Mining Frequent Distributions in Time Series

José Carlos Coutinho[1,2](✉), João Mendes Moreira[2], and Cláudio Rebelo de Sá[1]

[1] University of Twente, Enschede, The Netherlands
c.f.pinhorebelodesa@utwente.nl
[2] University of Porto, Porto, Portugal
{up201404293,jmoreira}@fe.up.pt

Abstract. Time series data is composed of observations of one or more variables along a time period. By analyzing the variability of the variables we can reveal patterns that repeat or that are correlated, which helps to understand the behaviour of the variables over time. Our method finds frequent distributions of a target variable in time series data and discovers relationships between frequent distributions in consecutive time intervals. The frequent distributions are found using a new method, and relationships between them are found using association rules mining.

1 Introduction

Time series data is constituted by a set of observations, each of them recorded at a specific time [5]. It can be defined by one or more variables that are measured in each observation. Keeping track of the variable's values along the time allows to study their variability and possibly obtain patterns that repeat or that are correlated.

This paper addresses the problem of finding patterns in time series data, in the form of distributions, and discovering the relationships between them in consecutive time intervals. After dividing the dataset in equal-sized time windows, it measures the distance between the target distributions using the Kolmogorov-Smirnov (KS) distance. When the distance between two distributions is below a threshold, we consider these distributions to be the same pattern. Whenever a distribution is not similar to any previously discovered pattern, we consider it a new pattern. In the end, we only keep the patterns (distributions) that are frequent according to a minimum support defined by the user.

Finally, we use association rules mining to find relationships between patterns on consecutive time windows. After defining a minimum support and confidence for the rules, the method will pair the time windows that are consecutive, labeling the pattern in the previous window as *antecedent* and the pattern in the next window as *consequent*. The method obtains the rules between antecedents and consequents which have values of support and confidence above the minimum.

After running the method, we could obtain distinct patterns of players' speed in football data and of price in electricity and AWS data. In the three cases, the method could find interesting association rules between the patterns found.

H. Yin et al. (Eds.): IDEAL 2019, LNCS 11872, pp. 271–279, 2019.
https://doi.org/10.1007/978-3-030-33617-2_28

2 Background and Related Work

Kolmogorov-Smirnov Statistical Test (*KS test*) is a statistical test which measures the equality of one-dimensional probability distributions. The KS statistic can be used to quantify the distance between two empirical distribution functions. Jorge *et al.* [7] also used this statistical test in their Distribution Rules method to measure the distance between a distribution of a target variable and a reference distribution.

Association Rules Mining is an area of Data Mining which studies ways of finding relationships between items in a dataset [1]. The relationships come in the form of implications $X \implies Y$, which are referred as *rules*. X and Y are itemsets, where X is the *antecedent* and Y the *consequent*. To measure the relevance of the rules, we check if the itemset $X \cup Y$ is observed frequently enough (*support*) and if the rule is verified frequently enough (*confidence*).

EP-MEANS proposed Henderson *et al.* [6], can be used to find patterns in the variability of one variable. This method clusters probability distributions regarding a target attribute. It is based in the K-means clustering algorithm [4] and the Earth Mover's Distance [8]. EP-MEANS has to pass through the data multiple times in order to get the centroids that will represent each pattern, which can be time consuming.

SPAM (Sequential PAttern Mining), proposed by Ayres *et al.* [3], can be used to find relationships between sequential time intervals. This method finds frequent itemsets sequences by building a lexicographic tree of sequences. Since we are only interested in finding the relationships between patterns in two consecutive timesteps, we chose not to use this method, for simplicity.

3 Proposed Method

We propose Frequent Distributions, a method which discovers frequent distributions of a variable, which we refer to as *profiles*. This method also uses association rule mining to look for relationships between the profiles in consecutive time intervals.

