Deriving trajectory embeddings from global positioning system movement data

by

Armand Graaff (u16038569)

Submitted in partial fulfillment of the requirements for the degree

MSc Advanced Data Analytics

In the Department of Statistics

In the Faculty of Natural and Agricultural Sciences

Univeristy of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

December 2022

Acknowledgements

I would like to thank Alta De Waal, my research supervisor, for continuously supporting me throughout the completion of this work. I would also like to thank Johan Joubert, for providing the GPS driving data that was used in Chapter 5.

Abstract

Analysing unstructured data with minimal contextual information is a challenge faced in spatial applications such as movement data. Movement data are sequences of time-stamped locations of a moving entity analogous to text data as sequences of words in a document. Text analytics is rich in methods to learn word embeddings and latent semantic clusters from unstructured data. In this work, the successes from probabilistic topic models which are used in natural language processing (NLP) were the inspiration for applying these methods on movement data. The motivation is based on the fact that topic models exhibit characteristics which are found both in clustering and dimensionality reduction techniques. Furthermore, the inferred matrices can be used as interpretable topic distributions for movement behaviour and the lower dimensional embeddings generated by the LDA model can be used to cluster movement behaviour.

In this work various existing techniques for trajectory clustering in the literature are explored and the advantages and disadvantages of each method are considered. The challenges of trajectory modelling with LDA are examined and solutions to these challenges are suggested. Lastly, the advantages of using LDA compared to traditional clustering techniques are discussed.

The analysis in this work explores the use of LDA to two use cases. Firstly, the ability of LDA to infer interpretable topics is explored by analysing the movement of jaguars in South America. Secondly, the ability of the LDA to cluster movement trajectories is investigated by clustering driver behaviour based on real world driving data. The results of the two experiments show that it is possible to derive interpretable topics and to cluster movement behavior of trajectories using the LDA model.

I, *Armand Graaff,* declare that this mini-dissertation (100 credits), which I hereby submit for the degree Magister Scientiae in Advanced data analytics at the Univeristy of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature:

Date: 05-12-2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"We must be careful not to confuse data with the abstractions we use to analyze them." - William James

Mobility can be seen as a significant element that contributes to the ongoing development of society. Data on the movement of individuals, vehicles, animals and objects are being collected at an ever increasing rate. The advances in GPS equipped devices such as smart phones, smart watches, vehicle tracking devices and animal collars, as well as the wide scale availability of, and access to commercial satellites, means that the collection of positioning data that is generated by the movement of objects has become easier than ever. It is, however, challenging to analyse large trajectory data sets to extract meaningful insights.

The US-military made the use of accurate GPS technology available to the public in the year 2000, and since then, development of wireless communication and wireless sensing have been flooding institutions and researchers with data containing time-varying geographic positions [41]. Researchers and analysts have shown that these datasets constitute a valuable resource in the form of human movement behaviour. Analysing these datasets can lead to important insights and solutions in multiple domains like the aviation industry, which uses movement data for navigation and collision avoidance. Other fields where the analysis of movement data is useful include farming activities, commercial fishing, urban planning, surveillance, sport scene analysis, behavioural ecology, and security [32]. Thanks to the recent developments in GPS and sensor technologies, the large scale collection of the changing latitude-longitude points, which can be seen as independent discrete objects, became technically and economically feasible [56].

Trajectory data analysis is useful in an increasing number of applications, which aims at universal comprehension and management of complicated scenarios which involve moving objects [56]. The explosion of big trajectory data has led to very active research in the field of trajectory analysis [30]. From a machine learning (ML) perspective, trajectory data lends itself to clustering and unsupervised learning, as it is

very seldom labelled due to its dynamic nature and sheer size. This places the emphasis of trajectory analysis on discovery of patterns. As an example, vehicle driving is a complex behavioural task with many interacting features, which makes it difficult to model or predict. Evans et. al (1985) [18] stated that human behavioural patterns is a pervasive phenomenon in traffic systems, and that this can possibly have a great influence on the effect of safety measures. In another example of behavioural analysis of trajectory data, Long et al. (2016) [39] identified four kinds of extreme transit behaviors: early birds, night owls, tireless itinerants, and recurring itinerants. What makes this transit behaviour study of interest to the work in this paper is that it followed a semantic approach by setting working definitions for each behaviour type and then identified extreme travelers from an in-vehicle smart card dataset. Also, Long et al. (2016) [39], grouped movement of people into clusters based on similar movement behaviour, which is something that will be explored in the application chapters of this work.

Semantic trajectory analysis uses contextual data to enrich trajectory data. This contextualisation can include street names, points of interest (POI) and transport types if such data is available. This semantic information is extracted jointly with the mobility data (trajectories) and the underlying geographic and other domain-specific data [61]. This enrichment creates the opportunity to apply many more ML techniques which can learn from contextual data. The word *semantics* immediately draws attention to the field of natural language processing (NLP). Grounded on the distributional hypothesis, developed by Harris (1945) [34], which states that words in similar contexts have similar meanings, the NLP community explored numerous methods of deriving word representations [35]. One of the proposed methods is to represent words as dense vectors. It has been shown that these representations, referred to as "word embedding", perform well across a diverse range of tasks such as question-answer pairing and conversational agents (chatbots) [59] [55] [1].

Word embeddings are lower dimensional representations of words in line with the manifold hypothesis, which states that higher dimensional data lie within lower dimensional forms embedded inside higher-dimensional spaces [19]. Word embeddings took the artificial intelligence (AI) world by storm, producing state-of-the-art models trained on billions of word pairs. GPT-3 [11] is arguably the most powerful language model up to date. GPT-3 (Generative Pre-Trained Transformer) is a third generation, autoregressive language model which utilises deep learning methods to produce human like text [21]. In the context of movement data, one can think of a trajectory as a collection of semantic events in the same way one may think of a document as a collection of words. Once this analogy has been established, we have access to a wealth of NLP models which are specialised to model temporally related events.

We are confined to the availability of semantic information in the data. Telematics data, for example, have inherent spatial and temporal properties, often derived from GPS receiver devices or wearables. The resulting datasets are large tables of point locations in time, with fields for identifying the moving objects

and carrying attributes such as speed, track and other semantic information.

From an application point of view, the ability to record continuous movement is a first foundational step in managing movement data, but satisfying application requirements usually call one to go a step further. In other words, many applications require a more organised way to record movement, that is, movement needs to be recorded as a temporal sequence of points, each lat-lon point following the previous time interval on the object's lifespan. Time-wise, these points lie between the departure point and the destination point [56].

In this work, an NLP technique, in particular, Latent Dirichlet Allocation (LDA), was applied to trajectory data with semantic information.

The scope of this work was limited to the movement of discrete objects [5], of which the spatial positions can be represented by points. This work does therefore not take into consideration the movement of large units that can change in shape and size, such as ocean currents or clouds. The methods applied in this work does not lend itself to the analysis of these types of moving objects.

1.1 Motivation

The motivation for this work is based on the fact that topic models exhibit characteristics which are found bin oth clustering and dimensionality reduction techniques. The geo-spatial points along a trajectory can also be regarded as similar to words in a document, especially if the data points contain additional semantic information. This analogy opens up a wide range of NLP methods which can be applied to trajectory data. The main benefits of using LDA to analyse trajectory data comes down to two points. Firstly, the inferred matrices can be used as interpretable topic distributions for movement behaviour and secondly, the lower dimensional embeddings generated by the LDA model can be used to cluster movement behaviour.

The goal of this work is therefore not to compare the clustering results of the LDA model to that of other techniques in this field, but to show that latent Dirichlet allocation can be used in order to obtain interpretable topics from unstructured movement data and that LDA can also be used in order to cluster movement data. The ability of LDA to detect movement patterns will also be investigated as part of the clustering application of this work.

Chapter 2

Literature study

In this chapter, existing research on the fields of trajectory analysis, and the application of NLP techniques on trajectory data are explored and discussed.

2.1 Analysis of trajectory data

Movement data in the form of trajectories are particularly heterogeneous. Movement datasets can differ in a variety of different aspects [26]. These include, but are not limited to, temporal resolution (frequent/constant to sparse), spatial resolution (refined to rough), spatial dimensions (2D to 3D), movement constraints (area and/or network-constraint or not), movement models (Lagrangian or Eulerian) [54], tracking system (cooperative or uncooperative) [64], differences in size of the datasets, the availability of semantic information in the data and privacy constraints when it comes to collection and analysis or distribution of the data.

In geographical literature, the availability of movement data was traditionally often limited to information about the movement between the origin and the destination (OD flows). On the other hand, modern sources of data contain more detailed episodic or quasi-continuous information about the movement data [25, 5]. This type of data is known as trajectory data. Demsar et al. (2015) [16] defined trajectory data as a discrete time series of measured locations. This is important to note since this frequent and discrete data is required for the use of LDA models.

With the rise in the amount of global positioning system (GPS) data, there was a boom in the collection of GPS data, which was followed by the capturing and storing of movement data. The quantum geographical information system (QGIS), first released in 2002, is an open source system that offers standard

GIS functionality, with a variety of mapping features and data editing. This QGIS system was developed by Gary Sherman, and thoroughly researched by Anita grazer [25]. Over the last couple of decades these GIS systems have proved to be agile and powerful tools in many academic, civic and political disciplines [44]. With the rise of wearable GPS technology such as smart watches or smartphones, the next logical step was to track the movement of the GPS points/geo-locations in the form of trajectories.

The large scale capture of the movement data of individuals and objects, was made possible by access to low cost GPS devices. The knowledge of movement patterns of consumers became useful tools for companies to drive their business value. Google can for example, track the movement of all their customers using the Google Maps application and use this data to provide better navigation for customers, especially in busy cities. Zoologists can track the movement behaviour of animals over time to determine the impact of different environmental factors on the movement behaviour of the animals [24, 27]. Other examples include the use of hiker movement patterns obtained from GPS data to improve the management of parks [42] and a systematic review of the contextual factors on rugby league match running, focusing on the complexity of analysing this type of GPS data [14].

Ferrante et al. (2018) [20], devised an extensive framework for the analysis of cruise passenger movement at different destinations. The researchers built the framework to improve the understanding of cruise passenger behaviour at different stops. Studies also bring to mind the tracking and analysis of visitors at large theme parks such as Disney world, in order to efficiently distribute traffic around the park to decrease the amount of bottlenecks forming at different parts of the park and to enhance the experience for both the tourists and the staff.

Tracking companies can provide fleet management as a service to trucking companies, so that these companies can track whether their drivers are driving according to their standards, and even to suggest ways in which these trucks can operate in a more fuel efficient manner. Needless to say, there are endless applications in the analysis of traffic data.

2.2 Trajectory clustering techniques

Clustering is a popular branch of unsupervised learning that is used in many machine learning projects to handle a large amount of data. Clustering is used for the grouping of similar entities into bundles, or clusters, and usually employs distance functions as the measure of similarity. The aim of trajectory data clustering is to use the movement characteristics of trajectories to group a trajectory dataset into a finite

number of clusters. If the clustering technique performed well, the trajectories that are clustered together will exhibit similar movement characteristics, and will be different than the movement characteristics in other clusters [41]. In this section various clustering techniques that are used in trajectory analysis are explored, and the advantages and disadvantages of the different techniques are discussed.

Since trajectories of moving objects can be seen as spatio-temporal constructs, usually of a complex nature, and are characterised by diverse non-trivial properties [53], they contain many potentially relevant characteristics. These characteristics include the geometry of the earth, the object's spacial position, the life span of the object/trajectory, and the dynamics of the object (the manner in which the object's spatial location, direction/heading, speed/acceleration, and other point related characteristics of the movement which change over time). Since trajectories have so many rich characteristics that can be analysed, trajectories are well suited to clustering.

Some of the most common trajectory clustering methods are extentions of classical clustering techniques, with the important exception that the distance (or similarity) functions are properly defined to meet the needs of trajectory data. The majority of these algorithms can be split into 3 main categories [41].

1. Hierarchical algorithms like BIRCH [65].
2. Partitioning algorithms like k-means [38, 40].
3. Density-based algorithms like DBSCAN [17].

Firstly, the hierarchical methods order the objects in a multi-level structure containing clusters and sub-clusters. Based on loose similarity requirements, the procedure clusters trajectories at a higher level, while by tightening similarity requirements, sub clusters are found. BIRCH (Balanced iterative reducing and clustering using hierarchies) [65], mentioned above, is an example of one of these hierarchical algorithms. Secondly, in the partitioning methods, all trajectories are arranged into a pre-defined number of clusters. These techniques start by forming random partitions, which is then iteratively refined, where trajectories move between clusters at each iteration. K-means [38, 40] is an example of a partitioning algorithm. Lastly, density-based algorithms work by partitioning the trajectories based on their density, the cluster starts at one trajectory, and keeps growing as long as new objects are present in the neighbourhood. The cluster is deemed valid if the total amount of trajectories in the cluster surpass a certain threshold. DBSCAN (Density-based spatial clustering of applications with noise) [17] is an example of a density based clustering algorithm.

The clustering of trajectories can pose a number of challenges when applying traditional clustering techniques mentioned above. Olive et al. (2020) [48] identified multiple factors that make clustering trajectories particularly challenging and argues that traditional clustering techniques are not well-suited to trajectory data, these factors are listed below.

1. The functional nature of trajectories make it difficult to define an appropriate distance function between trajectories.

2. Trajectory data is often available as data points in high-dimensional space, and traditional distance metrics lose their precision when applied to high dimensional data (a.k.a. the curse of dimensionality) [31].

3. The processing of large amounts of open trajectory data, require tremendously efficient and highly scalable trajectory clustering algorithms.

Focusing on the second point, conventional trajectory clustering techniques that rely on traditional similarity (or distance) functions may be ineffective when defined directly on high dimensional data. This is because, when applied to trajectories directly, the distance (or similarity) metrics used in these algorithms are not as effective in capturing the richer dependencies which are potentially present in the lower-dimensional latent space. Therefore classical clustering techniques (k-means [38], DBSCAN [17], and BIRCH [65]) are ineffective when attempting to cluster trajectory data.

To avoid the issues with these traditional clustering techniques, Olive et. al (2020) [48], developed a deep trajectory clustering technique with autoencoders in order to analyse air-travel trajectories landing at Zurich airport. This algorithm embeds trajectories into the latent spaces to enable the clustering of the trajectories.

In addition to the deep trajectory clustering by Olive et. al (2020) [48], more trajectory specific clustering techniques have been proposed, mostly by modifying statistical and probabilistic models. For example, Gaffney et al. (1999) [23], developed a clustering approach based on mixture models, which clusters trajectories together based on the likelihood that they are generated by a common representative trajectory. Alone et al. (2003) [3], developed an approach that models trajectories as chains of transitions between locations and uses a hidden Markov model (HMM) that is most suited to the trajectories to model a cluster.

In the most basic sense, trajectories are clustered by taking each trajectory in a set of trajectories and allocating it to a cluster. There are, however, cases where the aim is not to cluster a set of trajectories, but one trajectory [41]. In these cases, the goal is to cluster individual points on a single trajectory, in order to characterise the positions along the trajectory. In the study by Palma et al. (2008) [50], the researchers cluster points on a single input trajectory to discover the stopping points in it. Looking at trajectories on an even lower level, different segments of trajectories can be clustered separately, if researchers are interested in different geo-locations that the trajectories crossed, for example, if the similarity of trajectories is measured by visiting similar places, clustering can be applied on segments of trajectories [41]. The TraClus clustering algorithm, developed by Lee et al. (2007) [33] followed this approach, by proposing a

partition-and-group framework, which works by partitioning a trajectory into a collection of line segments, and then grouping similar line segments.

2.2.1 The use of distance functions in clustering of trajectories

Rinzivillo et al. (2008) [53] advocated the use of a wide range of distance functions which address various properties of trajectories and present the method of progressive clustering, allowing analysts to include distance functions as part of analyses. Using this procedure, this analytical process is broken into a series of steps [5]. Clustering is applied at every step using one distance function. This offers a few advantages. Firstly, the analyst can refine clustering results, secondly, the analyst can consolidate multiple distance functions containing different semantics and lastly, the analyst can use a step-wise approach to build a thorough comprehension of different attributes of the trajectories. Various distance functions that are suited for trajectory clustering are suggested by Andrienko et al. (2007) [4].

Distance functions can also be used to calculate the distances between lower-dimensional embeddings, which are obtained by applying abstraction on high-dimensional trajectories. The distances between these embeddings often yield better clustering results when compared to applying distance functions on raw trajectory data [48]. This will be explored in one of the application sections of this work.

2.3 Semantic trajectory analysis

When comparing trajectories in any solution, the context in which the trajectories have been captured should always be considered, and analysts should ensure that the contexts are the same for the trajectories in question [5].

Most GPS trackers capture raw trajectory data *i.e.* data in the form $<x,y,t>$, where x and y is the latitude and longitude of an object, and t is the time point. The t parameter can also be used for altitude (z), in which case it would be <x,y,z>, but for most trajectory analysis studies, the time parameter provides more useful information. In semantic trajectory analysis, the goal is to augment the raw trajectory data by adding another data point to capture semantic information.

Semantic trajectory data refers to any additional data that accompanies the raw trajectory data, this can include additional information regarding the activity, transport mode, etc [2]. Understanding how and why vehicles, individuals and animals move, the locations that they visit, the frequency or patterns in which they visit certain locations, and the resources used, are important in decision making. Applications such as mobile health, the monitoring of road traffic, and animal data ecology, require semantic trajectory

data to enable the rich and detailed portrayal of moving objects. For example, when analysing traffic data, it would be useful to add data like street names, vehicle speed and/or heading data, such as turns and stops.

A study by Albanna et al. (2015) [2], focused on the combination of semantic data and raw trajectory data in the analysis of movement data. It is important to note that, GPS trackers capture only raw information, and other semantic information often does not exist and can therefore not be used in the analysis. When incorporating semantic information in the analysis, semantic data should be captured from the outset, making clear that the data will be used for analysis. For example, if the aim is to analyse the driving behaviour of a person or group of people, and heading data like stops, turns, and speeding are used, these heading events should be captured along with the GPS trajectory data.

The addition of semantic information strengthens the analysis of data and eases the detection of semantically implicit patterns and behaviours. In a study by Chu et al (2014)[13], the geo-locations of a massive taxi trajectory dataset were transformed to street names, which reflected contextual semantic information. The trajectory data of each taxi was studied as a document that contained the street names that the taxi traversed, which enabled the semantic analysis of a document corpora, with NLP techniques such as topic modelling.

Yan et al. (2013) [61] developed a semantic method which progressively turns raw mobility data into semantic trajectories that can immediately be used in analyses and applications. The model computes trajectories at different levels, ranging from basic high-level location feeds, to low-level semantic behaviours. In this work, different ways to extract semantic information from raw trajectory data will be experimented with. This semantic information will be used in a LDA model to cluster moving objects.

There are many more examples which showcase the potential benefits of analysing movement data. The nature and sheer volume of this type of data brings to mind a few machine learning techniques.

Clustering similar trajectories is a common method used to discover similar driving patterns and behaviours. It furthermore results in a latent representation of each cluster, which - depending on the clustering algorithm - can hold semantic information and interpretability. Important surveys focused on trends and research in the field of trajectory data mining and clustering include [41], [63] and [8].

De Almeide et al. (2020) [52] also pointed out the need to investigate behavioural data as opposed to raw trajectories only. Behavioural aspects in the data are contextual or semantic information such as points of interest, transport type and street names. The enhancement of raw trajectories with semantic annotation

is referred to as semantic trajectories [51].

2.4 Trajectory clustering techniques using LDA

The principal machine learning technique that was tested in this work, is Latent Dirichlet Allocation (LDA), which is a NLP technique. There have been a couple of studies done on trajectory analysis with LDA. One of these studies analyses the analysis of taxi movement by building a corpora of the street names that the taxi's traversed, and applying topic modelling on that dataset [13]. Chu et al. (2014) [13], devised a new visual analytics system, by mapping GPS lat-lon coordinates to street names, and studying each trajectory as a document containing the street names that taxi's traversed. This technique can be seen as semantic trajectory analysis since it can replace or supplement spatio-temporal GPS trajectories with contextual information.

