has 18 lines with 334 stations, of which 53 are transfer stations. In 2015 it had an operating length of 554 km, with 5,024 vehicles running. [4] Its network is split by two operators: the state-owned Beijing Mass Transit Railway Operation Corp (operating 15 lines), and the joint Hong Kong venture Beijing MTR Corp (operating 3 lines).

Beijing's subway has an average daily traffic volume of 9.11 million passengers, with a maximum recorded volume of 11.66 million passengers. As such, it is the second busiest metro system in the world, providing 3,410 million annual journeys. Compared to the service provided in 2012, the system observed a 39% increase in usage by 2014. It is also the second longest metro network, surpassed by Shanghai by only 21 km. [25]

Bicycles: Beijing first implemented public bicycle systems in 2012. As of 2015, in total, 67,000 bikes are available for rental with 2,700 pick up/drop off points spread across the city. [4]

1.3 Smart cards and Big Data

Smart cards present us with a straightforward way of massively collecting daily data. In the last years, smart card systems have become more popular in the Transportation domain, making it possible to monitor travelers transactions and facilitating fare collection. Several cities have implemented such systems, for example the Octopus card in Hong Kong[6], Oyster card in London [3], OV-chipkaart in The Netherlands [7], and Yikatong card in Beijing [5], to name a few.

In Beijing, over 90% of public transit users are smart card holders. There is a significant incentive for using the Yikatong smart card since bus rides are heavily subsidized (users have only to pay 50% of the full price)[12]. Moreover, the Yikatong smart card system is also integrated with taxi, electricity and sewage payments, making it convenient to use as a general paying method.

Data quantity Placed in context, public transit systems serve at least hundreds of users daily, where a typical user performs several trips a day, every day. On the specific case of Beijing, there are hundreds of thousands of smart cards gathering between 5 and 16 million records (trips) a day, among a large complex network containing thousands of routes and tens of thousands of stops.

Data quality However, though smart cards exponentially increase the quantity of data, they do not completely guarantee its quality. As is, some aspects of the trips cannot always be faithfully recorded but are inferred (for example, the transfers between the subway system when no check-in/out is done at changing trains). Furthermore, some fields are sometimes simply missing or incorrectly recorded due to malfunctions and situations out of control.

Given the large amounts of data collected and its nature, the analysis of such becomes challenging. Transit smart cards are capable of recording spatiotemporal information at an individual level over long periods of time. This generates a large volume of historical data that only tailored big data techniques can deal with.

1.4 Project motivation

This project performs an interdisciplinary study between the areas of Artificial Intelligence and Metropolitan Transportation. It is focused on introducing data mining techniques to a data rich domain.

The area of Artificial Intelligence is able to provide dozens of prediction algorithms. Though constantly under refinement, it is time for state-of-the-art techniques to be applied, tested and validated under real and large impact situations to test their ability to deal with noisy streams. Comparably, given the ever growing complexity of urban mobility, domain experts must focus on analyzing trends and insights instead of curating and making sense out of raw data. As such, introducing these state-of-the-art techniques into the Metropolitan Transportation domain can aid to unravel massive human behaviors and reveal patterns and trends in mobility.

1.4.1 Societal context

Identifying and analyzing mobility patterns may have different goals, from description, prediction or prescription, all of which affect their stakeholders directly.

A descriptive analysis determines how people use the public transit. It can pinpoint chaotic hotspots in the city, peak hours, popular routes or other behaviors. A predictive analysis investigates how will people use the transit in the future or under new circumstances. For example, public transport usage projections in the years to come directly affects environmental models trying to improve air and water quality, energetic demand or other natural and economic resources.

Finally, a prescriptive analysis focuses on how should the different stakeholders deal with mobility behaviors. For example, the government as well as transport management and operators would gain invaluable spatial and temporal insights regarding commuters' behaviors. This insight may lead to tangible results, including policies for increasing the efficiency of the public transit network, adjustable travel fares tailored to the most relevant mobility patterns, incentives to relieve peak hours and thus traffic congestion, urban planning for residential and industrial land use, and others.

Given that Beijing has a widely spread data collection system, combined with formidable institutions capable of introducing new measures in their public transportation network, this city is an excellent use case where the results of an in-depth study can generate actionable plans and bring benefits in the short and long term.

Furthermore, the social context of Beijing presents specific opportunities for improvements where the full power of data analysis and its impact can be tested. For example, the city of Beijing faces a large imbalance between residential and working areas. Due to urban expansion, most residents have been forced to move to suburban areas due to the lack of affordable housing, regardless of having their work environments within the six Ring Roads [33]. Investigating and targeting this group could alleviate the pitfalls of long distance commuting.

1.4.2 Scientific context

Mobility patterns in metropolitan areas follow complex swarm behaviors. Based on individual travels and routines, travelers exhibit distinguishable characteristics on a larger scale. Both individual and collective levels of understanding are crucial for Transportation experts. In order to explore both levels, Metropolitan Transportation studies typically focus of the usage of surveys. These surveys are targeted to reach travelers on an individual level, while large scale indicators and aggregated data are taken to investigate their collective behavior.

These methods have several disadvantages. On the one hand, surveys are costly to implement, and in general have problems related to small non-representative samples. Even when these problems are escaped, the usual quality versus quantity trade off is present, reducing the confidence of the collected information. On the other hand, large scale measurements (i.e. total passenger flow) miss the interactions between individuals that cause the collective behavior.

On top of this, an important consideration on the Metropolitan Transportation domain is that the data collected by smart cards is unlabeled. This means that traveling behaviors are not assigned to known specific categories, making it hard to validate and evaluate. Typically, this issue is address by asking some sample users -via surveys- how they categorize themselves (for example, if they consider themselves to be commuters) and then extrapolating this profile to new users. However, self-reported data by itself has bias problems, therefore introducing noise or false patterns.

Fortunately, the field of pattern recognition has seen major development in the last years. Nowadays, there exist machine learning and other data mining methods specialized in analyzing disaggregated complex information. Data analysis can be as general is specialized as needed, producing reliable and comprehensible information and visualizations. Furthermore, unsupervised tools have arisen that find patterns based on the data alone, thus being independent from the aforementioned biases.

1.5 Thesis organization

The thesis is organized as follows: On the next section we perform a *Literature review* to explore previous work on mining smart card transit data. We also summarize current representation and pattern recognition methodologies for dealing with complex spatiotemporal data.

Subsequently, we establish the *Research framework* where we explicitly state the objectives and research questions of this project. As a result, we limit the project's scope and clearly define the most important terms to be used.

We continue to describe the *Methodology* thoroughly. This consists of an extensive description of the data and its characteristics, our proposed 3 dimensional representation for spatiotemporal data, and the data mining approach to follow, including supervised and unsupervised learning techniques for dimensionality reduction and pattern recognition.

Following this, we identify three distinct stages of the project: *Data preparation and preprocessing*, *Commuters identification*, and *Traveling behavior clustering*. In the *Data preparation and preprocessing* section we describe the pipeline for processing raw data, extracting trip attributes and finally creating the proposed 3D representation of an user's traveling behavior.

The section on *Commuters identification* describes a supervised learning approach for classifying labeled data, using feature selection and ensemble models. Its counterpart, the section on *Traveling behavior clustering* describes an unsupervised learning approach to recognize similar traveling behaviors, using feature extraction (by means of an autoencoder) and clustering algorithms.

Finally, we gather conclusions regarding the proposed representation. We compare both supervised and unsupervised approaches and explore future work opportunities.

2 Literature review

In this section we look at studies within the last decade that are related to smart card transit data. First, we summarize the approach and the most relevant findings of each paper in order grasp a broad view of the Transportation domain. Secondly, we explore representations for spatiotemporal data and compare the way traveling behavior is usually represented in the Transportation domain, and other types of representations available in the Artificial Intelligence domain. Thirdly, we discuss techniques for pattern recognition through supervised and unsupervised learning.

2.1 Data mining on transit card data

With the introduction of smart card systems in large cities, several studies have aimed to extract knowledge from the large amounts of data collected. Many of this studies focus on analyzing traveling behavior, which is regarded as a spatiotemporal mobility pattern. Though different in their methodology, results concerning commuters are duplicated across studies. As the spatial and temporal regularity of commuters' travel behavior is evident in their smart card data, they pose an excellent opportunity of study.

Morency et al. study spatio-temporal variability in Canadian smart card data. On the one hand, they examine spatial variability by measuring the number of distinct stops a smart card user visits, and the frequency of each stop. On the other hand, they examine temporal variability by clustering the boarding times of each type of smart card. Using these features, they observe the week to week variability for each of the five types of transit card available (Adult-interzone, Adult-express, Adult-regular, Elderly and Student). Their findings show that commuter types of cards visit a smaller range of bus stops compared to non-commuter types. Therefore, a small number of stops account for a high proportion of commuter's boardings. Additionally, commuters have the highest proportion of zero-boarding days on weekends [16].

Bhaskar et al. are concerned with passenger segmentation using Australian smart card data. First, they perform a two level DBSCAN algorithm for investigating spatial patterns, where the first level clusters Destination stops and the second level clusters Origin stops. From this they extract frequent Origin-Destination (O-D) pairs. Separately, they applied DBSCAN to temporal features to determine most frequent boarding times. As such, they characterize each user by the percentage of journeys they perform between the regular O-D, and the percentage of journeys they perform during their habitual times. Users with at least 50% spatial and temporal regularity are thus classified as transit commuters; while users with no evident spatial or temporal pattern are classified as irregular passengers. The authors find that while most (64%) of the passengers riding the public transit are irregular passengers, it is transit commuters who bring the most (46%) revenue. Furthermore, they find that irregular passengers prefer high frequency routes significantly more than transit commuters, arguing that commuters are usually on a time habit, and thus are more willing to check and adapt to public transit timetables. [2]

Tu et al. follow a supervised learning approach to classify public transit users in Beijing as commuters or non-commuters. In order to produce labeled data, they convey an online survey asking for travel patterns and smart card ID. Matching the ID to the journeys recorded by smart card during the span of one week, they collect records associated to 978 travelers. The classification is then performed by a Support Vector Machine (SVM), which reaches up to 94.24% accuracy. [24]

Langlois et al. present an innovative representation for smart card data. Using four weeks worth of data from London Oyster cards, they represent the card information as a time-ordered sequence of inferred activities. 11 clusters are found and characterized by evaluating socio-demographic variables like age, employment, annual household income, children per household and vehicles per household. The authors further grouped the clusters under "working day", "home bound", "complex activity pattern" and "interrupted pattern" categories. Their findings show that four clusters, grouped under the "working day" category have significantly different activities during weekdays as compared to weekends, with some avoiding transit during the weekends and others visiting different areas. Four more clusters, grouped under the "home bound" category, are characterized by staying mostly at their primary area and low number of traveled days. [11]

One of the latest work on the field corresponds to Ma et Al. The objective of their work is to determine a scoring function for travelers that can correctly identify them as commuters, or non-commuters.

In their work, they cluster stops using an improved DBSCAN algorithm. They engineer features for representing the frequency in which travelers follow spatio-temporal patterns. Travelers are then clustered according to these features following the ISODATA algorithm. As an output of the clustering, optimal cutoff levels in the scoring function were determined. As a result, evaluating a traveler does not depend on clustering centroids, but only on calculating the commuting score. This, as expressed by the authors, reduces computing time and treats each traveler independently from the others, which is not true for clustering algorithms [12].

A common practice, as used by [12], [11], and [16] is to divide the day into -hourly or half-and-hour-time bins. Bhaskar et al. recognize this as a problem in the field, by pointing out that this design choice segregates journeys from 9:59 AM and 10:01 AM even though they intuitively belong to the same behavior.

2.1.1 Volume of data

The volume of data collected by smart card systems is massive and is usually impossible to analyze all of it at once. The volume of the samples analyzed by previous work ranges from hundreds of smart cards to tens of millions of smart cards, leading to up to hundreds of millions of individual smart card transactions. The details of the revised literature are summarized in Table 1.

Authors	Year of publication	Records	Unique smart cards	Time span
Morency et al. [16]	2007	2.2 million	7,118	277 days
Ma et al. [13]	2013	Unknown	3 million	one week
Ortega [20]	2013	65 million	5.7 million	one week
Bhaskar et al. [2]	2015	34.8 million	1 million	4 months
Tu et al. [24]	2016	8,067	978	one week
Langlois et al. [11]	2016	3 million	33,026	four weeks
Ma et al. [12]	2017	364 million	18 million	one month

Table 1: Volume of data analyzed by different authors

Given the limit on how many records can be examined per study, researchers usually face the decision to reduce the dataset to a manageable size. As such, there exists a trade off between the number of unique smart cards and the time span of the collected data. Some researchers, like Ortega [20], decide to analyze a large population over short periods of time. Others, like Bhaskar et al. [2] choose to explore long term behavior thus having to reduce the population size.

However, it is worth noting that the total number of records studied has increased overtime. This most likely is due to the trends of increased computational power and the design of optimized mining algorithms. A clear example is the study by Ma et al. [12] published just this year that was able to include data of a significantly large population over a month.

2.2 Representing spatiotemporal data

As Marr puts it, representations make explicit different types of information implicit in entities [15]. Thus, representations mainly differ in the information they describe and the way they describe it. Usually, representations are generated to achieve a information processing goal. Thus, the value of a representation depends on the purpose of the task it will be used for.

Data representation is one of the fundamentals in data mining. Ideally, the representation of a data point is comprehensive of its underlying unique factors and leaves out unnecessary or noisy information. Furthermore, the format for the representation must be akin to the types of information that data mining algorithms can process. Therefore, finding suitable representations for complex concepts like space and time is not an easy task.

2.2.1 Traditional feature engineering

Human mobility is intrinsically tied to spatio-temporal properties. Still, the greatest amount of studies analyze public transit journeys by separating spatial features from temporal features. Furthermore, in general scalar aggregated features are used for users characterization. Some examples are:

- **Frequency indicators:** number of traveled days [2] [11] [12], number of journeys [2], number of times a stop was visited [16], number of days with zero boardings [16], most frequent home/work stop [12], most frequent home/work route [12], most frequent departure time from home/work [12], number of trips to the most frequent home/work stop[12], number of trips following the most frequent home/work route [12], number of trips during most frequent departure time from home/work [12]

- **Range/coverage indicators:** distinct stops visited [16], spread of days between the first and last journey [11]

- **Calendar-based indicators:** observed day [16], day of week [16]

Though popular among the Transportation domain, hand engineered features may present great disadvantages. While these features are intuitive and semantically meaningful for Transportation specialists, they do not always represent distinctive properties of users or their public transit journeys. Therefore, the time invested in designing and producing features may not always payback in relevant findings.