3.1 Finding Frequent Distributions

Let us define a univariate time series dataset ($\mathcal{D}$) as a table with n rows and 3 columns that come in the format $\{value, t, entity\}$, where $t \in T$ and $entity \in E$. We call $E = \{entity_1, \dots, entity_k\}$ the *entities*, where k is the number of entities represented in $\mathcal{D}$, and we call $T = [t_{first}, t_{last}]$ the *timesteps*, where t_{first} and t_{last} are the first and last timesteps registered in $\mathcal{D}$. Each row $r_{t,entity_i}$ represents an observation for one entity at a specific timestep. It can be formally defined as $r_{t_x,entity_i} = \{value_{(x,i)}, t_x, entity_i\}$.

A profile (*pf*) is defined as an empirical distribution of the values of a variable during a time interval of size *wsize*. $\mathcal{D}$ will have a fixed set of profiles of the variable ($PF_{variable} = \{pf_1, ..., pf_n\}$) and the variability of variable can switch between the different profiles over time. For example, when looking at records of electricity consumption, we will probably have $PF_{consumption} = \{pf_{day}, pf_{night}\}$. For pf_{day}, the distribution will include lower values, as opposed to pf_{night}, which will include higher values due to the need for artificial lighting during the night.

The Frequent Distribution mining approach iteratively discovers new profiles of a target variable (*target*) in the time series data. It starts by splitting the data into *time windows* of size *wsize*, and then makes one pass through them sequentially. For each time window $tw = [t_0, \ldots, t_{wsize}]$, we observe $Dist_{target}(tw, entity)$ for each $entity \in E$ and try to assign each $Dist_{target}$ to a profile. This is done by checking the *distance* between the distributions and the discovered profiles. Any distance distribution metric can be used, but, for simplicity we use the Kolmogorov-Smirnov statistical test as in [7]. A distribution is assigned to a known profile *pf* if $distance(Dist_{target}, pf) < \theta$. The value of θ is in the range $[0, 1]$ and is defined by the user. If $distance(Dist_{target}, pf_i) \geq \theta, \forall pf_i \in PF_{target}$, where PF_{target} is the set of profiles, then the new profile $Dist_{target}$ is added to PF_{target}. Finally, the profiles in PF_{target} which have a support below *minsupp* are discarded. The *minsupp* is decided by the final user. The pseudocode for Frequent Distributions is shown in Algorithm 1.

```
Input: target, wsize, $\theta$, minsupp
begin
    profiles_list; // Initialize with the initialization method described
    foreach time window tw of size wsize do
        foreach entity do
            $entity\_distribution \longleftarrow Dist_{target}(tw, entity)$;
            $is\_distribution\_distinct \longleftarrow True$;
            foreach pf in profiles_list do
                if $distance(entity\_distribution, pf) < \theta$ then
                    $is\_distribution\_distinct \longleftarrow False$;
                    $pf.count \longleftarrow pf.count + 1$;
                else
                    Add entity_distribution to profiles_list;
                end
            end
        end
    end
    $frequent\_profiles \longleftarrow$ All $profile$ in $profiles\_list$ where
      $profile.count > minsupp$;
    return frequent_profiles;
end
```

Algorithm 1. Frequent Distributions algorithm

3.2 Initialization of the Profiles Set

For this method to work, first we need to initialize *PF*. The initialization is done by observing $Dist_{target}(tw_{first}, entity)$ for each $entity \in E$. Then, we calculate the *distance* between those distributions and put them in a symmetric matrix, where the column and row indexes correspond to the entity indexes in *E*. This matrix is then binarized. Values above θ are set to 1, which represent the distributions which were different from each other, and all other values are set to 0. Finally, we group the distributions which have a 0 in the binarized matrix. In each group, the distribution whose entity has the lowest index will be added to PF as a profile.

3.3 Combining with Association Rules Mining

We can combine the Frequent Distributions with Association Rules mining by adding extra steps to the method. The objective is to obtain association rules that, for each entity, measure the transition between profiles in consecutive time windows. We will refer to the consecutive time windows as tw_x and tw_{x+1}, where tw_{x+1} is the window that immediately follows tw_x. For example, it would be interesting to observe that, for a given target attribute, a profile A is always followed by a profile B in the next time window.

