CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Comparing Time Series Segmentation Methods for the Analysis of Transportation Patterns with Smart Card Data

Li He
Bruno Agard
Martin Trépanier

June 2017

CIRRELT-2017-28

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Comparing Time Series Segmentation Methods for the Analysis of Transportation Patterns with Smart Card Data

Li He, Bruno Agard, Martin Trépanier*

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematics and Industrial Enginnering, Polytechnique Montréal, 2500, chemin de Polytechnique, Montréal, Canada H3T 1J4

Abstract. Time series clustering is important in the analysis of action. In the domain of transportation, it is especially important. It allows an understanding of people's activities within a time period. In this article, a method is presented for the segmentation of time series that takes advantage of cross-correlation distance, hierarchical clustering, and considers the separation of positive/negative correlations in order to understand temporal patterns of users. This method consists of two steps: (1) Combining cross correlation distance and hierarchical clustering to obtain cluster groups, and (2) dividing these groups into smaller sized groups by separating cross correlation parameters, "correlation coefficient" and "lag". Considering that dynamic time warping is a common method to measure time series distance, the clustering results are compared between dynamic time warping distance and cross correlation distance. After a small pedagogical example, we develop a program by R to validate the method on a real data set. The results of the real data demonstrate that this method precisely segments the time series. This comparison result also demonstrates the advantage of using cross correlation distance in the domain of public transportation.

Keywords. Cross correlation, time series segmentation, hierarchical clustering, public transportation data, customer behavior.

Acknowledgements. The authors wish to acknowledge the support of the Société de transport de l'Outaouais (STO) for providing data, the Thales group and the Natural Sciences and Engineering Research Council of Canada (NSERC project RDCPJ 446107-12) for funding.

---

* Corresponding author: martin.trepanier@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2017

1 INTRODUCTION

The extraction of customer behavior in public transit systems is extremely dynamic. Having a better understanding of travellers' patterns is helpful in predicting the demand for transportation (Joh and al., 2006). Many automatic systems take advantage of smart card technology that generates and stores gigabits for data about day-to-day activities of users (Pelletier and al., 2011). The time series technique is widely used in transportation forecasting (Chen et al., 2009). Many authors have suggested metrics, tools, algorithms and methods to help better understand the mobility of users during different time periods.

Public transit smart card data can be used for an analysis of users' behaviors (Pelletier and al., 2011). Some analyses have been done to cluster smart card users' behaviours using data mining (Agard and al., 2006). A tool is developed to measure the similarity of spatiality, with which card users whose routes are similar can be identified (Ghaemi and al., 2015). With regards to temporality, a method is designed to mine the pattern of card users' daily frequency (Nishiuchi and al., 2013). However, considering that it requires an efficient method for time series segmentation, no review has yet to analyse the daily profiles of users using data mining.

One largely adopted strategy in the research is to subdivide the entire population in subgroups to analyse a smaller (but representative) amount of behaviors. Input data is then the activity of each individual customer, which is explained by subgroup activities.

In most of those examples, a user is represented by a vector that describes his or her activity in the transit network. This vector could be considered as a time series and take advantage of advances in this domain. A time series represents a collection of values obtained from sequential measurements over time. Time-series data mining stems from our natural ability to visualize the shape of data (Esling and Agon, 2012).

In data mining, there are some traditional methods to measure the distance of samples, such as Euclidean distance (Berkhin, 2006), Manhattan distance (Bakar and al., 2006), and others. The most commonly used distance measures in classification or change detection approaches are derived from the Minkowski distance (Jain and al., 1999), as it is a generalization of both the Euclidean distance and the Manhattan distance (Lhermitte and al., 2011). All of these distance-calculating methods have been implemented in R (Buchta, 2015). However, these methods do not contain a conception of time process. When using these methods to calculate the distance of two given time series, it considers the difference value between values of only one time point, but not of some near-time points. However, "time" is a continuous variable and methods that do not consider the values of near-time points are ineffective when classifying time series.

Some other distance measure methods can deal with near-time points, such as cross correlation distance (Liao, 2005) or dynamic time warping distance (Berndt and Clifford, 1994). With these methods, it is possible to calculate time series distance in an appropriate way. Knowing the distance of any two time series is not enough to segment them. To divide all of the time series into some clusters, a hierarchical clustering method is necessary (Langfelder and al., 2008).

Our objective is to design a model by combining and deriving current algorithms. Therefore, first of all, cross correlation and hierarchical clustering are combined to design a method. Then, the algorithms developed by cross correlation and dynamic time warping are compared. Finally, with the method designed, a real database is tested to see how it performs.

The next section provides a state of the art on the segmentation methods, distance metrics, and then the application of data mining in public transit smart card data.

2 STATE OF THE ART

2.1 Segmentation

A cluster is a collection of data objects so that one object is like one another within the same cluster and dissimilar to the objects in other clusters. Segmentation is grouping a set of data objects into clusters. A good clustering method will produce high quality clusters with high intra-class similarity and low inter-class similarity. The quality of a clustering result depends on both the similarity measure used by the method and its implementation (Subbiah, 2011). There are several major approaches to segmentation.

(1) Partitioning algorithms

Partitioning algorithms is to construct various partitions and then evaluate them by certain criterion. The mean idea is to find a partition of k clusters that optimizes the chosen partitioning criterion given a k. The two main heuristic methods are k-means and k-medoids (Subbiah, 2011). In the first one, each cluster is represented by the center of the cluster. While in the second one, each cluster is represented by one of the objects in the cluster. The Partitioning algorithms have their advantages and limits when treating a time series. For example, the k-means can deal with large datasets, but it is only available with Euclidean distance as it defines “mean” by Euclidean distance.

(2) Hierarchical algorithms

Hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis that seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types (Rokach and al.,2005): agglomerative and divisive. For the first one, each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy. This is a "bottom-up" approach. For the second, all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. It is a "top down" approach. Compared to partitioning algorithms, hierarchical clustering is available for a variety of distances but it cannot deal with a large dataset. This requires us to choose a segmentation algorithm that satisfies the demand in the specific case.

(3) Other algorithms

There are still other methods such as the density-based, grid-based and model-based methods. These methods are designed based on certain specific functions.

Furthermore, all methods described necessitate computing a distance between the different elements (between two objects, between an object and a group, between two groups). Unfortunately, there is no universal distance measure, and it is necessary to adapt the metric for many real-world problems. In the following section, the specific situation of distance between time series is presented, which is a relevant model for the mobility profile analysis of users.

2.2 Distance Between Time Series

A time series is a set of observations x_t, each one being recorded at a specific time t. A discrete-time series (the type to which this article is primarily devoted) is one in which the set T_0 of times at which observations are made is a discrete set (Brockwell and Davis, 2016).

Comparing it to the other vectors, a time series contains a relationship among the time t itself. For example, for a time series $x_1, x_2, x_3,\ldots, x_n$, the corresponding specific time $t_1, t_2, t_3,\ldots, t_n$, t_1 is closer to t_2 than t_n regarding time, regardless of the value of x_1, x_2 and x_n.