In [13], the street names in each trajectory represent the words in the LDA corpora, and they are used to discover group movement patterns. This methodology did not consider trajectory direction, which was addressed by Liu et al (2019) [37] by combining adjacent street names in the trajectory as bigrams.

Cao et al. (2019) [12] propose another NLP technique that can be modified to analyse trajectory embeddings. They define a person's habit signature unit transition (H_u) as a feature representing the typical specific locations a person visits at a specific time slice. H_u can be defined in the format $(u_i, h_1^{p_1}, h_2^{p_1}, ..., h_m^{p_1})$, where u_i is a unique identifier of a person, and $h_i^{p_i}$ are the different recorded locations for that person. Cao et al. (2019) [12] defined the similarities in natural language and a person's H_u in the following points.

- Both can be regarded as time-dependent series.
- Both can be approximated by context
- There is a large scale of data available for both NLP and signature habit transition from which their characteristics can be learned.
- The frequency distributions of natural language and habit unit are very similar.

From these similarities, a likeness can be drawn between learning representation for signature habit trace and word embeddings. Hence, an algorithm was proposed to learn an individuals' habits from the trajectory, inspired by the methodology of word2vec, the technique was called habit2vec [12].

Meriono [43] investigated naturalistic driving by constructing a behavioural profile for drivers using LDA.

Since LDA is usually applied in text-based studies, the driving data of drivers was converted into “document” and “occurrences” by applying symbolic time-series abstraction methods to the driving data.

In the work of this thesis, natural driving behaviour will also be clustered using NLP techniques. The key differences in the methods explored in this paper and existing methods are given below.

1. Features recorded in the driving data, for example speed, rpm, emissions, etc, were clustered individually using k-means clustering, and concatenated together to form different "words" at each point in the trajectory.
2. Lower dimensional embedding were obtained from these words using Latent Dirichlet allocation (LDA).
3. The similarity between the lower dimensional embeddings were calculated using statistical distance metrics such as the Bhattacharyya and Jensen-Shannon distance metrics.
4. A simulation study involving millions of iterations was executed in order to test the method thoroughly.

2.5 Software

Some of the software that already exists include MovingPandas, which is a trajectory analysis package, written by Anita Grazer [1]. The moving pandas library is written on top of the GeoPandas package, which is a spatial data analysis tool. As the name suggests, the MovingPandas and GeoPandas libraries are based on the popular Pandas Python library. The development of MovingPandas started as a plugins idea for QGIS data analysis in 2018. The resulting plugin (called Trajectools) was first published in 2019. However, it became clear that the core trajectory handling classes should be extracted into a separate library so that it can be used outside of the QGIS context. For data visualisation, MovingPandas uses Matplotlib for the static plots and hvplot for the interactive plots.

MovingPandas makes it straightforward to compute movement characteristics, including the length, duration, speed and direction of trajectories. It also has the capability of overlaying the trajectories on a actual map, so that you can see the trajectory in terms of the real world terrain. By leveraging the existing functionality within the Python data analysis ecosystem, such as the handling of time-series data by Pandas, and spatial data analysis by GeoPandas and close integration with Holoviews [2] (a package which makes data visualisation a lot easier) to enable interactive plots, MovingPandas can focus on it’s

[1] https://anitagraser.com/movingpandas/
[2] https://holoviews.org/

core functionality, which is dealing with the challenges that are specific to movement data.

Chapter 3

Theory

In this chapter the theoretical concepts that were used to apply LDA to trajectory data to infer interpretable topics and to cluster movement trajectories are discussed. A formal definition of a trajectory is given as well as a discussion on the statistical concepts that will be used to model the trajectories.

There are a number of challenges when analysing GPS trajectory data which form the motivation for the approaches developed in this chapter.

1. Trajectories as a set of $\langle x_k, y_k, t_k \rangle$ points are continuous data. In order to consider NLP techniques such as Latent Dirichlet Allocation (LDA), it makes sense to have discrete data which can be counted. This problem is addressed by discretising the continuous data, k-means clustering can be used for the discretisation of the features of the trajectories.

2. There can be a large variance in the dimensions of observations in a trajectory. The problem of varying dimensions is typically addressed by a kernel, and in the case of count data, Fisher kernels are often used [57]. This problem can also be solved by deriving a count-vector representation of trajectories, such as a Bag-of-words (BOW) vectorisation which is commonly used as an input to an LDA model.

3. Movement data often only contains spatio-temporal data in the form $\langle x_k, y_k, t_k \rangle$ i.e. $\langle lat, lon, time \rangle$ points, with no additional semantic information on the data, such as speeds, distances covered and geo-spatial landscape. This problem is addressed by applying abstraction to the data in order to extract meaningful features like speed, acceleration and headings. Furthermore, a basic unit needs to be defined, which is analogous to words in documents.

4. Trajectory data along with semantic information are often available in the form of data points in high-dimensional space and traditional distance metrics lose their precision when applied to high

dimensional data (a.k.a. the curse of dimensionality) [31]. This problem is addressed by deriving lower dimensional embeddings for the trajectories and using these embeddings for clustering.

Before discussing the challenges that are outlined above, a proper definition of a trajectory is given.

3.1 Trajectory definition

There is no standardised terminology in the discipline of movement data analysis. Each application domain and research group have their own terms for trajectory, point, node, path, travel and segment, which are all used interchangeably to refer to the same or different concepts related to movement. The definition of a trajectory varies between studies. For example, Demsar et al. (2015) [16] defined trajectory data as a discrete time series of measured locations, while Alvares et al. (2007) [22] and Baglioni et al. (2009) [6] defined a trajectory as a sequence of moves and stops. Spaccapietra et al. (2008) [56], on the other hand, described a trajectory as movement data that is structured into countable semantic units.

In terms of the mathematical definition of a trajectory, in the most basic sense, a trajectory can be defined as a line that a moving object follows through a geometric space. A trajectory is formally represented in this research project as $T = \langle p_1, ..., p_k \rangle$, where p_k is the time-ordered k^{th} point in T. The basic trajectory data only contains spatio-temporal information $\langle x_k, y_k, t_k \rangle$. This representation can be extended to contain an identifier variable in which case p_k is defined as a quadruple $\langle id_k, x_k, y_k, t_k \rangle$ [41]. The identifier id is a useful container to store additional information about individual positions in a trajectory such as speed, acceleration, and other semantic information. The representation p_k can also be extended further to store many features at once. This trajectory definition forms the basis of the kernel representation which will be discussed later on.

3.2 Addressing the challenges of trajectory clustering

3.2.1 Discretisation of continuous trajectory data.

As explained above, movement data in the form of points and semantic features along a trajectory are continuous. In order to use the LDA model, these features need to be discretised. The continuous data points can be discretised based on k-means clustering, or in certain cases, domain expertise and simple data exploration can be used to divide the features into meaningful clusters or "bin".

3.2.2 Count-vector representation

As mentioned above, there can be great variance in the dimensions of observations of a trajectory. For example, a vehicle journey can vary from visiting the local supermarket to travelling long cross country distances. In both the time and distance covered, these two journeys will result in two completely different trajectories based on the amount of data points in the respective trajectory. Because of this diversity between trajectories, a kernel has to be defined to meaningfully compare two different trajectories. This chapter explores the use of count-vector representations, discussed in Section 3.3, to address the problem of variation in the dimensions of trajectories.

3.2.3 Applying abstraction to extract semantic information

While movement data contains important information on the optimisation of positional and trajectory related infrastructures and services, in isolation it does not contain the required semantic embedding that would make fully automated machine learning (ML) analysis possible [4]. Semantic trajectory analysis is focused on the analysis of trajectories in context. At the moment, most of the methods developed for trajectory analysis focus on the spatio-temporal properties of movement trajectories without giving much attention to semantics. Therefore the high level semantic properties of trajectory data is not a very well investigated field. These techniques provide sub-optimal clustering results because they do not take into account other features that characterise movement/trajectory data. As part of doctoral work, Yan et al. (2009) [62] addressed this area, with the aim of combining semantic concepts and statistical computational methods, for the purpose of trajectory/movement data analysis. Chu et al. (2014) [13] provide a semantic trajectory model which considers spatio-temporal features (geo-location) and semantic trajectory units (for example turns and stops). The core focus of the trajectory analysis will therefore shift from movement data only, to semantically rich trajectories [51]. If movement data does not contain semantic information, a solution could be to apply abstraction to the data in order to extract meaningful features which can be used for clustering. In Chapter 4, abstraction was applied to raw spatio-temporal data in order to extract semantic information including distance, speed and acceleration and this semantic information was used to create interpretable clusters for the trajectories of jaguars.

3.2.4 Lower dimensional embeddings

Trajectory data is often only available in a high dimensional space. Many traditional trajectory clustering algorithms which are defined on the high-dimensional trajectory data (spatio-temporal information and semantic information) tend to miss information present in the lower dimensional latent space, and

will therefore struggle to cluster trajectories effectively. These clustering techniques often use traditional distance functions as a measure of similarity between clusters which can not detect differences between trajectories present in the lower dimensional latent space. In Chapter 5, lower dimensional embeddings that were obtained from an LDA model were used to create clusters. In that case, differences in driver behaviour were detected in a lower dimensional latent space. Since the embeddings are the topic distributions for each of the trips, these clustering results were achieved by defining distance functions, that calculate the differences between distributions, between the embeddings of the trips of drivers. The distances served as a measure of similarity between trips. The drawback of using the lower dimensional embeddings to cluster trajectories is the difficulty of interpreting the clusters obtained without expert knowledge in the application domain.

The clustering techniques which are discussed in this chapter enrich the definition of a trajectory as the trace of a moving entity not only containing geometric spatio-temporal features (the changing of a geometric landscape), but also semantic features (the context and meaning of the movement). By using this definition of a trajectory, geographic information and application domain knowledge can be combined to interpret clustering results[62].

Figure 3.1 illustrates the different levels at which GPS movement data can be analysed. Starting with the raw lat-lon points which do not contain much information, moving to a spatio-temporal trajectory that contains some information on the trajectory, such as heading, and start/stop points. Below that, the image shows a semantically enriched trajectory, containing information such as speed, acceleration, distance, type of vehicle used, street names, and where the trajectory stopped. All of this semantic information can be used to improve trajectory clustering results. In Chapter 4, spatio-temporal trajectories were changed into semantic trajectories by applying abstraction to the data. Lastly, at the bottom, a lower dimensional embedding is displayed, this is the lower dimensional embedding obtained for the first trip in Chapter 5. In Chapter 5 lower dimensional embeddings were derived from semantic trajectories using the LDA model, and these embeddings were used to cluster the data.

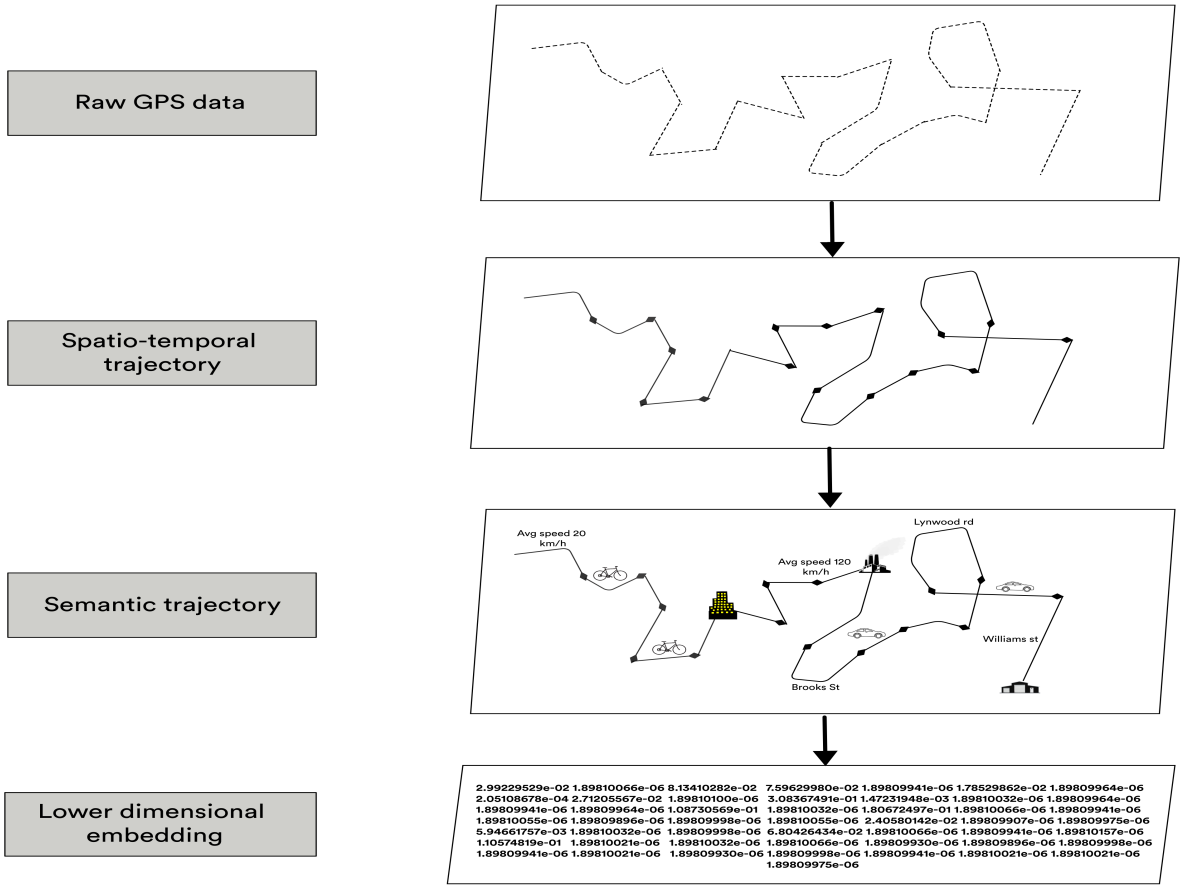


Figure 3.1: Schematic of the different levels that trajectories can be analysed at.

3.3 Count-vector representation of trajectories

Mathematically, vectors are geometric objects which have magnitude and direction. In NLP, vectors can be seen as a means by which words are projected onto a mathematical space, while preserving the semantic information contained in these words.

When considering a large collection of trajectories, each trajectory contains a varying number of geo-temporal events. Using machine learning terminology, each trajectory represents an object, or observation which can be clustered. Furthermore, it is assumed that such an object can be represented as a fixed-size feature vector in the form $\mathbf{x}_i \in \mathbb{R}^D$ [45]. Therefore, the first challenge is to convert the unstructured and often noisy trajectories in a mathematically computable form. If the structure of the trajectories is ignored and only the occurrences of events are counted, the result is a vector representation $\mathbf{x}_i$, where x_{ij} is the number of times that event j occurs in trajectory i. In the field of NLP this is known as the *Bag-of-Words* (BOW) vectorisation and an analogy between trajectories and documents becomes clear:

A trajectory is a grouping of events, just as a document is a grouping of words. The GPS signals (lat-lon points) can be compared with the words in the document. Similar analogies were drawn in [43, 13, 36].

In `gensim`, algorithms such as LDA use vectors as an input, this is mainly because the back end of these algorithms contain mathematical operations involving matrices. Therefore, what was formerly represented as a string, needs to be represented as a vector. In the case of trajectory clustering, points on a trajectory are continuous data points, which need to be discretised in order for the occurrences of certain events to be counted.

3.4 Trajectory modelling with LDA

The study on trajectory modelling indicates a focus on the clustering of trajectories. The soft clustering algorithm of LDA is investigated in this work. The techniques used in Chapter 4 follow a similar approach to Meriono et al. (2018) [43], which was discussed in Section 2.4 and focus on the explainability and interpretability of the obtained clusters. While in Chapter 5, a new strategy is applied, in which the lower dimensional embeddings obtained from the LDA model are used to detect differences in driver behaviour, and interpretability if the clusters is not considered.

The count-vector representation described in Section 3.3 opens a whole field of NLP algorithms that can be applied to trajectory modelling. The successes of probabilistic topic models serves as an encouragement for the use of these models on GPS location data. The motivation is based on the fact that topic models exhibit characteristics which are found both in clustering and dimensionality reduction techniques. Furthermore, their sparse representation make them efficient methods for data compression. Topic modelling is an unsupervised learning algorithm, such as clustering and dimensionality reduction techniques. Topic modelling discovers latent topics in large collections of documents, which is called a corpus. Therefore a corpus can be analysed without having any prior knowledge regarding their context. Topic models do not take document structure into account, but only count the occurrences of unique words in each document. Thus, the input to a topic model algorithm is a document $\times$ word vectorisation of the corpus. When applied to trajectory modelling, each trajectory will be represented as a document, and each lat-lon point on the trajectory will be represented as a word.

Latent Dirichlet Allocation is arguably the most widely used topic modelling algorithm and has been successful in the field of text analytics [10]. Based on the analogy between documents and trajectories, the LDA generative process for trajectory modelling is described as follows:

1. For all topics, randomly choose $\phi_k \sim Dirichlet(\beta)$.
2. For each trajectory, randomly choose a topic distribution, $\theta_m \sim Dirichlet(\alpha)$.

3. For each event, e_{mn}, in trajectory m:

   a) Randomly choose a topic assignment,
   $z_{mn} \sim Multinomial(\theta_m)$.

   b) Randomly choose an event,
   $e_{mn} \sim Multinomial(\phi_{z_{mn}})$.

The output of the LDA algorithm can best be described by its graphical model in Figure 3.2, where nodes represent variables and plates represent repeated structures. The variable z is the topic assignment for each word. The placement of z within the N plate illustrates the admixture effect of LDA, allowing each trajectory to contain multiple different topics in different proportions. One can also interpret the β and θ variables in terms of matrix factorisation as shown in Figure 3.3. LDA decomposes the original *trajectory* $\times$ *event* matrix into two lower dimensional matrices: The *event* $\times$ *topic* matrix (β) is a dimensionality reduction of the full representation of all the trajectories. The *topic* $\times$ *trajectory* matrix (θ) is a soft clustering assignment of trajectories to topics. The β matrix allows for the inference of interpretable topics from the data, this is explained in more detail in Section 3.5 and is explored in Chapter 4, while the θ matrix allows for the soft clustering of trajectories, which is explained in more detail in Section3.6 and is explored in Chapter 5.

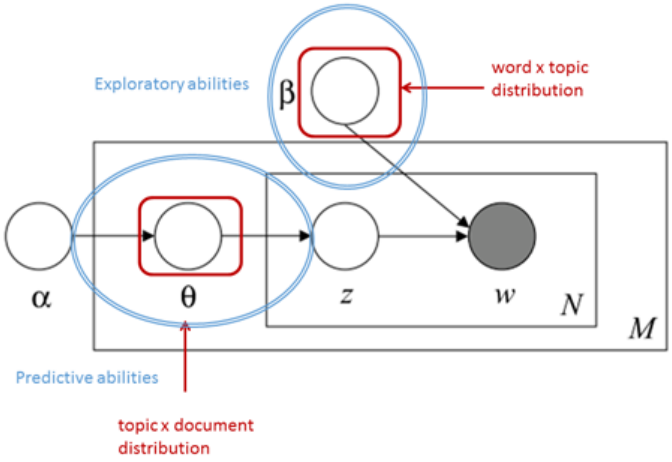


Figure 3.2: LDA graphical model. The grey node w represents the only observable variable in the model. In this case, it is the events in the trajectories. The β parameter represents the *event* $\times$ *topic* matrix and the θ parameter represents the *topic* $\times$ *trajectory* matrix.

Figure 3.2, shows the LDA graphical model for the normal LDA model, *i.e.* the LDA model used for text analytics. Figure 3.3 shows what the β and θ matrices looks like in the context of trajectory modelling.

