A Categorical Clustering of Publishers for Mobile Performance Marketing

Susana Silva[1], Paulo Cortez[1], Rui Mendes[2], Pedro José Pereira[1], Luís Miguel Matos[1], and Luís Garcia[3]

[1] ALGORITMI Centre, Department of Information Systems, University of Minho, 4804-533 Guimarães, Portugal
pcortez@dsi.uminho.pt, a64871@alunos.uminho.pt

[2] ALGORITMI Centre, Department of Informatics, University of Minho, 4710-057 Braga, Portugal
rcm@di.uminho.pt

[3] OLAmobile, Spinpark, 4805-017 Guimarães, Portugal
luis.garcia@olamobile.pt

Abstract. Mobile marketing is an expanding industry due to the growth of mobile devices (e.g., tablets, smartphones). In this paper, we explore a categorical approach to cluster publishers of a mobile performance market, in which payouts are only issued when there is a conversion (e.g., a sale). As a case study, we analyze recent and real-world data from a global mobile marketing company. Several experiments were held, considering a first internal evaluation stage, using training data, clustering quality metrics and computational effort. In the second stage, the best method, COBWEB algorithm, was analyzed using an external evaluation based on business metrics, computed over test data, and that allowed an identification of interesting clusters.

Keywords: Categorical Clustering, Mobile Marketing, Big Data

1 Introduction

The mobile performance marketing industry is currently experiencing a great evolution due to the growth of mobile device usage (e.g., tablets, smartphones). In this industry, compensation only occurs when an ad performs well (e.g., purchase). There are several Demand-Side Platforms (DSP) that match users to ads. In this market, user traffic is generated by publishers, that have a popular free website or app (e.g, news, game), capable of attracting a vast audience, and that is financed by dynamic link ads. Each time a user clicks on an ad, the DSP redirects the user to a marketing campaign from an advertiser. When there is a conversion (e.g., purchase), the DSP returns a percentage of the advertiser revenue to the publishers [10]. From the publisher's point of view, the Return On Investiment (ROI) is highly relevant, since there are several costs for maintaining and improving their service (e.g., producing content, rent digital space, technological costs). Thus, a key decision factor is the expected revenue when deciding about joining or leaving a particular DSP. Often, this involves the analysis of several performance marketing metrics adopted by the industry, including [6, 19]: Click Through Rate (CTR), the rate between the number of advertisement clicks and the number of impressions; the Conversion Rate (CVR), the percentage of ad clicks (redirects) that originated a conversion sale (e.g., purchase); and client Lifetime Value (LTV), the amount of revenue generated per sale.

Currently, most DSPs only provide global metrics, computed using all publishers. In this work, we explore a clustering approach to automatically group publishers into similar profiles, such that more informative and realistic revenue metrics could be provided. As a case study, we work with recent data from OLAmobile, a global mobile marketing company. This DSP generates big data with its volume and velocity properties (e.g., millions of redirects and thousands of sales per hour).

Although clustering is a popular data mining approach in many real-world applications [2], its usage in mobile performance marketing is scarce, since studies are mostly focused on user CTR or CVR response prediction [6, 20]. The current clustering approaches on advertising data include, mostly, understanding user behaviour. In particular, organizing user's search terms by subject, using clustering, and then relating it with user's intention to purchase and CTR [14]. In [15] the authors use k-means on user data, ending up with 10 user profiles, and mapping each profile to the types of advertisement that profile is more likely to interact with. In a different context, clustering was also applied to model users' behaviour in e-commerce web-pages, with the purpose of optimizing these websites and providing personalized recommendations to users, using click-stream data [18]. More recently, a study on click-stream data used clustering and biclustering to group users [11]. Furthermore, the authors stated that click-stream data clustering is a recent and challenging problem for many applications, with the main difficulty being related to grouping categorical data sequences, since most traditional clustering algorithms are not applicable to categorical data. Moreover, the problem complexity increases when handling big data, which is the case of mobile performance marketing.

In the mobile marketing performance industry, there is a lack of studies concerning clustering approaches related to publishers. This is probably due to the clustering complexity associated with the amount of data produced by this industry. There are vast amounts of records, most variable attributes are categorical and several of these attributes have a large cardinality (e.g., with dozens or hundreds of levels). This study fulfills this gap by addressing a categorical data clustering approach that is capable of grouping publishers in different clusters based on their similarities. In particular, we first compare five distinct clustering algorithms using two internal (training) metrics and also execution time. The comparison is executed using one week of data and a rolling window evaluation. The best clustering method is then selected and evaluated in terms of an external (test) metric that includes several business goals, showing its potential value in this domain.