Various distance metrics exist to measure the (dis)similarity between two vectors (He and al., 2017). In this part, four types of distance are compared: Euclidean distance, Manhattan distance, cross correlation distance and dynamic time warping distance. Among them, the first two distances cannot be used to measure the distance between time series, but the cross correlation distance and dynamic time warping distance are designed to compare the (dis)similarity of time series.

2.2.1 Euclidean and Manhattan distance

Euclidean distance is the straight-line distance between two points in Euclidean space (Deza, 2009). Let x_i and v_j each be a P-dimensional vector. The Euclidean distance is computed as (Liao, 2005):

$$d_E = \sqrt{\sum_{k=1}^{P}(x_{ik} - v_{jk})^2} \quad (1)$$

Manhattan distance is computed between the two numeric series using the following formula (Mori and al., 2016):

$$d_M = \sum_{k=1}^{P}\left|x_{ik} - v_{jk}\right| \quad (2)$$

According to functions (1) and (2), for both distances, the result of distance would not be changed if the order of k is changed; for example, if the values of k_1 and k_2 are exchanged, the distance is the same. However, a time series contains the relationship among the time t itself; this is a characteristic that makes time series different from other vectors. For a time series, if the values of k_1 and k_2 are exchanged, the distance result should change. Therefore, the Euclidean distance and Manhattan distance are not suitable for time series.

2.2.2 Cross correlation distance

Distance is based on the cross correlation between two time series (Mori and al., 2016). The similarity of two time series is measured by shifting one time series to find a maximum cross-correlation with another time series. The cross correlation between two time series at lag k is calculated as:

$$CC_k(X,Y) = \frac{\sum_{i=0}^{N-1-k}(x_i-\bar{x})(y_{i+k}-\bar{y})}{\sqrt{(x_i-\bar{x})^2}\sqrt{(y_{i+k}-\bar{y})^2}} \quad (3)$$

Where $\bar{x}$ and $\bar{y}$ are the mean values of the series. Based on this, the distance measure is defined as:

$$d_{CC}(X,Y) = \sqrt{\frac{(1-CC_0(X,Y))}{\sum_{k=1}^{max} CC_k(X,Y)}} \quad (4)$$

In R, the distance measure can be calculated by using a function. This function will return the distance between two time series by specifying two numeric vectors (x and y) and maximum lag.

2.2.3 Dynamic Time Warping

Dynamic time warping is a popular technique for comparing time series, providing both a distance measure that is insensitive to local compression and stretches and warping, which optimally deform one of the two input series on the other (Giorgino, 2009). The method to calculate the dynamic time warping distance is as follows (Berndt and al., 1994):

$$S = s_1, s_2, \dots s_i, \dots, s_n \quad (5)$$

$$T = t_1, t_2, \dots t, \dots, t_n \quad (6)$$

The sequences *S* and *T* can be arranged to form a n-by-m plane or grid, where each grid point, (*i*,*j*), corresponds to an alignment between elements s_i and t_j. A warping path, *W*, maps or aligns the elements of *S* and *T*, such that the "distance" between them is minimized.

$$W = w_1, w_2, \dots w_i, \dots, w_n \quad (7)$$

That is, *W* is s sequence of grid points, where each w_k corresponds to a point $(i,j)_k$.

To formulate a dynamic programming problem, a distance measure between two elements is indispensable. Two possible distance measures are usually used for a distance function *d*. They are the magnitude of the difference (8) or the square of the difference (9),

$$d(i,j) = |s_i - t_j| \quad (8)$$

$$d(i,j) = (s_i - t_j)^2 \quad (9)$$

Once a distance measure is selected, the dynamic time warping problem can be defined as minimization over potential warping paths based on the cumulative distance for each path, where *d* is a distance measure between two time-series elements.

$$DTW(S,T) = \min w \left[\sum_{k=1}^{P} d(w_k)\right] \quad (10)$$

2.3 Use of Data Mining in Public Transit Smart Card Data

The data collected from an automatic collection system (in this article, smart card data,) can be used to understand characteristics of public transit card users (Pelletier and al., 2011). Based on temporal analysis (Ghaemi and al., 2016) and spatial analysis (Ghaemi and al., 2015), the public transit card user's temporal patterns and spatial patterns are discovered and analysed. Data mining even helps to predict user demand (Nuzzolo and al., 2016). Moreover, by using data mining, especially the segmentation technique, a methodology has been developed to analyse the quality of transit service level (de Oña and al., 2015). However, all of this research is based on each of the smart card user's transactions, not their behaviour in a day. The time series segmentation technique helps develop a method in which to analyse a smart card user's daily profile, so that public transit authorities can offer better service that will satisfy passengers' daily requirements.

2.4 Synthesis

In this section, the definition of time series is introduced and then four distances measuring the (dis)similarity of vectors are presented; among them are two distances that are designed for time series. Segmentation methods are also presented along with their advantages and limits. However, no tool that is adequate for the segmentation of time series could be found in the literature. Therefore, the following section will propose a methodology in this direction, and in section 4, an example is introduced to show how this methodology works.

3 PROPOSED METHODOLOGY FOR THE SEGMENTATION OF TIME SERIES

3.1 Algorithm design

The following three step algorithm is proposed for the segmentation of time series, as presented in Figure 1.

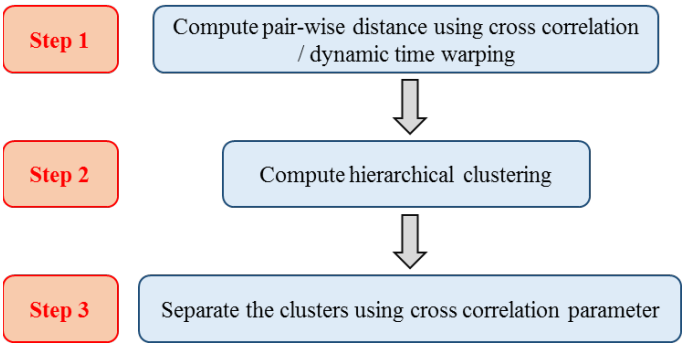


Figure 1

Step 1 The input data is time series. In this step, pairwise distance with cross correlation distance is computed. The output is a distance matrix for each of any two time series. As mentioned in part 2.2.1, Euclidean distance and Manhattan distance are not suitable for measuring time series distance. Dynamic time warping and cross correlation are both candidates for computing time series distance, then dynamic time warping can also be applied in this step.

Step 2 The input is the pairwise distance matrix between all observations, computed in step 1. A hierarchical clustering is computed to cluster the time series using a distance matrix of cross correlation or dynamic time warping. The results of hierarchical clustering are usually presented in a dendrogram. Finally, a dendrogram is obtained from which the clusters can be selected. The output is the clusters, including all of the observation points. At the end of this step is the result of series segmentation by dynamic time warping. For cross correlation, a finer result will be obtained after executing step 3.

Step 3 Each cluster is split using the cross correlation parameters: correlation coefficient and max lag. They will be explained in sections 3.2 and 3.3.

In conclusion, through step 1 and 2, time series segmentation with dynamic time warping will be done. However, to obtain a finer result with cross correlation, it needs to go through all 3 steps.