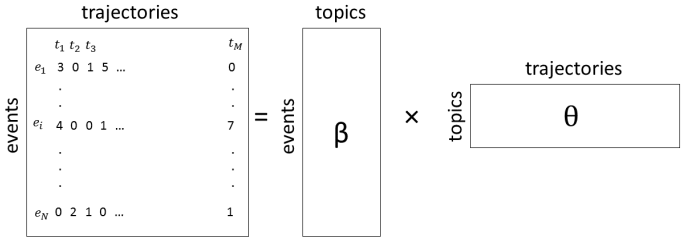


Figure 3.3: Matrix factorisation interpretation of LDA for trajectory modelling.

A problem that arises with the inference of the LDA model is the computation of the posterior distribution of the hidden variables given a trajectory [10]. This posterior distribution is intractable and need to be approximated using optimisation or sampling methods. The `gensim` package in Python uses a variational Kalman filtering approach [9] to compute the posterior variance. The core approximation code for the posterior is based on the online variational Bayes (VB) algorithm, developed by Hoffman et al. (2010) [28].

The input data for an LDA model should be in a count-vector representation format. In the context of text data, a document is a collection of words. Only the occurrences are counted and the structure of the document is ignored, hence a Bag-of-Words (BOW) vectorisation is created. In a similar manner, a BOW vectorisation is created for trajectories in the analysis and based on these vectorisations, the topic $\times$ trajectory matrix is obtained from the LDA model. The lower dimensional embeddings of each individual trajectory correspond to the rows in the topic $\times$ trajectory matrix. These embeddings are the topic distributions for each trip.

In Chapter 5, distance functions which calculate the distances between distributions, were defined and used to calculate the distances between the embeddings for each of the trips. These distances were then used as a measure to determine the similarity of the trajectories. Based on this similarity, clusters for the trajectories were obtained. The distance metrics that were used are defined later in this chapter in Section 3.6.2.

In the first part of this chapter, a theoretical overview of LDA as applied to trajectory modelling was given.

Next, an overview will be given on how LDA models were used, as well as how these models were evaluated. The application of the matrices, defined in Figure 3.3, and how they can be applied to obtain different results, will also be discussed.

3.5 Interpretability of topics

As explained earlier, the *event* × *topic* matrix (β) will be used to interpret the clusters. The probability distributions of the word in each topic will be evaluated in order to interpret the topics that the model created.

3.5.1 Top-topics per trajectory

As part of an analysis using topic modelling, it is possible to determine which topics relate more to which document, or in the case of trajectories, which topics relate more to which trajectory. Topic modelling obtained from the LDA model creates soft clusters for the trajectories, which indicate the probabilities with which each of the topics are contained in each of the trajectories. In other words, the probability for each of the trajectories to belong to the different topics can be calculated.

3.6 Clustering of trajectories

As explained earlier, the topic × trajectory matrix (θ) will be used to create soft clusters for the trajectories. Each row in this (θ) matrix is the LDA word embedding for each of the trajectories, which can be used to cluster the trajectories.

3.6.1 LDA word embeddings

Word embeddings derived from massive, unstructured corpora, are powerful tools for the detection of semantic regularities in natural language [15]. In the trajectory context of LDA, the lower dimensional embeddings are the rows of the topic × trajectory matrix, denoted as θ in Figure 3.3, these word embeddings are also known as topic distributions. An advantage of using lower-dimensional embeddings to cluster trajectories is that the embeddings allow the distance functions to capture information in the lower dimensional latent space. This is a major advantage when compared to traditional clustering algorithms, which do not take into account differences between trajectories present in the lower dimensional latent space, causing them to lose precision or to give poor clustering results.

3.6.2 Distance between the word embeddings

There are many distance measures that can be used to compare the similarity of two (or more) probability distributions. In this work two different distance measures will be used to measure the similarity between

topic distributions.

3.6.2.1 The Bhattacharyya distance

The Bhattacharyya distance measure is a popular distance measure that is used to compare the similarity of two probability distributions. Advantageous to the application in Chapter 5, it can also be used to calculate the separation of classes in classification problems. In an application to classify urban trees, Ouma et al. (2006) used the Bhattacharyya distance to measure the similarity between multi-scale wavelet sub-bands [49].

The Bhattacharyya bound is a upper bound of the Bayes error and can be defined as:

$$U = \nabla_\theta \log P(X|\theta) \tag{3.1}$$

where θ is a set (vector) of parameters.

For discrete probability distributions p and q over a domain X, the Bhattacharyya distance metric is defined as:

$$D_B(p,q) = -\log(BC(p,q)) \tag{3.2}$$

where $0 \leq D_B \leq \infty$, and:

$$BC(p,q) = \sum_{x \in X} \sqrt{p(x)q(x)} \tag{3.3}$$

is the Bhattacharyya coefficient [7], with $0 \leq BC \leq 0$.

3.6.2.2 Jensen-Shannon

The Jensen-Shannon (JS) divergence is a symmetrised and smoothed version of the Kullback-Leibler (KL) divergence $D(P||Q)$. In terms of the KL divergence, the JS divergence can be defined as:

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M) \tag{3.4}$$

where $M = \frac{1}{2}(P+Q)$.

Without taking the KL divergence into account, the Jensen Shannon divergence can be calculated as follows.

$$JS(p;q) = \frac{1}{2}\int \left(p(x)log\frac{2p(x)}{p(x)+q(x)} + q(x)log\frac{2q(x)}{p(x)+q(x)} \right) d\mu(x) \tag{3.5}$$

where p(x) and q(x) are the two respective distributions [47].

The Bhattacharyya and Jensen-Shannon distance metrics were used because the word embeddings are topic distributions. Other distance metrics including euclidean and cosine distance are not as effective in calculating the distances between distributions and were therefore excluded from the study.

The Kullback-Leibner (KL) divergence was also considered as a similarity measure since it can calculate the similarity between distributions, however the asymmetric nature of the KL divergence made it unfit for this application. The KL measure is asymmetric in the sense that the KL distance from f(x) to g(x) is generally not the same as the KL distance from g(x) to f(x).

3.7 Topic model evaluation

Topic modelling is an unsupervised approach with the main purpose to discover topics without labelled observations. As a result, the evaluation of the model is challenging as no ground truth exists. One evaluation metric for topic models is the predictive likelihood of held-out trajectories, when given a trained model - the aim is to obtain a high likelihood [60]. The marginal distribution of a trajectory is [10]:

$$p(\mathbf{w}|\alpha,\beta) = \int p(\theta|\alpha)\Big(\prod_{j=1}^{N} \sum_{z_j} p(z_j|\theta)p(w_j|z_j,\beta) \Big) d\theta_i. \tag{3.6}$$

The probability of a held-out trajectory can be calculated using Eq. 3.6. By taking the product of the marginal probabilities, the probability of a corpus is obtained as:

$$p(\mathcal{C}|\alpha,\beta) = \prod_{i=1}^{M} \int p(\theta_i|\alpha)\Big(\prod_{j=1}^{N_i} \sum_{z_{ij}} p(z_{ij}\theta_i)p(w_{ij}|z_{ij},\beta) \Big) d\theta_i. \tag{3.7}$$

where $\mathcal{C}$ denotes the corpus.

Equations 3.6 and above relate to Figure 3.2, where the β parameter represents the *event* $\times$ *topic* matrix and the θ parameter represents the *topic* $\times$ *trajectory* matrix. The variable z is the topic assignment for each word. As Figure 3.2, makes clear, this LDA representation is made up of three levels. The α

and β parameters are corpus-level parameters which are assumed to be sampled once in the procedure of generating a corpus. The variables θ_i are document-level variables, sampled once per document. Finally, the variables z_{ij} and w_{ij} are word level variables and are sampled once for each word in each document [10].

3.7.1 Perplexity

Perplexity is perhaps the most popular metric in language modelling [46]. Normalised by the number of words in a document, the perplexity score is calculated based on the probability of a validation set to evaluate the performance of a LDA model, in layman's terms, the perplexity score is an indication of how well a model can predict the next word in a text, based on the context of the previous words. The perplexity can be defined as

$$PP(\mathbf{W}) = P(\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_m)^{(\frac{-1}{m})}$$

where PP refers to perplexity, and $\mathbf{W}$ refers to word. P is the probability estimate assigned to document words. A model is considered optimal if the model, with a given number of topics, minimises the perplexity score [46].

A theoretical foundation of trajectory analysis using LDA techniques is now established. In the following chapters practical examples of applying the techniques to real world datasets are explored.

Chapter 4

Trajectory topic inference with LDA applied to jaguar movement data

4.1 Background

Animal movement data is usually collected with collars and sensors which provide information about animal movement behaviour such as migration, foraging and resting [24]. Meta-data is often available and can include features such as location, gender, age and weight of animals, which further provide insight into individual, as well as relational patterns. In more general terms, trajectory data allows for the study of moving objects, such as vehicles, humans or animals, which spatial location changes over time [52].

In this chapter, the method of analysing GPS data by using LDA is explored. The movement of animals is analysed (in this case jaguars) by applying abstraction on the data in order to extract semantic information.

The aim of this chapter is to showcase how the event $\times$ topic matrix β, in Figure 3.3 can be used to generate interpretable topics from movement data.

4.2 Research approach and research question

4.2.1 Research approach

Abstraction was applied to the data in order to gain semantic information by calculating the speeds of the jaguars. The continuous speed variable was discretised. This was done by using different speed bins to assign letters to each of the data points in the dataset. After this, a Bag-of-Words (BOW) vectorisation was obtained, and LDA was applied on the BOW to obtain the clusters. This is done by discretising continuous data into bins which can then be counted.

4.2.2 Research question

Is it possible to extract semantic information from raw jaguar movement data, and can this semantic information be used to infer interpretable topics from the jaguar movement trajectories using LDA?

4.3 Data

4.3.1 Description of the the dataset

The data, collected by Thompson et al. (2021) [58], consists of GPS telemetry data of 117 jaguars (54 males and 64 females), amounting to 134690 lat-lon locations. The jaguars were monitored in 5 countries in South America. Trajectories represent a set of GPS signals and 117 of these trajectories can be obtained by grouping the GPS locations of each jaguar.

Thomson et al (2021) [58] utilised this data in order to examine the topographic and environmental factors that are associated with jaguar home range size, and movement patterns. Garcia et al. (2021) [24] analysed this same data and grouped the jaguars into three groups based on movement behavior. The three groups are resting, transit and foraging. The contribution of this chapter extends the work of Garcia et al. (2021) [24] into the domain of computational analysis methods, in particular the NLP method of Latent Dirichlet Allocation (LDA).

Figure 4.1 depicts the telemetry data locations of the different jaguars, as it was captured across South America. Figure A.1 in Appendix A, which was generated in Python, shows a more detailed scatter plot of 6 individuals along a riverbed. The dataset is freely available at the following link [1].

[1] Full link given above https://esajournals.onlinelibrary.wiley.com/doi/10.1002/ecy.2379

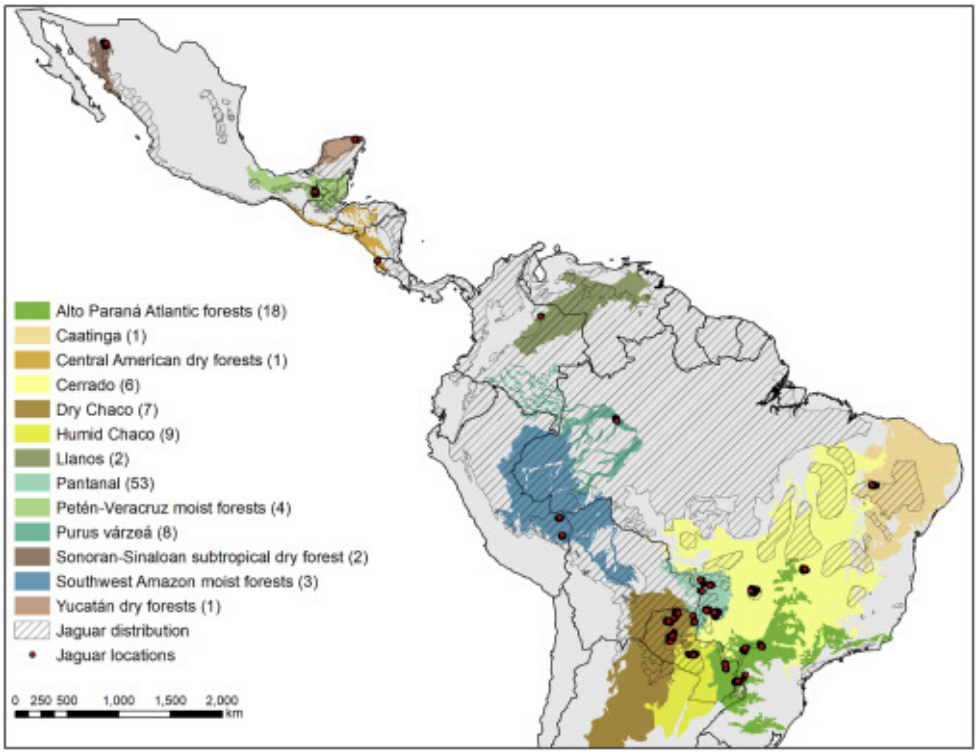


Figure 4.1: Telemetry data locations of the jaguars. This image was obtained from [58].

4.3.2 Pre-processing

As mentioned in Section 1, animal GPS collars do not always exhibit semantic enrichment, which is the case of the jaguar movement data under investigation. Garcia et al. (2021) [24] classified the states of jaguar trajectories as follows:

1. Resting was defined as behaviour with a long time span but a short distance covered.
2. Transit was defined as behaviour with a short time span and a short distance covered.
3. Foraging was defined as behaviour with a long time span and a large distance covered.

Therefore Garcia et al. (2021) clustered the distances manually based on distance and time. Since $speed = \frac{distance}{time}$, in this work the two features of distance and time were combined by using the speed that jaguars travelled.

The speed, which will serve as semantic information, was calculated from distances between geo-locations. To calculate the distances between geo-locations, the Haversine distance formula was used, which calculates the distance between any 2 points on a sphere. The formula is given by

$$2r \times arcsin\left(\sqrt{sin^2(\frac{\psi_1 - \psi_2}{2}) + cos(\psi_1)cos(\psi_2)sin^2(\frac{\lambda_2 - \lambda_1}{2})}\right),$$

where ψ_1 and λ_1 are the 2 latitude coordinates, ψ_2 and λ_2 are the 2 longitude coordinates and r is the radius of the earth, which was added as 6 371 km. The dataset was truncated to include only values larger than 1 km\h since the goal is to analyse movement behaviour. Speeds greater than 80 km\h were considered as outliers and were removed. The speed values calculated sequentially between each lat-lon for each individual jaguar are shown in Figure 4.2.

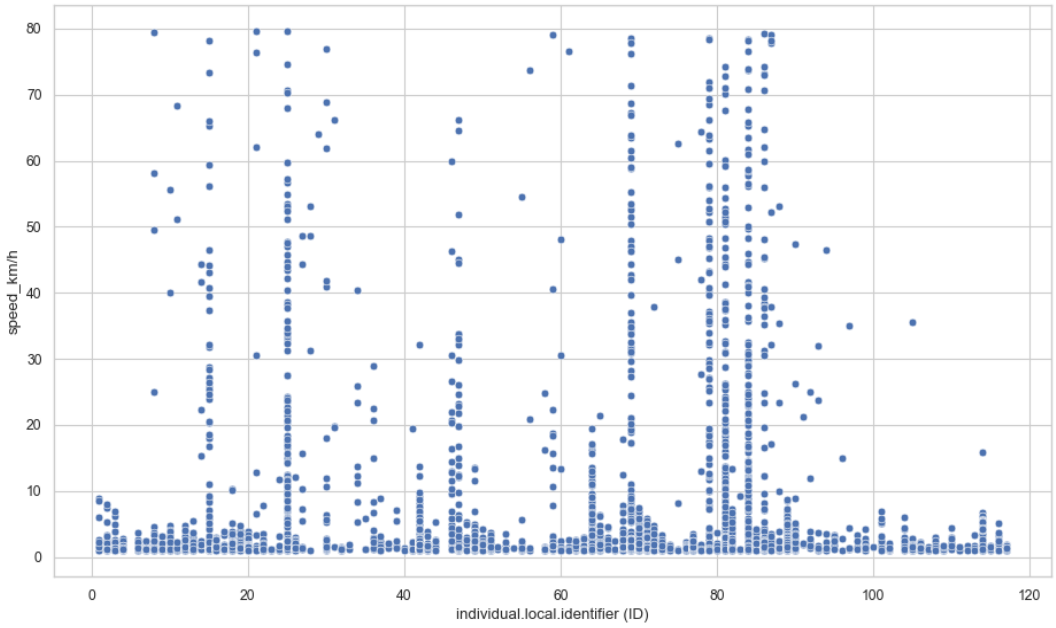


Figure 4.2: All the calculated speeds for each of the 117 jaguars.

From Figure 4.2 it can be seen that most of the values are concentrated below 10 km\h, but there are several trajectories where the individuals moved at much greater speeds. The goal is to see if these trajectories can be clustered together and interpreted using the LDA model.

4.3.3 Feature extraction

LDA is a count model, which requires discrete data. In order to discretise the speeds, as a starting point the data was binned by using k-means clustering, and iteratively adjusted by hand until the class imbalance was not too big. More specifically, it was adjusted so that the biggest classes (classes A and B), do not contain more than 30% of the total lat-lon observations, this was done in order to prevent obtaining results that are skewed in the direction of the bigger classes. This resulted in seven bins which are illustrated in Table 4.1 with the corresponding speed ranges and frequencies. These bins resemble words in NLP, and are the most basic unit in the dataset.

Table 4.1: Descriptions of the speed bins used.

Description	Speed range	N
A	1 – 1.27 km\h	2396
B	1.27 – 1.8 km\h	2439
C	1.8 – 2.79 km\h	1603
D	2.79 – 5.56 km\h	931
E	5.56 – 7.82 km\h	164
F	7.82 – 28 km\h	327
G	28 – 79.7 km\h	302

The discretisation of the continuous speed values leads to the use of the BOW vectorisation which requires the counts of each bin in each trajectory. Due to inconsistent time intervals between GPS signals, all speed bins of each jaguar were combined as a trajectory. In other words, a trajectory contained all the movement data for an individual jaguar. This results in the matrix (β) left of the equation in Figure 3.3. For the jaguar movement dataset, the dimensions of the BOW matrix is 117×7. The 'vocabulary' size is thus seven, since there are only seven speed bins.

The next step in the process is to apply LDA to the BOW matrix. The dimension size of the dataset is seven, which is relatively low when compared to typical text datasets which can reach dimensions of hundreds of thousands. Although a grid search would typically be performed at this stage to select the optimal number of topics, given the small vocabulary size, a total of three topics were chosen. The use of three topics, also ensured that the results could be compared to that of Garcia et al. (2021) [24], which was mentioned earlier. The goal was to compare the clustering results to see if a similar result to Garcia could be obtained from LDA. The `gensim` package in Python was used to calculate the vectorisation and to apply the LDA algorithm.

4.4 Results

4.4.1 Experimental results

This section presents the results of the LDA experiment on the data described in the previous section. LDA produces two matrices as shown in Figure 3.3. The topic × event matrix gives a description of the latent topic space inferred from the data. The trajectory × topic matrix provides a soft clustering matrix, which will be utilised for clustering in Chapter 5. In Chapter 4, when referring to clusters, we refer to the dominant topic of each trajectory based on the soft clustering results, which can be seen as the cluster it was assigned to.

From the topic × event matrix, the composition of each topic can be analysed. A topic is a probability distribution over events, and if the events are sorted according to probabilities, the events with the highest probabilities provide a good description of the topic. This is possible because LDA produces sparse latent vectors as opposed to the dense vectors produced by word embedding algorithms. It is left to the user of the model to further interpret and describe the topics, which is demonstrated in the following paragraphs. The three probability distributions are displayed in Figures 4.3 - 4.5. These figures, as well as Figure 4.6, show the differences between the inferred topics. LDA is a soft clustering technique and its use may result in a variation in topic × trajectory distributions for different initialisations. Furthermore the probability contribution of the dominant topic and the second highest topic may have similar values as can be seen in Table 4.2.

