Spatio-temporal data streaming with affinity propagation

Nasrin E. Ivari[a]*, Monica Wachowicz[a] Tamara Agnew[b], Patricia A.H. Williams[b]

[a] Department of Geomatics, University of New Brunswick, Fredericton, Canada
[b] Digital Health Design Lab, Flinders University, Australia
* nasrin.eshraghi@unb.ca

Keywords: affinity propagation, landmark time window, e-counter data, Internet of Things

Introduction

Spatio-temporal data stream clustering is a growing research field due to the vast amount of continuous georeferenced data streams being generated by IoT devices. Carnein and Trautmann (2019) provide an extensive review on stream clustering algorithms, outlining an overall strategy that is based on a two-phase clustering approach; having an online phase which uses a time window model to capture the data streams and then computing micro-clusters (i.e. preliminary clusters within each time window). The second phase is carried out offline as the micro-clusters are re-clustered to generate the macro-clusters after the entire stream data is processed. Distance-based algorithms such as CluStream using the pyramidal time window model (Aggarwal et al. 2003) and DenStream using the damped time window model (Cao et al. 2006), are widespread approaches used in stream clustering.

Very few attempts have been found in the literature in clustering data streams using the Affinity Propagation (AP) approach proposed by Dueck and Frey (2007). Zhang et al. (2008) introduced the STRAP algorithm as an extended AP using sliding time windows for clustering text data streams. By using the same data streams, but integrating affinity propagation with a decay density method using pyramid time window model, Zhang et al. (2013) developed the APDenStream algorithm. Recently, Sui et al. (2018) proposed the ISTRAP algorithm, which detects the evolution of micro-clusters in imagery stream data focusing on their emergence, disappearance and re-occurrence.

This research work proposes a novel DSAP (Data Stream Affinity Propagation) algorithm using the landmark time window model for clustering people counting data streams obtained from an experiment deployed in an indoor space. To the best of our knowledge, streaming AP algorithms have not yet been applied for clustering e-counter data streams. We demonstrate the potential of DSAP for understanding the effect of an intervention campaign in motivating the usage behaviour of stairways in a building.

Method

Data streams are a continuous infinite sequence of data points where each data point contains sensor measurements, their location and a timestamp. The landmark time window model separates the data points based on a set time interval (e.g. 1 hour) or an event (e.g. every 10 steps), where after a landmark is reached, new data points start to be captured using the next time window. The temporal resolution of a time window is usually a-priori defined based on the application requirements. In the landmark time window, each data point is of equal importance in the computation of the clusters.

The proposed DSAP algorithm takes as an input the similarities between data points. It aims to identify exemplars among data points and generate clusters around these exemplars. It is considered as a k-center algorithm because it randomly separates some data points as representatives of clusters, known as cluster heads. The objective of DSAP is to maximize the sum of similarities between the data points and their cluster heads. The algorithm computes four matrices. The similarity matrix estimates the similarity between each data point and finds the most suitable cluster head. The responsibility matrix contains values that correspond to how likely is one data point to be a member of a cluster. The availability matrix defines how available one data point is to be an exemplar. Finally, in the criterion matrix, the sum of the availability matrix and responsibility matrix is calculated.

Four main hyperparameters are used to improve the performance of the proposed DSAP algorithm: *1) preferences* for each data point that is more likely to be chosen as an exemplar. *2) convergence_iter* is the number of iterations with no change in the number of estimated clusters that stop the convergence. *3) damping factor* (between 0.5 and 1) is the extent to which the current data point is maintained relative to incoming data points (weighted 1 = damping); and *4) max_iter* defines the maximum number of iterations.

For the online phase, the data streams of an IoT device arrive as a continuous sequence of data points that are accumulated using the landmark time window model. Each time window has the same timeframe. The micro-clusters are computed for every time window until the data stream ends. This phase is important since the timeframe and temporal resolution of the time windows have a direct impact on the computation of the cluster heads of the micro-clusters, and therefore, play an important role in finding the macro-clusters in the next phase. After the streaming has ended and all the cluster heads of the micro-clusters have been computed, macro-clusters are generated by re-clustering these cluster heads. The DSAP algorithm is again used to compute the final macro-clusters and their respective cluster heads.

