Cluster Analysis and EDA of Taxi Demand Data

Zilong Wang[a,1] **and Athanasios Lolos**[a,1]

[a]Georgia Institute of Technology

This manuscript was compiled on October 28, 2020

We analyse the Taxi Service Trajectory dataset (1) in the UCI Machine Learning Repository, which was used in a prediction challenge in ECML KDD 2015, and attempt a more in depth exploratory data analysis (EDA) stage. Chief amongst these methods used are the dynamic time warping (DTW) metric along with hierarchical clustering to group demand trends together, along with other classical dimensionality reduction tools such as principal component analysis (PCA). We hope that we will get a better feel of the exact nature of the dataset before attempting to replicate a portion of the paper by (2) in order to verify their assumptions with regards to the data generating process.

Dynamic Time Warping|Principal Component Analysis|Vector ARIMA|Hierarchical Clustering

This is the project proposal for Fall 2020 ISyE 7405's project using the style of PNAS's template. In the following sections, we provide a brief description of the dataset, along with a quick rundown of the methods we propose to use.

Description of Dataset

For this project we will use an accurate dataset that describes the busy trajectories (the trajectories when a customer uses the taxi) performed by all the 442 taxis running in the city of Porto, in Portugal (from 01/07/2013 to 30/06/2014). These taxis operate through a taxi dispatch central, using mobile data terminals installed in the vehicles.

Dataset Headers. Each data sample corresponds to one completed trip and contains a total of nine features that are described below:

1. **TRIP_ID**:(String) It contains a unique identifier for each trip.
2. **CALL_TYPE**: (Char) It identifies the way used to demand this service. It may contain one of three possible values.
   - i **'A'** if this trip was dispatched from the central.
   - ii **'B'** if this trip was demanded directly to a taxi driver at a specific stand
   - iii **'C'** otherwise (e.g. a trip demanded on a random street).
3. **ORIGIN_CALL**: (integer) It contains a unique identifier for each phone number which was used to demand at least one service. It identifies the trip's customer if CALL_TYPE= 'A'. Otherwise, it assumes a NULL value.
4. **ORIGIN_STAND**: (integer) It contains a unique identifier for the taxi stand. It identifies the starting point of the trip if CALL_TYPE= 'B'. Otherwise, it assumes a NULL value.
5. **TAXI_ID**: (integer) It contains a unique identifier for the taxi driver that performed each trip.
6. **TIMESTAMP**: (integer) Unix Timestamp (in seconds). It identifies the trip's start.
7. **DAYTYPE**: (char) It identifies the daytype of the trip's start. It assumes one of three possible values.
   - i **'B'** if the trip started on a holiday or any other special day (i.e. extending holidays, floating holidays, etc.)
   - ii **'C'** if the trip started on a day before a type-B day
   - iii **'A'** otherwise (i.e. a normal day, workday or weekend)
8. **MISSING_DATA**: (Boolean) It is FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing.
9. **POLYLINE**: (string) It contains a list of GPS coordinates (i.e. WGS84 format) mapped as a string. The beginning and the end of the string are identified with brackets. In addition, each pair of coordinates is also identified by brackets [LONGITUDE, LATITUDE]. The list contains one pair of coordinates for each 15 seconds of trip. The last list item corresponds to the trip's destination, while the first one represents its start.

Other Details. From the CALL_TYPE feature we see that essentially there are three main ways to pick-up a passenger. The first way is that a passenger goes to a taxi stand stand and picks-up a taxi (CALL_TYPE= 'B'). The second way is that the passenger calls the taxi network central and demands a taxi for specific location/time (CALL_TYPE= 'A') and finally a passenger may pick a vacant taxi on any street at random (CALL_TYPE= 'C').

Overview of Methods

Dynamic Time Warping. Finding methods to "better" cluster time series is still an ongoing effort. As opposed to time

Significance Statement

Public datasets are now commonly reused for different works once introduced, it is therefore important to curate and properly document them with extensive EDA for future referencing and reproducibility.

Please provide details of author contributions here.

Please declare any competing interests here.

[1]A.O.(Author One) contributed equally to this work with A.T. (Author Two) (remove if not applicable).

[2]To whom correspondence should be addressed. E-mail: author.twoemail.com

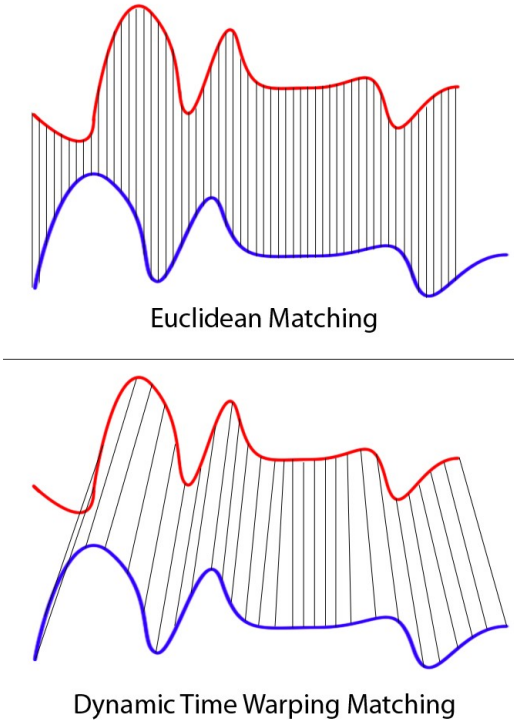

Fig. 1. Euclidean Matching v.s. DTW Matching

invariant observations, where individual points are clustered with each other, due to the temporal component, where we now have a sequence of observations for each individual, clustering individuals together is now much more difficult. The Dynamic Time Warping (DTW) (3) algorithm, which was first used in speech pattern discrimination, is a commonly used and robust algorithm that attempts to match two univariate time series data by the similarity of their "shape" and "trend". As opposed to Euclidean distance matching. An illustration can be seen in Figure 1, where DTW has a more natural match. DTW can also account for different lengths (different starting and end times) and is hence a very natural choice.

Once the dissimilarity matrix between each time series observation is generated, the sequences are typically clustered together from the bottom up via hierarchical clustering.

Hierarchichal Clustering. NOTE: We leave out the description for now as the instructor will cover this later in class. We wish to be consistent in terminology and description

Vector ARIMA. The Auto Regressive Integrated Moving Average (ARIMA) model is a commonly used method in statistics and econometrics in forecasting time series. It is a very general model with some key properties being the "Auto Regressive" (AR) property, where future predictions can be done by regressing off previous lagged observations, and the moving average (MA) property, where the data can be de-trended to account for seasonality and general exogeneously induced trends. Vector ARIMA is an extension of this method where instead of forecasting univariate time series, panel data (repeated observations of longitudinal data / multiple time series) are forecasted together. The authors used various fitting windows (lag periods) and different distributional assumptions in order to forecast 30 min ahead at every time point.

ACKNOWLEDGMENTS. N/A

1. Taxi Service Trajectory prediction challenge, ecml pkdd 2015 data set (https://archive.ics.uci.edu/ml/datasets/Taxi+Service+Trajectory+-+Prediction+Challenge%2C+ECML+PKDD+2015#) (2015) Accessed: October 28, 2020.
2. L Moreira-Matias, J Gama, M Ferreira, J Moreira, L Damas, Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intell. Transp. Syst.* **14**, 1393–1402 (2013).
3. TK Vintsyuk, Speech discrimination by dynamic programming. *Cybernetics* **4**, 52–57 (1968) Russian Kibernetika 4(1):81-88 (1968).