This case can be compared with the trends seen in Computer Vision. A few decades ago, most approaches for Image Understanding were focusing on designing features to describe them (i.e. SIFT). However, after the rapid development of Neural Networks in the last years, the most successful Vision applications are based on learned features. As Nithin and Sivakumar explain, hand crafted features are time consuming, fragile and incomplete, thus being outperformed by automatically extracted features which learn better the underlying representations in images [17].

2.2.2 Feature extraction

Feature extraction refers to the creation of features that represent the underlying characteristics of data. These features are automatically created, solely from the data, in either statistical or learned ways. One popular way for extracting features is by using methods for dimensionality reduction, which beyond finding representations further tackles the curse of dimensionality.

Principal Component Analysis

One of the most robust algorithms for this is Principal Component Analysis (PCA) which is a mathematical tool used across several domains. By doing matrix manipulation, PCA extracts eigenvalues and eigenvectors from a given dataset. The top eigenvectors represent the ways in which the data points are more different from each other.

An isolated work related to this was performed by Langlois et al. Following a unique methodology for engineering features, first they represent the travel data per user using a three dimensional matrix where x represents the day in the four week period, y represents the hourly time bin, and z represents the area where the inferred activity took place, encoded as a one hot vector. The authors perform PCA for dimensionality reduction, based on Eagle and Pentland's eigenbehaviours [9]. An analysis of the average correlation of the first 13 components, results in the selection of the first 8 components as the most informative and stable. The projections of a user sequence onto these components (called weights) constitute the features to be clustered using k-means. [11]

Autoencoders

Following the same main principle as the first neural networks, autoencoders are highly connected networks that map high dimensional data to low dimensional spaces. Primarily used for images, the goal of an autoencoder is to deconstruct an input image onto a representation, and reconstructing the image again, with a minimum loss of information. Each of these are called the Encoder and Decoder modules, respectively. Together these are learning modules that tune its parameters until achieving sufficient performance.

Much research has been done on autoencoders, leading to several variations of them. Denoising autoencoders result in a more robust algorithm, since they get a corrupted image as input, but aim for reconstructing the original image. Therefore, the autoencoder does not simple map one instance to a representation, but truly learns the significant characteristics present in the data [17].

To the best of our knowledge, these techniques have not been introduced to the Transportation domain.

2.3 Pattern recognition on spatiotemporal data

2.3.1 Classifying algorithms

The domain of Metropolitan transportation faces a specific problem: although smart card systems have allowed massive collection of data, this data is not labeled regarding commuting behaviors. Additionally, obtaining labels for smart card data is expensive and unreliable, since it has to be acquired through surveys or interviews. Furthermore, even when labels are obtained, the amount of labels obtained is often insufficient for big data analysis. It is due to these reasons, that most studies are inclined to used unsupervised learning techniques.

One of the few studies that uses labeled data corresponds to Tu et al. They obtain 978 labeled records, with an almost equal distribution of records over both classes (49.18% related to commuter samples and 51.82% related to non-commuter samples). They solve the issue of limited samples by selecting a model that is not heavily affected by sample size: Support Vector Machines. Using a linear kernel, and 7:3 ratio for train-to-test sets division, their results report a 94.24% accuracy over a test set of 295 samples.

2.3.2 Clustering algorithms

If labeled data is not available, then unsupervised learning techniques must be applied. There is a large variety of clustering algorithms available nowadays, however not all of them are suitable for all types of data and purposes.

Hierarchical clustering

Langlois et al. use agglomerative hierarchical clustering for areas clustering. In order to infer the user-specific activities, all stops or stations visited by each user are clustered by merging the two closest areas until a threshold distance is reached. Their algorithm also considers the distance between stops and the frequency of travel between them. Therefore, different activities are likely to be associated with different areas [11].

Partitional clustering

The K-means algorithm is the most widely used method for partitional clustering. It requires having a predefined number of clusters to fit the data to.

Morency et al. use K-means for clustering hourly boarding times according to card type. They apply Hamming distance (representing the percentage of data between two elements) and a combination of batch and online updates. Through empirical tuning, they select to find four clusters per card type. It is worth noting that by using a card-day unit, they allow a card to belong to a different cluster according to the day of travel. As every card type is composed of four boarding patterns, travelers are not restricted to follow a routine everyday, but can exhibit different behaviors on different days. For example, the Adult-regular card type contains a 9:00AM-and-5:00PM-boarding cluster and a no-boarding cluster. Thus, a user of this card could belong to the first cluster on weekdays and to the second cluster on weekends. [16]

Bhaskar et al. apply K-means for binary classification purposes. As such, they classify frequent and infrequent transit users, using the number of traveled days and the number of journeys made as features. Unfortunately, K-means performs poorly since no distinct clusters are evident. The most likely cause for the previous is the strong correlation between traveled days and journeys, combined with the authors oversight of whitening and standardization techniques. [2]

Langlois et al. use K-means to find clusters of activity sequences. They employ specialized sampling techniques, like bootstrapping, to deal with big data. Moreover, they tune the algorithm parameters using the DB-index, which is the ratio of the within cluster distances to the across cluster distances. They find two optimal number of clusters (4 and 11), out of which they select the largest to provide the most detailed segmentation. They further perfection the algorithm by using k-means++initialization over 150 replications. Additionally, this paper acknowledges that clustering techniques are sampled based, which means different samples may find different optimal solutions. The authors validate their approach by analyzing the stability of the clusters over samples obtained at different points in time. By extracting the same number of clusters and fitting the samples to each set, they find that 91% of users are assigned to their equivalent clusters. [11]

Density based clustering

Density based algorithms excel at dealing with anomalies, since they ignore low density areas and interpret them as noise. They do not required a redefined number of clusters and adapt to find clusters of any size. The required parameters for DBSCAN are a maximum reach distance ϵ and the minimum number of points per cluster.

Bhaskar et al. use three DBSCAN algorithms to cluster Origin stops, Destination stops, and boarding times. For each of the previous, they tune the algorithm parameters by fixing a domain reasonable ϵ (1000 m walking distance or 5 min variance in boarding time), and selecting the minimum points by comparing the percentage of data considered to belong to any cluster as opposed to data considered to be noise given the par-specific parameters [2].

Ma et al. use an improved DBSCAN algorithm to cluster bus/subway stops. In their approach, abnormal stops are not considered noise, but are allowed to be re-clustered by splitting large clusters into several smaller clusters.

Though clustering algorithms are common in the field, they are not always used for classifying users. For example, Bhaskar et al. use density based clustering for engineering regularity features. However, the classification of users is rule-based according to which feature (spatial or temporal regularity) is stronger in each user [2]. Morency et al. use partitioning clustering to characterize existing user categories according to their boarding times [16].

As a conclusion, we note that while there has been research applying basic clustering and classification algorithms, most studies lack further specialized data mining techniques for preprocessing data, tuning algorithms parameters, and/or visualizing results.

3 Research framework

The underlying goal of this project is to find an accurate spatiotemporal representation for public traveling behavior while accounting for big data constraints and the inherent data nature. The main two objectives are:

Objective 1 To identify commuters based on their routine patterns.

Objective 2 To group users with similar travelling behaviors.

Combined, these objectives characterize public transit users in the city of Beijing. For this project we decide to use one month worth of smart card data, since we believe it to be a long enough period to see different traveling patterns, while keeping the data at a manageable level.

3.1 Research questions

The main objectives is further broken down into answering the following research questions:

1. How can spatiotemporal features be analyzed as a unit?
2. What are the most relevant features when identifying commuters?
3. How accurately can commuters and non-commuters be identified using an ensemble model?
4. How many distinct behaviors are present among public transport users in Beijing?
5. How does feature selection and feature extraction compare to each other in the transportation domain?

3.1.1 Definition of terms

A commuter is a public transit user whose smart card data reveals repeatable patterns in time and space. Though commuters are usually associated with Monday to Friday 9:00am to 5:00pm schedules, in this work we extend the definition to any routine travel pattern. This flexibility allows us to include travelers with stable yet rare commuting schedules, such as night workers, weekend workers and evening workers.

A trip is a sequence of smart card transactions, including transfers, performed by the same user to travel from an origin to a destination. A trip is also represented as a record in the data, as it will be further explained in Section 4.1

A transfer is a change in transportation mode, or a change in vehicles whenever a smart card has to be checked within the same transportation mode. Transportation modes include Bus, Subway, and Bike.

We make the assumption that smart card IDs and users have a one to one relationship, meaning each user has exactly one card and each card is used by exactly one user. As discussed with domain expert Quian Tu, although some people may own more than one card, this is a minority. Thus, the assumption holds for the majority of travelers.

3.2 Scope and structure

This project is divided three main stages:

PART I: Prepare and preprocess the data using Big Data techniques In this part we focus on research question 1. Techniques for cleaning, knowledge extraction, categorization, patching and standardization are used and tailored to the data. From this, we build an appropriate 3 dimensional representation for each user's traveling behavior. This part corresponds to Section 5.

PART II: Classify commuters versus non-commuters by using an ensemble model In this part we focus on research questions 2 and 3. First, we perform feature selection in order to identify the most informative features and disregard redundant information. An extensive analysis of spatiotemporal properties is done, combining transportation domain knowledge, machine learning and statistical tools. Subsequently, we create a classifier using ensemble models and discuss its performance. This part corresponds to Section 6

PART III: Users clustering according to patterns in their travel behaviors. In this part we focus on research question 4. First, we do feature extraction with the goal of reducing the dimensionality of the data. This is done via a convolutional autoencoder. Finally, we cluster the low dimensional representation using k-means clustering and do cluster analysis to understand the underlying pattern of each cluster. This part corresponds to Section 7

Figure 3 displays a flowchart for the stages and their connection.

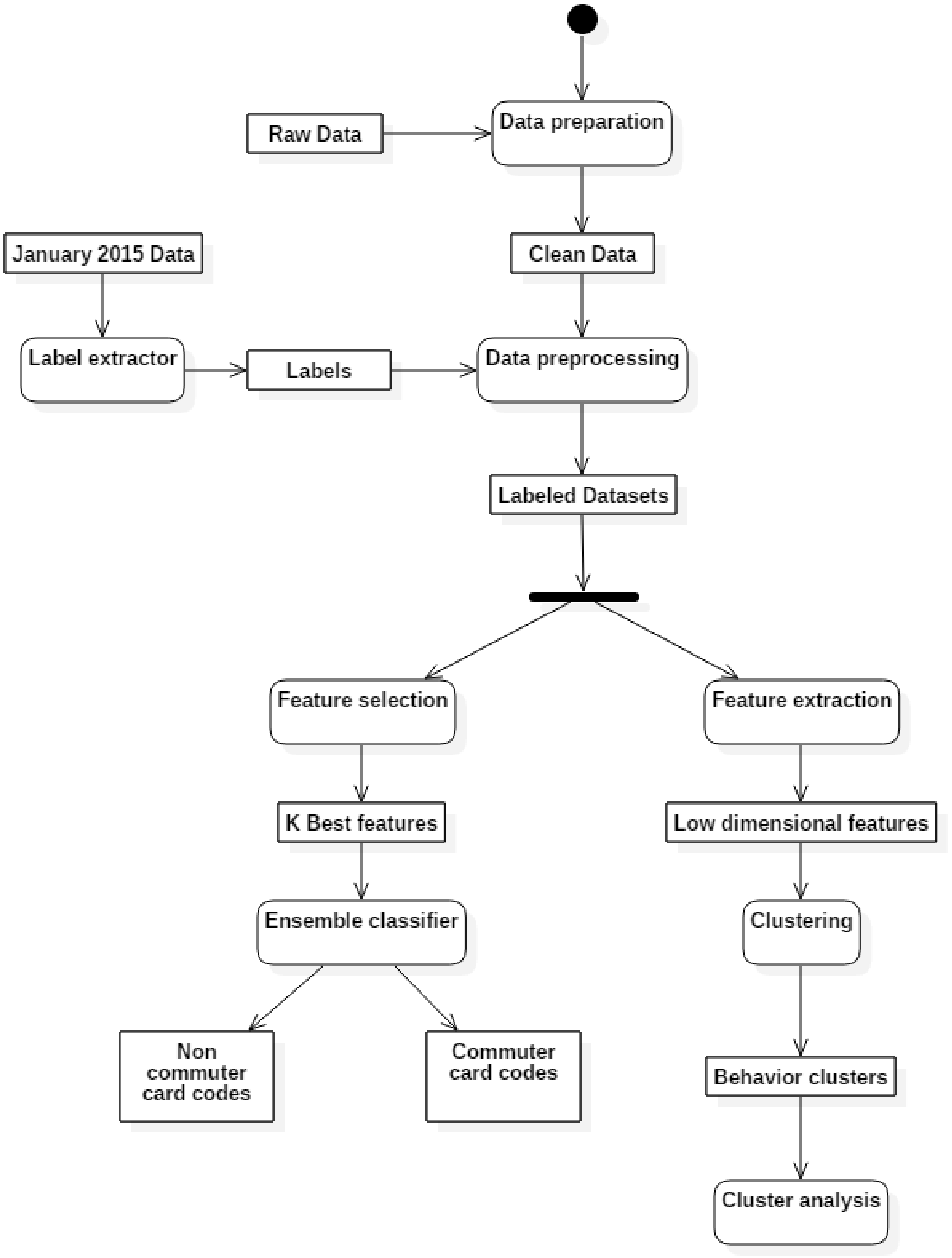


Figure 3: Project flow

4 Methodology

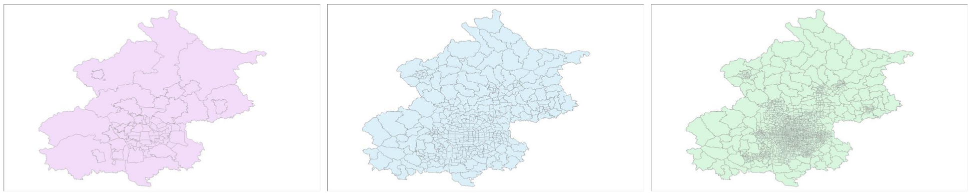
4.1 The data

The data used in this project is provided by government agency of Beijing Transportation Operations Coordination Center (TOCC), facilitated by the College of Metropolitan Transportation at Beijing University of Technology. Every record in the data represents a trip performed by a specific smart card. As such, it contains the following data fields:

- Data date: Year, month and day that the trip was made
- Card code: Card identification number
- Path link: Mode of transportation. B stands for bus, R for subway, Y for bicycle. Transfers between modes are shown by a dash. [1]
- Travel time: Time spent in vehicles, measured in milliseconds
- Travel distance: Distance traveled, measured in meters as performed by route.
- Transfer number: Number of changes in travel mode during the trip.
- Transfer total time: Total time spent in transfer, measured in milliseconds
- Transfer average time: Time spent in transfer, divided by number of transfers. Measured in milliseconds
- Start/End time: Time stamp of when the trip started/ended. Date and time up to milliseconds precision
- On/Off small traffic area: Integer ranging from 1 to 1911
- On/Off middle traffic area: Integer ranging from 1 to 389
- On/Off big traffic area: Integer ranging from 1 to 60
- On/Off ring road: Integer ranging from 1 to 6
- On/Off area: Integer ranging from 1 to 18
- ID: record identification number created by joining the following: hour of the beginning of the trip | time stamp of beginning of the trip | card code performing the trip
- Transfer detail: Mode of transportation, as well as line/route number and stations for boarding and alighting. More detail provided in Section 5.2.2

Full privacy of card users is ensured, as there is no personal data linking card codes to specific individuals.