Discussion of the Results

The proposed DSAP algorithm was implemented in Python 2.7 programming language, using the free machine learning library scikit-learn. The snippet below illustrates the pseudocode of the implemented DSAP algorithm for a timeframe of 60 minutes. An experiment was conducted by deploying e-counter sensors at six different stairways located at the Tonsley building at the Flinders University, Australia. These e-counters consist of an infrared transmitter and a receiver which are triggered when the signal is interrupted by a person passing in between the two. The event-triggered data streams consisted of data points having attributes such as <*stairway location, timestamp, number of people being counted, sensor id*>. The experiment was conducted over three months from March 18th to June 23rd 2019. However, the Easter break has taken place during the experiment, and therefore, the data streams collected during these days have not been included in the clustering analysis.

Algorithm 1 DSAP Algorithm

Data: Data Points: $E = (E_1, E_2, ..., E_n)$ for computing micro clusters;

Require hyper_parameters: preference, damping, max_iter, convergence_iter

Initialize: Landmark time window (size T_s = 60 minutes)

Similarity Matrix: S ∀ i, k: s(i, k) = 0

Availability Matrix: A ∀ i, k: a(i, k) = 0

Responsibility Matrix: R ∀ i, k: r(i, k) = 0

Function `Affinity_Propagation` (*Data_points*) **:**

$S \; \forall \; i, k: s(i,k) = -||x_i - x_k||^2$

while *r(i,k) and a(i,k)* $\neq$ *convergence* **do**

Updating R:

$r(i, k) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \{ a(i, k') + s(i, k') \}$

Updating A:

$(i, k) \leftarrow \min\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\{0, r(i', k)\}$

non-diagonal A:

$a(i, k) \leftarrow \sum_{i' \neq k} \max(0, r(i', k))$

for *7a.m.* $\leq T_s \geq$ *7p.m.* **do**

Function `Affinity_Propagation` (E) **:**

Result: Set of cluster heads for computing macro clusters:

$P = (P_1, P_2, ..., P_n)$

Function `Affinity_Propagation` (P) **:**

Result: Macro Clusters:

$C = (C_1, C_2, ..., C_n)$

Figure 1 provides an overview of the accumulated hourly counting of people in the building. The first month was used to generate a baseline, before the motivation campaign has taken place to encourage stair usage in the second month. The third month collected the e-counter data streams to determine whether the campaign has had a positive impact on people's behaviour.

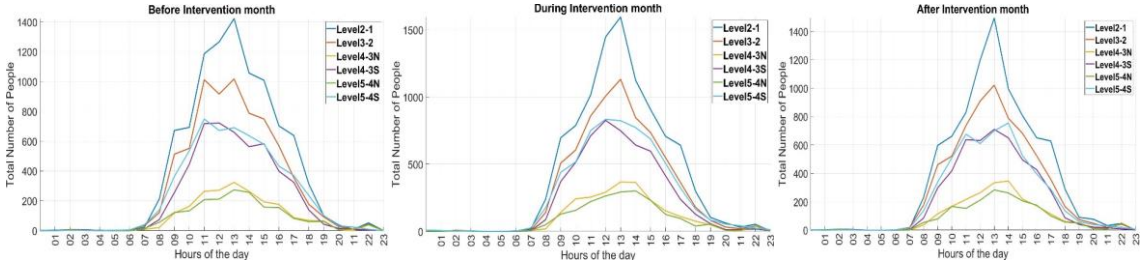

Figure 1: Accumulated hourly data distributions for different stairways.