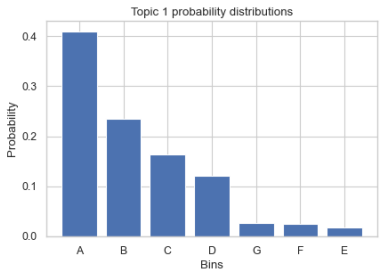


Figure 4.3: Resting.

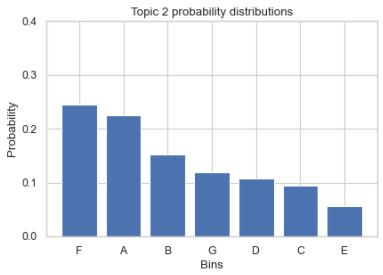


Figure 4.4: Foraging.

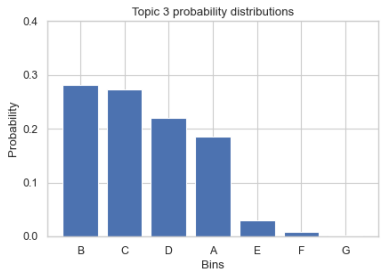


Figure 4.5: Transit.

4.4.1.1 Topic 1 - Resting

Topic 1 can be characterised as the resting topic, as the three events with the greatest probability, shown in Figure 4.3, are the three lowest speed bins, namely A, B and C and higher speeds have a much lower low probability. This is also the topic that can be seen most clearly from the figures. When looking at Figure

4.6, topic 1 is represented by the light cream colored dots and it is clear that most of the trajectories that remained at low speeds throughout their life span, have the resting topic as the dominant topic. The other trajectories in Figure 4.6 also have values at lower speeds, but almost all of these trajectories also contain higher speed values. The assumption is that the trajectories that remained at low speeds, and are not cream colored, were clustered incorrectly by the LDA model. Similarly, topic 1 trajectories (cream colored) that extended to high speeds, are assumed to be clustered incorrectly based on the dominant topics.

4.4.1.2 Topic 2 - Foraging

Topic 2 is characterised as the Foraging topic. Foraging includes activities where the jaguar is searching for food, including hunting, which typically occurs at high speeds. It can be seen from Figure 4.5, that trajectories in this topic has the highest probability for speed bin F, which is between 7.82 km\h and 28\h. The speed bins directly following F are A and B, meaning that the topic has high probabilities for lower speeds as well. Speed bin B is again closely followed by speed bin G, the highest speed bin. This behavior can be seen from the trajectories represented by the black dots on Figure 4.6, where there are high concentrations of black dots at lower speeds, and then suddenly shooting up toward 80 km\h. This is a possible indication of hunting behaviour.

4.4.1.3 Topic 3 - Transit

Figure 4.5 is characterised as the transit topic, since the speed bins seem more erratic. The bins with the highest probability are B and C, similar to the resting bin, but speed bins D and A also have high probabilities. Meaning the jaguars in this topic therefore had a high probability of moving at speeds between 1 km\h and 5.56 km\h. This movement behaviour can serve as an indication that the jaguar is between the states of resting and foraging and therefore, in the transit state. The difference between topic 2 and topic 3 is not clear on Figure 4.6.

At face value, one might be tempted not to attach much value to the inferred topics and interpretation. What makes this an interesting result, is that the topics obtained, and the interpretation thereof, is similar to the results that Garcia et al. (2021) [24] obtained by grouping jaguars into the 3 categories of resting, transit and foraging by association rule mining based on location points and times (date/time). The results were similar with regards to the sizes of each of the three clusters. Therefore, by following a completely unsupervised approach, LDA inferred similar movement behaviour by using semantic information abstracted from the raw data containing only GPS locations and their respective timestamps.

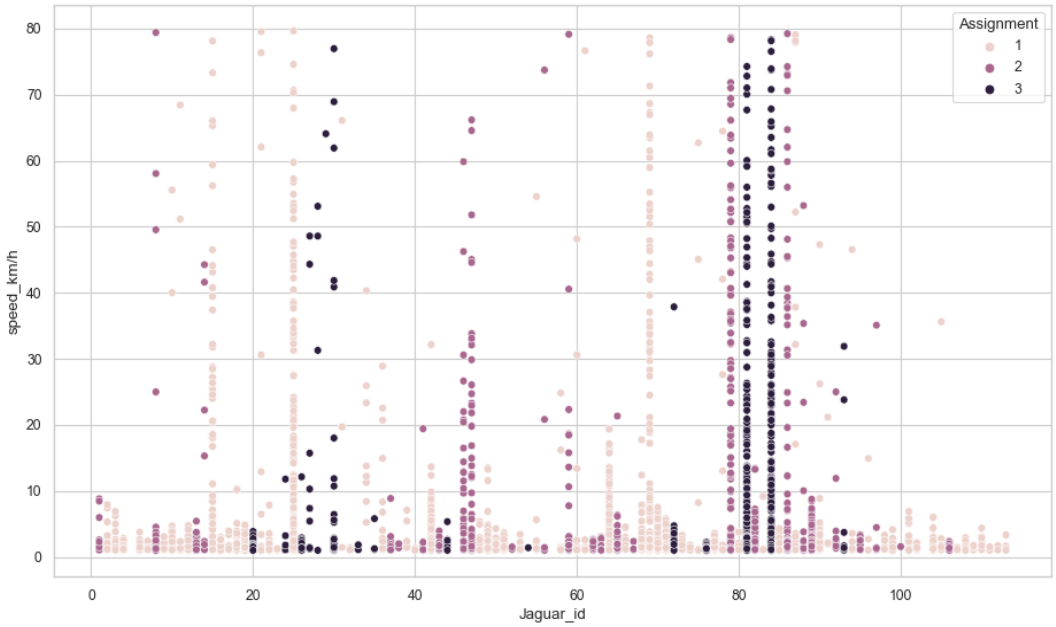


Figure 4.6: Jaguar speeds with 117 jaguar trajectories assigned to their respective dominant topic.

4.4.2 Dominant topic per jaguar

In Subsection 4.4.1, the interpretation power of the topic × event matrix was illustrated. In this subsection, the focal point was the trajectory × topic matrix. One of the practical applications of this matrix is to determine the dominant topic of a given document (or in this case, dominant trajectory). The probabilities of the topics indicate to which extent a topic is represented in each trajectory. Table 4.2 provides an extraction of dominant topics associated with trajectories. Table 4.2 shows the topic contributions for each topics for jaguars 1,2,4, 12 and 15, these jaguars were chosen to show a wide range of results obtained for the trajectories. From Table 4.2 it is clear that there are certain instances where the dominant topic is clearly visible, trajectories 2 and 4 are examples of this, with topic contributions close to 1 for their dominant topics, and topic contributions close to 0 for the other topics, these trajectories were clearly clustered into certain topics. On the other hand, in certain cases like trajectories 1, 12 and 15, the top topics were not as clearly defined, where the topic contributions of some of the topics are very close in value. It is in these cases, where trajectories may be clustered into the wrong group, like some of the trajectories in Figure 4.6.

Table 4.2: Dominant topics for 5 random trajectories (trajectories 1,2,5,26 and 59).

Trajectory	Dominant topic	Topic 1 contribution	Topic 2 contribution	Topic 3 contribution
1	2	0.000665	0.799085	0.200249
2	1	0.998891	0.000555	0.000555
5	1	0.684882	0.314964	0.000154
26	3	0.001107	0.001107	0.997785
59	2	0.000624	0.998752	0.000624

As a summary, Table 4.3 displays the number of trajectories that are clustered into each topic.

Table 4.3: Counts of dominant topics across all trajectories.

Topic	Number of trajectories
Resting	69
Foraging	27
Transit	17

4.4.3 Model evaluation

The performance of the LDA model was tested by running the model 1000 times and calculating the average perplexity score as well as a 95% confidence interval for the perplexity score of the model. The average perplexity score was obtained as -1.54444 and the 95% confidence interval of the perplexity score was obtained as (-1.54575, -1.54315). This low perplexity score indicates a better performance of the model, which is encouraging. The small confidence interval also indicates a very small variance in the results of the model, which is a good sign.

In Figure 4.7, a visualisation is shown for the three topics. Topic 1 is highlighted, showing the estimated term frequency within topic 1 in red, as well as the overall term frequency in blue. From the figure it can be seen that there is a big separation between the topics when plotted on the PC1 and PC2 axes. This is an indication that the model was able to distinguish well between the three topics. The size of the circles represent the relative statistical weight of the topics, *i.e.* the circle size indicates is the prevalence of the particular topic in the corpus of trajectories. Therefore, from Figure 4.7 it can also be seen that the resting topic (topic 1) was the most prevalent topic.

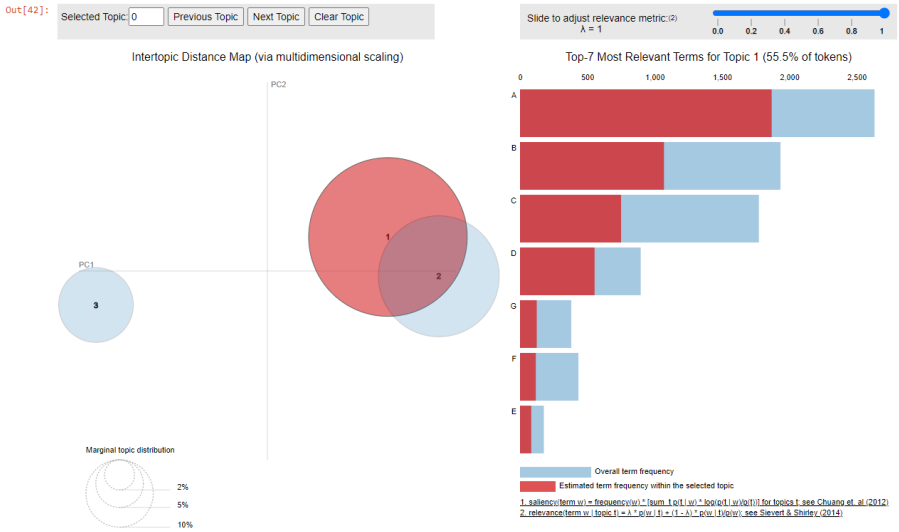


Figure 4.7: PyLDAvis for the three topics. The bubbles on the left are a representation of the size of the topics and their distances to each other are shown as obtained by multi-dimensional scaling. The bars on the right of the figure show the word term space, giving a visual representation of the term frequencies for each bin. The bins are ranked by their importance [43].

4.5 Conclusions

In this chapter, the trajectories of jaguars were analysed and the use of the β matrix, defined in Section 2.4 to infer interpretable topics from the data was demonstrated. Semantic information (in the form of jaguar speed) was extracted from the trajectories, and these speeds were discretised before being used as an input to the LDA model.

The model was able to obtain clusters, similar to those found by associative rule mining by Garcia et al. (2021) [24]. Therefore LDA, which is an unsupervised machine learning technique, was able to generate clusters which yield similar results to those obtained by Garcia et al. (2021) [24].

Although all jaguars showcase movement behaviours associated with each respective topic, the frequency of each behaviour may differ. This provides the opportunity to label individual jaguars in terms of their behaviour profile, and to gain insight on the relation between jaguars. The sparse matrix representation (or embedding) makes the topics interpretable and thus transparent. LDA is a generative model, which facilitates the inference of topic distributions for new observations. The rich analysis and interpretation of movement data, made possible by applying the LDA model, is encouraging.

Please find the full code for this chapter on my github as Jaguar movement topic interpretation.ipynb. Please note that in order to protect the IP of this project, the repository is set to private, please contact

me at armandgraaff@gmail.com, if you would like access to the repo.

Chapter 5

Trajectory clustering with LDA applied to GPS driving data

5.1 Background

In this application, embeddings (topic distributions) inferred from LDA models were used to cluster drivers based on features that characterise driver behaviour. The data involves 30 trips along a predetermined urban route in Pretoria, South Africa. The data was captured by the Centre for Transport Development at the Department of Industrial and Systems Engineering at the University of Pretoria in an effort to calculate the Real Driving Emissions (RDE) of a specific vehicle [29]. The data contains the pollutant concentrations of CO, CO_2 and NO_x, the ambient conditions, and vehicle diagnostics collected from different sensors mounted to the car during the trips.

The goal of this application was to see if embeddings inferred from the topic $\times$ trajectory matrix θ, shown in Figure 3.3 could be used to cluster individual driver behaviour. The Bhattacharyya and Jensen-Shannon distance metrics were used to calculate the distances between the obtained topic distributions for each of the trips. The driver identity for each trip is labeled, thereby enabling hypothesis testing to determine if the trip embeddings between drivers differ significantly or not.

5.2 Research approach

In the dataset, each lat-lon point is an observation and a total of 20 features are captured at each observation. See Table 5.1 for a comprehensive list of the features in the dataset. Similar to the jaguar

dataset described in Chapter 4, the features are continuous, multi-dimensional variables.

As discussed in Chapter 3, in order to make use of LDA the occurrences of the features must be counted. Recall that the input format for LDA is a Bag-of-Words (BOW), which is a count matrix, where each cell represents the frequency of a word in each document. In this implementation, each trajectory/trip is regarded as a document. In the raw dataset, the continuous feature variables must be discretised. We used k-means clustering in order to cluster each feature variable, these clusters are the bins needed for the next step of constructing the words, see Table 5.2, which shows the clusters created for the features speed, rpm, and CO_2.

Once discretised, these features were combined to create "words" at each observation, where each word can be counted, as it would in the typical BOW matrix. This combination of features, henceforth referred to as feature combinations (FC's) creates rich semantic information at each lat-lon observation (see Tables 5.3 and 5.4 to see the FC's used in the analysis). A BOW vectorisation was created for each trip, and the LDA model could be applied directly. Furthermore, different combinations of features were experimented with and the results are displayed.

The inferred topic $\times$ trip distribution is a lower dimensional embedding of the count-vector, described in Section 3.3. Throughout this chapter, the terms "embedding" and "topic distribution" are used interchangeably. Recall from Section 3.2.4, an embedding for a single trip t_i, is the vector θ_i from the θ matrix. In contrast to the application in Chapter 4, where the focus was on interpretability of the topic assignments to the trips, the embeddings in this implementation are used to investigate the ability to cluster driver behaviour using LDA. The distances between the embeddings of each of the trips was used as a measure of how similar the trips are. These distances were measured by calculating the Jensen-Shannon and Bhattacharyya distances between the topic distributions for the different trips.

Hypothesis tests were set up for each of the drivers individually. In each one of these hypotheses, the probability to randomly obtain a distance equal to or smaller than the average distance between the 10 trip embeddings of the driver himself is calculated. The urn model sampling method, described in more detail in Appendix B was applied to these distances in order to compare them for the individual drivers in an effort to determine if the model was able to differentiate between the drivers based on the features in the dataset.

The research question is, is it possible to distinguish between driver behaviour of individual drivers using lower dimensional embeddings contained in the topic $\times$ trajectory matrix, which was derived from vehicle driving data?

See Section 5.3 for a full description of the data.

5.3 Data

Three drivers each did 10 trips with a Ford Figo (a small hatchback vehicle). The trips were done in an effort to capture Real Driving Emissions (RDE) to certify that the emissions of a vehicle are within the acceptable standards while driving under real world conditions [29]. Each of the drivers followed the exact same 62 km route to limit variance of external factors that can affect driver behaviour (change in altitude, highway driving vs street driving, etc) as much as possible. Each of the driving features mentioned in Table 5.1 was captured at every second of the trip. So for example, trip 1 has 6279 observations, relating to 6279 seconds of the trip, or 1 hour, 44 minutes, and 39 seconds to complete the trip. Depending on traffic conditions, the other trips were completed within a similar time span.

Figure 5.1 shows the trajectory of the route taken by the drivers.

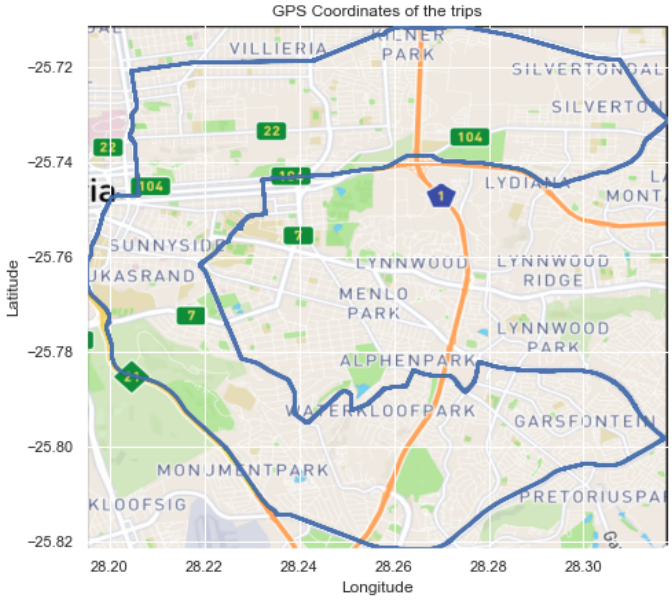


Figure 5.1: Trajectory outline of the vehicle trip used to gather the data.

Table 5.1 provides detail on all of the vehicle data metrics that were collected as part of the research done by Joubert et al. (2021) at the University of Pretoria [29].

Table 5.1: Field descriptions.

Metric	Description	Unit of measure
date	Date and time in GMT + 2 (South African Standard Time).	Date time
trip	Trip identifier, sequentially starting at one (30 Trips total). One trip is a single field test completing a single route.	Integer value.
driver	Driver number, sequentially starting at one (3 Drivers total).	
load	Additional load (balast) added to the vehicle.	kg
gps_lat	Latitude in WGS84.	decimal degrees
gps_lon	Longitude in WGS84.	decimal degrees
gps_alt	Altitude above sea level.	metre
gps_speed	Vehicle speed derived from the GPS unit.	km/h
humidity	Ambient humidity.	%RH (relative humidity)
pressure	Ambient air pressure.	mbar
temp	Ambient temperature.	°C
speed_vehicle	Vehicle speed as recorded from OBDII port.	km/h
throttle	Absolute throttle position.	%
rpm	Vehicle rpm as recorded from OBDII port.	rpm
air_fuel_ratio	Air/fuel ratio of the gas sample.*	
CO_2_mass	Instantaneous mass CO_2.*	g/s
CO_mass	Instantaneous mass CO_2.*	g/s
NO_x_mass	Instantaneous mass NO_x.*	g/s
dist	Distance between current data point and previous data point (one second earlier). *	m
dist_from_start	Total distance covered by the current trip.*	m
accel	Acceleration	
CO_2	The CO_2 emitted per metre (m).	CO_2_mass/m
CO	The CO emitted per metre (m).	CO_mass/m
NO_x	The NO_x emitted per metre (m).	NO_x_mass/m

* Calculated fields. Table adapted from Joubert et al (2022) [29].

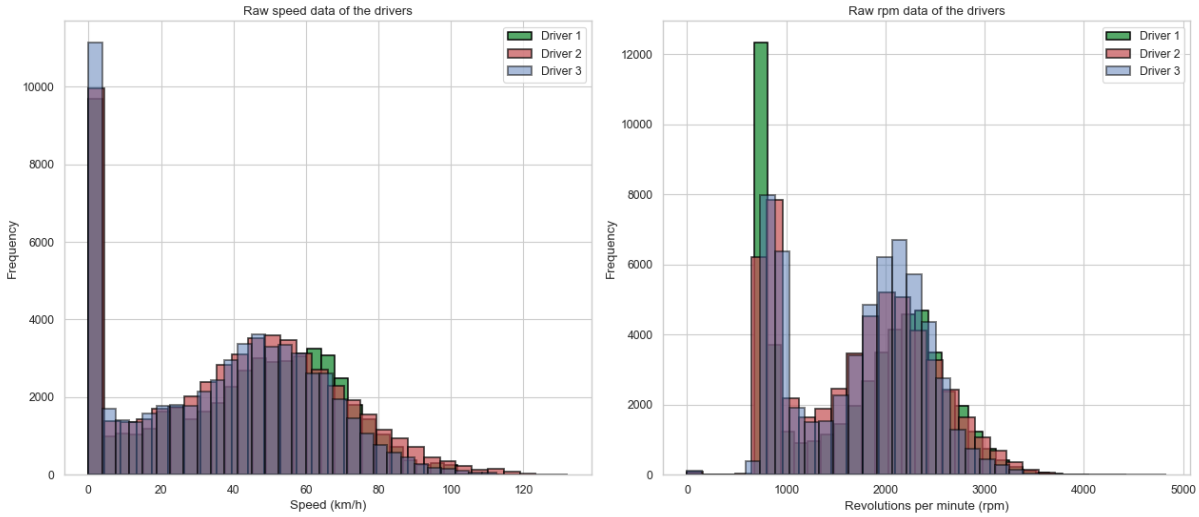


Figure 5.2: Visualisation of raw data for the speed and rpm of the different drivers.