This document is organized as follows: Section 2 discusses data collection, clustering methods and the evaluation procedure; Section 3 describes the experiments performed and analyzes the results obtained; finally, Section 4 draws the main conclusions and discusses future work.

2 Materials and Methods

2.1 Data Collection

For the data collection process, the organization under study, OLAmobile, provided a web-service enabling the connection to their data center, where data is divided in two streams: redirects and sales. The first one is related to data generated by user clicks on advertisements, where each click originates a record, while the latter is from redirects that originated a purchase, with each record containing also the information about the corresponding redirect. It is important to notice the great discrepancy existing between the number of redirects and the number of sales, since OLAmobile deals with millions of hourly records concerning redirects and only approximately 1% culminate in a purchase. Moreover, due to computational limitations associated with our server for this project, it was impossible to retrieve all redirect records from OLAmobile, thus, our ratio between collected redirects and sales was much higher (around 36%) than the real one. In Section 2.3, we adjust the CVR metric to realistic values (around 1%) by adopting an α correction factor.

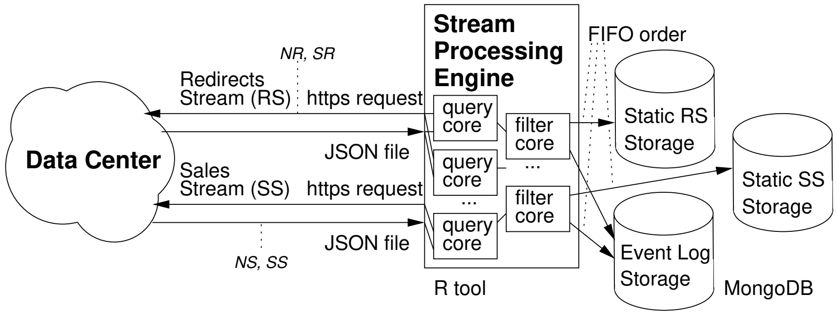

Fig. 1. Representation of the data collection process

Figure 1 represents the data collection process that was carried out and it consists in a multi-core system constantly gathering the most recent records concerning both redirects and sales (NR and NS denote the amount of redirects and sales events requested, while SR and SS represents the sleep time between two consecutive requests). The first core was constantly collecting records regarding redirects, the second core was collecting sales while a third core was filtering data from both redirects and sales. Our filtering process consisted in selecting best mode events and clients from Europe. The best mode is related with product campaigns that have obtained a minimal performance in a testing mode and it corresponds to most DSP traffic. After filtering, data was stored in two different MongoDB databases (one for redirects and another for sales) with a predefined size, using a FIFO (First In First Out) writing process, and, once the database was full, the data was stored in two static files. This process was performed during 7 days, from 31^{th} October to 6^{th} November 2017, ending up with 48,900 records with 17,600 concerning sales and nearly 31,300 regarding redirects.

After collecting and storing data, we joined both redirect and sales events into a single file. Table 1 shows a summary of the analyzed data attributes. In particular, the first six clustering attributes were identified by OLAmobile as interesting to associate with distinct publisher profiles and were used to estimate internal (int.) metrics, while the last two attributes were used to compute business metrics in the external (ext.) evaluation.

Table 1. Description of analyzed data attributes

Feature	Use	Description
vertical	int.	ad type selected by the publisher, with 4 levels (e.g., VOD - video on demand, mainstream)
country	int.	code of country, with 45 levels (e.g., Russia, Spain)
idos	int.	operating system, with 10 levels (e.g., Android, iOS)
account	int.	account type, with 11 levels (e.g., network, advertiser, app)
hardware	int.	type of device, with 3 levels (smartphone, tablet, smart TV)
network	int.	type of network access (mobile, WiFi)
target	ext.	if there is a conversion (no sale, sale)
revenue	ext.	publisher revenue if sale (in EUR)

2.2 Clustering Methods

There is an increasing need for clustering algorithms to support categorical data, since most of the traditional clustering algorithms were built on the assumption that they would work only with numerical data, while most real-world problems deal with other types of data, such as categorical, temporal or structural [1]. Several challenges are associated with clustering of categorical data, namely, the lack of ability to understand how similar two different classes are. Thus, clustering algorithms for numerical data cannot be directly applied to this type of problems [2].