In order to do this, first we have to have a record of the profiles observed for each pair $(tw, entity)$. Afterwards, we need to find the frequent itemsets that will be used in the association rules mining. Each itemset will be composed of a pair of consecutive profiles: a previous profile, observed in tw_x, and a next profile, observed in tw_{x+1}. So, we need to scan through the time series and obtain, for each (tw_x, tw_{x+1}), the pair of consecutive profiles. After obtaining all the itemsets in the last step, we use the Apriori algorithm [2] to find the itemsets that are frequent and afterward use association rules mining to find the association rules. An itemset is considered frequent if the support for that itemset is higher than a user-defined minimum support. Also, in our approach, an association rule will only be considered relevant if its confidence value is higher than the user-defined minimum confidence threshold.

We could have used Sequential Pattern Mining [3] to obtain these relationships. However, in order to simplify, we chose to use association rules mining.

4 Results

To test our method, we use data from 3 sources. The first source, which we call Source A, contains football (soccer) spatiotemporal data. The data has multiple datasets, each one representing the xy positions of the players during one match, and from the xy positions we could calculate the speed of the players. The second and third sources are the well-known Electricity and AWS datasets[1]. Given the big number of instance types on the AWS dataset, we decided to use a subset of the data corresponding to the *m4.large* instance type and the Linux/UNIX operative system.

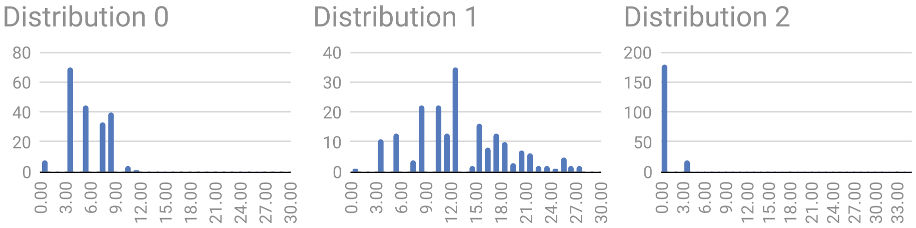
[1] https://moa.cms.waikato.ac.nz/datasets/.

In the datasets from Source A, we used the Frequent Distributions to look for speed profiles of players and relationships between the profiles (*target* = speed). In the Electricity dataset, we focused on the electricity prices (*target* = price). In the AWS dataset, the focus was the server prices (*target* = price). For that, we tested different values in the parameters. In all experiments , θ values varied between 0.6-0.9, and *minsupp* value was 0.01. Also, all rules with lift less or equal to 1 were discarded. In experiments with Source A's data, we used 100, 200 and 600 as the *wsize* value; with the Electricity dataset, we used 48, 96 and 144; and with the AWS dataset, we used 1440; In Source A, the *wsize* values correspond to a number of seconds times 10 (for example, $wsize = 50$ corresponds to a 5-s time window). In the Electricity dataset, the *wsize* values correspond to the number of half hours in the time window (for example, $wsize = 48$ corresponds to 48 half hours, which is the same as a day). In the AWS dataset, the *wsize* values correspond to the number of minutes in the time window.

In all experiments we chose the *wsize* according to the time length of the profiles we wanted to obtain. We observed that the smaller the *wsize*, the more profiles are found. This can be explained by the fact that having smaller windows implies that each distribution will include fewer examples. Fewer examples lead to more variability between the distributions observed, which in the end translates into finding more profiles.

In the experiments made, the values for θ were chosen empirically. However, since we are looking for distinct profiles, we set the minimum distance to 0.6. On the other hand, when the distances are too big (>0.9) only very few distinct profiles were found. This can be explained by the fact that a higher θ implies that more distributions will be considered similar to each other.

From the results of Source A, we can observe 3 main profiles. (Figure 1). There is one profile for standing still/walking (Fig. 1c), a second profile for slow running (Fig. 1a) and third profile for bursts of speed (Fig. 1b). This third profile has more variability, where the speed observed ranges from 0 to 22 km/h. This can mean that, in this profile, the player is not always running but increases and decreases his speed considerably.



(a) Profile 0, supp=49% (b) Profile 1, supp=37% (c) Profile 2, supp=14%

Fig. 1. Speed profiles of players obtained in Source A, with $wsize = 200$ and $\theta = 0.7$.