The traffic zones (small, middle and big areas) are administrative divisions by the Beijing Municipal Institute of City Planning and Design (BICP). They are specific in different degrees, as shown in Figure 4. In general, the division principles correspond to the geopolitical environment and administrative planning, for example roads, villages and others. The 6 ring road and 18 areas districts are administrative divisions by the Beijing Municipal Government. The division is unique in Beijing. The 18 districts and counties are shown in Figure 5. According to domain expert PhD. Liang Quan, these divisions are sufficiently informative for traffic analysis [22].



(a) Big zones

(b) Middle zones

(c) Small zones

Figure 4: Traffic zone division

[1] Example: B-B represents a Bus to Bus transfer.

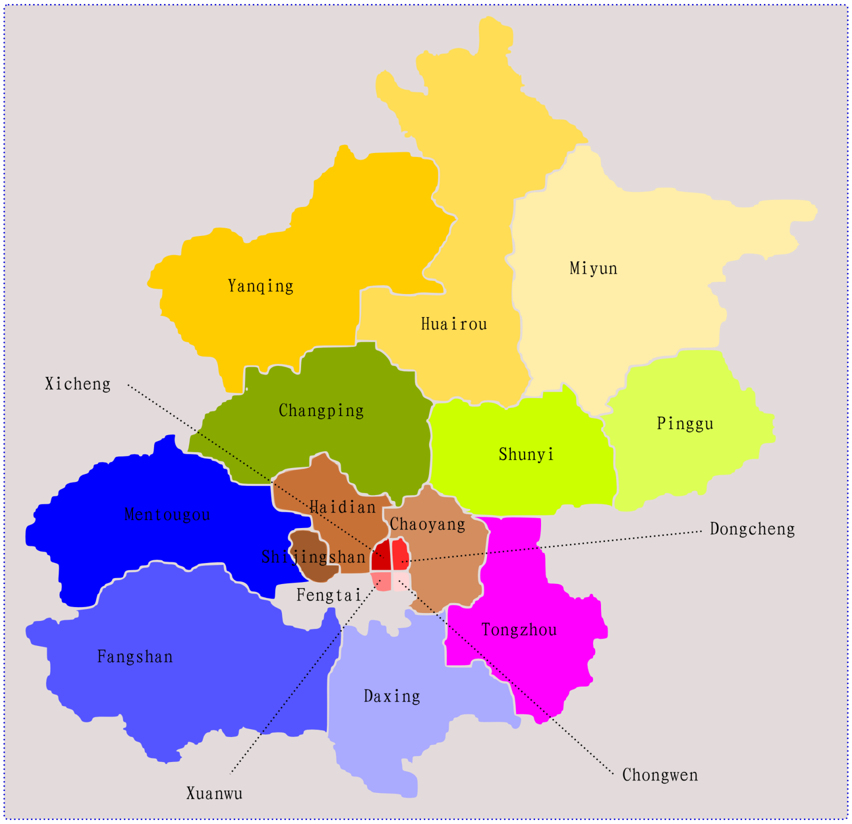


Figure 5: Beijing's Districts and its Counties

Every day, more than 13 million trips are performed, with approximately 5 million corresponding to subway trips, 8 million corresponding to bus trips, and 100,000 corresponding to bicycle trips. All records corresponding to one day are saved in a single csv file.

In this project we examine one month worth of data, corresponding to November 2015. The month of November is chosen because it does not overlap with holidays and has a relatively stable weather, hence diminishing the variance between bicycle and bus/subway traveler preferences. this, we hypothesize, will maximize the data quality. As a result, the data to be analyzed is divided into 30 csv files, requiring a total storage space of 53GB.

4.1.1 Special considerations

The previous description corresponds to the data as delivered by the TOCC. As such, it is the result from processing the raw records at the collecting phase. Some special considerations concerning this processing are explained below:

Travel distance by bike: Since bicycles do not have predefined routes, the distance cannot be directly recorded. However, it is inferred by using the travel time and a static average speed for cyclists.

Subway transfer: Transfers between subways lines of the same operator cannot be tracked since a single check-in gives access to the traveler to all the subway network. In order to infer the transfer detail, the A* algorithm is used to calculate the most likely transfer sequence, given the boarding and alighting stations. Furthermore, similarly to the bicycle missing information, the transfer time inside the subway system cannot be directly recorded. Using a static average walking speed and the known distance in transfer stations, the transfer time is calculated.

Transfer information: The path link and transfer number fields are extracted from the transfer detail field. Similarly, the transfer average time is calculated from the transfer total time and transfer number fields.

4.1.2 Labeled and unlabeled data

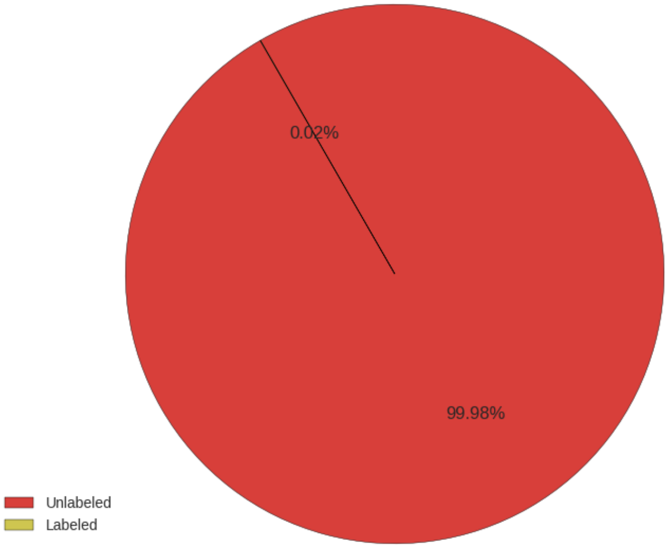


Figure 6: Relation between labeled and unlabeled data.

Commuter classification: Classification is a supervised learning task, where every training data sample requires an associated label determining its true class. In case of commuter classification, this translates to having smart card codes associated with either a "commuter" or "non-commuter" label. Such data is expensive to obtain and limited in principle, since it obtained by asking public transit users directly if they are commuters or not. Thus, in general, annotated data is not available, and labeling new records falls beyond the scope of this project.

As a solution for the above, we take advantage of the dataset used by Tu[24]. This dataset corresponds to trip records performed during a week in January 2015, and it contains labels for 978 smart cards, collected and validated via surveys. The original dataset distribution is composed by:

- 6439 records of 481 commuters
- 1628 records of 497 non-commuters

As mentioned in Section 4.1, this project considers data corresponding to November 2015. Since both datasets correspond to the same year, we believe the labels to be reasonably relevant for both datasets. As such, in order to construct an extended labeled dataset, we take the 978 labeled smart card codes from Tu's dataset, and search for their corresponding records in our dataset. As shown in Figure 6, a great disadvatage of this is that it tremendously reduces the size of the dataset to less than 1% of its original size.

The reduced labeled dataset is used for Part II (Section 6) of this project.

Behavior clustering: Clustering, being an unsupervised learning task, does not require labels. Therefore, for this task we attempt to use the full dataset. Though rich in its contents, this poses some computational challenges which will be further explained in Section 7.

4.2 Spatio-temporal representation

One of the main contribution of this project is the novel representation of public transit travel behavior. We consider that the travel behavior of a public transit user is composed by a) the distribution of trips over time, and b) the spatial, temporal and general attributes of each of the trips.

On the one hand, the distribution of trips over time is a complex continuous distribution. However, according to the Transportation domain standard, it can be approximated in a discrete manner. For this purpose, we divide each day into one hour bins. Then, we consider that each bin can be "occupied" by a trip, or empty if there was no traveling at that moment.

On the other hand, each trip possesses different attributes of different nature. Fields like *Travel time* and *Travel distance*, for example, have continuous positive values within a reasonable range specific to the city of Beijing. In contrast, fields like *Traffic areas* and *Ring road*, for example, contain categorical values. Therefore, the information from smart card transactions is processed to extract relevant the attributes.

We propose a 3 dimensional data structure to contain the monthly travel information of a public transit user. The conceptualization is pictured in Figure 7.

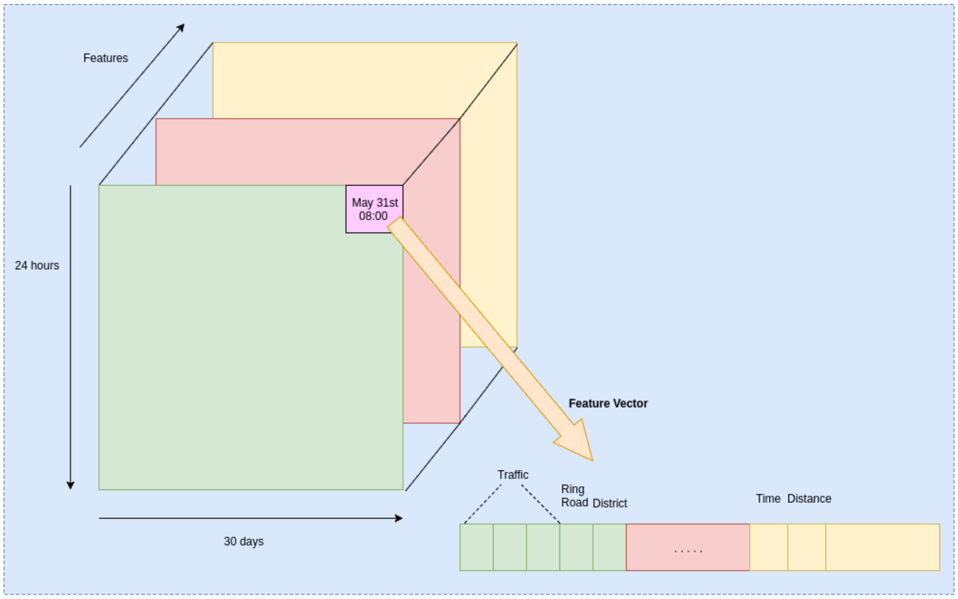


Figure 7: Spatio-temporal data structure.

Inspired by Langlois et al. [11], the x-y plane of our representation constructs a temporal structure between days of the month and hours of the day. The crucial advantage of this structure lies in its local properties. Similar to the case of Image Processing, in this representation a temporal pixel is simultaneously influenced by what happened in the previous/following hours (y axis), and on the previous/following days (x axis).

As for the z axis, each layer contains a trip attribute. In Figure 7 boarding spatial attributes are portrayed in green, alighting spatial attributes are portrayed in red, and other types of general attributes (such as travel time, travel distance, transfer number, transfer total time, etc) are portrayed in yellow.

Therefore, each temporal pixel may contain a trip feature vector, which expands several layers deep. Considering that even regular public transport users do not usually perform more than 6 trips a day, the proposed representation is sparse, since only a few time pixels are populated with trips.

4.3 Dimensionality reduction

The proposed representation is directly proportional to the number of attributes used to describe a trip. As it is explain in Section 5.5, we extract 26 attributes in a trip. Considering we have 30 days worth of data, and 24 hourly time bins in a day, this leads to $24 \times 30 \times 26 = 18,790$ temporal pixels to represent one user. Given the high dimensionality and the sparsity of the structure, we need perform dimensionality reduction in order to avoid the typicl effects of the curse of dimensionality.

4.3.1 Feature selection

One of the simplest ways for reducing the number of attributes in a dataset is to perform feature selection. This technique consists of evaluating each feature's influence in making predictions or classifying samples. The strongest features are then selected to be part of the final dataset, while the least informative or redundant features are disregarded.

When performing feature selection, there are two main choices to be made: the amount of features to be kept and the approaches to evaluate each feature. In this project, the first aspect is tackled using the *Best k* algorithm, which fixes the amount of features to be kept to *k*, regardless of the initial number of features present in the data. The second aspect is tackled via an assortment of algorithms, from statistical tests -such as correlation tests, and ANOVA f-test-, machine learning techniques -such as Trees classifiers-, to domain knowledge from Transportation domain specialists.

4.3.2 Feature extraction

An alternative way of performing dimensionality reduction is by mapping high dimensional spaces to a low dimensional space. Though in most cases the mapping causes some loss of information, there exists mathematical transformations optimized for reducing the loss. Two of the most common techniques to do so are performing single value decomposition, as used by PCA, or work under the Universal Approximation Aroblem, as neural networks.

In this project, we perform the mapping through an autoncoder. Autoencoders follow the principles of neural networks, and focus on the task of encoding and decoding an image minimizing the loss but not abstracting the noise. As a network, autoencoders may become as complex as needed, allowing for any number of layers and any types of activation functions.