3.2 Cross Correlation Parameters

Cross correlation distance contains two important parameters:

(1) Correlation coefficient: The Pearson correlation coefficient measures the strength of the linear association between two variables (Sedgwick and Philip, 2012). A correlation coefficient close to +1 or -1 represents a strong correlation. However, the correlation coefficient can be positive and negative. In this way, even though the distance between them is small, two time-series may present an opposite pattern.

(2) Max lag. This argument represents the maximum delay accepted to compare one time series to another. If it is not set, this parameter will make sure that the first value of a time series matches with the last value of another time series. Then, the distance calculated may omit many time points, and the distance calculated cannot reflect the true dissimilarity of the two time series.

After obtaining the distances of any two time-series, the hierarchical clustering method can assemble all of the series to some clusters by comparing the close distance of these series. However, all of the elements in each cluster may have different correlation coefficients and difference lag. They need to be separated to obtain a better result in which all of the time series in a cluster have the same correlation coefficient and same lag.

3.3 Separating Clusters by Parameters

Figure 2 presents how these parameters impact cluster result. The top figure of Figure 2 shows all the time series in a cluster. The distance of any two series is 0 according to the functions (1) and (2). By watching all the series, it is not easy to understand the relation between any two time-series; therefore, it is necessary to separate these series by parameters.

The value of the correlation coefficient varies from -1 to 1. In this case, a time series T_1 is selected so that its correlation coefficient is positive. For any other time series in the same cluster, if it has a positive correlation coefficient comparing to T_1, then its correlation coefficient is positive; otherwise it will be negative.

The time series are first separated by the correlation coefficient. If series 1 and 3 are positive, then series 2 is negative. Therefore, the pattern of series 2 is opposite to that of series 1 and 3. Then a separation is done by maximum lag in the middle left figure of Figure 1. The bottom figures of Figure 1 show the results. If a lag of series whose peak occurs in the time point 3 is 0, then lag of series whose peak occurs in the time point 4 is 1. In this way, the relation between series 1, 2 and 3 in the cluster is obtained by cross correlation distance and the hierarchical method.

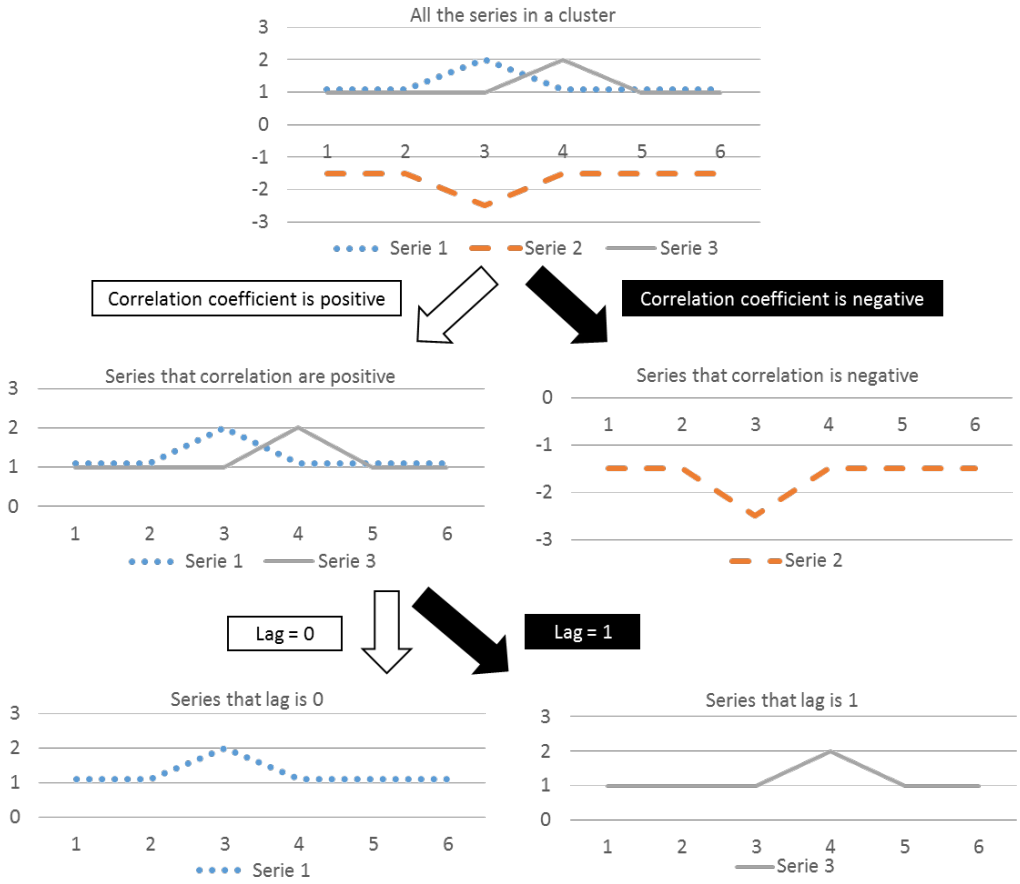


Figure 2: Separating by parameters in a cluster

3.4 Implementation

Figure 3 shows the implementation. It contains 3 main steps:

Step 1 First, in some cases, a pre-treatment is needed to deal with the original database. For example, series whose values are all 0 or "NA" are removed. However, giving all of the values a scale is not necessary; or, some values that are treated are not original. Then, the cross correlation distance is computed to calculate the dissimilarity of any two time series.

Step 2 Compute hierarchical clustering method. At the end of this step, the clusters in which the correlation coefficient and lag are not separated are obtained.

Step 3 Separate the clusters using the cross correlation parameter (correlation coefficient and lag). The most important part is in the right rounded corner rectangle:

Step 3.1 Firstly, a function "Find_Base_Ts" is applied to all of the series in a certain cluster. This function will return a base time series whose correlation is positive and lag is 0.

Step 3.2 Based on this time series, another function "Find_Max_CCF" is applied. This function will return correlation coefficients and lags relating to the base time series of all the other time series in a cluster. With the correlation coefficient and lag, a minimum cross correlation distance between the base time series and a given series in the cluster can also be obtained. For the correlation coefficient, it is used to segment time series with positive or negative coefficients; the symbol (positive or negative) is enough in that case. Therefore, the value -1 and +1 represent the positive and negative correlations.

Finally, three values are obtained for a time series: (1) Cluster, in which this time series has the best correlation with the other time series in the same cluster. (2) Correlation, the symbol of the base series. (3) Lag, the best one with which a minimum cross correlation distance can be obtained.