Regarding the relationships between profiles, it was found that is common for players to switch from the running profile to the slow running one (Rules 1, 2, 3 and 4 of Table 1). Also, it was found that was common for players to switch from the standing still/walking to the slow running profile (Rules 5 and 6 of Table 1). It was also found that was common for players to keep in the slow running profile (Rules 7, 8 and 9 of Table 1). All rules show that these phenomenons happened more than 50% of the times for multiple players across some of Source A's experiments. This reveals that, in the Source A dataset, players have a tendency to keep in the slow running profile or to return to it after being in a different profile.

Table 1. Most important rules found in Source A's data

Rule_id	Antecedent	Consequent	Support	Confidence	Lift	No. of players	wsize	θ
1	Profile 1	Profile 0	3–16%	50–58%	1.01–1.09	8	100	0.7
2	Profile 1	Profile 0	3%	50%	1.02	1	100	0.8
3	Profile 1	Profile 0	7–21%	50–53%	1.01–1.02	5	200	0.7
4	Profile 1	Profile 0	5%,11%	66%,73%	1.07,1.11	2	600	0.6
5	Profile 2	Profile 0	6–20%	50–53%	1.01	3	100	0.7
6	Profile 2	Profile 0	6%	50%	1.01	1	200	0.7
7	Profile 0	Profile 0	25–31%	51–56%	1.01–1.1	12	100	0.7
8	Profile 0	Profile 0	25%	50%	1.01	1	200	0.7
9	Profile 0	Profile 0	26–38%	51–62%	1.01–1.03	18	600	0.6

In the Electricity dataset, in the experiment with $wsize = 144$ and $\theta = 0.8$, there were found 7 profiles. The profiles are shown in Fig. 2.

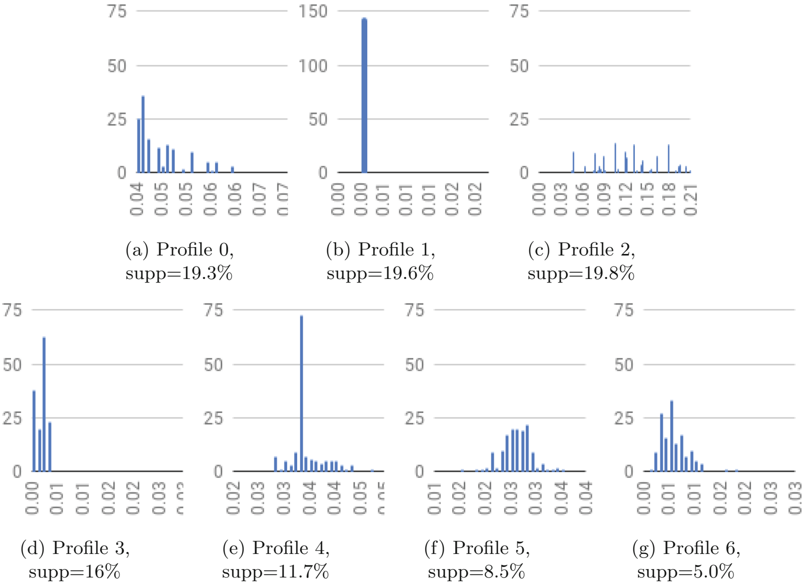
The rules found were only relative to the Victoria state, and are shown in Table 2. The rules show that, in more than 60% of the times that Victoria state was in the profile 3, it remained in that profile. This profile is one of the profiles which includes the lowest prices. This means that, when the price of electricity is low for 2 days ($wsize = 144$), it is likely to continue for the next 2 days.

Table 2. Association rules found in the Electricity dataset relative to the Victoria state

Rule_id	Antecedent	Consequent	Support	Confidence	Lift
1	Profile 3	Profile 3	24%	62%	1.58

In the AWS dataset, in the experiment with $wsize = 1440$ and $\theta = 0.9$, there were found 7 profiles. A temporal representation of the values of each profile is shown in Fig. 3.