Taking advantage of the local properties of the proposed representation, we decide to apply stacked convolutional filters followed by a final fully conencted layer that outputs features of a more manageable dimensionality. The end result will be used as features for clustering commuters in Part III (Section 7) of this project.

4.4 Pattern recognition

4.4.1 Ensemble models for classification

There is a large variety of algorithms for performing classification tasks. Each algorithm may focus on different samples when learning the task, and so it has strengths and weaknesses different from one another.

Ensemble models explore the idea of combining several non-correlated prediction methods that might correct each other in order to reach a better classification accuracy. Furthermore, ensemble models are robust and modular. Therefore, starting from a few weak classifiers, assembled via aggregation methods, the model can grow larger or more complex as needed.

This project constructs an ensemble model for classifying commuters. As proven by Tu [24], weak classifiers, like a Support Vector Machines (SVM), are sufficient to identify commuting behavior up to a 94% accuracy. Aiming to increase accuracy, but preventing overfitting, this project extends Tu's model with other similar weak classifiers such as decision trees, Bayesian classifiers, Gaussian Processes, multilayer perceptrons, and others. Their individual predictions are ensembled via the majority vote rule.

4.4.2 Clustering

For the third part of this project we perform behavior clustering. To this end, we apply the *K means* algorithm on low dimensional features extracted by a convolutional autoencoder. The *K means* algorithm is the standard algorithm used in the Transportation domain, and it consistently achieves satisfactory results.

4.5 Big Data considerations

The amount of data gathered for this project requires specific computational resources. While programming the learning tasks, measures are taken to optimize the code. For example, the code is written in modules that work with one day file at the time, and save checkpoints at every relevant step. Additionally, the code is compatible with cluster computing and SLUM jobs are created for its processing.

Furthermore, parallel computing techniques are implemented wherever the whole dataset goes into memory, for example when extracting trip attributes from the records. As shown in Figure 8, the time required for applying a single filter reduces drastically if the number of parallel workers equals the number of cores in the machine used.

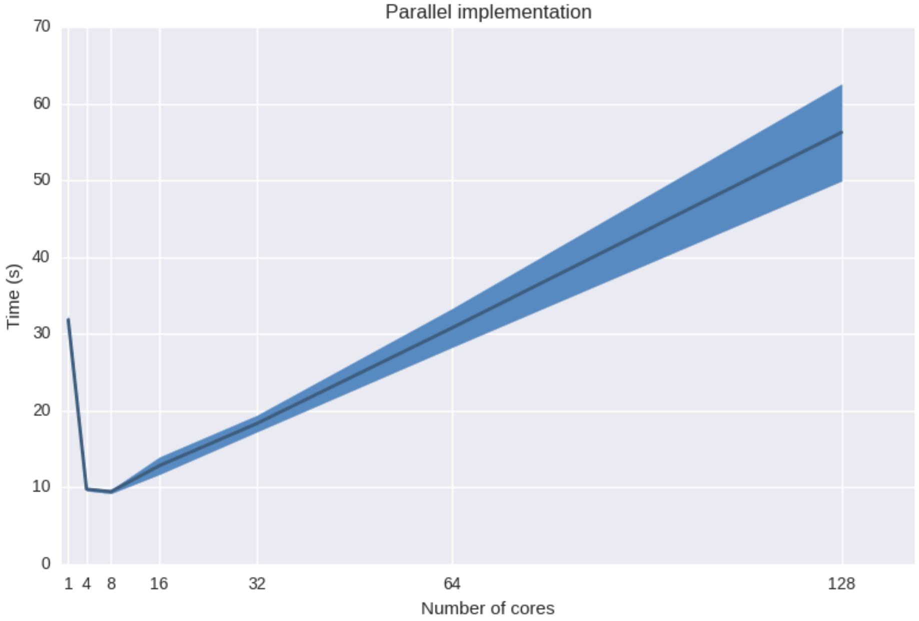


Figure 8: Performance in parallel computing.

The computational resources are provided by Vrije Universiteit Amsterdam. The results of this project are obtained by using DAS5 (The Distributed ASCI Supercomputer 5) [1].

5 Data preparation and preprocessing

The data acquired for this project corresponds to November 2015. In total, 219,452,319 trip records are collected for the month. Figure 9 shows the volume of records per day, color coded by day of the week.

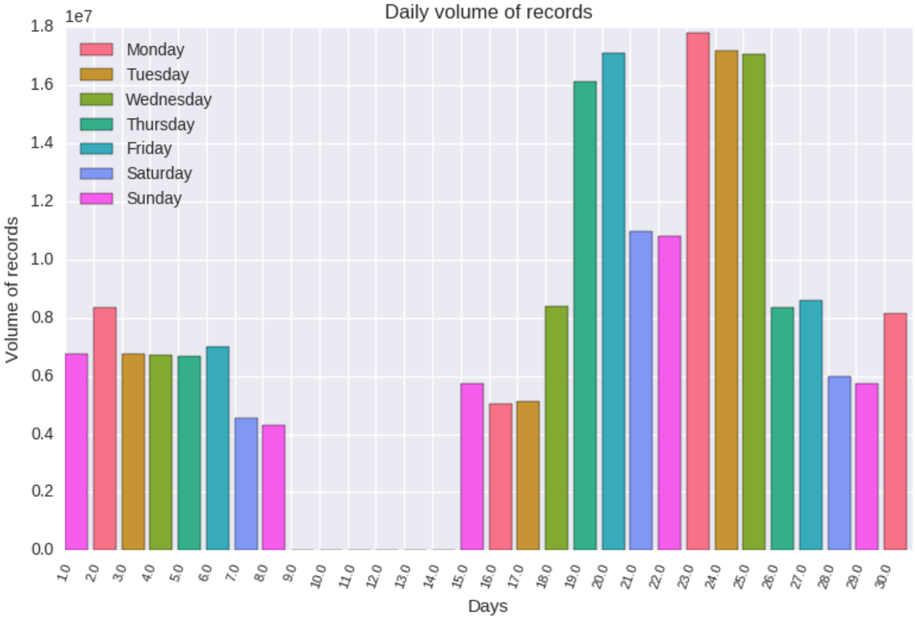


Figure 9: Performance in parallel computing.

The figure illustrates that, in general, Saturday and Sunday have less records compared to their corresponding weekdays. For example, for the first week of the month corresponding to days between the 2nd and the 8th, the weekend days (7th and 8th) have only about 4.5 million records, while the weekdays (from the 2nd to the 6th) have at least 6.5 millions each. The fourth week of the month, from the 23rd to the 29th, shows similar behavior with weekends having 5.5 million records but weekdays surpassing 8.5 million records.

Yet, the pattern for the third week of the month is more complex. While the weekdays (21st and 22nd) have less records than the previous couple weekdays (Thursday 19th and Friday 20th), the first three weekdays are not as busy, with Monday having only 5 million records.

With regards to the second week of the month, from this plot we note a gap of six days with 0 records collected. Due to defective collection methods, the data from November 9th to November 14th is not available.

The figure also shows an increase of public transit usage between Thursday 19th and Wednesday 25th. According to domain expert PhD. Liang Qu, the heavy usage is explained by weather conditions. During those days the weather was rainy and snowy; therefore there were more passengers than usual days.

5.1 Cleaning

As first step for preparing the data for its mining, we eliminate faulty records. To this end, we apply the following four filters:

1. Eliminate records with empty fields: ~7.26% records eliminated
2. Eliminate records with incomplete travel details: ~0.76% records eliminated
3. Eliminate records with travel time $<= 0$: <0.01% records eliminated

4. Eliminate records with travel distance <= 0: ~9.06% records eliminated

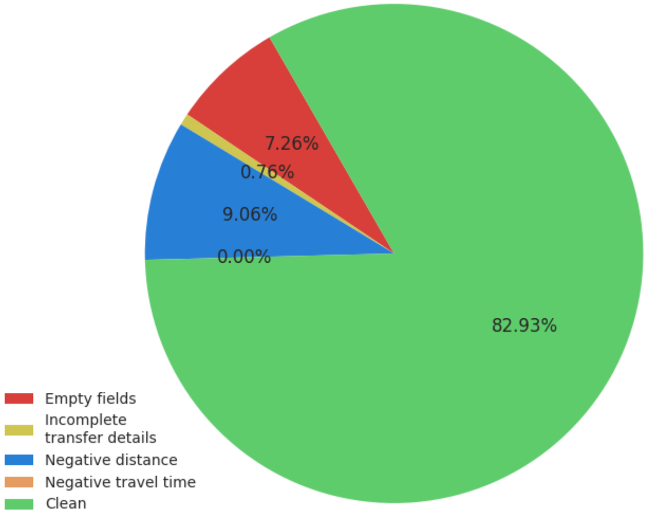


Figure 10: Reasons for eliminating records.

The percentage of data eliminated by each filter is shown in Figure 10. Removing faulty data reduces the dataset to ~82.93% of its original size. After cleaning, the dataset contains 182,033,638 trip records.

5.2 Extraction

From the raw records, we can further extract attributes to complement the trip information. For example, taking the date of the travel we can further obtain two important attributes. The "Day" attribute, corresponding to the number of day on the month, from 1 to 31. The "Weekday" attribute is a number from 1 to 7 corresponding to the day of the week, starting on Monday.

Furthermore, we calculate the "Number of trips" attribute which counts the trip records each user has in a day. Figure 11 shows the distribution of this attribute on a random weekday. We note that most people perform two trips per day, followed by people who perform only one trip a day.

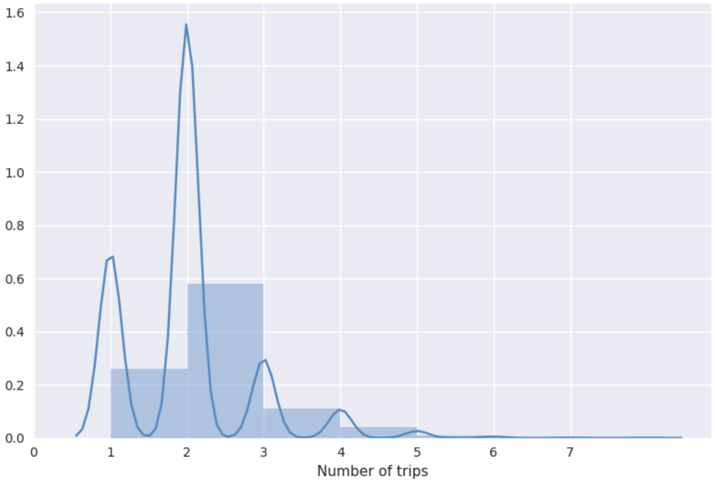


Figure 11: Number of trips distribution.

Other fields, such as "Start/End time" and "Trip details" require more specialized algorithms for attribute extraction.

5.2.1 Time bins

Using hourly time bins is standard practice in the field and has proven sufficient to examine temporal data [11] [12] [16]. Therefore, in this project we follow the same technique and extract the hour of the start and end of each trip.

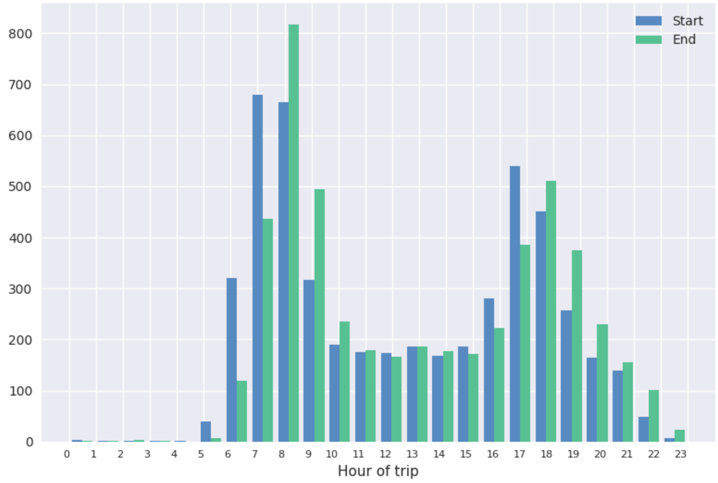


Figure 12: Distribution of start/end hours for trips.

From Figure 12, we note that our data follows the expected distribution for the domain, showing clear morning and evening peaks. Furthermore, we note that boarding and alighting patterns during the peaks hours are shifted by one hour. For example, the morning peak for boarding is between 7:00 am and 8:00 am, while the alighting peaks at 8:00 am and 9:00 am. Similarly, the evening peak happens between 5:00 pm and 6:00 pm for boarding, but between 6:00 pm and 7:00 pm for alighting.

5.2.2 Trip parsing

The trip details obtained from the records are in Chinese, with descriptors containing a combination of numbers and text. In order to extract boarding/alighting route features, the descriptors must be parsed. We parse the trip details using a combination of two techniques: regular expressions and the construction of an ID vocabulary.

Regular expressions

Since a trip may include transfers, we define a trip to be composed of one or many rides. Each ride is carried out in a single travel mode. In order to obtain the elements of each ride we look at the descriptor pattern according to its travel mode.

$$\begin{aligned} BIKE &= (bike.STOP - STOP)\\ SUBWAY &= (subway.LINE : STOP - LINE : STOP)\\ BUS &= (bus.ROUTE(DIRECTION - DIRECTION) : STOP\\ &\quad - ROUTE(DIRECTION - DIRECTION) : STOP) \end{aligned}$$

where the upper-case text corresponds to placeholders for ride elements, the lower-case text corresponds to the English translation of the descriptor in Chinese, and the punctuation (parentheses, dots, colons and dashes) correspond to separators between ride elements.