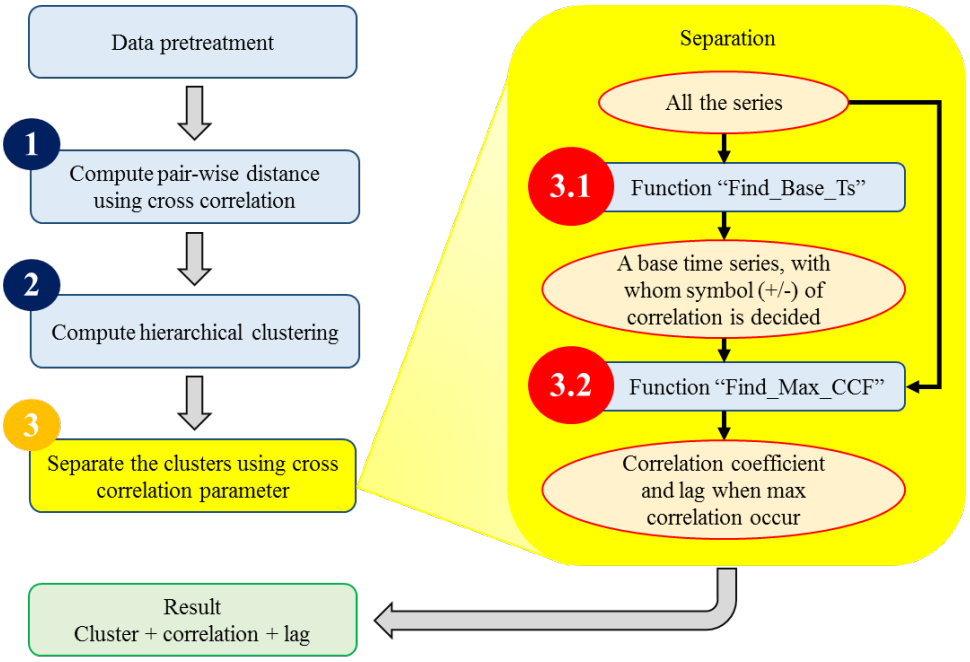


Figure 3: Algorithm implement

4 COMPARISON BETWEEN CROSS CORRELATION AND DYNAMIC TIME WARPING DISTANCE

The state of the art shows that dynamic time warping is also a pertinent tool to calculate the distance between time series. It is interesting to compare the segmentation results through a dynamic time warping and cross correlation method. Therefore, the advantages of cross correlation for time series segmentation will be shown in this section.

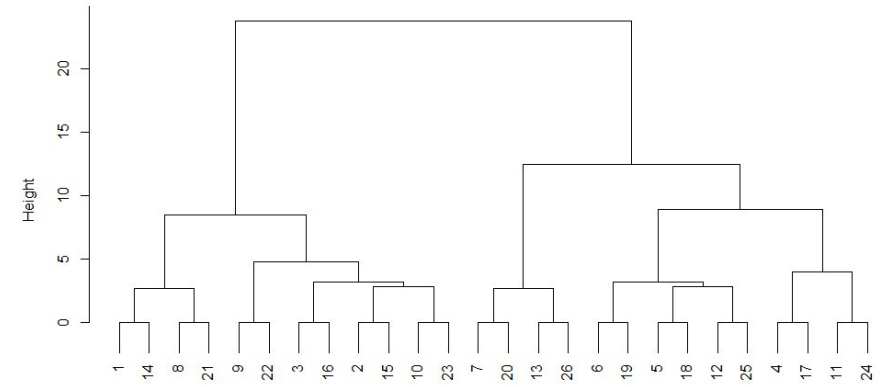
In section 3, a method using cross correlation distance to calculate distances between time series has been designed. Through the algorithm presented in Figure 3, time series can be segmented by cross correlation. However, another method, dynamic time warping, can also be used to measure the dissimilarity of a time series distance. Therefore, it is necessary to compare these methods.

To this end, the following simple example has been designed.

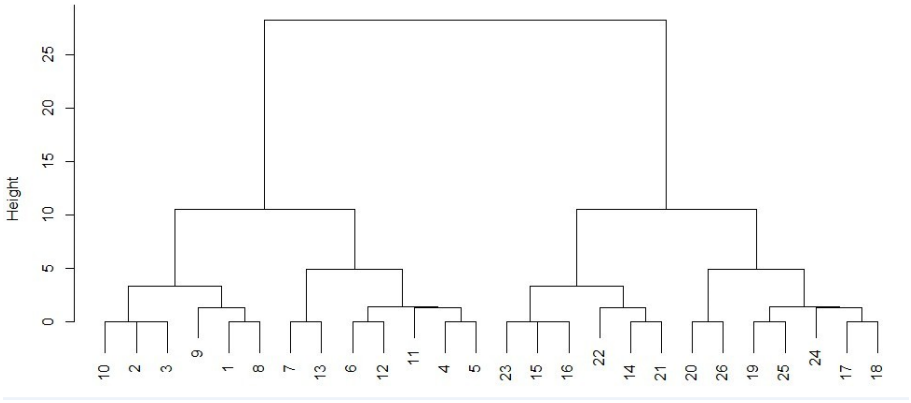
In this part, a sample, which contains 26 time series, is tested. All of the values in these time series are 0 or 1, as shown in Table 1. In this table, the series 1 – 13 is the opposite of the series 14 -26. “Opposite” here means that if a value of a certain time point is 0 in a series, then the value in the same time point in another series is 1.

Table 1: 0 - 1 sample data

	V1	V2	V3	V4	V5	V6	V7
1	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0
5	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	1
8	1	1	0	0	0	0	0
9	1	0	1	0	0	0	0
10	0	1	1	0	0	0	0
11	1	0	0	0	1	0	0
12	0	0	0	0	1	1	0
13	0	0	0	0	0	1	1
14	0	1	1	1	1	1	1
15	1	0	1	1	1	1	1
16	1	1	0	1	1	1	1
17	1	1	1	0	1	1	1
18	1	1	1	1	0	1	1
19	1	1	1	1	1	0	1
20	1	1	1	1	1	1	0
21	0	0	1	1	1	1	1
22	0	1	0	1	1	1	1
23	1	0	0	1	1	1	1
24	0	1	1	1	0	1	1
25	1	1	1	1	0	0	1
26	1	1	1	1	1	0	0



(a)



(b)

Figure 4: Hierarchical clustering dendrogram
(a) by cross correlation distance (max lag = 1)
(b) by dynamic time warping (window = 1)

Figure 4(a) is the dendrogram of hierarchical clustering by cross correlation distance, in which lag is 2. All of the series are cut into 5 clusters as shown in Table 2 (in the column "lag =2"). Each result consists of a number, a plus or minus sign, and another number. The first number means the cluster, and the sign means whether the correlation coefficient is positive, and the second number. For example, for the time series 22 in the column "lag =2" of Table 2, the result of a cross correlation with lag of 2 is the cluster 1, with a negative correlation coefficient. Thus, it is presented as "1 (cluster) – (negative) 2 (lag)".

Figure 4(b) is the dendrogram of hierarchical clustering by dynamic time warping distance, in which the parameter window is 2. All of the series are cut into 6 clusters, as shown in Table 2 (in the column "window =2").

In table 3, both cross correlation distance and dynamic time warping are considered. For cross correlation distance, the parameter lag varies from 1 to 2, and for the dynamic time warping, the parameter window varies from 1 to 2. The test of calibration helps to understand the sensitivity of each parameter and metric.