From Figure 5.2, it is clear that in the high-dimensional trajectory data (raw data), the speed and rpm values of the drivers are not very different. Some differences can be seen, for example, for Driver 1 (green) the rev ranges close to idle (1000 rpm) were observed more often despite spending less time at rest, suggesting that the driver coasted more often. Some other small differences can also be seen from this plot, yet this data would be impossible to cluster when applying unsupervised clustering techniques to the data directly. This is why the use of lower dimensional embeddings is a crucial part of this research, since the LDA model is able to identify differences between the driver behavior based in the lower dimensional latent space, and cluster the trajectories in this way.

5.3.1 Pre-processing and feature extraction

Now that an overview of the raw data is given, the pre-processing that needed to be done in order to apply the LDA model can be explained. K-means clustering was used to discretise the data in order to create the feature combinations (FC's) that contain specific semantic information. For a detailed explanation on this procedure, see Appendix C. In other words, different segments/bins were created of the value range for each feature in the spatio-temporal time series data, in which each of the segments were represented by an alphabetical letter. Each word that was created is therefore a concatenation of the feature segment values, observed at that particular lat-lon point. Each feature combination (FC) can be thought of as the vocabulary of the unique occurrences of these words. See the Table 5.2 for examples of the discretisations obtained for some of the features.

A cluster of size 6 was chosen for the k-means discretisation, this number proved to be sufficient and provided enough clusters in order to create meaningful words. A different number of clusters could have

Table 5.2: Bin description of speed, rpm, and CO_2.

Discretisation of speed, rpm, and CO_2					
Speed		rpm		CO_2	
Description	Range (km \h)	Description	Range	Description	Range (emission mass per metre)
A	0 – 11.4	A	0 – 1110	A	-1.36 – -0.253
B	11.4 – 29.4	B	1110 – 1590	B	-0.253 – 0.311
C	29.4 – 44.7	C	1590 – 1980	C	0.311 – 0.849
D	44.7 – 58.8	D	1980 – 2320	D	0.849 – 1.47
E	58.8 – 76	E	2320 – 2730	E	1.47 – 2.82
F	76 – 132	F	2730 – 4830	F	2.82 – 7.37

been chosen, but in the interest simplicity, 6 clusters were used. Using a higher number of clusters should give even more information on the clusters present in the data, however, in the interest of parsimony, a lower number of clusters were used. An empirical study on the number of cluster can be considered for future work.

Table 5.3 shows which features were included in each FC.

Table 5.3: Feature combinations that were included in the sampling procedure.

Feature combination list		
Feature combination (FC)	Features included in the FC	vocabulary size
FC 01	speed	6
FC 02	rpm	6
FC 03	acceleration	6
FC 04	speed, rpm	36
FC 05	speed, throttle	36
FC 06	speed, acceleration	36
FC 07	speed, rpm, acceleration, throttle	931
FC 08	CO_2, CO, NO_x	115
FC 09	acceleration, CO_2	35
FC 10	speed, CO_2, CO, NO_x	432
FC 11	acceleration, CO_2, CO, NO_x	415
FC 12	speed, rpm, acceleration, throttle, CO_2, CO, NO_x	6711

Table 5.4: The concatenated words for observation 5000 for each FC.

Feature combination examples		
Feature combination (FC)	Example word	vocabulary size
FC 01	speed-D	6
FC 02	rpm-D	6
FC 03	accel-B	6
FC 04	speed-D_rpm-D	36
FC 05	speed-D_throttle-B	36
FC 06	speed-D_accel-B	36
FC 07	speed-D_rpm-D_accel-B_throttle-B	931
FC 08	co2-B_co-A_nox-A	115
FC 09	accel-B_co2-B	35
FC 10	speed-D_co2-B_co-A_nox-A	432
FC 11	accel-B_co2-B_co-A_nox-A	415
FC 12	speed-D_accel-B_rpm-D_throttle-B_co-A_co2-B_nox-A	6711

Table 5.4 shows the words created at the 5000th observation of the second trip by driver 1. The vehicle speed was 57.2 km\h, thus it makes sense that a lot of the features fall into clusters C and D.

In total, the experiment stored 167 561 observations, *i.e.* the experiment stored 167 561 seconds of driving time, equalling about 46.54 hours of driving data. Each of the trips varied slightly in time, for example, $\text{driver1}_{trip1} = 6279$ seconds ≈ 1.74 hours, and $\text{driver2}_{trip5} = 5853$ seconds ≈ 1.6275 hours.

5.4 Experimental design

In this experiment, the goal was to look at what combination of features gives the "word" embeddings with the biggest separation between the drivers. The experimental design follows the following steps.

5.4.1 Building the LDA model and constructing the distance matrix

Step 1: Pre-processing.

Step 1 is the pre-processing step, outlined in Subsection 5.3.1. In this step each of the features in the raw

dataset are clustered into 6 clusters using k-means. These discretised features are then used to build the Bag-of-Words vectorisations for each of the FC's. See Appendix C for a full explanation on how these words were constructed using R-studio, as well as a link to the code on Github.

Step 2: Train the LDA model for each FC.

In this step the Bag-of-Words (BOW) vectorisations that were created in the previous step are used to calculate the LDA model. This was done by following these steps: [1]

1. Order the words by trip.
2. Use this data to create an event dictionary of the words.
3. Use the event dictionary to calculate the trajectory vectorisations.
4. Use the trajectory vectorisations and the event dictionaries to build the LDA model.

The LDA model was created by using the `gensim` package in Python.

Step 3: Calculate the trajectory embeddings

In this step the trained LDA model generated in the previous step was used to infer the trajectory embeddings for each of the 30 trips. We therefore obtain a $30 \times n$ matrix, where n is the number of topics used, containing the topic by trip distributions of each trip. The size of the matrix therefore depends on the number of topics used. If 5 topics are used, the matrix would contain 30 rows and 5 columns, if 100 topics were used, the matrix would contain 30 rows and 100 columns.

Step 4: Construct the distance matrix and define populations.

After topic distribution for each trip was obtained, the distance between the topic distributions (embeddings) for each of the trips were calculated. The result of this calculation is a symmetric 30×30 matrix, containing the distances between each of the topic distributions for the different trips. The trips for each driver were ordered, therefore trips 1-10 were the trips for Driver 1, trips 11-20 were the trips for Driver 2, and trips 21-30 were the trips for Driver 3. In the distance matrix below, $d_{1,2}$ is the distance (difference) between the embeddings for trip 1 and trip 2, these distances, explained in Section 3.6.2, were used as a measure of similarity between the trips. The 30×30 distance matrix, henceforth referred to as the **Population** is defined as follows:

[1] For the sake of simplicity, the process is explained in singular form, but keep in mind each of these steps are repeated for all the FCs.

$$\text{Population} = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & ..... & d_{1,30} \\ d_{2,1} & d_{2,2} & d_{2,3} & ..... & d_{2,30} \\ . & . & . & . & \\ . & . & . & . & \\ d_{11,1} & d_{11,2} & d_{11,3} & ..... & d_{11,30} \\ . & . & . & . & \\ . & . & . & . & \\ d_{21,1} & d_{21,2} & d_{21,3} & ..... & d_{21,30} \\ . & . & . & . & \\ . & . & . & . & \\ d_{30,1} & d_{30,2} & d_{30,3} & ..... & d_{30,30} \end{bmatrix}$$

where the first 10 rows, trips 1-10, henceforth referred to as **Population 1** are the distances from the trips of Driver 1 to all of the other trips, including Driver 1 himself. Population 1 is defined as follows:

$$\text{Population 1} = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & ..... & d_{1,30} \\ d_{2,1} & d_{2,2} & d_{2,3} & ..... & d_{2,30} \\ . & . & . & . & \\ . & . & . & . & \\ d_{10,1} & d_{10,2} & d_{10,3} & ..... & d_{10,30} \end{bmatrix}$$

Population 2 and **Population 3**, are defined in a similar manner, with rows 11-20, and 21-30 of the Population distance matrix being the trips for Driver 2 and Driver 3, respectively.

We therefore have that:

$$Population = Population1//Population2//Population3$$

where // indicates row concatenation of the populations.

This matrix is defined as the Population since it contains all of the distance metrics that are of interest and it will be the basis of the urn model sampling, which is done in the next step. This Population matrix will be a symmetric matrix since the distance between trip 1 and trip 2, will be the same as the distance between trip 2 and trip 1, *i.e.* $d_{1,2} = d_{2,1}$.

Furthermore, definitions are established for smaller subsections of the Population matrix, containing the distances between only the 10 trips for Driver 1, 2 and 3 respectively, these subsections are referred to as **Population 1:1**, **Population 2:2**, and **Population 3:3**. In other words, Population 1:1 contains the distances between the trips for Driver 1 and himself, again, Population 2:2 and Population 3:3 are defined in the same way, for Driver 2 and Driver 3 respectively. Thus, if Population 1, is the first 10 rows in the

population matrix defined above, Population 1:1, would be the first 10 rows, and the first 10 columns of the population matrix, *i.e.* the first 10 columns of Population 1.

See Figure 5.3, which shows the structure of the Population and sub populations, contained in the distance matrix.

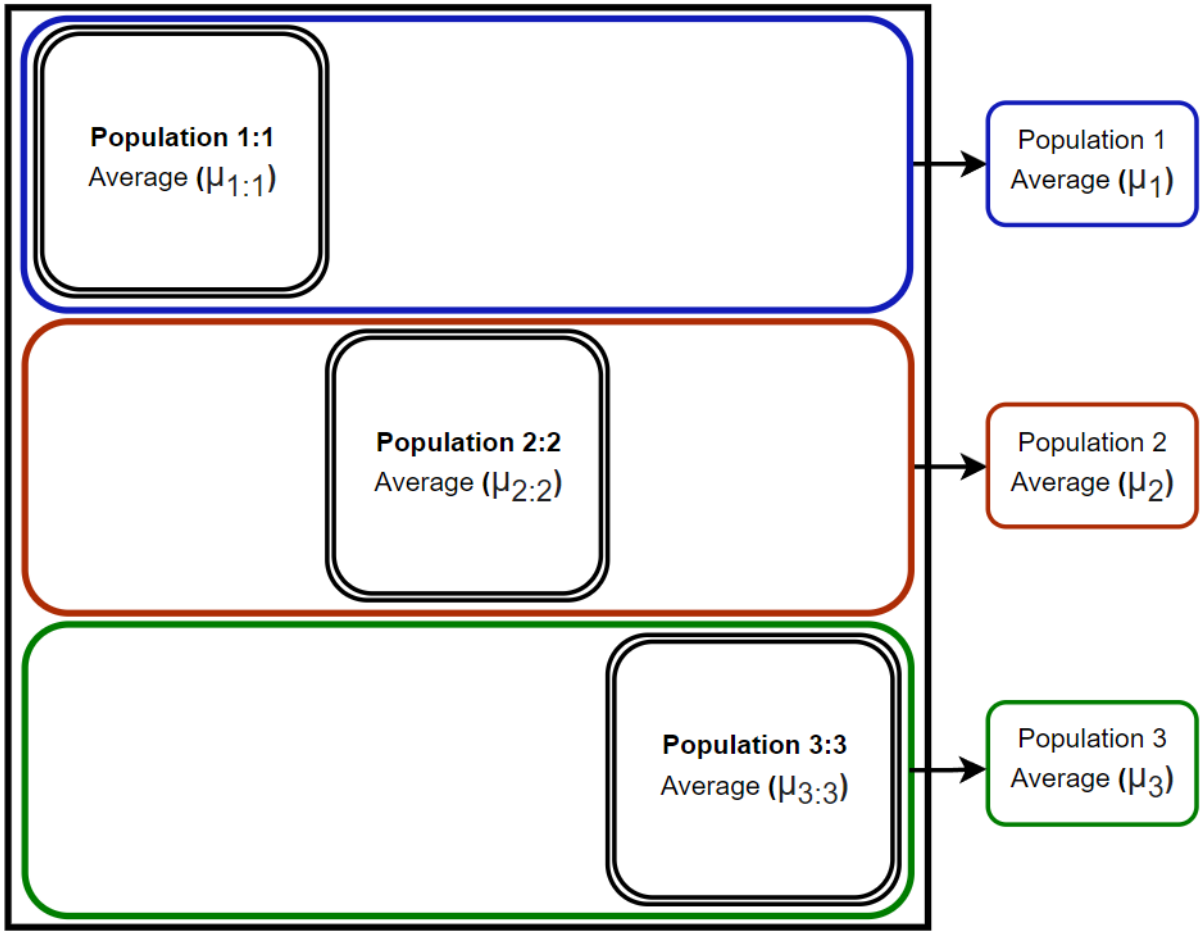


Figure 5.3: Visualisation of the structure of populations in the data.

Now that these distances are known, the aim is to establish if the difference between the trips of one driver and his own trips is lower than the difference between the trips of same driver and the trips of the other drivers.

The obvious way to test this is by means of a hypothesis test. Since the assumption of a normal distribution can't be made, the method of sampling was used for hypothesis testing, which is described in the next few steps.

5.4.2 Sampling

Step 5: Sampling and hypotheses testing.

With regards to Figure 5.3, the blue block, which is Population 1 (the population for Driver 1), is the portion that was sampled for the hypothesis test of Driver 1, using the urn model, the urn model re-sampling method is explained in detail in Appendix B. In short, values were sampled from Population 1, after which it was tested if the average of the 10 000 sampled values from Population 1 is smaller than the average of the values in Population 1:1. After 10 000 iterations, the resulting p-value is the proportion of times that this was the case. Naturally, a small p-value is expected if the average distances in Population 1:1 is smaller than the average distances contained in Population 1.

The same procedure was repeated for Driver 2 and Driver 3, using the red and green blocks respectively. Since the distance matrix in Figure 5.3 is symmetric, the blue, red and green blocks could have been defined on the Population matrix vertically as well, and the values in the populations would have been the same. In the implementation, this distance matrix can be seen as a heat map in Figure 5.5, the symmetry of the distance matrix can clearly be seen on the heat map as well.

The hypothesis tests for each of the drivers in this work are defined as follows:

$H_0 : \mu_{i:i} \geq \mu_i$

$H_A : \mu_{i:i} < \mu_i$

For i = 1,2,3 for Driver 1, Driver 2 and Driver 3 respectively and where μ_i is the sample average of Population i and $\mu_{i:i}$ is the average of Population i:i. Therefore, the null hypothesis for Driver 1 is that the average of Population 1:1 is greater than or equal to the sample average of Population 1 and this hypothesis will be rejected if the average of population 1:1 ($\mu_{1:1}$) is significantly smaller than the average of the sampled values (μ_1). The same logic follows for Driver 2 and Driver 3.

Therefore, the hypothesis tests if the embeddings (or topic distributions) of the trips of one driver and the embeddings of the trips of another driver will be significantly different compared to the difference of the trips of a driver with himself. The results will be tested at a significance level of 1%. Therefore, the null hypothesis that the means are the same will be rejected if the p-value is less than 0.01.

The sampling procedure was repeated 100 times, and the averaged results are given. Furthermore, these hypothesis tests were conducted for all of the FC's, for all the drivers.

5.5 Results

For each FC at each number of topics, the final averaged results of 100 models are added in tables for the implementations which used the Bhattacharyya and Jensen-Shannon distance metrics, respectively.

As explained before, a lower p-value means that the topic distribution for the driver was separated more from the other drivers, so a lower p-value means that the model picked up a greater separation/difference/distance between the driving behaviour present in the trips.

In Tables 5.5 - 5.14, p1, p2 and p3, are the p-values for the hypothesis tests of Driver 1, Driver 2 and Driver 3, respectively. With the 1% level of significance, the results were regarded as significant if the p-value was less than 0.01, the p-values that were regarded as significant are marked in **bold** in Tables 5.5 - 5.14.

5.5.1 Bhattacharyya

The results for the **Bhattacharyya** implementation is given in Tables 5.5 - 5.9:

Table 5.5: Results for Bhatacharyya implementation with 5 topics.

Bhattacharyya (5 topics)			
Feature Combination	p1	p2	p3
FC 01	0.139004	0.145068	0.134583
FC 02	0.026185	0.039556	0.021644
FC 03	0.200006	0.266165	0.283411
FC 04	**0.000059**	**0.001034**	**0.000476**
FC 05	**0.000026**	**0.001002**	**0.000281**
FC 06	0.034951	0.051496	0.033035
FC 07	**0.000008**	**0.000206**	**0.000114**
FC 08	0.090204	0.158781	0.044302
FC 09	0.158800	0.165194	0.050315
FC 10	0.020988	0.058462	0.014944
FC 11	0.062472	0.137225	0.023006
FC 12	**0.003765**	0.049775	0.011280
P-value Averages	0.06137	0.08950	0.05145

Table 5.6: Results for Bhatacharyya implementation with 10 topics.

Bhattacharyya (10 topics)			
Feature Combination	p1	p2	p3
FC 01	0.088371	0.062519	0.088537
FC 02	0.010922	0.015026	0.005727
FC 03	0.131532	0.176245	0.260220
FC 04	**0.000033**	**0.000134**	**0.000074**
FC 05	**0.000020**	**0.000204**	**0.000105**
FC 06	0.011911	0.024987	0.010960
FC 07	**0.000010**	**0.000040**	**0.000031**
FC 08	0.046443	0.128471	0.019199
FC 09	0.097014	0.131685	0.025566
FC 10	0.013382	0.023481	**0.003985**
FC 11	0.045335	0.117677	0.015969
FC 12	0.015900	0.051138	0.024916
P-value Averages	0.03841	0.06097	0.03794

Table 5.7: Results for Bhatacharyya implementation with 20 topics.

Bhattacharyya (20 topics)			
Feature Combination	p1	p2	p3
FC 01	0.049068	0.033309	0.045046
FC 02	**0.007218**	**0.006372**	**0.003722**
FC 03	0.093748	0.147962	0.201044
FC 04	**0.000018**	**0.000061**	**0.000035**
FC 05	**0.000011**	**0.000106**	**0.000080**
FC 06	**0.002709**	0.013403	**0.004083**
FC 07	**0.000005**	**0.000028**	**0.000042**
FC 08	0.048310	0.118708	0.021487
FC 09	0.055886	0.117338	**0.008880**
FC 10	**0.007262**	0.027910	**0.002680**
FC 11	0.037053	0.105076	0.016918
FC 12	**0.009191**	0.038883	0.013821
P-value Averages	0.02587	0.05076	0.02649

Table 5.8: Results for Bhatacharyya implementation with 50 topics.

Bhattacharyya (50 topics)			
Feature Combination	p1	p2	p3
FC 01	0.025190	0.014400	0.028270
FC 02	**0.006169**	**0.004665**	**0.002390**
FC 03	0.055966	0.130974	0.124931
FC 04	**0.000017**	**0.000033**	**0.000018**
FC 05	**0.000008**	**0.000052**	**0.000053**
FC 06	**0.001380**	**0.007144**	**0.002489**
FC 07	**0.000006**	**0.000015**	**0.000014**
FC 08	0.027836	0.095927	**0.004204**
FC 09	0.038058	0.099071	**0.006928**
FC 10	**0.008116**	0.023029	**0.001782**
FC 11	0.029204	0.113683	**0.005912**
FC 12	**0.007147**	0.028228	0.013060
P-value Averages	0.01659	0.04310	0.01584

Table 5.9: Results for Bhatacharyya implementation with 100 topics.

