Extracting Knowledge from Stream Behavioural Patterns

Ricardo Jesus[1], Mário Antunes[2], Diogo Gomes[1,2] and Rui Aguiar[1,2]
[1]*DETI, Universidade de Aveiro, Aveiro, Portugal*
[2]*Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal*

Keywords: Stream Mining, Machine Learning, IoT, M2M, Context Awareness.

Abstract: The increasing number of small, cheap devices full of sensing capabilities lead to an untapped source of information that can be explored to improve and optimize several systems. Yet, as this number grows it becomes increasingly difficult to manage and organize all this new information. The lack of a standard context representation scheme is one of the main difficulties in this research area (Antunes et al., 2016b). With this in mind we propose a stream characterization model which aims to provide the foundations of a new stream similarity metric. Complementing previous work on context organization, we aim to provide an automatic organizational model without enforcing specific representations.

1 INTRODUCTION

The advent of cheap devices full of sensors and networking capabilities lead to, among other things, the rate at which data is created and made available increase significantly. It happens that there is a very large amount of knowledge waiting to be harvested from these flows of data, making the need to properly conduct analysis on them of great importance. The cornerstones of this connectivity landscape are the Internet of Things (IoT) (Wortmann et al., 2015) and machine-to-machine (M2M) (Chen and Lien, 2014). Context-awareness is an intrinsic property of IoT and M2M scenarios. The data gathered by these devices has no value in its raw state, it must be analysed, interpreted and understood. Context-awareness computing plays an important role in tackling this issue (Perera et al., 2014).

As discussed in previous publications (Antunes et al., 2016b) analysing these data sources can improve efficiency, help optimize resources or even detect anomalies. The following examples illustrate the importance of context information in IoT/M2M scenarios. Fusing data from several sensors makes it possible to predict a driver's ideal parking spot (Suhr and Jung, 2014). Projects such as Pothole Patrol (Eriksson et al., 2008) and Nericell (Mohan et al., 2008) use vehicular accelerations to monitor road conditions and detect potholes. TIME (Transport Information Monitoring Environment) project (Bacon et al., 2011) combines data from mobile and fixed sensors in order to evaluate road congestion in real time.

These projects provide valuable insight about the potential of sensor data in advanced IoT/M2M scenarios. However, many of these projects follow a vertical approach. This has hindered interoperability and the realisation of even more powerful scenarios. Another important issue is the need felt for a new way to manage, store and process such diverse machine data; unconstrained, without limiting structures and with minimal human interaction. With this in mind we proposed a data organization model optimized for unstructured data (Antunes et al., 2016b; Antunes et al., 2016a) that organizes context data based on semantic and stream similarity.

In this paper we tackle the issue of propagating classification tags based on stream similarity. We propose a general method for stream characterization, that can be either used for classification or generation. The end game is to use the previously mentioned model to organize sensor streams based on their patterns and improve the efficiency of our context representation model.

In 2 we detail our context organization model. 3 will address our stream characterization model. Future work is addressed in 4 while initial results are evaluated in 5. Finally, discussion and conclusions are presented in 6.

Jesus, R., Antunes, M., Gomes, D. and Aguiar, R.
Extracting Knowledge from Stream Behavioural Patterns.
DOI: 10.5220/0006373804190423
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 419-423
ISBN: 978-989-758-245-5

2 CONTEXT ORGANIZATION MODEL

Context information is an enabler for further data analysis, potentially exploring the integration of an increasing number of information sources. Common definitions of context information (Abowd et al., 1999; Winograd, 2001; Dey, 2001) do not provide any insight about its structure. In fact, each device can share context information with a different structure. E.g. sensory and location information can be used to characterize an entity context, yet the two can have different structures. One important objective of context representation is to standardize the process of sharing and understanding context information. However, nowadays no widely accepted context representation scheme exists; instead there are several approaches to deal with context information. These can be divided into three categories: i) adopt/create a new context representation, ii) normalize the storing process through ontologies or iii) accept the diversity of context representations.

We accepted the diversity of context representation as a consequence of economic pressures, and devised a bottom-up model (Antunes et al., 2015; Antunes et al., 2016b; Antunes et al., 2016a) to organize context information without enforcing a specific representation. Our organization model is divided into four main parts, as depicted in 1.

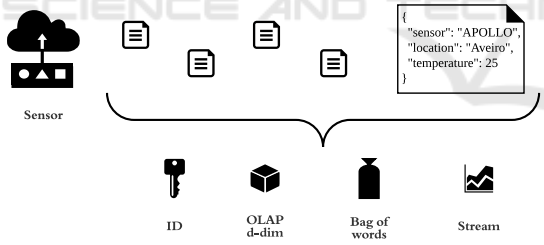

Figure 1: Context organization model based on semantic and stream similarity.