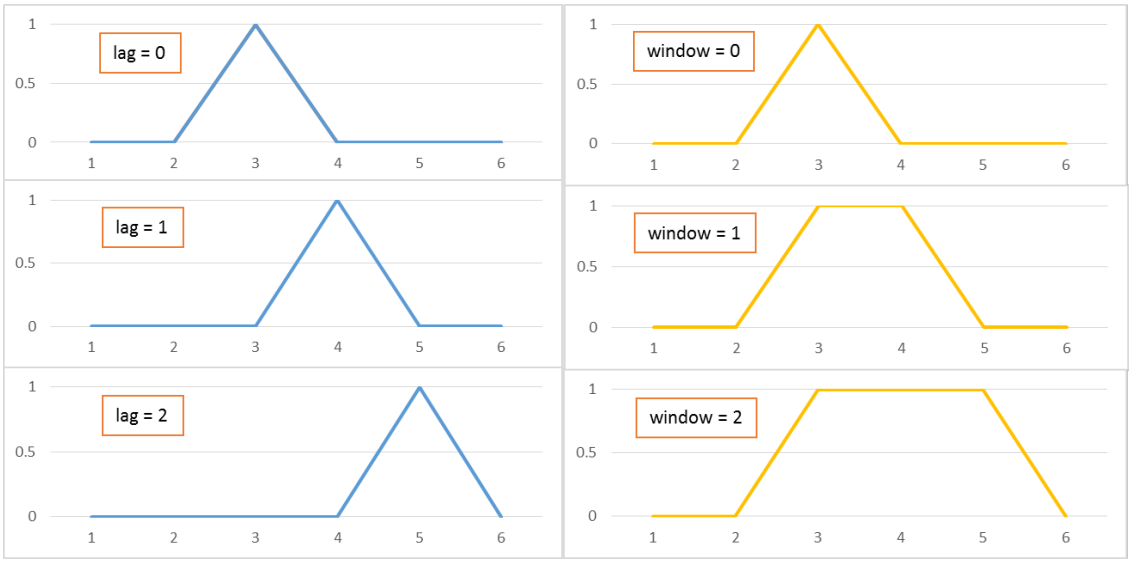
Table 2 : Sample result of cross correlation distance and dynamic time warping

								max lag = 1	max lag = 2	window = 1	window = 2
1	1	0	0	0	0	0	0	1+0	1+0	1	1
2	0	1	0	0	0	0	0	2+0	2+0	2	2
3	0	0	1	0	0	0	0	2+1	2+1	2	2
4	0	0	0	1	0	0	0	3+0	3+0	3	2
5	0	0	0	0	1	0	0	5+0	5+0	3	3
6	0	0	0	0	0	1	0	5+1	5+1	3	3
7	0	0	0	0	0	0	1	4+1	4+1	4	4
8	1	1	0	0	0	0	0	1+1	1+1	1	1
9	1	0	1	0	0	0	0	2+1	1+2	1	1
10	0	1	1	0	0	0	0	2+0	2+0	2	2
11	1	0	0	0	1	0	0	3+1	3+1	3	1
12	0	0	0	0	1	1	0	5+0	5+0	3	3
13	0	0	0	0	0	1	1	4+0	4+0	4	4
14	0	1	1	1	1	1	1	1-0	1-0	5	5
15	1	0	1	1	1	1	1	2-0	2-0	6	6
16	1	1	0	1	1	1	1	2-1	2-1	6	6
17	1	1	1	0	1	1	1	3-0	3-0	7	6
18	1	1	1	1	0	1	1	5-0	5-0	7	7
19	1	1	1	1	1	0	1	5-1	5-1	7	7
20	1	1	1	1	1	1	0	4-1	4-1	8	8
21	0	0	1	1	1	1	1	1-1	1-1	5	5
22	0	1	0	1	1	1	1	2-1	1-2	5	5
23	1	0	0	1	1	1	1	2-0	2-0	6	6
24	0	1	1	1	0	1	1	3-1	3-1	7	5
25	1	1	1	1	0	0	1	5-0	5-0	7	7
26	1	1	1	1	1	0	0	4-0	4-0	8	8

The same 0 – 1 sample data is used for the comparison. To better test the data, it is important to choose pertinent parameters for each method. Figure 5 illustrates two parameters for each of the methods (Giorgino, 2009) .

For cross correlation distance, the parameter max lag is a maximum delay that a time series can be shifted. The max lag 1 and 2 is used to test 0 – 1 sample data. When shifting time series, it has a risk of omitting the first and last value of the two time series.

For dynamic time warping, the parameter window represents the maximum delay that a time series can be warped. The windows 1 and 2 are used to test 0 – 1 sample data.



(a) (b)

Figure 5: Calibration for time series distance calculate method – (a) lag for cross correlation, (b) window for time warping

4.1 Comparison Between Cross Correlation and Time Warping when Maximum Lag and Window are 1

In Table 2, even for the same time series when using a different method and parameter, the result can be different. For example, for the time series 9, when using cross correlation distance, if the max lag changes, the result will be different. Then, it is of interest to discover the difference with the different parameter and different method.

Based on Table 2, given a method (CCD or DTW), the size of a given cluster can be known, and the intersection size of two methods or parameters can also be known. For example, for the time series in the cluster 1+ of cross correlation distance (with max lag = 1; to match the results between cross correlation and dynamic time warping, the lag in the cross correlation is omitted. Therefore, “1+” is discussed here, instead of “1+1” or “1+0”. They are all to be seen as “1+”), there are two time series that correspond in cluster 1 of dynamic time warping (with window =1). In Table 2, these two time series are series 1 and 8. Table 3 is built with this information.

In Table 3, the horizontal axis is the size of each group of dynamic time warping. The parameter window is 1. The vertical axis is the size of each group by cross correlation. The parameter lag is also 1.

Almost the same result can be obtained by using these two methods. Except for group 3 and group 7 of dynamic time warping, these groups can be divided into two groups when using cross correlation. Groups 1 and 5 of time warping have minor differences with cross correlation. The other groups containing two methods are all the same. Moreover, the group sizes that are given by time warping contain large numbers (in the Table 3); cross correlation divides these groups into smaller numbers, so that all of the group sizes are more uniform.

In conclusion, when parameters are small enough, the results of dynamic time warping and cross correlation are almost the same, except for cross correlation, which has finer segmentation.

Table 3: Comparing cross correlation (max lag = 1) and time warping (window =1)

Sum of group		Time warping								
		1	2	3	4	5	6	7	8	Total
Cross correlation	1+	2								3
	2+	1	3							3
	3+			2						2
	4+				2					2
	1-					2				3
	2-					1	3			3
	3-							2		2
	4-								2	2
	5+			3						3
	5-							3		3
	Total	3	3	5	2	3	3	5	2	26

* 1+ represent the group number 1 of cross correlation with positive correlation coefficient.

4.2 Comparison Between Cross Correlation (max lag = 1 and max lag = 2)

Table 4 shows another result of the comparison. Both axes show the size of each group by cross correlation. The difference is that the horizontal axis shows the result of lag equalling 2. The vertical axis demonstrates the result of lag, equalling 1.