Bhattacharyya (100 topics)			
Feature Combination	p1	p2	p3
FC 01	0.025442	0.010125	0.017742
FC 02	**0.005334**	0.003575	**0.002099**
FC 03	0.042088	0.101235	0.111890
FC 04	**0.000025**	**0.000025**	**0.000021**
FC 05	**0.000007**	**0.000065**	**0.000066**
FC 06	**0.000916**	**0.005991**	**0.002046**
FC 07	**0.000004**	**0.000048**	**0.000058**
FC 08	0.026212	0.096515	**0.004864**
FC 09	0.031852	0.089839	**0.004543**
FC 10	**0.007587**	0.020673	**0.002412**
FC 11	0.019836	0.100617	**0.004579**
FC 12	0.012084	0.029140	**0.008505**
P-value Averages	0.01428	0.03815	0.01324

5.5.2 Jensen-Shannon

The results for the **Jensen-Shannon** implementation is given in Tables 5.10 - 5.14:

Table 5.10: Results for Jensen-Shannon implementation with 5 topics.

Jensen-Shannon (5 topics)			
Feature Combination	p1	p2	p3
FC 01	0.123822	0.104895	0.124121
FC 02	0.027313	0.028808	0.016190
FC 03	0.174145	0.201469	0.242810
FC 04	**0.000067**	**0.000891**	**0.000252**
FC 05	**0.000043**	**0.000760**	**0.000263**
FC 06	0.029483	0.051427	0.041066
FC 07	**0.000031**	**0.000247**	**0.000110**
FC 08	0.072759	0.135959	0.038358
FC 09	0.102357	0.142991	0.036887
FC 10	0.013743	0.033540	0.010471
FC 11	0.083446	0.146366	0.038786
FC 12	0.010509	0.072658	0.016365
P-value Averages	0.05314	0.07667	0.04713

Table 5.11: Results for Jensen-Shannon implementation with 10 topics.

Jensen-Shannon (10 topics)			
Feature Combination	p1	p2	p3
FC 01	0.062240	0.041092	0.050106
FC 02	**0.008486**	0.010507	0.004215
FC 03	0.099279	0.161899	0.195033
FC 04	**0.000029**	**0.000149**	**0.000100**
FC 05	**0.000013**	**0.000191**	**0.000116**
FC 06	0.004599	0.018349	0.008778
FC 07	**0.000012**	**0.000025**	**0.000034**
FC 08	0.052536	0.113937	0.025362
FC 09	0.061624	0.111563	0.015709
FC 10	**0.005586**	0.017316	**0.002543**
FC 11	0.029827	0.087417	0.010229
FC 12	**0.004108**	0.054275	0.017213
P-value Averages	0.02736	0.05139	0.02745

Table 5.12: Results for Jensen-Shannon implementation with 20 topics.

Jensen-Shannon (20 topics)			
Feature Combination	p1	p2	p3
FC 01	0.033317	0.021049	0.027088
FC 02	**0.006356**	**0.005770**	**0.002428**
FC 03	0.060136	0.105669	0.144043
FC 04	**0.000021**	**0.000069**	**0.000057**
FC 05	**0.000011**	**0.000102**	**0.000066**
FC 06	**0.002096**	**0.009933**	**0.002750**
FC 07	**0.000008**	**0.000029**	**0.000018**
FC 08	0.040263	0.089403	0.017142
FC 09	0.029737	0.081662	**0.007187**
FC 10	**0.003104**	0.015647	**0.002687**
FC 11	0.021191	0.084134	**0.009651**
FC 12	**0.006012**	0.044829	0.018279
P-value Averages	0.01685	0.03819	0.01928

Table 5.13: Results for Jensen-Shannon implementation with 50 topics.

Jensen-Shannon (50 topics)			
Feature Combination	p1	p2	p3
FC 01	0.022054	0.011141	0.019596
FC 02	**0.005959**	**0.003800**	**0.002067**
FC 03	0.029769	0.084062	0.090304
FC 04	**0.000016**	**0.000047**	**0.000028**
FC 05	**0.000011**	**0.000083**	**0.000056**
FC 06	**0.001059**	**0.006686**	**0.002283**
FC 07	**0.000009**	**0.000016**	**0.000028**
FC 08	0.023588	0.070768	**0.006356**
FC 09	0.023596	0.071067	**0.005697**
FC 10	**0.003757**	0.013698	0.001810
FC 11	0.019365	0.063843	**0.007062**
FC 12	**0.006006**	0.025563	0.016385
P-value Averages	0.01127	0.02923	0.01263

Table 5.14: Results for Jensen-Shannon implementation with 100 topics.

Jensen-Shannon (100 topics)			
Feature Combination	p1	p2	p3
FC 01	0.018708	**0.009271**	0.01409
FC 02	**0.005097**	**0.003215**	**0.001974**
FC 03	0.020159	0.059759	0.079711
FC 04	**0.000037**	**0.000041**	**0.000028**
FC 05	**0.000007**	**0.000074**	**0.000068**
FC 06	**0.000916**	**0.004678**	**0.001902**
FC 07	**0.000004**	**0.000028**	**0.000043**
FC 08	0.023716	0.064049	**0.005536**
FC 09	0.018200	0.065571	**0.005119**
FC 10	**0.003013**	0.013513	**0.002070**
FC 11	0.017312	0.056644	**0.006125**
FC 12	0.019383	0.024351	0.018490
P-value Averages	0.01055	0.02510	0.01126

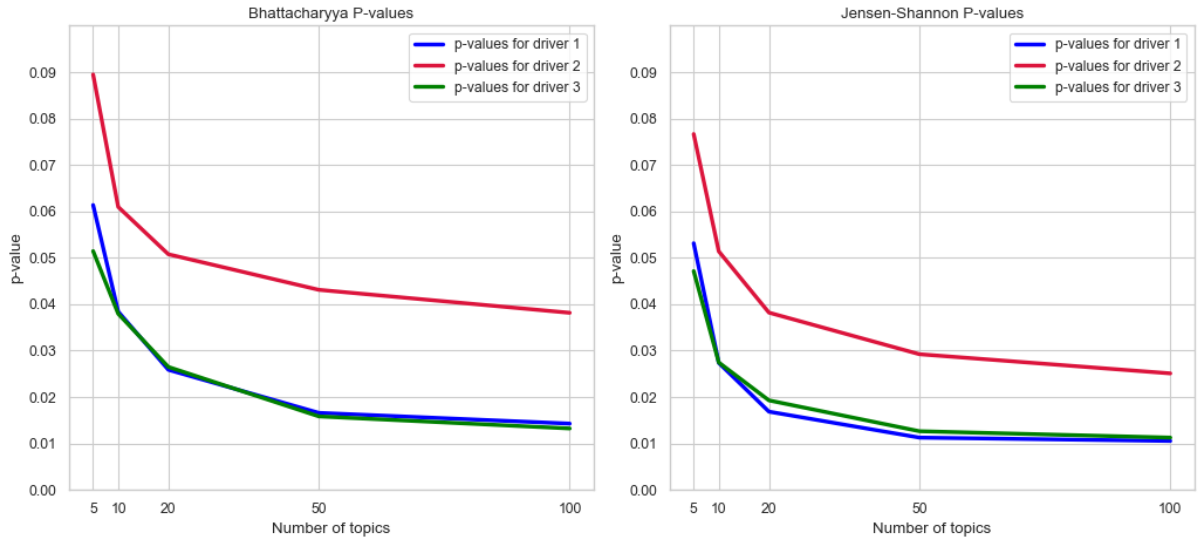


Figure 5.4: Side by side comparison of Bhattacharyya and Jensen-Shannon results, showing how the average p-values decrease as the number of topics increase.

In Figure 5.4, the average p-values for all of the 12 FC models at each of the number of topics is displayed, for each driver. It is clear that, as the number of topics increase, the p-values decrease, this means that when a higher number of topics are used the clustering results improve. Upon close inspection, it is clear that, for both the Bhattacharyya and Jensen-Shannon implementations, these graphs make an elbow curve. This is because increasing from 5 topics to 10 topics, and from 10 topics to 20 topics, there is a sharp decrease in the p-values, indicating substantially better clustering performance of the LDA model. After that, increasing the number of topics only provides marginally lower p-values. Therefore, when aiming to create the most parsimonious model, a model with 50 topics should be chosen, since the added complexity of using more topics, does not justify the slightly better clustering results. For example, for FC 07, to create a model with 5 topics takes about 1.97 seconds, a model with 50 topics takes about 14.33 seconds, and to create a model with 100 topics, takes about 27.46 seconds. So it takes noticeably longer to create the models with more topics because of the added complexity.

5.5.3 Visualisation of the best obtained model

The heat map shown in Figure 5.5 represents the distance matrix that was defined in Section 5.4 *i.e.* it is the population of distances that were sampled from. Population 1, Population 2 and Population 3, explained in Section 5.4 are surrounded in red bounding boxes, and Population 1:1, Population 2:2 and Population 3:3 are surrounded in green bounding boxes. These red and green colours represent the values that were sampled, and correspond to the colours of the density plots of Figures 5.6 - 5.8.

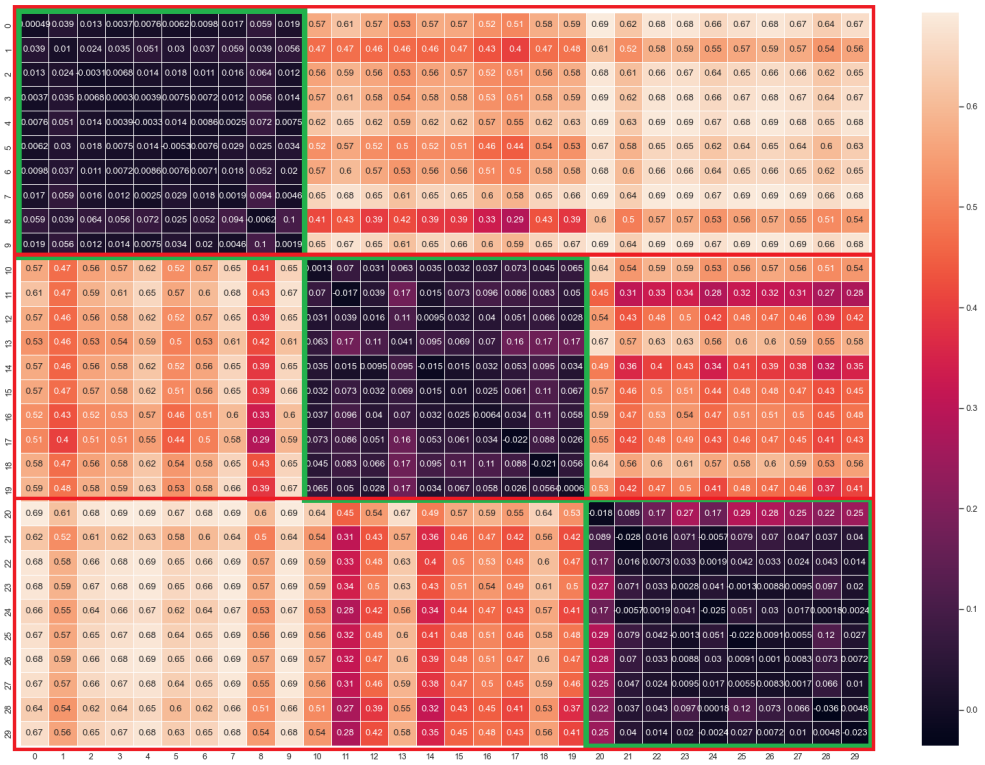


Figure 5.5: Heat map of the calculated distances for the best performing model. The row indexing starts at 0 and ends at 29 because python indexing starts at 0 (0 is the first trip).

The distances in populations 1:1, 2:2 and 3:3 , are notably darker compared to the rest of the distances, and darker blocks contain lower values. This is as expected, since if the distances between the embeddings for the 10 trips of each driver is smaller compared to the distances between embedding of the trips of different drivers, it follows logically that there are higher similarities between the embeddings of the trips of each driver's own sub population (Population 1:1, Population 2:2 and Population 3:3), and hence, there is a higher similarity in driver behavior.

With Figure 5.5 at hand, the calculation of the p-values can be explained again. For Driver 1, by sampling, 10 values were drawn randomly from the upper red bounding box (Population 1), and the average was calculated. This was repeated 10 000 times, and all of the values are added in a list. The proportion of these values that were smaller than the average of the upper left green bounding box (Population 1:1), was calculated. This calculated value is the probability to randomly get a value from Population 1 that will be smaller than the average of Population 1:1. This calculated value is the p-value for Driver 1 (p1), as seen in Tables 5.5 to 5.14. The process was repeated for Driver 2 and Driver 3.

Each FC, at each number of topics, for each distance metric, hence each row in Tables 5.5 to 5.14, had its

own unique distance matrix like the one in Figure 5.5, from which there was sampled in order to calculate the p-values.

The heat map in Figure 5.5 is the heat map of one of the best models, more specifically, it is the model for FC 07, using the Jensen-Shannon distance metric, at 100 topics, shown in row 7 of Table 5.14, FC 07 contained the features speed, rpm, acceleration and throttle, and these combined features provided the best clustering results across all the different models tested. It should be noted that the combination of only speed and rpm, or only speed and throttle, which was FC 04 and FC 05 respectively, provided very good results as well.

The results in Figures 5.6 to 5.8 indicate the respective densities of the distances between the topic distributions for each of the drivers, corresponding to the model shown in the heat map in Figure 5.5. In Figure 5.6, the green density plot, is the density of 10 000 sampled averages from the green bounding box for Driver 1 (Populations 1:1), shown in the heat map in Figure 5.5. Recall in the sampling procedure, only values from Population 1 are sampled, and compared to the average of Population 1:1, thus the green density plot containing 10 000 sampled averages from Population 1:1 was only created to aid in the visualisation of the density plots. The red density plots are the densities of 10 000 sampled averages from the corresponding red bounding boxes for driver 1 (Population 1) and it is the values from this red density plot that were used to calculate the p-values. A similar approach was followed to create the density plots for Driver 2 and Driver 3, shown in Figure 5.7 and 5.8, respectively.

5.5.4 Analysis of the density plots.

In Figure 5.6, a clear difference can be noticed between the distribution of the distances for Population 1:1 (green), and Population 1 (red). The p-value (p1) is the percentage of red values, that is to the left of the green dotted line. The green dotted line is the average of Population 1:1 and the red dotted line is the average of Population 1. Similarly, in Figure 5.7, the green dotted line is the average of Population 2:2, and the red dotted line is the average of Population 2. The same logic follows for Driver 3 in Figure 5.8.

In all three the density plots for the three drivers of the best model (Figures 5.6 to 5.8), corresponding to the heat map in Figure 5.5, it can be seen that there is a large difference between the distributions of the sample values for the sub populations 1:1, 2:2 and 3:3, and their sampling populations *i.e.* populations 1, 2 and 3 respectively. On all three of the plots it can be seen that the distances between a drivers' own trips are much smaller than the distances between the trips of different drivers, and that almost no values in the red density plots lie left to the green dotted lines. This is why the p-values for this model are so small. The model was therefore able to provide an excellent clustering result for the driver behavior of

the three drivers in the dataset. Subsection 5.5.5 explores the results of some of the other models that were tested as part of the simulation.

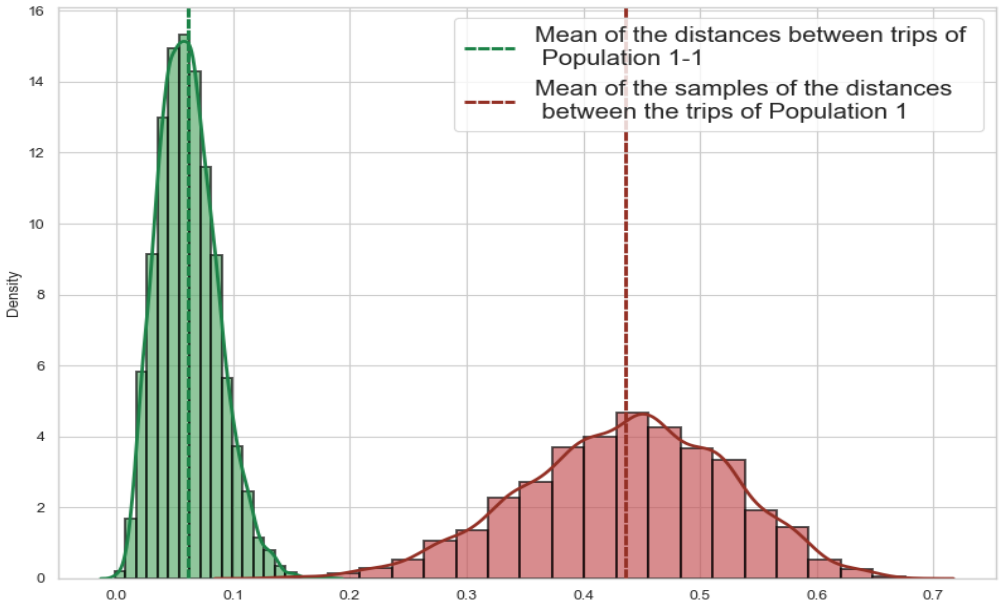


Figure 5.6: Density plots of the best model for Driver 1.

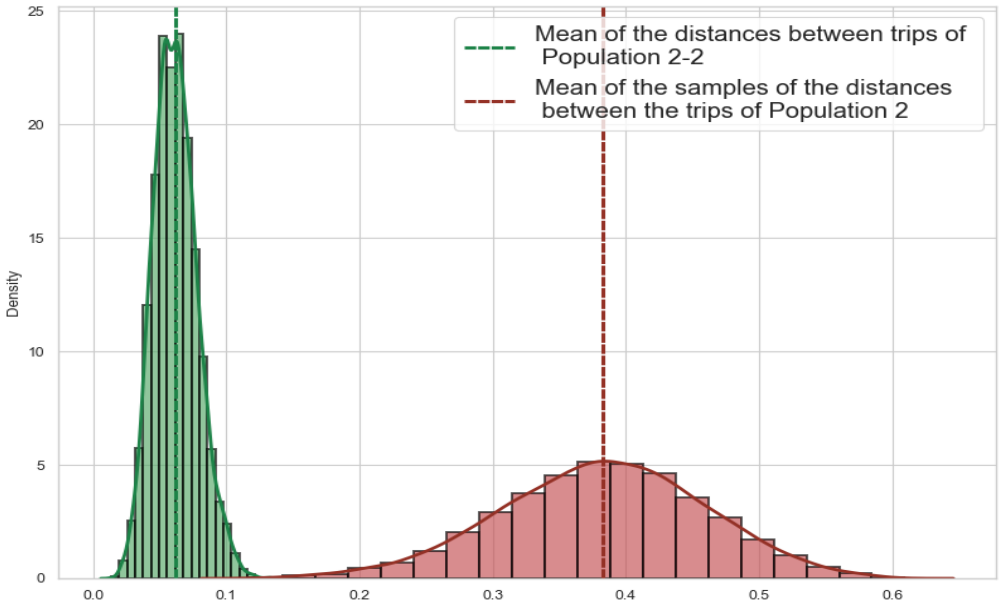


Figure 5.7: Density plots of the best model for Driver 2.