The first two parts represent the structured part of our model and account for the source ID and fixed d-dimensions respectively. These d-dimensions allow human users to select information based on time, location or even other dimensions, and can be understood as an OLAP cube helping in the process of filtering information. The remaining parts of our model extract information from the content itself and organize it based on semantic and stream similarity. Our work on semantic similarity can be found in the following publications (Antunes et al., 2016b; Antunes et al., 2016a). The first steps towards a stream similarity model are given in this paper.

3 STREAM CHARACTERIZATION BASED ON MARKOV CHAINS APPROACH

This section will address two different but related ideas. First, will present our proposed approach for stream characterization based on Markov Chains and the rationale behind it. Second, will elaborate on a stream generator which uses this previously mentioned model. Actually the characterization model was first devised entirely for the purpose of stream generation. A realistic generator help us improve the validity and repeatability of our evaluations. Despite its origins, the model has several advantages and merits of its own.

3.1 Stream Characterization

Our approach is to model a stream's behaviour by knowing how probable it is for, at a given time instant x_{i-1} with a value of y_j, a stream at the time x_i have a value of y_k. We represent this with

$$P_i(y_k|y_j)$$

meaning the probability of having some value at a time instant x_i knowing its immediate predecessor. For the remainder of this paper we will call the succession of a value to the one following it (along the x axis) a jump or transition.

Considering a perfect scenario where there is no noise nor errors, most events would thus happen in a very predictable manner (i.e. without major variances). We could then argue that using the method above and knowing all the probabilities of all the jumps along the period of the event, we could represent it with quite high confidence. For the sake of argument, consider that we had at our disposal such a probability function as expressed above, and we were given a sequence of values representing an event. We would like to compute the similarity (S) between the sequence of values and the probability function.

This can be achieved by verifying all the values of P_i for all transitions within a sequence's period, and either averaging them or using some other statistical indicator to get a representative, normalized value of the overall resulting probabilities. For example:

$$S = \frac{1}{n}\sum_{i=1}^{n} P_i$$

The probability function assigns high or low values to each jump of the sequence based on how well it relates to the events expressed by the probability function

itself. If the sequence's values were off the event's, then the overall probability would be low. On the other hand if it was high, then we could be confident that this sequence is similar to the event represented by the function.

The problem arises as we notice that this perfect scenario is not possible in practical cases, and thus if we intend to use such a function as the one described above to represent a stream, we need to make a few changes to its definition, so as to answer to the following issues:

1. Streams representing the same events more commonly than not vary widely, for several reasons. Such as noise, location, time of day, etc.;
2. It is impractical, due to time and space constraints, to have a function mapping every set of points $((x_i, y_j), (x_{i+1}, y_k))$ that might appear in a stream;
3. Along the lines of the previous item, it is not reasonable to consider the continuous and/or infinite domain associated with most events (which would imply considering infinite values).

Our proposal solves these issues by overlaying a grid-like structure over the different values a stream takes along its period, effectively turning each (x_i, y_j) in the preceding discussion into a gap (as depicted in 2). This gives rise to two other values that are now to be considered, Δx and Δy. Each representing the resolution of their corresponding axis.

Issue 1 can be solved by overlaying multiple streams representing a same event, and computing the probabilities that arise from their transitions. Issues 2 and partially 3 are solved by now considering jumps' areas instead of single values, in a sense discretizing both a stream's domain and codomain. By the law of large numbers and assuming that those streams do follow a pattern (even if with noise and/or erratic behaviour), one can be sure that eventually the probabilities will converge. Issue 3 can be further improved in the case of periodic streams. Given that most real scenarios are periodic to some extent, this property can many times be used. Splitting a stream according to its period and using it as the domain of the grid, it is possible to work even with infinite domains. Each stream's period is taken as a 1-period stream by itself.

This way we are capable of characterizing the underlying behaviour of some event, based on the behavioural patterns of some related streams. We say this method is Markov chains' based since it assumes that there is little to no knowledge lost by only considering direct transitions along the x axis. This means that we do not use all the previous values a stream took before a given x_i when computing the probability of being in some other area in the time slot following (with $x_{i+1} \equiv x_i + \Delta x$). This is done to minimize the computational complexity that would arise from doing so.