Unifying the mode-specific patterns, we describe a ride and a trip using regular expressions:

$$RIDE = (MODE.[LINE/ROUTE:]?STOP - [LINE/ROUTE:]?STOP)$$
$$TRIP = RIDE[- > RIDE]?$$

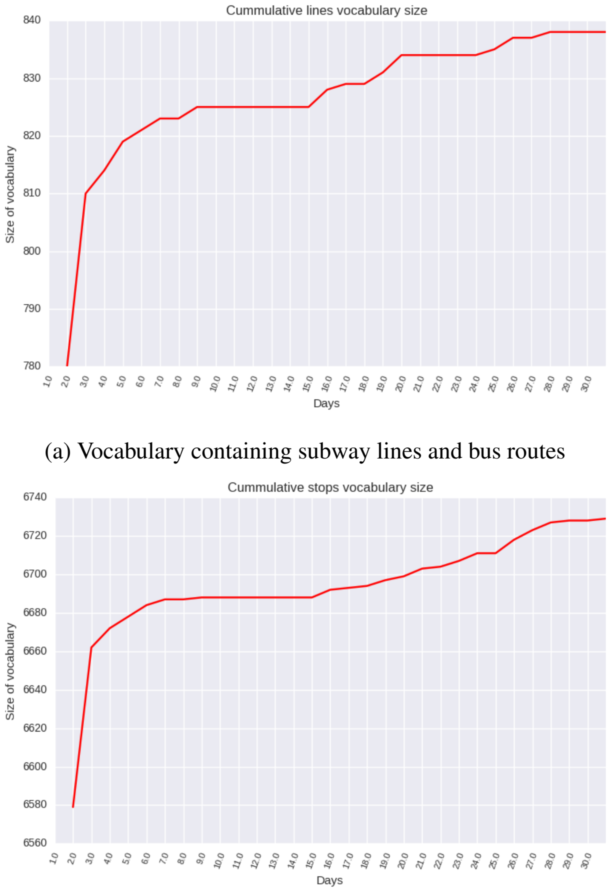
where elements surrounded by squared brackets and followed by a question mark (e.g. $[ELEMENT]?$) correspond to optional elements. Note that when parsing bus details, we disregard the route direction. This decision is motivated to fit both subway lines and bus routes to a single pattern, acknowledging that the direction of the route does not affect the path of the route itself.

ID Vocabulary

Once the elements of a trip are extracted, they must be labeled by unique numerical IDs. These IDs are not available from the TOCC, hence we label them using our own system. IDs are assigned incrementally by 1, meaning that the next available ID is equal to the size of the vocabulary. Furthermore, IDs start from number 1 since the value 0 has a special meaning in the proposed representation.

Two different vocabularies are created, the first one for subway lines/bus routes, and the second one for stops. Modes of transportation are assigned ID 1 for subway, ID 2 for bus and ID 3 for bicycle.

Usually, bus routes are identified by a number. However, in Beijing a single bus route number can be associated to different paths. For example, night buses, express buses and other special cases of a bus route may follow different paths even if they are described with the same number. For this reason we create a vocabulary with all unique parsed routes according to their full description and not only their number.



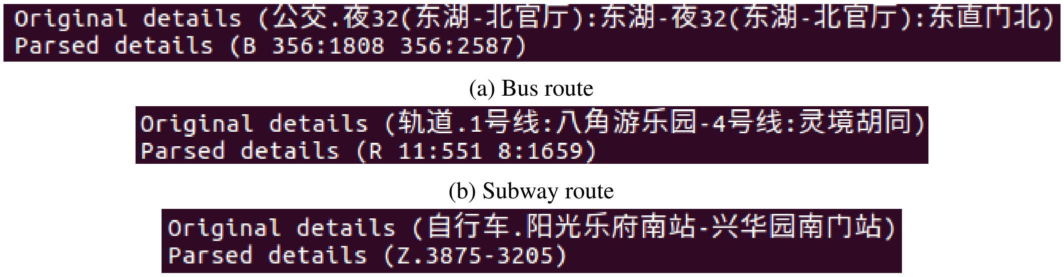
(a) Vocabulary containing subway lines and bus routes

(b) Vocabulary containing subway and bus stops, as well as bikes drop off spots.

Figure 13: Cummulative plots.

Vocabularies are created from the available data. As such, the first time a line/route/stop is seen it is assigned the next available numerical ID. Thus, the vocabulary is dependent on the order on which the records are processed. Moreover, the size of the vocabulary increases as we process more daily files. Figure 13 shows the size of each of the vocabularies. We find ~840 lines and ~6740 stops. Though different in size, both vocabularies exhibit the same growing behavior. 92% of lines and 97 % of stops are seen since on the first file, and by the 7th file we have seen 99% of both vocabularies.

Examples of parsed routes for each mode of transportation are shown in Figure 14.



```
Original details (公交.夜32(东湖-北官厅):东湖-夜32(东湖-北官厅):东直门北)
Parsed details (B 356:1808 356:2587)
```

(a) Bus route

```
Original details (轨道.1号线:八角游乐园-4号线:灵境胡同)
Parsed details (R 11:551 8:1659)
```

(b) Subway route

```
Original details (自行车.阳光乐府南站-兴华园南门站)
Parsed details (Z.3875-3205)
```

(c) Bike route

Figure 14: Examples for parsed and tokenized trip details.

5.3 Data patching

We note that the number of transfers and the path link fields of some records do not correspond to the information in their trip details. According to domain expert PhD. Tu Qiang, this must be recalculated [21].

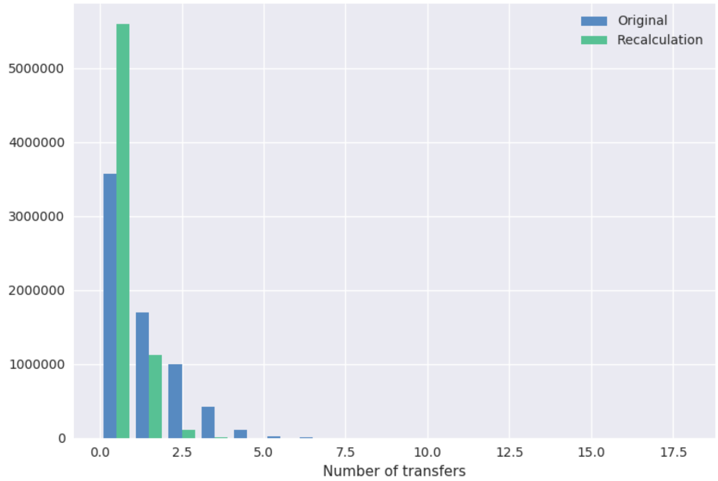


Figure 15: Transfer number distribution before and after recalculation.

Figure 15 shows the distribution of the number of transfers per trip before and after patching, illustrating that there is a significant correction in the distribution. The patched distribution shows that most trips are performed without transfers, which is consistent with other studies findings [2].

5.4 Standardization

Continuous values

In data mining, it is a standard practice to perform whitening. This technique eliminates correlations between features, which is desirable in most cases. However, for the domain of Metropolitan

Transportation some of these correlations are highly important and should not be discarded. This is the case of total travel time and distance, as shown in Figure 16. For this reason, we choose to only standardize the attribute but keep the correlations.

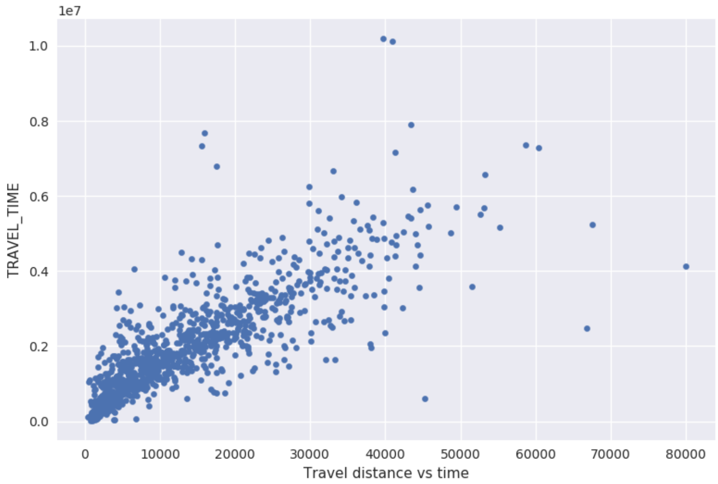
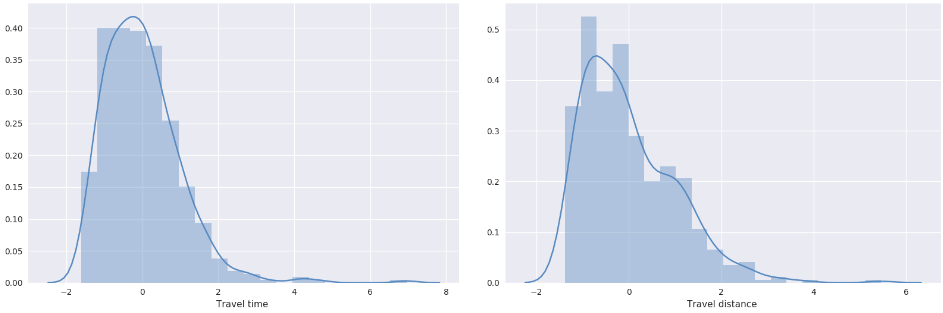


Figure 16: Travel distance vs travel time.

Travel time, travel distance, total transfer time and average transfer time are standardized by subtracting the mean of each distribution and forcing a unit standard deviation. We standardize every daily file separately, therefore preserving each day's characteristic distribution. This allows us to maintain distinct days separate, such as weekdays and weekends.



(a) Travel time

(b) Travel distance

Figure 17: Time and distance standardize distributions.

Figure 17 shows distributions for travel time and distance. Since the nature of the data prevents negative values (time and distance must be positive), the original distribution is truncated at 0. Standardization maintains the shape of the distribution, but shifts and contracts it to be closer to zero values.

The mean travel time is ~31.9 minutes, and the mean travel distance is ~13.5 kilometers.

Categorical values

As mentioned in Section 4.2, a trip contains both continuous and categorical attributes. Generally, categorical values must to be transformed into one hot encoding. This prevents problems with ordinal values which impose underlying structures on the data.

In the context of this project we find that a trip contains 12 categorical attributes. Furthermore, each of these contains its own range, with some of them having up hundreds or thousands of categories (i.e. Middle and Small traffic areas). For these reasons, one hot encoding is not a scalable option for this project.

5.5 Attributes

After the data is preprocessed, we collect 26 attributes that describe a trip. We divide them onto three categories: general attributes, temporal attributes and spatial attributes.

The general trip attributes are:

1. Number of day
2. Weekday
3. Number of trips
4. Travel time
5. Travel distance
6. Number of transfers
7. Transfer total time
8. Transfer average time

The temporal trip attributes are:

1. Start hour
2. End hour

Finally, the spatial attributes are:

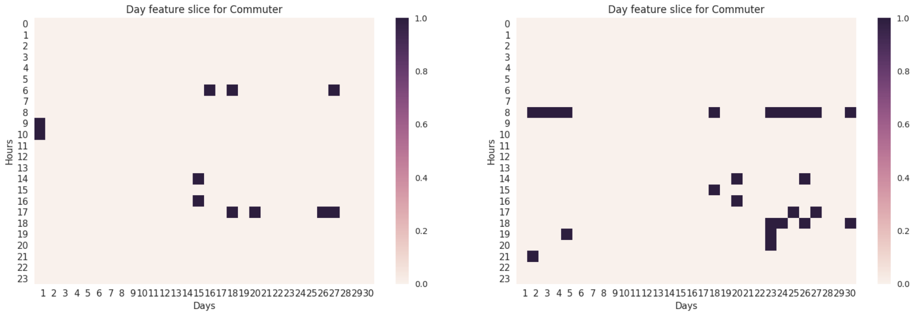
1. On/Off District
2. On/Off Small traffic area
3. On/Off Middle traffic area
4. On/Off Big traffic area
5. On/Off Ring road
6. On/Off Mode
7. On/Off Line/Route
8. On/Off Stop

5.6 User cubes

Figures 18 and 19 show the first slice for a couple random Commuters and a couple random Non-commuters.

For the case of Commuter 1, we identify a pattern between the 15th and 20th of November. We observe several trips at 6 in the morning and 5 in the evening, suggesting a typical working schedule. Non-commuter 1, in contrast, has an irregular distribution of trips.

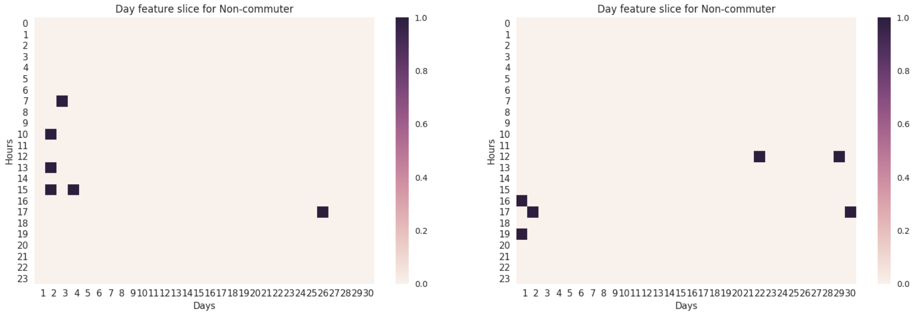
It is important to note that as the data between Monday November 9th and Saturday November 14th is missing. Thus, both cases show a gap in records during these days. However, we note that the gaps are not restricted to these days, and that the behavior is very irregular, even for the case labeled as commuter.



(a) Random Commuter 1.

(b) Random Commuter 2.

Figure 18: Sample cube slices for Commuters.



(a) Random Non-commuter 1.

(b) Random Non-commuter 2.

Figure 19: Sample cube slices for Non-commuters.

6 Commuters identification

In this part of the project we focus on the task of classifying public transit users into one of two categories: Commuter or Non commuter. To do so, examine the relevance of each pf the trips attributes and we reduce the dimensionality of the original user cubes by selecting the most informative feature slices. Subsequently, several weak classifiers are trained and evaluated. The best classifiers are selected and incrementally added to an ensemble model.The final model is selected according to the highest accuracy achieved.

6.1 Attributes correlation

First, we perform an exploratory study of the relationships among all the attributes of a trip. We calculate the correlation matrix of the trip attributes using the labeled dataset refer to in Section 4.1.2. This set contains 12,584 trips corresponding to 639 labeled card codes. These correspond to 353 Commuters, and 286 Non-commuters.