The results are almost the same. Moreover, the augmentation of max lag can lead to a more comparable size. This means that even though augmentation of the max lag will more significantly shift the time series, the best correlation coefficient should be matched when the max lag is equal to 1. Therefore, the max lag of 2 not only maintains good results that do not need to significantly shift time series, but it also makes the group size similar.

In conclusion, the result of a bigger lag has a minor change compared to the result of the smaller lag. This test shows that this parameter has a minor impact when it is increased. This means that it is easier to calibrate the algorithm when selecting cross correlation.

Table 4: Comparing cross correlation (max lag = 1 and max lag = 2)

size of group		Cross correlation (max lag = 2)										
		1+	2+	3+	4+	5+	1-	2-	3-	4-	5-	Total
Cross correlation (max lag = 1)	1+	2										2
	2+		3									3
	3+			2								2
	4+				2							2
	5+				1	3						4
	1-						2					2
	2-							3				3
	3-								2			2
	4-									2		2
	5-									1	3	4
	Total	2	3	2	3	3	2	3	2	3	3	26

4.3 Comparison Between Time Warping (window = 1 and window = 2)

Table 5 shows another result of the comparison. Both axes show the size of each group through dynamic time warping. The difference is that the horizontal axis shows the result of the window equal to 2. However, the vertical axis demonstrates the result of the window equal to 1.

Table 5 illustrates the results of the comparison. Unlike the comparison between cross correlation, almost all of the groups have been changed if the parameter window has been changed. This means that time warping is more sensitive when changing parameters. Even though a big change in this case can make a 50% change in the group (4 groups out of 8 have been changed in Table 3).

In conclusion, the parameter value augmentation of dynamic time warping has a more negative impact than the parameter value augmentation.

Table 5: Comparing time warping (window = 1 and window = 2)

Size of group		Time warping (window = 1)								
		1	2	3	4	5	6	7	8	Total
Time warping (window = 2)	1	3		1						4
	2		3	1						4
	3			3						3
	4				2					2
	5					3		1		4
	6						3	1		4
	7							3		3
	8								2	2
	Total	3	3	5	2	3	3	5	2	26

4.4 Comparison Between Cross Correlation (max lag = 2) and Time Warping (window = 1)

For a simple conclusion of 1.1 – 1.3, Figure 6 shows max lag 2 for cross correlation and window 1 for time warping. In this part, these two parameters are compared through two methods.

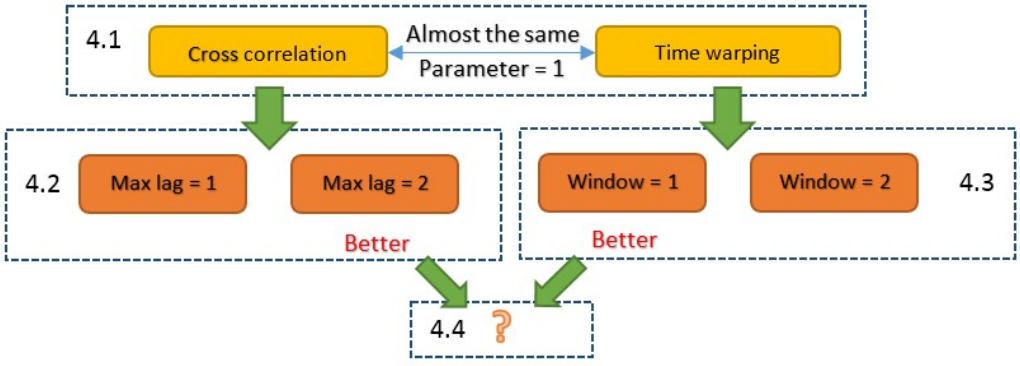


Figure 6: Comparison process

Table 6 shows the results of the comparison. It shows that if the parameter is carefully chosen for each of the methods, the result through two methods will be nearly the same. However, compared to time warping, the cross correlation is easier to calibrate:

In cross correlation, when a parameter max lag that is not too small is used, then the result can be the same as when a smaller parameter is set. This is also the result of time warping (if the parameter of time warping is small enough).

For time warping, when a parameter window is chosen, the result can be different than the case in which a smaller parameter is chosen. It is difficult to determine the window value. In fact, if a smaller value in real is chosen (for example, 1), some superior results may only be obtained with a window equal to 2. In this way, the best result will be lost.

Table 6: Comparing a cross correlation (max lag = 2) and time warping (window = 1)

Sum of group		Time warping (window = 1)								
		1	2	3	4	5	6	7	8	Total
Cross correlation (max lag = 2)	1 +	3								3
	2 +		3							3
	3 +			2						2
	4 +				2					2
	1 -					3				3
	2 -						3			3
	3 -							2		2
	4 -								2	2
	5 +			3						3
	5 -							3		3
	Total	3	3	5	2	3	3	5	2	26

4.5 Synthesis

After comparing the application of the cross correlation and time warping in 0 – 1 sample data, the cross correlation is determined to be better because of the following reasons:

(1) Cross correlation is easier to calibrate. As presented in Table 4, when using a cross correlation distance, the choice of parameters (lag) has a minor impact on the result, and almost the same result can be obtained when using 1 or 2 as the lag. However, as presented in Table 5, the choice of parameters (window) of dynamic time warping distance has a lager impact on the result than cross correlation distance. Therefore, the cross correlation is easier to calibrate.
(2) The result of cross correlation contains information on the correlation coefficient and lag. As presented from Table 3 to Table 6, besides the parameter “lag”, for each result of cross correlation, there is another factor “correlation coefficient” (positive or negative). That means the number of groups can be adjusted depending on our need. If a small number of groups is needed, the correlation coefficient can be combined to obtain a smaller number of groups. For example, group 1+ and 1- can be combined into group 1 if fewer groups are needed. On the other hand, separating the positive and negative correlation coefficient will give us more groups. However, the dynamic time warping distance has only one choice in the group number.
(3) Group size is more similar and better defined when using cross correlation. Table 6 illustrates the group size of each method. For cross correlation distance, all of the group sizes are 2 or 3. However, for a dynamic time warping, the group size is from 2 to 5. Therefore, the grouping of the cross correlation is more even than dynamic time warping.

5 APPLICATION FOR A TRANSPORTATION ANALYSIS OF SMARTCARD USERS’ PATTERNS

In this section, a real case and its results through the method developed in this paper is presented to show how this method can perform.

5.1 Presentation of the real problem

The dataset has been provided by the *Société de Transport de l'Outatouais* (STO), a transit authority serving 280,000 inhabitants in Gatineau, Quebec. The STO authority is a Canadian leader in user transit using smart card fare collection (Morency and al., 2007).

Table 7 demonstrates an excerpt of the raw smart card dataset; it contains a variable of a user's trip information. Apart from the card identification (which has been made anonymous), there is the ticket code (fare categories), the date and the time of the transaction, the line (route) number and the direction. All transactions are made on a bus network; the location of the transaction is also available (He and al., 2017).