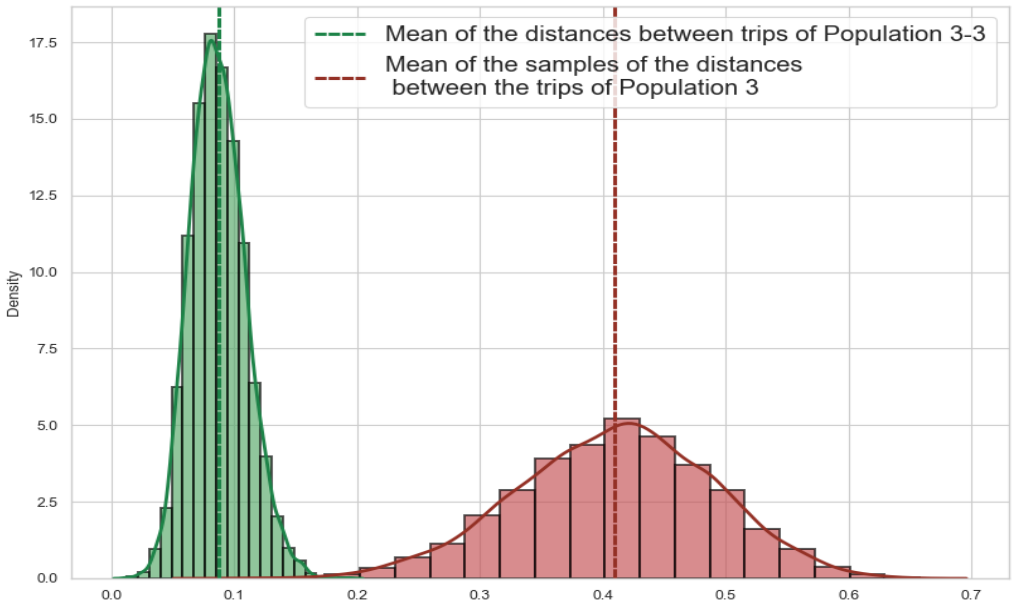
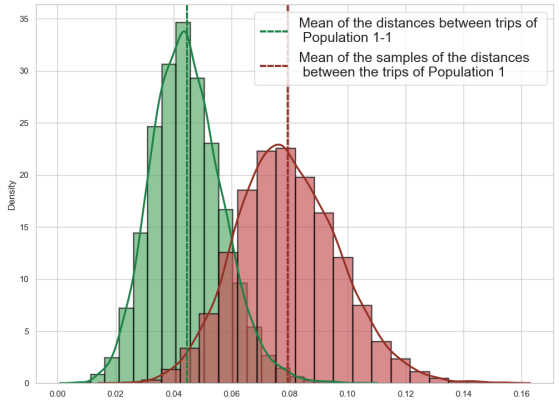


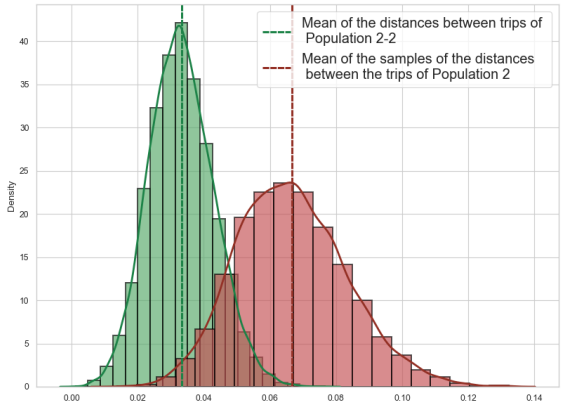
Figure 5.8: Density plots of the best model for Driver 3.

5.5.5 Discussion of other models

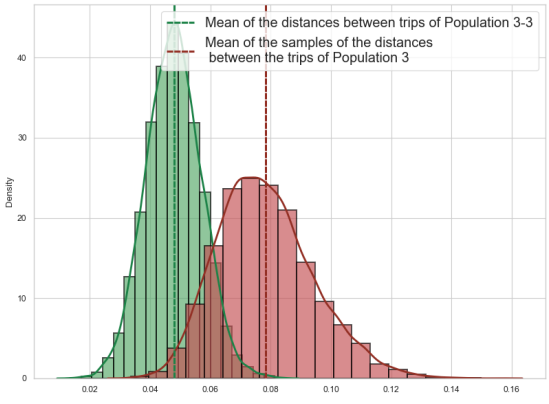
5.5.5.1 Model containing only speed data



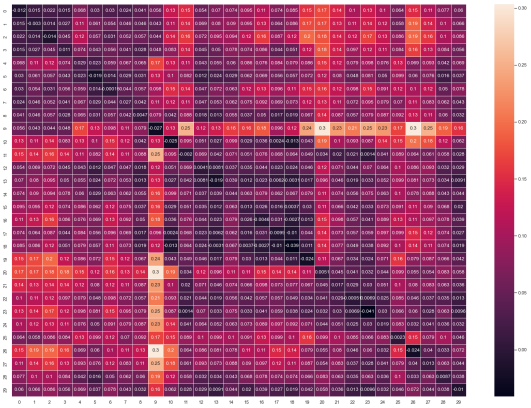
(a) Driver 1 density plot.



(b) Driver 2 density plot.



(c) Driver 3 density plot.

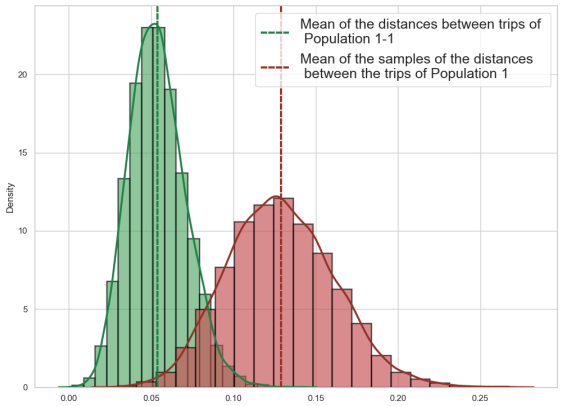


(d) Heat map of the speed only model.

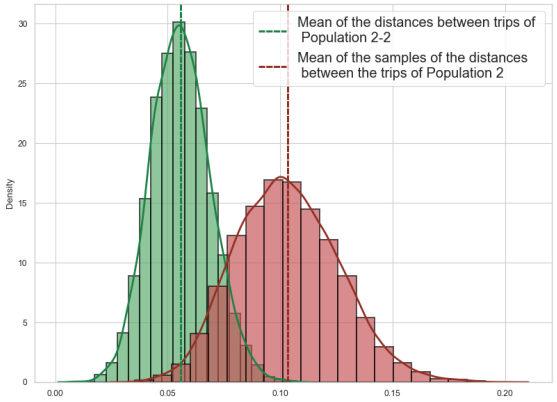
Figure 5.9: Plots for the model based only speed, (FC1) in Table 5.4.

Figure 5.9, shows how the model performed for FC 01, *i.e.* the model that only used the speed feature to cluster drivers. Since, as explained above, the most parsimonious model would be to use 50 topics. Figure 5.9, shows the result for 50 topics, for the Jensen-Shannon distance metric. The average p-values for this specific model, was after 100 iterations 0.022054, 0.011141 and 0.019596 for p1, p2 and p3 respectively. This is not a very good result, none of the p-values are significant at the defined 1% level of significance. It can be seen from the density plots for this model that the red and green densities overlap quite a bit more, compared to the best result shown above. Although one can see the population blocks for the different drivers from the heat map in Figure 5.9 (d), these blocks are not very defined, which also shows that the model was not able to pick up enough separation between the drivers.

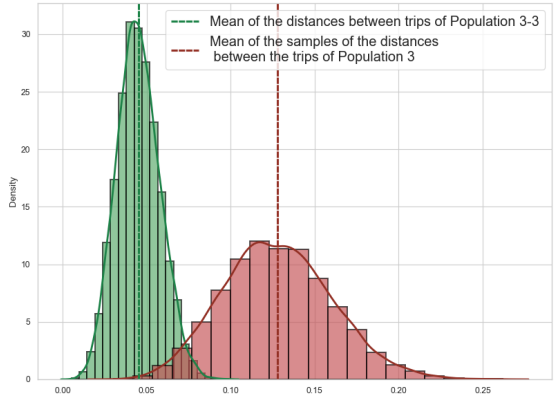
5.5.5.2 Model containing only rpm data



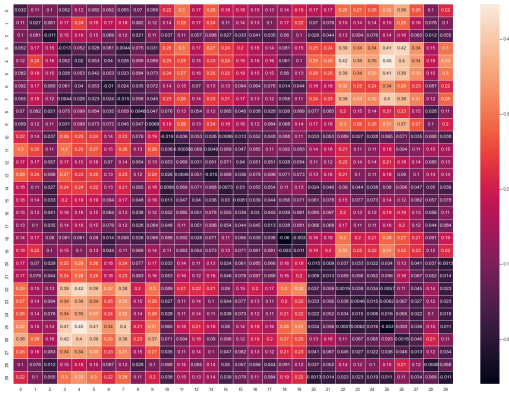
(a) Driver 1 density plot.



(b) Driver 2 density plot.



(c) Driver 3 density plot.

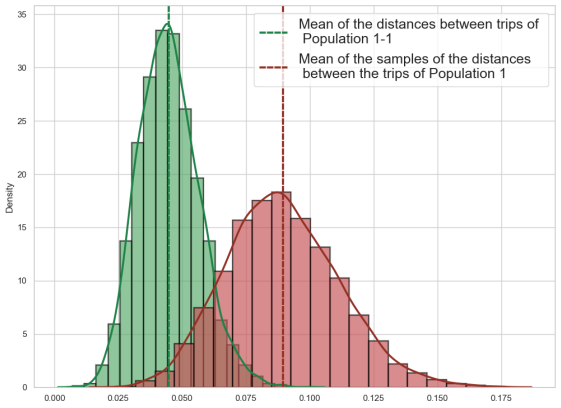


(d) Heat map of the rpm only model.

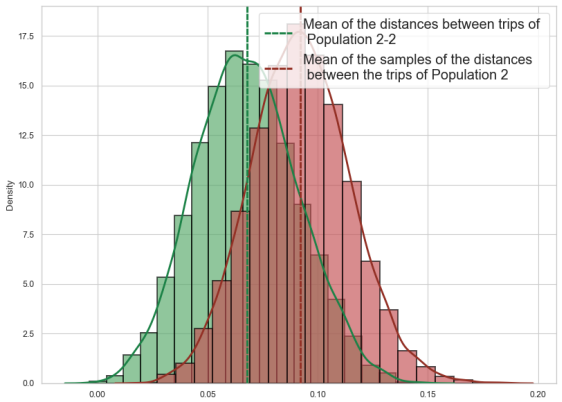
Figure 5.10: Plots for the model based only on rpm, (FC2) in Table 5.4.

Figure 5.10, shows how the model performed for FC 02, *i.e.* the model that only used the rpm feature to cluster drivers. Similar to the speed only model above, Figure 5.10 shows the result of the model, using 50 topics, for the Jensen-Shannon distance metric. From the density plots it can be seen that the rpm only model performed better than the speed only model, since, although marginally, the density plots for the rpm only model were more separated compared to the speed only model. Visually, a big difference can be seen on the heat map, with much clearer blocks for the populations of each driver, indicating better separation between the drivers. Also, the average p-values for each driver for this model after 100 iterations is 0.005959, 0.003800 and 0.002067 for p1, p2 and p3 respectively. All 3 these p-values are significant at the defined 1% level of significance.

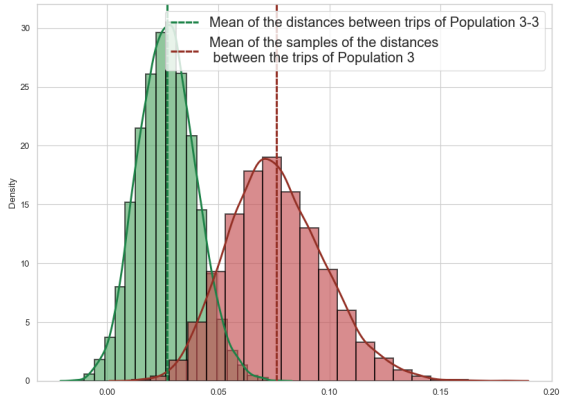
5.5.5.3 Model containing only emissions data



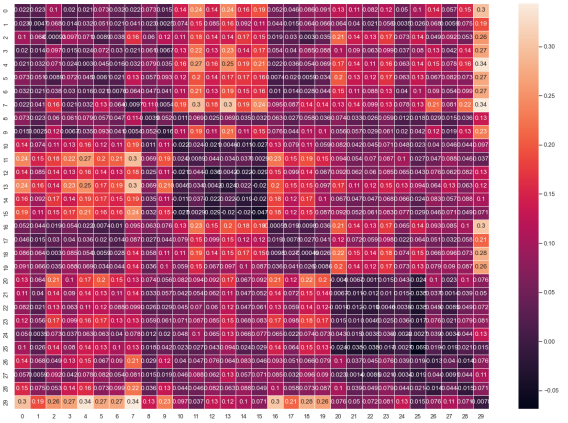
(a) Driver 1 density plot.



(b) Driver 2 density plot.



(c) Driver 3 density plot.



(d) Heat map of the emissions only model.

Figure 5.11: Plots for the model based only on emissions, (FC8) in Table 5.4.

Figure 5.11, shows how the model performed for FC 08, *i.e.* the model that only used the emissions features to cluster drivers. Once again, Figure 5.11 shows the result of the model using 50 topics, for the Jensen-Shannon distance metric. In this model, the features CO_2, CO and NO_x were combined in an experiment to see if the emissions values could provide good clustering results between the trajectories of the different drivers. The results, however show that this was not the case, from the density plots it is clear that there is a large overlap between the densities for the drivers. On the heat map, it is also clear that the model was not always able to differentiate between drivers. The distances for Driver 2 seemed to form two separate clusters, one for trips 10-15, and another one for trips 16-19, which is not a great result. This is also why there is such a large overlap of the density plots for Driver 2. The p-values for this model are 0.023588, 0.070768 and 0.006356 for p1, p2 and p3, respectively. The poor clustering result for driver 2 can also be seen from the higher value for p2. The emissions only model was able to cluster

Driver 3 well, the can be seen from the value of 0.006356 for p3, which is significant at the 1% level, as well as from the density plots for Driver 3 and the clearly defined block in the lower right corner of the heat map in Figure 5.11.

See the emissions only model in Figure D.1 of Appendix D, where only 5 topics were used. It can be seen that the model performed worse when only using 5 topics, showing the difference it makes in the clustering results when more topics are used.

5.6 Summary of results

Although the data was originally captured to study the emissions of vehicles, many other features related to driver behaviour were also captured, features like speed, rpm and throttle are all indications of driver behaviour and made this an ideal dataset to use for the application in this chapter.

From the p-values in Tables 5.5 to 5.14, the heat map of the distance matrix in Figure 5.5, and the density plots in Figures 5.6 to 5.8, it is clear that the model was able to pick up differences in the driver behaviour of the 3 drivers using lower dimensional embeddings obtained from the LDA model. Referring formally to clustering, the model was able to cluster trips 1-10, trips 11-20 an trips 20-30 into their own groups, from the pre-processing, we know that these were the trips of Driver 1, Driver 2 and Driver 3 respectively, since the trips were ordered per driver, starting with Driver 1. Therefore the model was able to cluster the trips according to their drivers correctly.

5.6.1 The use of the speed and rpm features in the models

One might be tempted to think the rpm and speed should give the same results since they should be highly correlated. This is not always the case, since on the road there is a small delay between the rpm rising and speed catching up to the rpm levels. Another major consideration to keep in mind is that the car has 5 gears, and the speed-rpm ratio for each gear is different. The correlation between gps_speed (the speed metric used in the models) and rpm across all the drivers are 0.708, which is high, but not as high as one might expect.

As explained in Subsection 5.5.5, the rpm only model provided better results, in terms of the p-values obtained. This shows that rpm provided richer spatio-temporal information and is a better indication of driver behaviour in this dataset, compared to the speed metric.

5.6.2 Combination of speed and rpm in one model

It was when these 2 features were combined, that we obtained the model with one of the best results. When the speed and rpm features were combined in FC 04, the model was able to provide excellent clustering results for the different drivers.

5.6.3 Combination of speed and throttle in one model

It is worthy to note that the combination of speed and throttle also provided great results, comparable to that of the speed and rpm model. Once again, the logical assumption would be that there is a high correlation between throttle and rpm, but the correlation between throttle and rpm across all drivers is 0.513, which is a lot lower than one might expect. Again, this may be due to the fact that, once more pressure is applied to the throttle, there is a delay before the car's rpm can get to the desired level, causing a lower correlation. Also, since the car has 5 gears, and the throttle-rpm ratio for each gear is different, so a high correlation should not be expected. Nevertheless, the throttle feature contributed significantly towards a model that provided meaningful clustering results.

5.6.4 Acceleration only model

The model that used only the acceleration feature (FC 03), provided one of the worst models in the analysis, even at 100 topics, the p-values for the model were not significant. Hence, in contrast to what was expected, when using the LDA model on this dataset, the acceleration feature did not provide as much information on driver behaviour, when compared to other features like speed, rpm and throttle. The model was able to somewhat cluster the trajectories of the drivers, but not as well as some of the other models tested.

5.6.5 Emissions only model

Similar to the Acceleration only model, the emissions model, containing a combination of the features CO, CO_2 and NO_x, which is FC 08, did not provide good clustering results. Interestingly, the model performed far better in separating Driver 3 from the other drivers, when compared to the results for Driver 2 and Driver 1. At the 100 topic model FC 08 had p-values of 0.023716, 0.064049, and 0.005536 for p1, p2 and p3 respectively. Therefore, the p-value for Driver 3 was significant, where the p-values for Drivers 1 and 2 was not even close to the significance level of 1%.

5.6.6 Comparing the two distance metrics

When looking at Figure 5.4, the results of the Bhattacharyya and Jensen-Shannon models can be compared quite easily. In comparison with the Bhattacharyya implementation, the results of the Jensen-Shannon implementation performed marginally better. This can be seen from the fact that the average p-values of the drivers for the Jensen-Shannon model were a bit lower at each topic level. Also, the elbow turn on the Bhattacharyya graph is not as sharp, when compared to the Jensen-Shannon graph, indicating that increasing the number of topics did not have such a strong effect in decreasing the p-values when compared to the Jensen-Shannon model.

5.6.7 Number of topics

As can be seen from Figure 5.4, the number of topics used in the model proved to be a significant factor in it's performance. The graph makes a clear elbow turn, indicating that, initially, increasing the number of topics had a significant effect on decreasing the average p-values for all the drivers by a substantial margin. In contrast, between 50 and 100 topics, increasing the number of topics only related to a marginal improvement in the results. For the best models, FC 04, FC 05 and FC 07, which were the speed-rpm, speed-throttle and speed-rpm-throttle-acceleration models respectively, the models performed well at all levels, from 5 topics to 100 topics. With more topics providing negligible improvements. In contrast, for the models that struggled at 5 topics, adding more topics to the mix allowed those models to perform exceptionally better, for FC 01 (the speed only model) for example, the p-values more than halved when 10 topics were used instead of 5, and halving again when 20 topics were used instead of 10. For that model, when 5 topics were used, the p-values were 0.12382, 0.10490 and 0.12312, for p1, p2 and p3 respectively, compared to 0.01871, 0.00927 and 0.01409 for p1, p2 and p3 respectively, when using 100 topics. This shows how much better the results get when more topics are used. The other FC's that struggled at 5 topics, also followed a similar pattern, with almost all the models providing good results at 100 topics.

5.6.8 Results for the individual drivers

Referring to Figure 5.4 once again, it is clear that the model performed similarly for Drivers 1 and 3, both having average p-values close to 0.01 at 50 and 100 topics. Driver 2, denoted by the red line, was clearly more difficult to cluster than the other two drivers for the majority of the models. For the best models (FC 04, FC 05 and FC 07), at 50 topics for the Jensen-Shannon model, the clustering results for Driver 2 was clearly significant, with p-values of 0.000074 and 0.000083 for p2 of FC 04 and FC 05 respectively, as can be seen on Table 5.13. Therefore when using the right model, Driver 2 was clustered correctly, but even in these models, the p-values for Driver 2 were higher, compared to the p-values for Drivers 1 and 3.

In Figure 5.4, the red line showing the p-values for Driver 2 is clearly well above the blue and green lines of Drivers 1 and 3, even when looking at the 100 topic level.

5.6.9 General notes

It is important to note that when capturing the data, the goal was not to look at driver behaviour and hence, the drivers did not set out to drive differently, meaning that the dataset is a very realistic representation of real word driving behaviour. Because of this, differences in driving behaviour picked up by the LDA embeddings can be considered as a significant result.