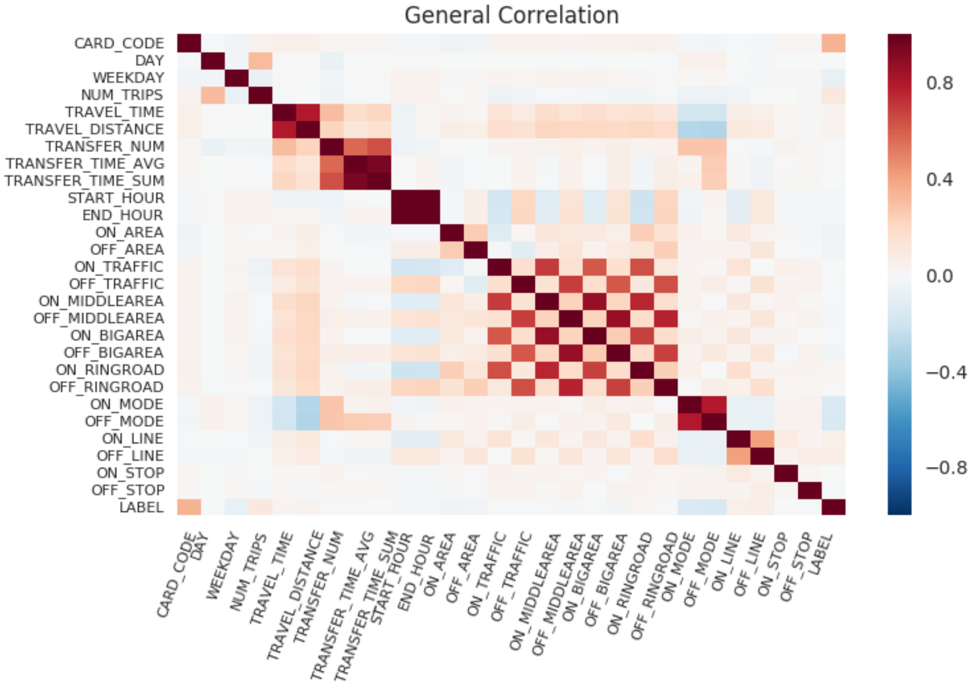


Figure 20: Attributes correlation to each other and to label. Color bar indicates the direction and magnitude of correlation.

From Figure 20 we note distinguishable blocks with specific patterns. We perform an analysis by attribute category in order to better understand the interdependence among attributes.

General

It is clear that travel time and travel distance are highly correlated. Rationally, this may be explained by the reduced offer for express trips in the public transit network of Beijing. As a counterexample, in Amsterdam it is possible to travel significant distances within the city in a fast manner by using the train. Alternatively, the correlation can also be explained by the vast public transit network in Beijing. Given the high inter-connectivity and large offer of routes, it is possible to travel from one point to another in a relatively straight way, thus avoiding detours or requiring to travel to major stations for transfers.

Furthermore, the transfer average and total times are also highly correlated. As examined in Section 5.3, most trips are have one transfer, which makes the relationship between average and total transfer time linear.

Temporal

In this block we see a very high positive correlation, represented by a dark red color. This indicates that the start and end hour for most trips are identical. Given that most trips are completed within the hour, this is coherent the data exploratory findings.

Spatial

With regards to the traffic areas and the ring road areas, we note that the boarding attributes share a stronger connection among themselves than with their correspondent alighting attributes. Since the traffic areas are hierarchical (small, middle and big divisions), it is logical for them to be correlated. Similarly, the ring road areas, though having a slightly different division, maintain a fixed relationship with the traffic areas and thus one can be inferred or approximated from the other.

Finally, we observe that the boarding and alighting mode of transportation are highly correlated, suggesting that most trips are performed using only one mode of transportation.

6.2 Feature selection

Feature selection is the process of choosing features that aid in the learning task at hand, and disregarding the information that is not helpful. From the correlation matrix we note that some attributes are redundant. Hence, feature selection can help to disregard those which are the least informative and reduce the dimensionality.

We evaluate each attribute via its correlation to the true class label, its importance according to an ExtraTrees classifier, its ANOVA f-value, and the jugdment of usefulness according to domain expert PhD. Liang Qu. Initially, a Chi Squared test was also performed, but was abandoned since it had significant bias in favor of continuous attributes over categorical attributes.

Each method's scores are normalized to sum up to 1. Then, we aggregate them and calculate the final score. The results are shown in Figure 21.

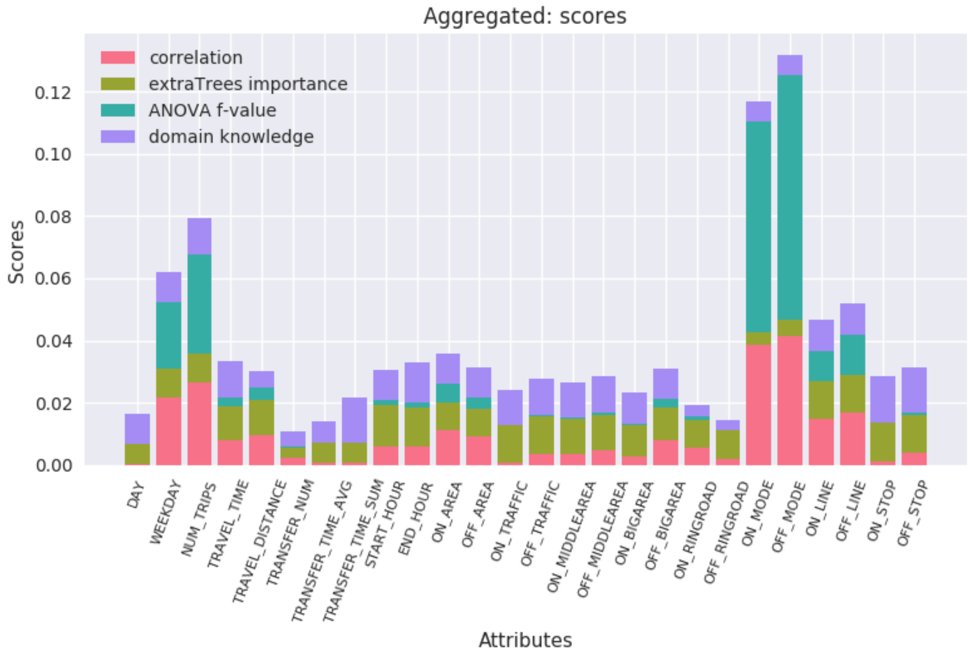


Figure 21: Attributes scores.

For the final set selection, we take a fixed amount of attributes from each category, in order to maintain a balanced set.

- From the *General attributes*, we select the best 3 out of 8: Number of trips, Weekday, and Travel time.
- Given that the size of the set is already reduced, we keep all the *Temporal attributes*. Formally speaking, we select the best 2 out of 2: Start hour and End hour.
- For the *Spatial attributes*, we design a system for pair selection. This mechanism ensures that if an attribute is selected, its boarding/alighting counterpart is also selected. As such, we select the best 2: On/Off mode, and On/Off line. The cubes are flatten in order for them to be fed into different weak classifiers.

A final set of 9 features is created. The cube slices corresponding to the selected features are kept in the user cubes and the rest disregarded, resulting in ~66% reduction.

6.3 Model

As suggested by Tu's results [24], the data is almost linearly separable. Thus, weak classifiers such may suffice to achieve a satisfactory accuracy (chance prediction is 50%).

We train four weak classifiers: one linear SVM, one Gaussian Process, one Gaussian Naive Bayes classifier, and one multilayer perceptron. We also train three simple ensemble models based on decision trees: a random forest, a model using the adaboost algorithm, and a model using bagging. The parameters for each classifier are shown in Table 2

Classifier	**Parameters**
SVM	Linear kernel, One vs all decision function
Gaussian process	RBF kernel
Gaussian Naive Bayes	Prior probabilities set according to data
Multilayer perceptron	100 layers, ReLu activation function
Random Forest	100 trees, max depth of 10
Adaboost of decision trees	100 trees
Bagging of decision trees	100 trees

Table 2: Accuracy for weak classifiers

The seven weak classifiers are trained and evaluated separately. The dataset to be used contains 639 user cubes to be classified into one of the two categories. We flatten the reduced user cubes since the classifiers require the data to be represented as vectors. For evaluation we split the dataset into 70% for training and 30% for testing, as evaluated by Tu. Furthermore, we evaluate using more robust techniques, such as cross validation over 5 folds.

We proceed to build an ensemble model based on a Majority Vote aggregation method. This method takes the prediction from each individual classifier, and select the class which was predicted by the most classifiers. When facing a tie among classes, the model selects the class based on ascending order. Since we perform binary classification, that means the default category is Non-Commuter.

Incrementally, we incorporating the best k classifiers and evaluate at each step. The final model is selected according to the largest accuracy using cross validation over 5 folds.

6.4 Experiments

The weak classifiers performance is reported by its weighted mean accuracy over both classes. We report two experiments: a) one run on a single train/test division of the dataset, and b) average of cross validation over 5 folds. Table 3 summarizes the results.

Classifier	Single run accuracy	Cross validation accuracy
SVM	78%	62% (+/- 4)
Gaussian process	79%	45% (+/- 0)
Gaussian Naive Bayes	77%	66% (+/- 7)
Multilayer perceptron	77%	66% (+/- 5)
Random Forest	85%	74% (+/- 7)
Adaboost of decision trees	80%	69% (+/- 5)
Bagging of decision trees	83%	73% (+/- 8)

Table 3: Accuracy for weak classifiers

With 74% accuracy, Random Forest is the strongest classifier. This is reasonable since Random Forests are generally competent when dealing with categorical information. Given that 6 out of the 9 trip features we kept after feature selection are categorical, this method adapts smoothly to our data.

The incremental ensemble model performance is reported by its average accuracy using cross validation over 5 folds.

Number of classifiers	Classifiers included	Cross validation accuracy
Best 2	Bagging of decision trees, Random forest	76% (+/- 6)
Best 3	Previous set + Adaboost of decision trees	73% (+/- 8)
Best 4	Previous set + Multilayer perceptron	74% (+/- 8)
Best 5	Previous set + Gaussian Naive Bayes	73% (+/- 8)
Best 6	Previous set + SVM	72% (+/- 8)
Best 7	Previous set + Gaussian Process	74% (+/- 8)

Table 4: Accuracy for incremental ensemble model

From the results we note that including more classifiers does not necessarily boost performance, but it may even affect it negatively. Regardless, ensemble models perform better than weak classifiers alone, as shown in Table 5. Complex ensemble models refer to models that mix different types of weak classifiers, while simple ensemble models refer to models based on decision trees only.

Model	Cross validation accuracy	Type of model
Best 2	76% (+/- 6)	Complex ensemble model
Best 4	74% (+/- 8)	Complex ensemble model
Best 7	74% (+/- 8)	Complex ensemble model
Random Forest	74% (+/- 7)	Simple ensemble model
Bagging of decision trees	73% (+/- 8)	Simple ensemble model
Best 3	73% (+/- 8)	Complex ensemble model
Best 5	73% (+/- 8)	Complex ensemble model
Best 6	72% (+/- 8)	Complex ensemble model
Adaboost of decision trees	69% (+/- 5)	Simple ensemble model
Gaussian Naive Bayes	66% (+/- 7)	Weak classifier
Multilayer perceptron	66% (+/- 5)	Weak classifier
SVM	62% (+/- 4)	Weak classifier
Gaussian process	45% (+/- 0)	Weak classifier

Table 5: Models and their type, ranked by accuracy

The model with highest accuracy corresponds to ensembling the best 2 classifiers: Random forest and Bagging of decision trees. In order to investigate how the two selected classifiers complement each other, we take a look at their confusion matrices.

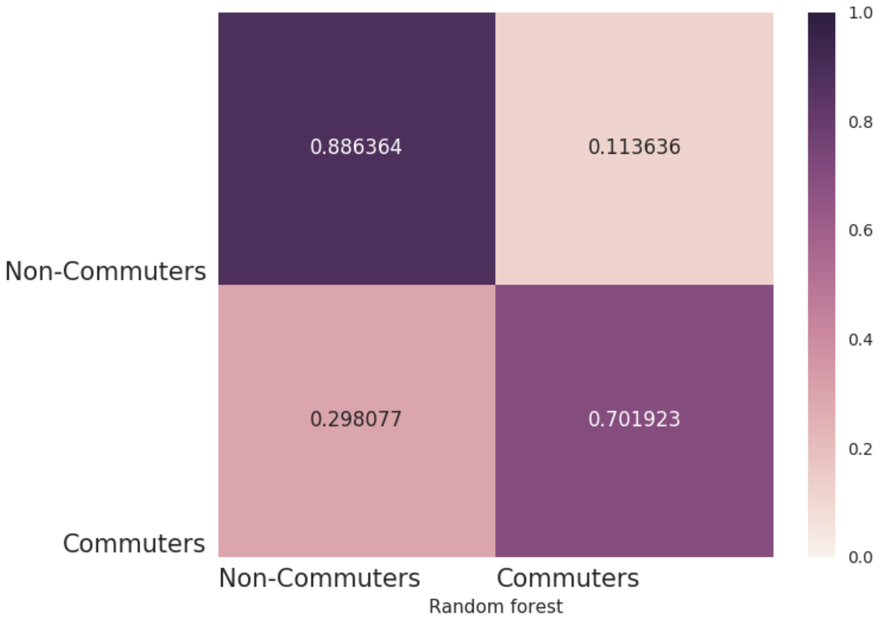


Figure 22: Confusion matrix for Random forest classifier.