Table 7. Excerpts of the raw smart card dataset (He and al., 2017,)

Card id	Ticket type	Date	Time	Line	Direction	Weekday	Stop id
1150629967111800	140	2013-09-03	65232	44	Sud	2	1140
1273590714804090	110	2014-09-02	71909	224	Sud	1	2801
1273590714804090	110	2014-09-02	154607	224	Nord	1	2610

The objective is to cluster card users' daily transactions into several groups, in which the boarding time of day of a user is like that of another user. For example, the boarding time of all of the transactions of user 1 on day 1 are the same as the boarding times of all the transactions of user 2 on day 2, but are not like the boarding times of all of the transactions of user 1 in day 2. Then, day 1 of user 1 and day 2 of user 2 will be in the same group, but day 2 of user 1 will be in another group.

5.2 Results

The method used transforms the smart card database into a 0 – 1 table (like Table 8) first, in which every line is a user's daily profile ("card id_date" combination), and every column is a time period, for example, the second column "05_30" means the period from 05:30 to 05:59. In the table, "1" represents that a transaction has happened in this time period. For example, for the user whose card id is 1150312817303160, in 2013-09-03, they have a transaction in the time period 05:30 – 05:59.

Table 8. Example dataset of users-day (0-1 table)

Combination	05_30	06_00	06_10	06_20	06_30	06_40	…
1150296033731200_2013-09-04	0	0	0	1	0	0	…
1150312817303160_2013-09-03	1	0	0	0	0	0	…
1150320729466490_2013-09-03	0	0	0	0	0	0	…

With Table 8, the distance of every two combinations is calculated by using the cross correlation distance. Both cross correlation and dynamic time warping are tested. For the cross correlation distance, the parameter "lag" is 2. For the dynamic time warping distance, the parameter "window" is 2.

Then, the cross correlation and dynamic time warping are computed and a distance matrix of any two combinations is obtained for each method. With that distance matrix, the combination ("user-date") is computed by using hierarchical clustering.

By observing the dendrogram, 11 groups and 6 clusters are cut for cross correlation and dynamic time warping. Finally, the sum of transactions for each cluster is calculated; this result is shown in Figure 7 and Figure 8.

In fact, there are 11 groups for cross correlation, but to better present the result, only 6 clusters are chosen in Figure 7 (the other groups do not impact the advantage of comparing to the result of

dynamic time warping). This figure shows how users' daily profiles can be separated. For example, for cluster 3, the users usually have a boarding transaction between 07:25 and 07:40 in the morning, and another boarding transaction between 17:00 – 17:30 in the afternoon. For cluster 6, the users usually have a boarding transaction between 05:30 and 06:20 in the morning, and another boarding transaction between 14:30 – 15:45 in the afternoon, and so on. As expected, the method developed can clearly separate time series into clusters.

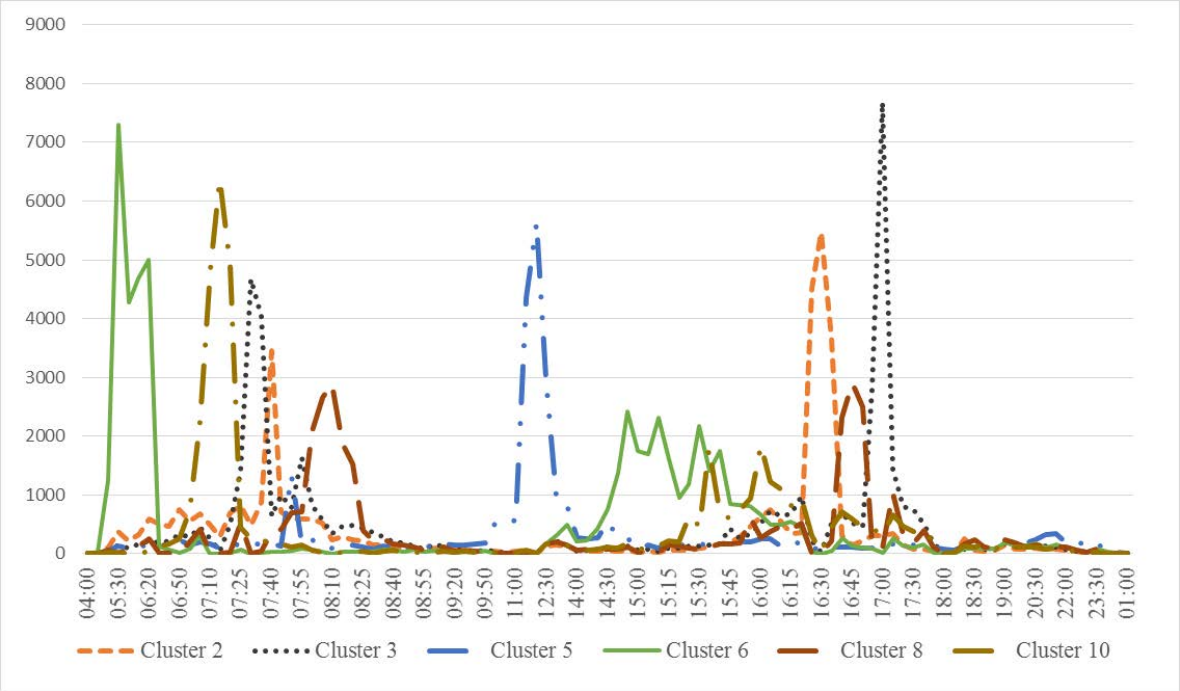


Figure 7: Sum of transaction time of each group (cross correlation)

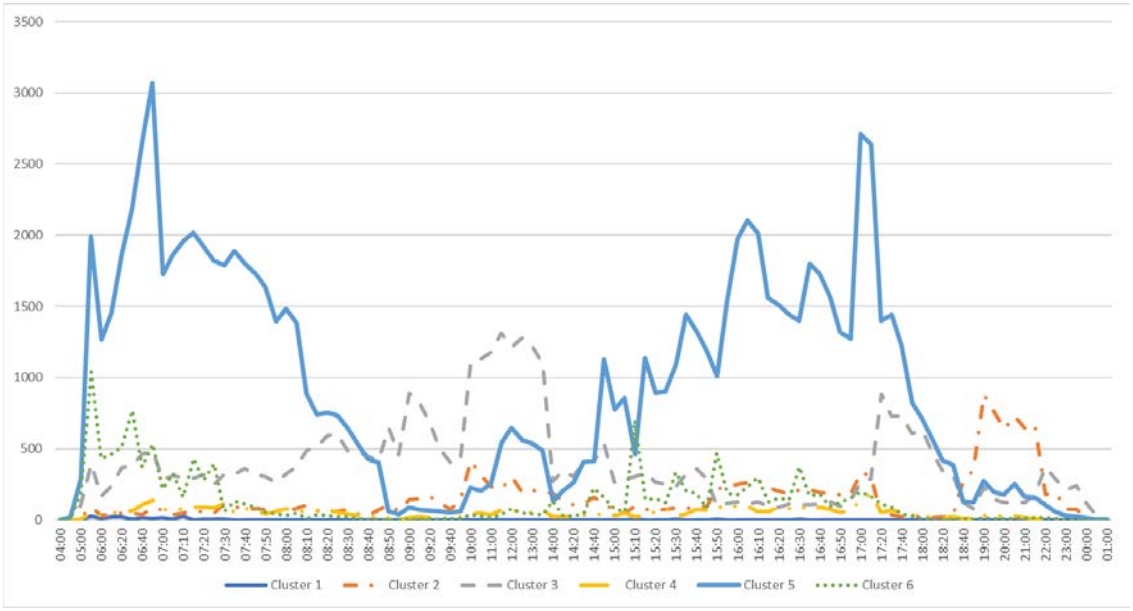


Figure 8: Sum of transaction time of each group (dynamic time warping)