The representation mentioned above can still have a problem: the notion of "area" itself. If it is too wide or too narrow, the model fails to capture the relevant pattern of the event. If any of Δx or Δy are taken too big to the event being represented, information about it will be lost. On the other hand, if these values are taken too small, the computation's complexity of the probabilities will start to degrade. Even worse, can make the whole representation too specific (commonly named overfitting).

In order to minimize this issue we propose to keep the following values associated to each slot, as shown in 2:

Probability vector This is the function which makes possible representing the nature of the stream using probabilities. Each P_i maps to the probability of jumping to the y_i following along the x axis (the transition).

Histogram of values Each slot maintains an histogram of values, allowing the model to identify which values are more commonly found within that slot. In a sense this adds another dimension to the model.

Other statistical values Other statistical values may be kept for further improvements. For example, keeping the average and the standard deviation of the values within the slot. They are both cheap computationally wise and may be of significance when evaluating how well a given point fits within the slot.

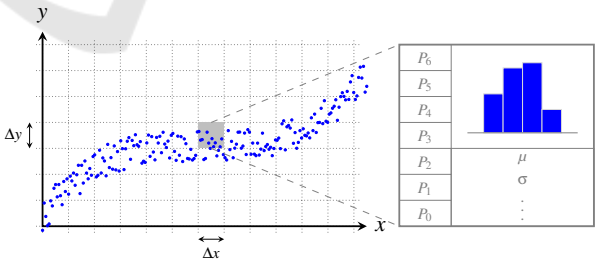

Figure 2: Structure proposed to represent stream information. A grid is overlayed over the streams, in order to build a matrix like structure where each slot contains a probability vector, an histogram of values, and other relevant statistical values (currently the mean and standard deviation of the values inside the slot). This figure is merely for illustrative purposes, none of the values represent real information nor the result of any kind of computation.

3.2 Generation of Streams

Initial work demanded the use of large datasets both to carry on tests and to validate the capability of representation of the model. This lead to the development of a stream generator general enough to be used in a wide class of streams. We wanted to use it to easily build synthetic datasets from real ones we had, but which were not as big as we needed.

Such generator would have to output plausible streams, and not just a stream which would for instance minimize the errors between itself and the set of streams given as examples. This constituted an opportunity to test our proposed representation. The internal structure of the generator is, thus, a matrix of slots, each with the values as described in 3.1. This matrix is built for each type of pattern we want to learn, from a set of streams representative of the pattern (e.g. temperature or humidity). After having the matrix built, we can traverse it (along its x axis) to generate streams hopefully similar to the underlying pattern of the ones which were previously presented.

Preliminary tests show the good capability of the generator to learn the most relevant motifs of the streams and be capable of generating realistic streams from the representation built. This is further discussed in 5.

4 FUTURE WORK

Further improvements to the model presented earlier are possible. Some of them are discussed bellow.

We believe it is possible to devise a metric to evaluate the similarity between a stream and our model. As mentioned in the beginning, our end game is to use the previously mentioned model to organize sensor streams based on theirs similarity and improve the efficiency of our context representation model. Using this similarity metric we can, based on a certain threshold, say which classification tags constitute the set of possible matches to the stream. Or we can even provide the set of k-strongest classification tags assignable to it. Once such set is known, more complex (computationally-wise) algorithms can be used in order to further carry on and narrow the search.

It is our belief that the integration of our fast labelling method with existing classification techniques will make organization across large stream-bases both possible, efficient and accurate. Our algorithm will serve as a strong filter, trimming the search space so that other techniques can proceed.

There is room to further improve our stream characterization model. Specially to cope with the variability associated with IoT/M2M scenarios. Some questions which are yet to be answered include: Is scale (along the y axis) important? If yes, in which cases and how to work with it? How to cope with time and location differences across the different sensors? How to automatically estimate a stream's period? We will continue our research on these topics and hopefully answer these questions in future publications.

5 PRELIMINARY RESULTS

The results shown in this section try to back our claims that our representation model is indeed capable of harvesting the most relevant features of the streams it was built with; for this we will use streams generated by our generator and compare them with real ones.

We have not finished a similarity metric for our stream characterization model. As such, we will use MSE (mean square error) and visual representations to evaluate the performance of our model. Given a set of (real) streams, in this case related to the temperature in a laboratory, we want to generate and validate another set of (synthetic) streams so that the later would be plausible elements of the former. By "plausible elements" we consider a human or other entity would have difficulty at telling them apart. Regarding the evaluation, we used k-cross validation. Each of the real streams used for computing MSE were not included in the set of training streams[1].