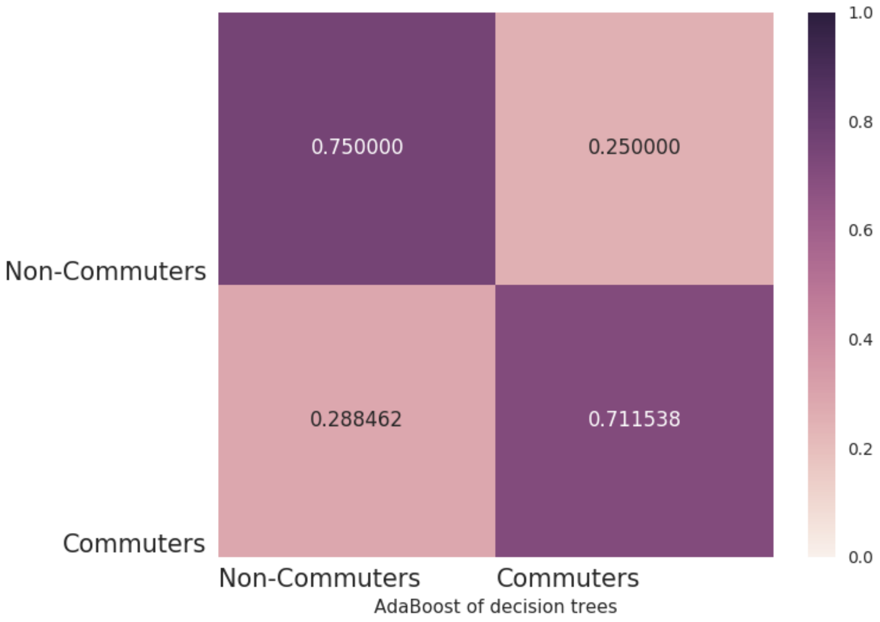


Figure 23: Confusion matrix for classifier based on bagging decision trees.

Interestingly, both classifiers have a similar behavior. Figures 22 and 23 show that both perform well in identifying commuters. Random forests achieves 70.1% closely outperformed by Adaboost with 71.1%. Similarly, they both perform even better when identifying Non-commuters, with 88.6% and 75.0% correspondingly.

6.5 Discussion

Dataset and representation

With the goal in mind of comparing our work to Tu's, we selected the same training parameters for the linear SVM classifier. However, Tu's results could not be replicated. We consider that this is a consequence of using a different dataset for training.

In terms of quantity, Tu's dataset was ~150% larger, having 978 labeled user card codes against ours with 639 labeled user card codes. Furthermore, the labels for user card codes were gathered during January 2015. Presumably, some users might have altered their behavior by changing their traveling patterns or stopped being active public transit users completely by November 2015. This brings noise to the labels in the dataset.

In terms of representation, Tu's dataset presents a simpler representation, including only one temporal and two spatial features for one week of public transit usage. In contrast, our representation includes a mixture of general, temporal, and spatial features gathered over a month. As a result, we suspect that that the dataset is not linearly separable anymore.

In order to investigate this hypothesis we visualize both representations using two dimensional t-Distributed Stochastic Neighbor Embedding (TSNE).

Figures 24 and 25 show each representation's manifold. Tu's straightforward representation gathers most commuters in three separate clusters. Non commuters are spread along a larger cluster, with some commuters blended in. In contrast, the reduced user cubes form only one cluster. Though most commuters align in the lower part of the space, and non commuters are present in the upper part of the space, both classes are virtually merged.

Thus, we confirm our hypothesis. It is clear that the classes are not linearly separable anymore, contrary to what was expected when starting the project. This suggests our representation is not fit for binary classification.

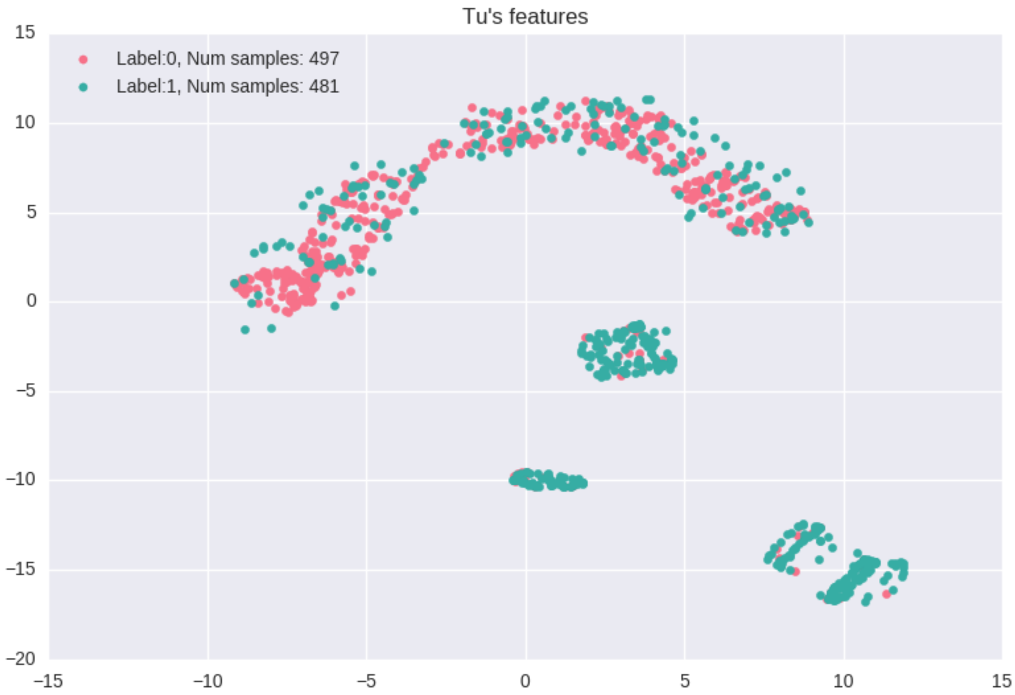


Figure 24: Visualization of samples and their labels after feature selection.

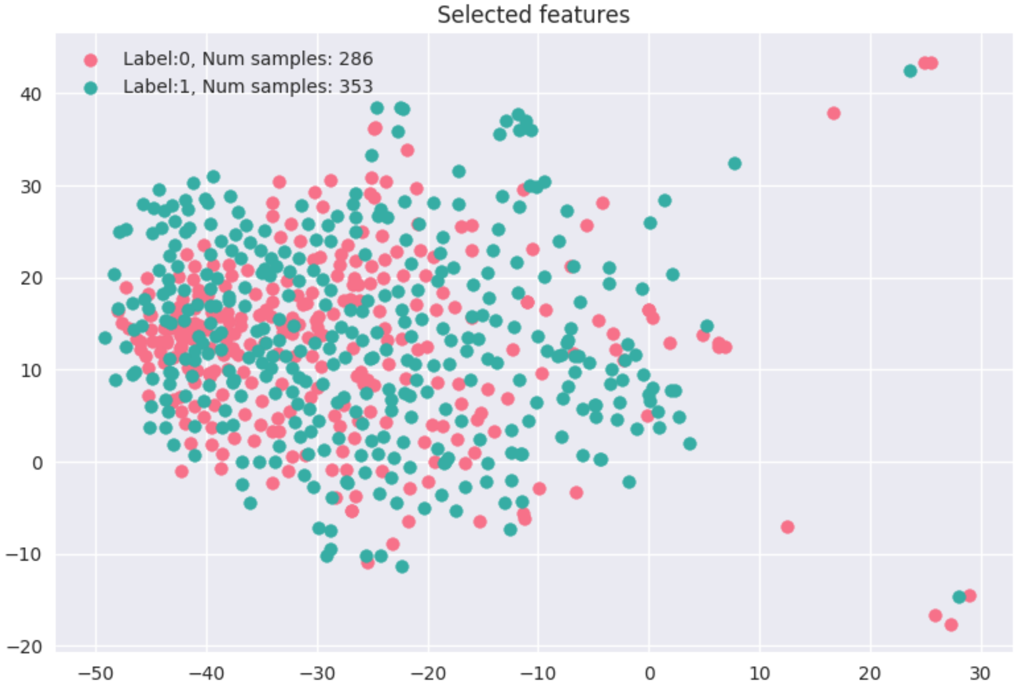


Figure 25: Visualization of samples and their labels after feature selection.

Ensemble models

The findings of this part of the project confirm the hypothesis that an ensemble model performs significantly better than a linear SVM, as designed by Tu. However, a specific type of ensemble model is required to maximize performance.

The final model consists of combining two simple ensemble models based on decision trees. This suggests that decision trees are ideal for our data due to its categorical nature. Furthermore, the model is ensembled by two levels. On a first instance, a large number of classifiers of the same type are aggregated. On a second instance, a small number of classifiers of a different type are aggregated. The result leads to a 10% increase in performance over any weak classifier alone.

However, though the chosen ensemble model maximizes performance, the diversity between the selected classifiers is unsatisfactory. Both classifiers have similar behavior and performance on each class. Ideally, two classifiers that differ in their strengths would benefit each other better.

In comparison the state of the art for commuters classification, our results fall short. Ma et al. achieve 94.1% detection accuracy; however, their dataset is small, containing 118 labeled samples only [12]. Tu et al. report 94.24% accuracy in detecting commuters. Yet, their evaluation technique contains only one test set thus making it prone to overfitting.

Furthermore, we believe that the missing information in our dataset might have affected the classifiers performance.

7 Commuters clustering

In this part of the project we perform an unsupervised learning task to cluster public transit users according to patterns in their travel behavior. This time we perform dimensionality reduction by encoding the original user cubes onto a low dimensional space by means of an autoencoder. Then, we cluster the user features and analyze each cluster's characteristics.

7.1 Feature extraction

As mentioned in Section 4.2, the proposed representation is sparse, hence dimensionality reduction has the potential to extract the relevant information and disregard noise. Furthermore, the proposed representation is local, in the same sense as images are. Combining these two characteristics, it is natural to think of convolutional filters as a tool to extract features from the user cubes.

7.1.1 Convolutional filters

We apply two dimensional filters to the three dimensional user cubes. Therefore, the x and y dimensions of the filter align with the day and time dimensions of the structures. Similarly to images and RGB channels, each of the feature planes is considered a channel in the convolutional filter.

The size of the filters is fixed to 3×3, since it leads to the best performance according state of the art techniques. The stride is set to 1 and we perform padding in order for the dimensionalities of the network to be suitable.

7.1.2 Autoencoder

Using Keras and Tensorflow as backend, we create an autoencoder. The network structure is shown below:

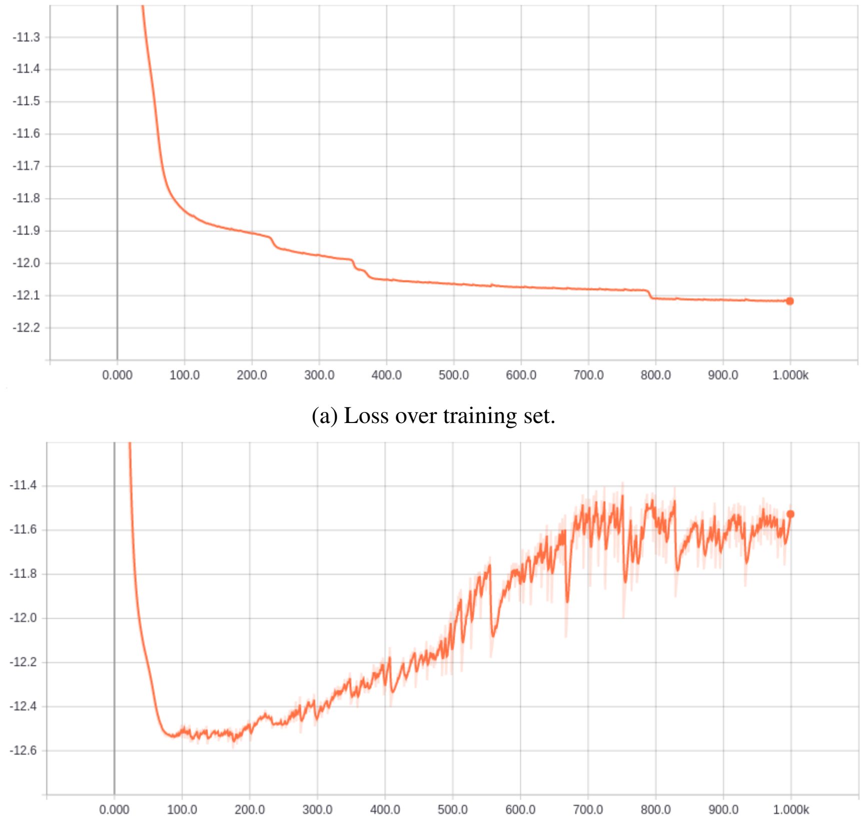
Type of layer	Filter size	Filter units	Stride	Activation function	Output dimensions
Input	None	None	None	None	[24, 30, 26]
Convolutional	[3, 3]	16	1	ReLU	[24, 30, 16]
Max Pooling	[2, 2]	16	1	None	[12, 15, 16]
Convolutional	[3, 3]	8	1	ReLU	[12, 15, 8]
Max Pooling	[3, 3]	8	1	None	[4, 5, 8]
Convolutional	[3, 3]	8	1	ReLU	[4, 5, 8]
Up sampling	[3, 3]	8	1	None	[12, 15, 8]
Convolutional	[3, 3]	16	1	ReLU	[12, 15, 16]
Up sampling	[2, 2]	16	1	None	[24, 30, 16]
Convolutional	[3, 3]	26	1	Sigmoid	[12, 15, 26]

Table 6: Autoencoder network structure

The first block corresponds to the encoder module, while the second block corresponds to the decoder module. The dimensionality of the encoded features is $4 \times 5 \times 8 = 160$. This represents more than 110% compression from the original dimensionality.

We evaluate the autoencoder using the binary crossentropy loss between the original user cube and the reconstructed user cube. As an optimizer we use Adadelta. We train on 60% of the data and validate on 40% of the data for 1,000 iterations. The loss over training and validation sets is shown in Figure 26.

We note that at around 200 iterations the validation loss begins to rise, even though the training loss keeps decreasing. This is a sign for overfitting since the autoencoder is learning the specific representations for the samples in the training set, but loses generality on the validation set. Thus, the final autoencoder is trained for only 200 iterations.



(a) Loss over training set.

(b) Loss over validation set.

Figure 26: Binary cross entropy loss in autoencoder.

7.2 Clustering

The K means algorithm has only one parameter: the number of clusters in which to allocate the data. As such, this parameter must be properly tuned in order for the algorithm to produce valuable results.