The experiment, as outlined above, was repeated 100 times for each of the 12 FC's, for 5, 10, 20, 50, 100 topics, which accounts to 6000 LDA models created for each distance metric, hence 12 000 LDA models were created in total. For each LDA model, and each driver, 10 samples were drawn, 10 000 times, meaning that 100 000 samples were drawn for each driver at each iteration, resulting in 300 000 samples drawn at each iteration. Therefore, in total 3.6 billion samples were drawn to test the models, at the Bhattacharyya distance metric, the simulation ran for 30078 seconds $\approx$ 8.355 hours to create the 6000 LDA models and for the Jensen-Shannon distance metric, the simulation ran for 31607 seconds $\approx$ 8.780 hours.

5.7 Conclusion

In this application the question of whether it is possible to cluster driver behaviour with LDA was answered. A simulation study was done, applying the LDA models in many different combinations, and the results show that when proper pre-processing is done on the semantic information available with the trajectory data, it is in fact possible to distinguish between driver behaviour using the lower dimensional embeddings obtained from the LDA model. When using the right combination of features and enough topics, the model was able to successfully cluster all of the trips for each of the drivers at a 1% level of significance. Therefore the topic $\times$ trajectories matrix θ, which was introduced in Chapter 3, provided embeddings that could be used to distinguish driver behaviour, based on the Bhattacharyya and Jensen-Shannon distance metrics.

Using the lower dimensional embeddings present in the LDA model, the model is able to detect differences present in the lower dimensional latent space, hence avoiding the problems that other traditional clustering techniques face. These problems were outlined by [48] and discussed in Section 2.2.

Please find the full code for this chapter on my github as Analysis of the emissions dataset final.ipynb. Please note that in order to protect the IP of this project, the repository is set to private, please contact

me at the email address armandgraaff@gmail.com, if you would like access to the repo.

Chapter 6

Conclusions

In this work, based on the similarities between text and movement data, the application of LDA to movement data was studied. Once this similarity was established, LDA provided a remarkable collection of methods and tools for the analysis of movement data. Trajectory clustering techniques with LDA utilise unsupervised techniques developed in the area of text analytics.

The results of the experiments in this work have shown that by using LDA, interpretable topics for trajectories can be obtained. Furthermore, trajectories can be clustered based on the lower dimensional embeddings of movement behavior that is semantically present in the trajectories. As far as we know, this is the first time that movement data have been clustered by using these statistical distance formulas to calculate the distance between the lower dimensional embeddings obtained from LDA.

Two use cases for LDA based on trajectory modelling were presented. Firstly, the technique was applied to the movement of jaguars in order to extract meaningful topics, and secondly, by clustering the driving behaviour of three individuals.

The limitations of this method include the fact that semantic information about the trajectory data is needed in order to perform clustering using LDA. If sufficient semantic information is not present in the data, semantic information can be derived by applying abstraction on the data, but these semantic features are often not as rich in information about movement behaviour, compared to semantic information captured with the lat-lon points.

Future work in this field can focus on extending the methodology established in this work to more domains by allowing for cross disciplinary collaboration between NLP specialists and domain experts in movement data fields. The analysis of trajectory clustering can be extended to trajectory classification, various text classification techniques can be explored, including Tf-Idf, Word2Vec and BERT. Future work can also

focus on the explainability of the LDA model, so that the technique developed in this work can be used and interpreted by domain experts. Efficient trajectory visualisation with packages such as moving pandas can also be explored as part of enhancing the explainability of these models.

Appendix A

Plot of the jaguar movement on a map

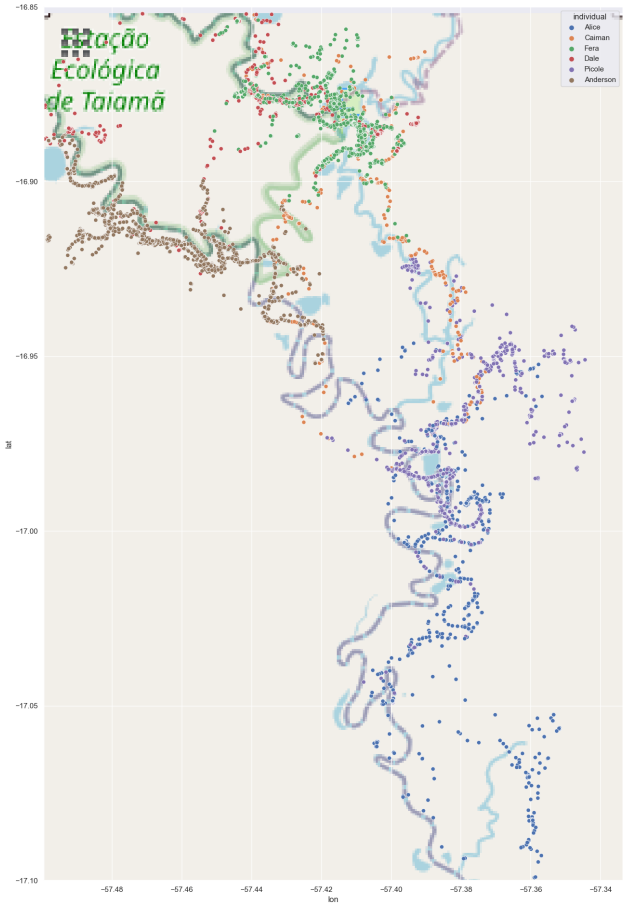


Figure A.1: Trajectories of 6 jaguars.

Appendix B

The Urn model

The urn model is a way to model real life issues as if they were problems that involve drawing balls out of an urn. The urn model used in this work can be described as follows:

In the application of this work, the "urns" in the urn model are represented by Population 1, Population 2 and Population 3, defined in Subsection 5.4. For the hypothesis test of Driver 1, Population 1 was used as the urn from which values were drawn. It is useful to note here that the sub populations *i.e.* Population 1:1, Population 2:2 and Population 3:3, were included in the urns, for example, Population 1:1 is included as part of Population 1. This is necessary since the urn model should sample from the entire population of values for each driver.

Steps

The steps of the urn model re-sampling procedure are explained in terms of Driver 1, but the same procedure was repeated for Driver 2 and Driver 3 as well.

Step 1:

Since each driver has 10 trips, a sample of 10 distances was randomly drawn from the urn. The mean of this sample was then calculated and added to a list. This process was repeated 10 000 times and in the end we have a list of 10 000 means. The mean of this list was then calculated, effectively calculating the "mean of means".

Step 2:

The mean of the distances between the 10 trips of Driver 1 was then calculated, note that no sampling

was done here.

Step 3:

The results for Driver 1, and all of the distances, can then be compared. The null hypothesis states that the mean of Population 1:1 will be greater than or equal to the mean of the sample averages from Population 1. The p-values were calculated as the proportion of values that were sampled from Population 1 that were smaller compared to the average of Population 1:1.

Step 4:

The means of population 1 and population 1:1 were plotted, along with the density plot for the distances between the trips for Driver 1 only (Population 1:1), and the density plot of the list containing the 10 000 sampled means from Population 1.

Appendix C

Discretisation of features in order to construct feature combinations

The feature combinations (FC's) were created from the raw data in the following manner using Rstudio. Recall that data was captured at every second of each trip, so there is a new observation at a new lat/lon point every second.

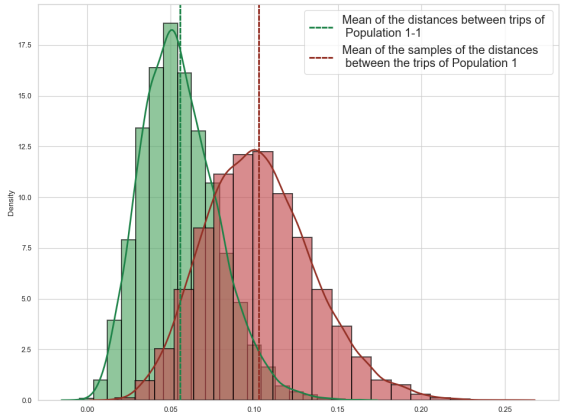
1. The raw data was loaded into Rstudio
2. The acceleration was calculated at each observation (lat-lon point).
3. The CO, CO_2 and NO_x emission metrics was calculated per distance, *i.e.* the emission metrics were divided by the distance travelled for that observation. So the result is emissions emitted per metre, calculated at each observation.
4. From the **arules** library in Rstudio, the **discretise** method was used to cluster each feature into 6 clusters by use of k-means clustering.
5. Each feature was given a "syllable", for example, the feature "speed", was given the syllable "speed-", "acceleration" was given the syllable "accel-", the rest of the features was given similar syllables.
6. For each FC, create "words" by concatenating the syllables of each feature included in the FC, with the bins that each feature are in at each observation.
7. Write the data to a csv file so that it can be imported into the python script.

Please find the full code on my github as Creating Feature Combinations.R. Please note that in order

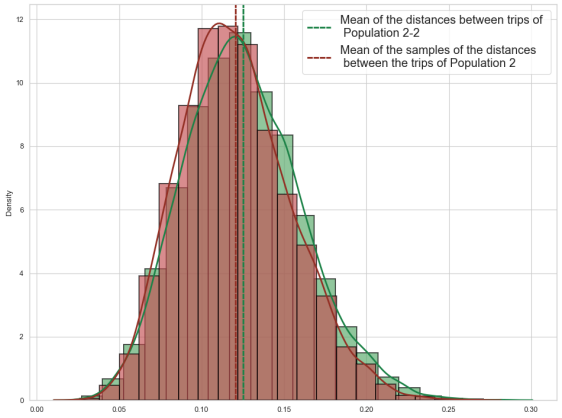
to protect the IP of this project, the repository is set to private, please contact me at the email address armandgraaff@gmail.com, if you would like access to the repo.

Appendix D

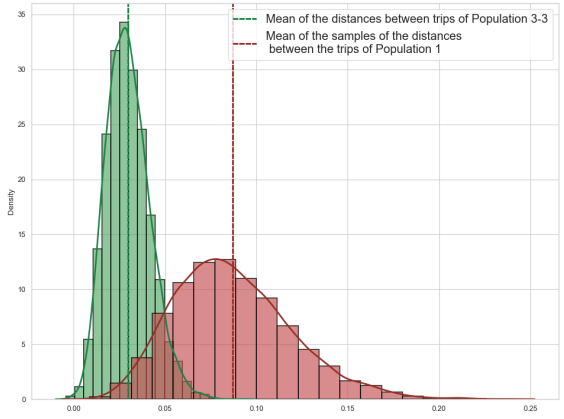
Emissions only model with 5 topics



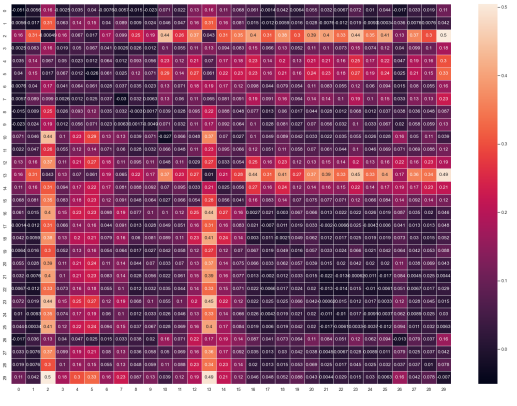
(a) Driver 1 density plot.



(b) Driver 2 density plot.



(c) Driver 3 density plot.



(d) Heat map of the emissions only model.

Figure D.1: Plots for the model based only on emissions, where only five topics were used, (FC8) in Table 5.4.

Bibliography

[1] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual NLP. *ArXiv Preprint ArXiv:1307.1662*, 2013.

[2] Basma Albanna, Ibrahim Moawad, Sherin Moussa, and Mahmoud Sakr. *Semantic Trajectories: A survey from Modeling to Application*, pages 59–76. 05 2015.

[3] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering clusters in motion time-series data. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[4] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *ACM SIGKDD Explorations Newsletter*, 9(2):38–46, 2007.

[5] Natalia Andrienko and Gennady Andrienko. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, 12(1):3–24, 2013.

[6] Miriam Baglioni, José Antônio Fernandes de Macêdo, Chiara Renso, Roberto Trasarti, and Monica Wachowicz. Towards semantic interpretation of movement behavior. In *Advances in GIScience*, pages 271–288. Springer, 2009.

[7] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[8] Jiang Bian, Dayong Tian, Yuanyan Tang, and Dacheng Tao. A survey on trajectory clustering analysis. *ArXiv Preprint ArXiv:1802.06971*, 2018.

[9] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, 2006.

[10] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[12] Hancheng Cao, Fengli Xu, Jagan Sankaranarayanan, Yong Li, and Hanan Samet. Habit2vec: Trajectory semantic embedding for living pattern recognition in population. *IEEE Transactions on Mobile Computing*, 19(5):1096–1108, 2019.

[13] Ding Chu, David A Sheets, Ye Zhao, Yingyu Wu, Jing Yang, Maogong Zheng, and George Chen. Visualizing hidden themes of taxi movement with semantic transformation. pages 137–144, 2014.

[14] Nicholas Dalton-Barron, Sarah Whitehead, Gregory Roe, Cloe Cummins, Clive Beggs, and Ben Jones. Time to embrace the complexity when analysing GPS data? A systematic review of contextual factors on match running in rugby league. *Journal of Sports Sciences*, 38(10):1161–1180, 2020. PMID: 32295471.

[15] Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 795–804, 2015.

[16] Urška Demšar, Kevin Buchin, Francesca Cagnacci, Kamran Safi, Bettina Speckmann, Nico Van de Weghe, Daniel Weiskopf, and Robert Weibel. Analysis and visualisation of movement: An interdisciplinary review. *Movement Ecology*, 3(1):1–24, 2015.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[18] Leonard Evans. Human behavior feedback and traffic safety. *Human Factors*, 27(5):555–576, 1985.

[19] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Joural of the American Mathematical Society*, 29(4):983–1049, 2016.

[20] Mauro Ferrante, Stefano De Cantis, and Noam Shoval. A general framework for collecting and analysing the tracking data of cruise passengers at the destination. *Current Issues in Tourism*, 21(12):1426–1451, 2018.

[21] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

[22] Andre Salvaro Furtado, Despina Kopanaki, Luis Otavio Alvares, and Vania Bogorny. Multidimensional similarity measuring for semantic trajectories. *Transactions in GIS*, 20(2):280–298, 2016.

[23] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 63–72, 1999.

[24] Suelane Garcia Fontes, Ronaldo Gonçalves Morato, Silvio Luiz Stanzani, and Pedro Luiz Pizzigatti Corrêa. Jaguar movement behavior: Using trajectories and association rule mining algorithms to unveil behavioral states and social interactions. *PLOS One*, 16(2):e0246233, 2021.

[25] Anita Graser. *Learning QGIS 2.0.* Packt Publishing Ltd, 2013.

[26] Anita Graser. MovingPandas: efficient structures for movement data in Python. *GIForum*, 1:54–68, 2019.

[27] Eliezer Gurarie, Russel D Andrews, and Kristin L Laidre. A novel method for identifying behavioural changes in animal movement data. *Ecology Letters*, 12(5):395–408, 2009.

[28] Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent Dirichlet allocation. *Advances in Neural Information Processing Systems*, 23, 2010.

[29] Johan W. Joubert and Ruan J. Grabe. Real driving emissions data: Isuzu FTR850 AMT. *Data in Brief*, 41:107975, 2022.

[30] Xiangjie Kong, Menglin Li, Kai Ma, Kaiqi Tian, Mengyuan Wang, and Feng Xia. Big trajectory data: A survey of applications and services. *IEEE Access*, PP:1–1, 10 2018.

[31] Mario Köppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, volume 1, pages 4–8, 2000.

[32] Patrick Laube. *Computational Movement Analysis.* Springer, 2014.

[33] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: A partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pages 593–604, 2007.

[34] Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31, 2008.

[35] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.

[36] Lyuchao Liao, Jianping Wu, Fumin Zou, Jengshyang Pan, and Tingting Li. Trajectory topic modelling to characterize driving behaviors with GPS-based trajectory data. *Journal of Internet Technology*, 19(3):815–824, 2018.

[37] Huan Liu, Sichen Jin, Yuyu Yan, Yubo Tao, and Hai Lin. Visual analytics of taxi trajectory data via topical sub-trajectories. *Visual Informatics*, 3(3):140–149, 2019.

[38] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[39] Ying Long, Xingjian Liu, Jiangping Zhou, and Yanwei Chai. Early birds, night owls, and tireless/recurring itinerants: An exploratory analysis of extreme transit behaviors in Beijing, China. *Habitat International*, 57:223–232, 2016.

[40] James MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297, 1967.

[41] Jean Damascène Mazimpaka and Sabine Timpf. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 2016(13):61–99, 2016.

[42] Erik W Meijles, M. de Bakker, Peter D Groote, and R. Barske. Analysing hiker movement patterns using GPS data: Implications for park management. *Computers, Environment and Urban Systems*, 47:44–57, 2014.

[43] Sebastiaan Merino and Martin Atzmueller. Behavioral topic modeling on naturalistic driving data. *Proceedings of BNAIC. Jheronimus Academy of Data Science, Den Bosch, The Netherlands*, 2018.

[44] Christopher C Miller. A beast in the field: The Google maps mashup as GIS/2. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 41(3):187–199, 2006.

[45] Kevin P Murphy. *Machine learning: A Probabilistic Perspective*. MIT press, 2012.

[46] Asana Neishabouri and Michel C Desmarais. Reliability of perplexity to find number of latent topics. In *The Thirty-Third International Flairs Conference*, 2020.

[47] Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10):1100, 2020.

[48] Xavier Olive, Luis Basora, Benoit Viry, and Richard Alligier. Deep trajectory clustering with autoencoders. In *ICRAT 2020, 9th International Conference for Research in Air Transportation*, 2020.

[49] Yashon O Ouma, TG Ngigi, and R Tateishi. On the optimization and selection of wavelet texture for feature extraction from high-resolution satellite imagery with application towards urban-tree delineation. *International Journal of Remote Sensing*, 27(1):73–104, 2006.

[50] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 863–868, 2008.

[51] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, et al. Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, 45(4):1–32, 2013.

[52] Damião Ribeiro de Almeida, Cláudio de Souza Baptista, Fabio Gomes de Andrade, and Amilcar Soares. *ISPRS International Journal of Geo-Information*, 9(2):88, 2020.

[53] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, and Gennady Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3-4):225–239, 2008.

[54] Peter E Smouse, Stefano Focardi, Paul R Moorcroft, John G Kie, James D Forester, and Juan M Morales. Stochastic modelling of animal movement. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1550):2201–2211, 2010.

[55] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Nnatural Language Processing*, pages 151–161, 2011.

[56] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Data & Knowledge Engineering*, 65(1):126–146, 2008.

[57] Qi Sun, Runxin Li, Dingsheng Luo, and Xihong Wu. Text segmentation with LDA-based Fisher kernel. In *Proceedings of ACL-08: HLT, Short Papers*, pages 269–272, 2008.

[58] Jeffrey J Thompson, Ronaldo G Morato, Bernardo B Niebuhr, Vanesa Bejarano Alegre, Júlia Emi F Oshima, Alan E de Barros, Agustín Paviolo, J Antonio de la Torre, Fernando Lima, Roy T McBride Jr, et al. Environmental and anthropogenic factors synergistically affect space use of jaguars. *Current Biology*, 31(15):3457–3466, 2021.

[59] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, 2010.

[60] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112, 2009.

[61] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic trajectories: Mobility data computation and annotation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):1–38, 2013.

[62] Zhixian Yan, Stefano Spaccapietra, et al. Towards semantic trajectory data analysis: A conceptual and computational approach. 3:23, 2009.

[63] Guan Yuan, Penghui Sun, Jie Zhao, Daxing Li, and Canwei Wang. A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, 47(1):123–144, 2017.

[64] Mingfeng Zhang and Hugh H T Liu. Cooperative tracking a moving target using multiple fixed-wing UAV's. *Journal of Intelligent & Robotic Systems*, 81(3):505–529, 2016.

[65] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. *ACM Sigmod Record*, 25(2):103–114, 1996.