During the online phase, the DSAP algorithm generated an average of four micro-clusters for each time window with a one-hour timeframe for a period from 7am until 7pm. Each time window contained a different number of triggered events that recorded how many people were using a specific stairway. The cluster heads of the micro-clusters were computed in each of these windows, and stored once the landmark time window passed over all the data points. They reveal that a large number of people have used the level 2-1, level 4-3 south, and level 5-4 south stairs. This can be attributed to the proximity of many classrooms to these stairways. After all the cluster heads were computed, the DSAP algorithm generated the macro-clusters as shown in Figure 2. The macro-clusters ultimately represent usage patterns that can be interpreted as regular and irregular behaviour. For example, the macro-clusters containing a few number of people taking the stairways were predominantly located at the lower levels of the building during the whole experiment. In contrast, irregular behaviour has occurred at level 3 of the building during the intervention month. Therefore, we can assume that the intervention campaign has had a positive impact on the people's motivation to use the

stairways. But the gradual decrease in the number of the people found in the macro-clusters after the intervention has also indicated that one month period might not be sufficient to bring about long term behavioural changes.

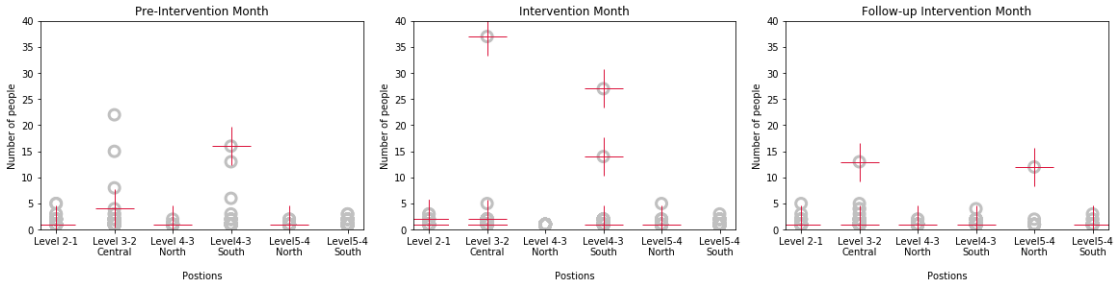

Figure2: Micro-cluster heads (grey circles) and their respective macro clusters heads (red crosses) before, during and after the intervention campaign.

Conclusions

In this paper we propose DSAP as a new streaming AP algorithm using the landmark time window model for clustering e-counter data streams. The DSAP algorithm is a flexible and easy to apply to IoT data streams for finding spatio-temporal patterns. However, small and medium volume of data streams are needed to maintain high speed performance when computing the micro-clusters. Towards addressing this issue, we will continue to explore other time window models (e.g. pyramidal and damped time window models) coupled with the DSAP algorithm.

Acknowledgements

This research was supported by the NSERC/Cisco Industrial Research Chair [Grant IRCPJ 488403-14].

References

Aggarwal, C. C., Philip, S. Y., Han, J., & Wang, J. (2003). A framework for clustering evolving data streams. In Proceedings 2003 VLDB conference (pp. 81-92). M. Kaufman.

Cao, F., Estert, M., Qian, W., & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. In Proceedings of the 2006 SIAM international conference on data mining (pp. 328-339). Society for industrial and applied mathematics.

Carnein, M., & Trautmann, H. (2019). Optimizing data stream representation: An extensive survey on stream clustering algorithms. Business & Information Systems Engineering, 61(3), 277-297.

Dueck, D., & Frey, B. J. (2007). Non-metric affinity propagation for unsupervised image categorization. In IEEE 11th International Conference on Computer Vision (pp. 1-8).

Sui, J., Liu, Z., Jung, A., Liu, L., & Li, X. (2018). Dynamic clustering scheme for evolving data streams based on improved STRAP. IEEE Access, 6, 46157-46166.

Zhang, X., Furtlehner, C., & Sebag, M. (2008). Data streaming with affinity propagation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 628-643). Springer.

Zhang, J. P., Chen, F. C., Liu, L. X., & Li, S. M. (2013). Online stream clustering using density and affinity propagation algorithm. In 2013 IEEE 4th International Conference on Software Engineering and Service Science (pp. 828-832). IEEE.