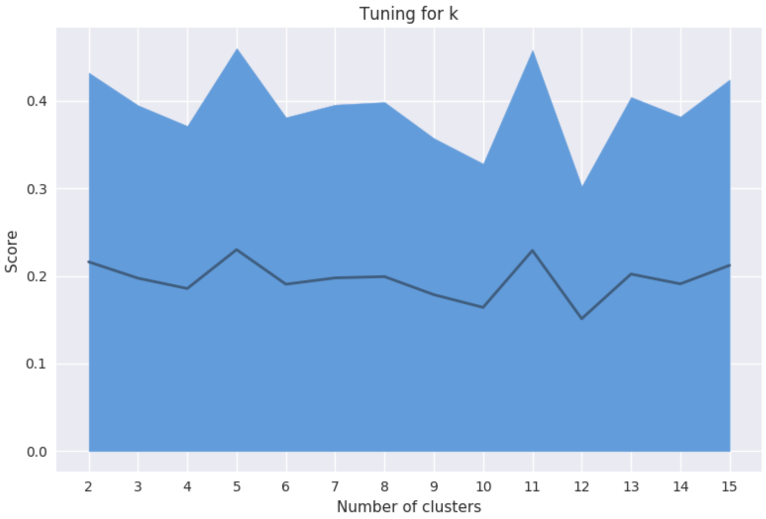


Figure 27: Silhouette average scores and their standard deviation.

In order to tune K, we split the data into train and test sets on a 6:4 ratio. Then, we test the performance on values for K from 2 to 15. The training set serves the purpose for calculating the clusters centers. The test set aids in evaluating the quality of the clusters, according to how test samples are assigned. For evaluation purposes we calculate the silhouette score.

The silhouette score is determined using the mean intra-cluster distance and the nearest-cluster distance (which the sample does not belong to). It ranges from 0 to 1, with higher scores corresponding to more condensed well-formed clusters. The score is measured per sample, and averaged over all the set.

Figure 27, shows the average score per K value, as well as the standard deviation in the samples scores distribution. The average score peaks at $K = 5$ and $K = 11$. We select 5 as the optimal number of clusters.

7.3 Cluster analysis

Qualitative analysis

Figure 28 shows the samples, as visualized by TSNE, and their corresponding cluster label. Though not completely separable, samples assigned to the same cluster according to their encoded representation indeed gather together in the visualization.

With regards to the distribution, clusters are significantly unbalanced. For instance, Cluster 0 has significantly less members than Cluster 1.

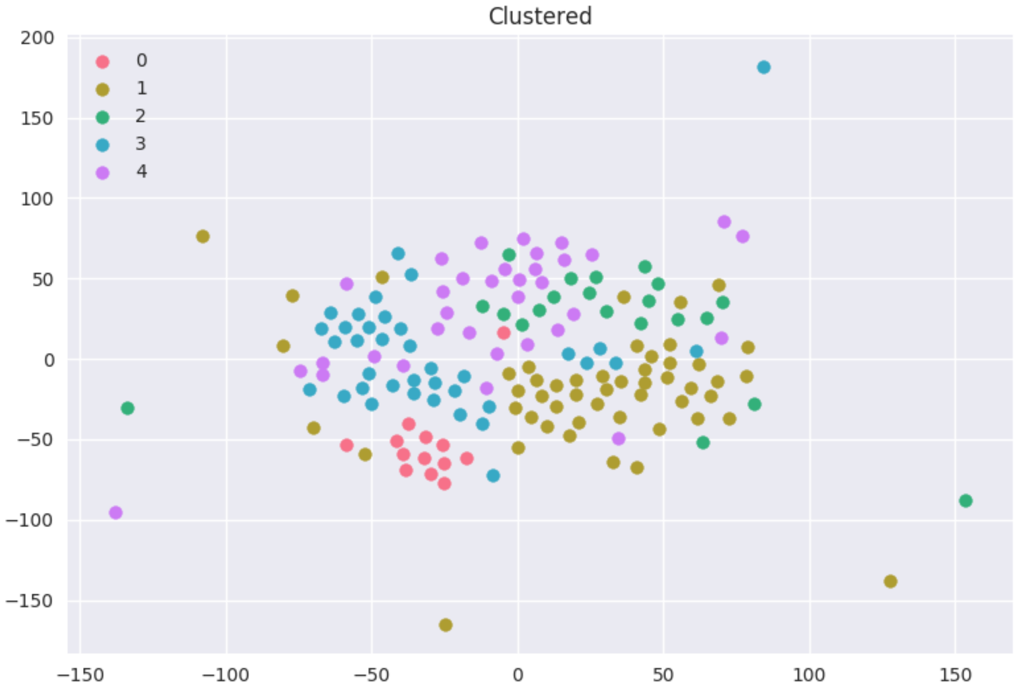


Figure 28: Visualization of samples and their labels according to clustering.

Quantitative analysis

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Number of cards	None	None	None	None	None
Percentage or cards	None	None	None	None	None
Number of Trips	None	None	None	None	None
Travel time	None	None	None	None	None
Travel distance	None	None	None	None	None
Start hour	None	None	None	None	None
End hour	None	None	None	None	None
On mode	None	None	None	None	None
Off mode	None	None	None	None	None

Table 7: Cluster analysis

7.4 Discussion

8 Conclusion and future work

In this project we examine public transit traveling behavior using data mining techniques. As the main contribution of this work, we propose a three dimensional representation that takes advantage of the local properties of weekly schedules, and incorporates up to 26 different trip attributes. Using this representation we compare supervised and unsupervised machine learning techniques for pattern recognition.

Three dimensional representation

The most significant advantage of the proposed representation is the ability to analyze spatiotemporal features as a unit. This is a result of representing the distribution of each user's trips overtime on a plane, and the spatial attributes in depth. As such, each temporal pixel may contain at most one trip, according to its boarding time. In our work we crop all boarding times to use the hour as a key. However, this may result in overlapping trips if more than one trip started at during the same hour (i.e. Trip 1 starts at 10:01 am, Trip 2 starts at 10:59 am). Possible simple solutions include to concatenate said trips, or create a finer temporal pixel grid.

However, the main drawback of our representation is its sparsity. The first major implication of this is the growth in computational resources needs, as three dimensional arrays require more storage space and ram requirements than the original raw files. In fact, the space requirements became unmanageable while attempting to analyze the whole dataset during the unsupervised task, leading to the decision to use only a reduced dataset.

The second major implication is the meaning of zero values in our representation. When constructed, all user cubes are "empty", which is signaled by having zero values in all its cells. However, some trip attributes may have zero values that do not necessarily mean empty fields. Thus, while considerable effort was put into avoiding zero values (i.e. in the creation of numerical IDs for modes, lines and stops), some attributes still contain them (i.e. start/end hours or number of transfers). Future work could investigate the effect of having a separate special value for empty fields (-1, for example).

Dimensionality reduction

Due to its sparsity, our representation requires compression. As such, we explore two approaches for dimensionality reduction. On the one hand, we perform typical feature selection by scoring each trip attribute according to a variety of techniques. On the other hand, we build and train a convolutional autoencoder that maps the high dimensional user representations to a low dimensional space.

The fundamental difference between these two approaches is that feature selection operates at a trip level, disregarding attributes that do not aid in identifying the correct user label. Meanwhile, the autoencoder has a more general view of the user behavior as it works at a user level. At every train iteration, the autoencoder attempts to deconstruct and reconstruct each user's behavior, without committing each one of them to memory. Thus, while feature selection compares attributes to each other, the autoencoder compares users to each other, thus abstracting the main similarities and differences among them.

To the best of our knowledge, three dimensional representations and autoencoding techniques have not yet been applied in the domain of Metropolitan Transportation. Therefore, there is still considerable amounts of future work in which this project can be expanded. Some suggestions are the application of three dimensional convolutional filters, using the feature slices as an extra dimension instead of channels. Extensions to the autoencoder are also interesting paths to follow, for example denoising autoencoders and variational autoencoders could be compared to this work's simple implementation.

Pattern recognition

Our findings show that ensemble models improve performance over any weak classifier alone. Furthermore, they illustrate that the required complexity of the model has a balance between same-type and different-type learning algorithms.

Beyond self-reported commuters, we find there are 5 types of distinct traveling behaviors in our dataset.

Use case relevance

This project analyzes one month worth of public transit data from the city of Beijing. Beijing poses a challenging use case due to the rapid urbanization of the last years. Furthermore, its public transit network is extensive and undergoes constant development in order to serve millions of passengers. As such, public transit users in Beijing may present traveling behaviors that are more complex than typical "commuter" definitions can describe.

Different from most studies on the field, by studying users over a longer timespan (one month compared to one week), we are able to discern more complex behaviors. However, data quality may be sacrificed while intending to grasp a long period of time, since it opens possibilities for data to be faulty. However, data acquisition methods grow as rapidly as the public transit network. We hope that in the future we have more complete data. This way we will be able to have a better understanding of commuter's behaviors.

References

[1] Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff. A medium-scale distributed system for computer science research: Infrastructure for the long term. *Computer*, 49(5):54–63, 2016.

[2] Ashish Bhaskar, Edward Chung, et al. Passenger segmentation using smart card data. *IEEE Transactions on intelligent transportation systems*, 16(3):1537–1548, 2015.

[3] Philip T Blythe. Improving public transport ticketing through smart cards. In *Proceedings of the Institution of Civil Engineers, Municipal Engineer*, volume 157, pages 47–54. Citeseer, 2004.

[4] Beijing Transportation Research Center. 2016 annual report on traffic development in beijing. `http://www.bjtrc.org.cn/InfoCenter/NewsAttach/2016%E5%B9%B4%E5%8C%97%E4%BA%AC%E4%BA%A4%E9%80%9A%E5%8F%91%E5%B1%95%E5%B9%B4%E6%8A%A5_20161202124122244.pdf`, 2016. Acessed on 21 April, 2017.

[5] Mo Lim Chan. *Tactical implementation model for the smart card payment system for metro operation*. PhD thesis, City University of Hong Kong, 2010.

[6] Patrick YK Chau and Simpson Poon. Octopus: an e-cash payment system success story. *Communications of the ACM*, 46(9):129–133, 2003.

[7] Gerhard de Koning Gans. *Analysis of the MIFARE Classic used in the OV-chipkaart project*. PhD thesis, Master's thesis, Radboud University Nijmegen, 2008.

[8] Kamran Ghasedi Dizaji, Amirhossein Herandi, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. *arXiv preprint arXiv:1704.06327*, 2017.

[9] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[10] US EIA. Energy information administration (2016), international energy outlook 2016, with projections to 2040. Technical report, DOE/EIA-0484, 2016.

[11] Gabriel Goulet Langlois, Haris N Koutsopoulos, and Jinhua Zhao. Inferring patterns in the multi-week activity sequences of public transport users. *Transportation Research Part C: Emerging Technologies*, 64:1–16, 2016.

[12] Xiaolei Ma, Congcong Liu, Huimin Wen, Yunpeng Wang, and Yao-Jan Wu. Understanding commuting patterns using transit smart card data. *Journal of Transport Geography*, 58:135–145, 2017.

[13] Xiaolei Ma, Yao-Jan Wu, Yinhai Wang, Feng Chen, and Jianfeng Liu. Mining smart card data for transit riders' travel patterns. *Transportation Research Part C: Emerging Technologies*, 36:1–12, 2013.

[14] Zidan Mao, Dick Ettema, and Martin Dijst. Commuting trip satisfaction in beijing: Exploring the influence of multimodal behavior and modal flexibility. *Transportation Research Part A: Policy and Practice*, 94:592–603, 2016.

[15] David Marr and A Vision. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*, 1(2), 1982.

[16] Catherine Morency, Martin Trepanier, and Bruno Agard. Measuring transit use variability with smart-card data. *Transport Policy*, 14(3):193–203, 2007.

[17] D Kanishka Nithin and P Bagavathi Sivakumar. Generic feature learning in computer vision. *Procedia Computer Science*, 58:202–209, 2015.

[18] OECD. Passenger transport (indicator). `10.1787/463da4d1-en`, 2017. Acessed on 10 April, 2017.

[19] Beijing Municipal Commitee of Communications and Beijing Transportation Reserch Center. Research on beijing's public transportation commuting transit network. https://www.esmap.org/sites/esmap.org/files/10282009102930_Beijing_Transport_finalReport.pdf, 2009. Acessed on 21 April, 2017.

[20] Meisy Andrea Ortega-Tong. *Classification of London's public transport users using smart card data*. PhD thesis, Massachusetts Institute of Technology, 2013.

[21] Tu Qiang. personal communication.

[22] Liang Quan. personal communication.

[23] World Population Review. Beijing population. http://worldpopulationreview.com/world-cities/beijing-population/, 2016. Acessed on 21 April, 2017.

[24] Qiang Tu, Jian-cheng Weng, Rong-Liang Yuan, and Peng-fei Lin. Impact analysis of public transport fare adjustment. *Traffic Engineering & Control*, 57(2), 2016.

[25] UITP. World metro figures, statistics brief. http://www.uitp.org/sites/default/files/cck-focus-papers-files/UITP-Statistic%20Brief-Metro-A4-WEB_0.pdf, 2015. Acessed on 21 April, 2017.

[26] Vukan R Vuchic. Urban public transportation systems and technology. 1900.

[27] Kunyan Wang, Qiong Luo, and Xueying Zang. Studies on ecological environmental carrying capacity in beijing, tianjin, and hebei region. In *Report on Development of Beijing, Tianjin, and Hebei Province (2013)*, pages 119–137. Springer, 2015.

[28] Jiancheng Weng, Yueyue Wang, Jianling Huang, and Ledian Zhang. Bus operation monitoring oriented public transit travel index system and calculation models. *Advances in Mechanical Engineering*, 2013.

[29] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016.

[30] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.

[31] Hefeng Zhang, Shuxiao Wang, Jiming Hao, Xinming Wang, Shulan Wang, Fahe Chai, and Mei Li. Air pollution and control action in beijing. *Journal of Cleaner Production*, 112:1519–1527, 2016.

[32] Shaojun Zhang, Ye Wu, Huan Liu, Ruikun Huang, Liuhanzi Yang, Zhenhua Li, Lixin Fu, and Jiming Hao. Real-world fuel consumption and co 2 emissions of urban public buses in beijing. *Applied Energy*, 113:1645–1655, 2014.

[33] Jiangping Zhou, Enda Murphy, and Ying Long. Commuting efficiency in the beijing metropolitan area: an exploration combining smartcard and travel survey data. *Journal of Transport Geography*, 41:175–183, 2014.