Figure 8 shows the segmentation results by using dynamic time warping. By comparing Figure 7 and Figure 8, the dynamic time warping is not effective in our case. Firstly, the size of cluster 5 is

so large that cluster 5 contains most of the transaction profiles, which can lead to an uneven size between all of the clusters. Secondly, comparing cluster 5 and cluster 6 in Figure 8, even though the size is different, the "peak hours" of these two clusters are almost the same. This means the users who have different behaviors cannot be separated by using dynamic time warping. Therefore, by comparing dynamic time warping, the cross correlation is much more effective in segmenting smart card users' transaction time behaviors,

6 CONCLUSION

6.1 Contribution

An analysis of smart card users' daily profiles needs a method that will segment time series. Because of the limitations of Euclidean distance, which do not work well on time series segmentation, a method has been designed by combining cross correlation distance and hierarchical clustering. After being implemented by R, this algorithm is compared to dynamic time warping distance; the comparison result shows that cross correlation works best if it is well calibrated. Data from a middle-sized public transit association is tested by using the method developed. The clear separation of card users' daily transaction times shows the feasibility of this method.

6.2 Limitations

With regards to this methodology, there are three limitations. The first is calculating time. The main time cost is calculated with the cross correlation distance by trying different parameters (lag for cross correlation and windows for dynamic time warping) for each parameter. For 1000 vectors, the time to calculate the distance is about 10 minutes. The second limit is the choice of metrics when dealing with transportation issues. In this case, the cross correlation is suitable because the delay of a smart card user's transaction time is like the parameter "lag" in the cross correlation distance. However, when dealing with other time series in transportation problems, another kind of distance that may be better for that case may need to be developed. Besides, with large sets of data, the pairwise distance matrix may be impossible to compute.

6.3 Perspectives

According to this limitation, there are two perspectives on the problem. First, regarding the calculation time, a new algorithm could be tested in order to avoid certain calculations in which the calculation of the cross correlation distance between certain vectors is cancelled out by assuming that the distance of these two vectors is too large, and it is impossible to group them into the same cluster. Secondly, regarding the distance metric, other distances that are not based on the Euclidean distance could be tested to match the different cases in transportation. For example, the Fourier transformation distance based on the analysis of fluctuation could be tested to explain the fluctuations in transactions, etc. Overall, the objective is to find the best metric for a specific transportation issue.

7 ACKNOWLEDGMENTS

The authors wish to acknowledge the support of the *Société de transport de l'Outaouais (STO*) for providing data, the Thales group and the National Science and Engineering Research Council of Canada (NSERC project RDCPJ 446107-12) for funding.

8 REFERENCES

Agard, B., Morency, C., & Trépanier, M. (2006). Mining public transport user behaviour from smart card data. *IFAC Proceedings Volumes*, *39*(3), 399-404.

Bakar, Z. A., Mohemad, R., Ahmad, A., & Deris, M. M. (2006, June). A comparative study for outlier detection techniques in data mining. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on* (pp. 1-6). IEEE.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71). Springer Berlin Heidelberg.

Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In *KDD workshop* (Vol. 10, No. 16, pp. 359-370).

Brockwell, P. J., & Davis, R. A. (2016). *Introduction to time series and forecasting*. springer.

Chen, C. F., Chang, Y. H., & Chang, Y. W. (2009). Seasonal ARIMA forecasting of inbound air travel arrivals to Taiwan. *Transportmetrica*, *5*(2), 125-140.

de Oña, R., & de Oña, J. (2015). Analysis of transit quality of service through segmentation and classification tree techniques. *Transportmetrica A: Transport Science*, *11*(5), 365-387.

Deza, M. M., & Deza, E. (2009). Encyclopedia of distances. In *Encyclopedia of Distances* (pp. 1-583). Springer Berlin Heidelberg.

Ghaemi, M. S., Agard, B., Nia, V. P., & Trépanier, M. (2015). Challenges in Spatial-Temporal Data Analysis Targeting Public TransportÕ. *IFAC-PapersOnLine*, *48*(3), 442-447.

Ghaemi, M. S., Agard, B., Trépanier, M., & Partovi Nia, V. (2016). A Visual Segmentation Method for Temporal Smart Card Data. *Transportmetrica A: Transport Science*, (just-accepted), 1-23.

Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in R: the dtw package. *Journal of statistical Software*, *31*(7), 1-24.

He, L., Trépanier, M., & Agard, B. (2017). *Evaluating the Impacts of a Bus-Rapid Transit on Users' Temporal Patterns Using Cross Correlation Distance and Sampled Hierarchical Clustering Applied to Smart Card Data* (No. 17-03711).

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

Joh, C. H., Timmermans, H. J. P., & Arentze, T. A. (2006). Measuring and predicting adaptation behavior in multidimensional activity-travel patterns. *Transportmetrica*, *2*(2), 153-173.

Langfelder, P., Zhang, B., & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: The Dynamic Tree Cut package for R. *Bioinformatics*, *24*(5), 719-720.

Lhermitte, S., Verbesselt, J., Verstraeten, W. W., & Coppin, P. (2011). A comparison of time series similarity measures for classification and change detection of ecosystem dynamics. *Remote Sensing of Environment*, *115*(12), 3129-3152.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, *38*(11), 1857-1874.

Meyer, D., Buchta, C., & Meyer, M. D. (2017). Package 'proxy'.

Morency, C., Trepanier, M., & Agard, B. (2007). Measuring transit use variability with smart-card data. *Transport Policy*, *14*(3), 193-203.

Mori, U., Mendiburu, A., & Lozano, J. A. (2016). Distance Measures for Time Series in R: The TSdist Package. R JOURNAL, 8(2), 451-459.

Nishiuchi, H., King, J., & Todoroki, T. (2013). Spatial-temporal daily frequent trip pattern of public transport passengers using smart card data. *International Journal of Intelligent Transportation Systems Research*, *11*(1), 1-10.

Nuzzolo, A., & Comi, A. (2016). Advanced public transport and intelligent transport systems: new modelling challenges. *Transportmetrica A: Transport Science*, *12*(8), 674-699.

Pelletier, M. P., Trépanier, M., & Morency, C. (2011). Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, *19*(4), 557-568.

Rokach, L., & Maimon, O. (2005). Clustering methods. In *Data mining and knowledge discovery handbook* (pp. 321-352). Springer US.

Sedgwick, P. (2012). Pearson's correlation coefficient. *Bmj*, *345*(7).

Subbiah, K. (2011). *Partitioning Methods in Data Mining*.