We found rules relatively to the instances of the two regions that correspond to Canada: *ca-central-1a* and *ca-central-1b*, which are shown in Table 3.



(a) Profile 0, supp=19.3%

(b) Profile 1, supp=19.6%

(c) Profile 2, supp=19.8%

(d) Profile 3, supp=16%

(e) Profile 4, supp=11.7%

(f) Profile 5, supp=8.5%

(g) Profile 6, supp=5.0%

Fig. 2. Profiles of the electricity prices. Note that the scales may be different between plots.

They show that, in more than 55% of the times that the prices of the canadian servers were in profile 3 or 6, they remained in that profile. These two profiles are the ones that include the lowest prices, which can mean that, when canadian servers have the lowest prices during one day ($wsize = 1440$), it is likely that the prices keep low on the next day.

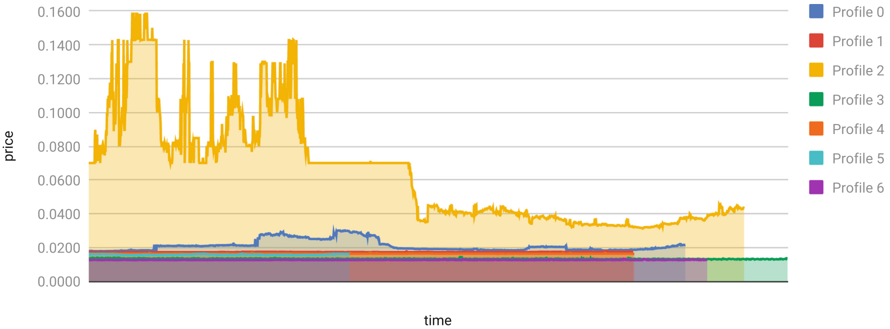


Fig. 3. Variation of the values of each AWS profile of price, relatively to all instances *m4.large*, Linux/UNIX, regardless of the region

Table 3. Association rules found in the AWS dataset relative to the Canadian region

Rule_id	Antecedent	Consequent	Support	Confidence	Lift	Region
1	Profile 3	Profile 3	13%	62%	2.55	ca-central-1a
2	Profile 6	Profile 6	10%	92%	8.52	ca-central-1a
3	Profile 3	Profile 3	13%	55%	2.16	ca-central-1b
4	Profile 6	Profile 6	9%	88%	8.50	ca-central-1b

5 Conclusion

We propose a preliminary study on what we refer as Frequent Distributions, a method to find distributions (profiles) of variables in time series data and relationships between them in consecutive time intervals.

As future we want to improve the initialization and obtain profiles with more than one variable. Also, sequential pattern mining could be used to obtain relationships that are not limited just to two consecutive time windows.

Acknowledgements. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project : UID/EEA/50014/2019

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26–28, 1993. pp. 207–216. ACM Press (1993). https://doi.org/10.1145/170035.170072
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB 1994, Proceedings of 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago de Chile, Chile, pp. 487–499. Morgan Kaufmann (1994)
3. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23–26, 2002, Edmonton, Alberta, Canada, pp. 429–435. ACM (2002). https://doi.org/10.1145/775047.775109
4. Bishop, C.M.: Pattern Recognition and Machine Learning, 5th edn. Springer, Information science and statistics (2007)
5. Brockwell, P., Davis, R.: Introduction to Time Series and Forecasting. Springer Texts in Statistics. Springer, New York (2013)
6. Henderson, K., Gallagher, B., Eliassi-Rad, T.: EP-MEANS: an efficient nonparametric clustering of empirical probability distributions. In: Wainwright, R.L., Corchado, J.M., Bechini, A., Hong, J. (eds.) Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13–17, 2015, pp. 893–900. ACM (2015). https://doi.org/10.1145/2695664.2695860

7. Jorge, A.M., Azevedo, P.J., Pereira, F.: Distribution rules with numeric attributes of interest. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 247–258. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_26
8. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Proceedings of the Sixth International Conference on Computer Vision (ICCV-1998), Bombay, India, January 4–7, 1998, pp. 59–66. IEEE Computer Society (1998). https://doi.org/10.1109/ICCV.1998.710701