3 depicts a comparison between real and synthetic data. Both plots represent the values from twenty different streams (real and synthetic accordingly). The generator was trained with around one hundred real streams. The MSE over an averaged set of twenty runs was 0.508777934. As can be seen, both curves are alike and the MSE mesure is small, which further suggests the representative power of the model used. Our representation seems capable of storing the shapes of the curves as probabilities of transitions. This representation can then be used to generate new streams.

While these preliminary results require more elaboration, we consider them useful as checkpoints which attest that our idea does have some foundations. How deep they are, that is something requiring further work (which we intend to carry on), but at least we see that they are present.

[1]The generator's parameters were:

$$T = 24\,\text{h};\ \Delta x = 10\,\text{min};\ \Delta y = 0.5\,^{\circ}\text{C}$$

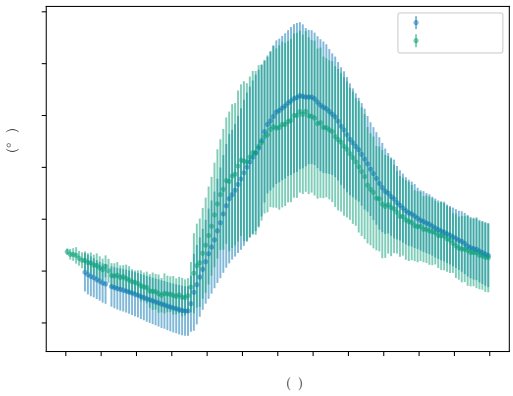

Figure 3: Real and generated streams plotted along with the deviation measured at each point.

6 CONCLUSIONS

We believe that our context organization model can be further improved by incorporating stream similarity metrics. While there are several academic works based on stream prediction and mining (Krempl et al., 2014), the same can not be said about stream similarity. Further work needs to be done to assert some ideas expressed on this paper, but our stream characterization model appear to be a viable option.

Meanwhile, the ability to generate streams resembling a given set of learning ones, can be useful in many situations. For instance, to generate large synthetic datasets where otherwise there is no specific generator available. Our general purpose generator has another big advantage. Improves the repeatability and validity of IoT/M2M and context-aware platforms. Currently these platforms use advanced machine learning algorithms to improve and optimize several processes. Having the ability to test them for a long time in a controlled environment is extremely important.

In future publication we will present an improved version of our stream characterization model and how to incorporate it into our context organization model.

ACKNOWLEDGEMENTS

This work was partially supported by European Regional Development Fund (ERDF) under grant agreement No. 7678 (Ref. POCI-01-0247-FEDER-007678) entitled “SGH - SMART GREEN HOME”, and research grant SFRH/BD/94270/2013.

REFERENCES

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Proc. of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307.

Antunes, M., Gomes, D., and Aguiar, R. (2015). Semantic features for context organization. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 87–92. IEEE.

Antunes, M., Gomes, D., and Aguiar, R. (2016a). Learning semantic features from web services. In *Future Internet of Things and Cloud (FiCloud), 2016 4rd International Conference on*. IEEE.

Antunes, M., Gomes, D., and Aguiar, R. L. (2016b). Scalable semantic aware context storage. *Future Generation Computer Systems*, 56:675–683.

Bacon, J., Bejan, A., Beresford, A., Evans, D., Gibbens, R., and Moody, K. (2011). Using real-time road traffic data to evaluate congestion. In Jones, C. and Lloyd, J., editors, *Dependable and Historic Computing*, volume 6875 of *Lecture Notes in Computer Science*, pages 93–117. Springer Berlin Heidelberg.

Chen, K.-C. and Lien, S.-Y. (2014). Machine-to-machine communications: Technologies and challenges. *Ad Hoc Networks*, 18:3–23.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7.

Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., and Balakrishnan, H. (2008). The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th International Conference on Mobile Systems,Applications, and Services*, pages 29–39.

Krempl, G., Žliobaite, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., and Stefanowski, J. (2014). Open challenges for data stream mining research. *SIGKDD Explor. Newsl.*, 16(1):1–10.

Mohan, P., Padmanabhan, V. N., and Ramjee, R. (2008). Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proc. of the 6th ACM conference on Embedded network sensor systems*, pages 323–336.

Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454.

Suhr, J. K. and Jung, H. G. (2014). Sensor fusion-based vacant parking slot detection and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 15(1):21–36.

Winograd, T. (2001). Architectures for context. *Hum.-Comput. Interact.*, 16(2):401–419.

Wortmann, F., Flüchter, K., et al. (2015). Internet of things. *Business & Information Systems Engineering*, 57(3):221–224.