Based on [1], there are several clustering methods that support categorical data, namely: K-modes, COOLCAT, ROCK, CLOPE and COBWEB. The K-modes algorithm results from k-means and aims to fill its inability to deal with non-numerical data, preserving its efficiency, by using dissimilarity measures for categorical objects, replacing the mean from the k-means algorithm for modes and using a frequency based method for modes detection [1]. Its main disadvantage is that it may produce locally optimal solutions that strongly depend on the initial modes and the objects' order in the dataset.

The COOLCAT algorithm is based on the same principles of k-modes but uses the clusters' centers instead of modes. This algorithm performs in an incremental way and it is capable of grouping new data without processing the whole dataset, which makes it suitable for clustering data streams, as well as categorical Big Data [17]. A main disadvantage is the fact that the data processing order influences clustering quality and, to increase the quality, an offline data sample must be reprocessed [7].

Previous clustering algorithms belong to partitioning methods, while the next ones are hierarchical. ROCK is a clustering agglomeration technique that uses the number of common neighbours between two objects as a connection measure, aiming to group objects with higher number of connections, which represents higher similarity between them [3].

The COBWEB algorithm produces a cluster dendrogram, which is a classification tree, that characterizes each cluster with a probabilistic description [16]. This algorithm uses an heuristic evaluation metric, termed category utility, to guide the tree's construction, and it has the ability of automatically adjusting the number of classes, without the need for the user's intervention.

Finally, the CLOPE algorithm was developed from the heuristic method of increasing the ratio between height and width of clusters histograms. It is considered to be fast and scalable in transactional databases and other data sources with a high number of dimensions [1, 17].

2.3 Evaluation

The experimental design includes two stages. In the first stage, using only clustering attributes and training data, we use internal objective metrics to select the clustering method and number of clusters. To get a more robust and realistic assessment, we adopt three clustering iterations in this stage, under a realistic rolling window evaluation [12]. The first four days are used in the first iteration. Then, the data is slided by one day, leading to the second iteration, and so on. Each day contains around 7,000 redirect samples, translating into a training window with around 28,000 samples. The internal metrics are computed using average rolling window values of two popular clustering metrics (silhouette and Dunn index) [2] and the computational effort. The silhouette varies between -1 (poor clustering) to 1 (excellent) [1, 5]. The Dunn index represents the ratio between the lower distance among observations on different clusters and the higher distance among objects within the same cluster. Its values range from 0 to 1, and higher values are better [5]. To compute the silhouette and Dunn index metrics we used the Gower distance measure, which is suitable for mixed (categorical and numeric) attributes and that ranges from 0 (identical) to 1 (most dissimilar)

[9]. To measure the similarity between two clustering results, we used the rand index, which presents the highest value of 1 when both clusterings are equal [4].

In the second stage, we apply a time order holdout split to the best clustering algorithm of the previous stage. The first four days of data are used to training and the last three days of data for testing. We compute silhouette and Dunn index metrics for the test data and also external business metrics, which allow the selection of interesting clusters. These clusters are then described in terms of their main characteristics in order to get feedback from human experts. The business metrics include: N - the total number of redirects associated with a cluster of publishers; CVR - the conversion rate for the cluster; and LTV - the average publisher revenue for the cluster when there is a conversion. Since our collected sampled data includes a ratio of sales that is much higher than the real market, we adjusted the conversion rate to its realistic version: $CVR = S/N \times \alpha$, where S is the number of cluster sales and $\alpha = 1/35.99$ is a correction factor, such that the global CVR value, over all data, is near 1%.

3 Results

We conducted all experiments using the R tool [13] and a Linux server with an Intel Xeon 1.70GHz processor with 56 cores and 64 GB of RAM. The packages used for cluster algorithms were **klaR** for k-modes, **coolcat** for COOLCAT, **cba** for ROCK and **RWeka** for both COBWEB and CLOPE. Furthermore, the **fpc** package was used for computing the clustering metrics.

The first experiments aimed to find a good number of clusters (K) and we tested four different possibilities: $K \in \{10, 20, 30, 40\}$. We only used the k-modes algorithm for this initial experiment, since it is easy to set K for this method and also it requires a reasonable computational effort. Table 2 presents the average clustering rolling window metric results. After analyzing Table 2, we decided to choose $K = 20$, since it accomplished good results with an acceptable computational cost and it is within the granularity level desired by OLAmobile.

Table 2. Results from tests using K-modes testing different numbers of clusters (best values in **bold**)

	number of clusters			
Metric	**10**	**20**	**30**	**40**
Silhouette	**0.18**	0.17	**0.18**	0.17
Dunn Index	0.17	0.18	0.18	**0.20**
Execution time (s)	**4579.31**	8133.19	12710.18	20048.49

After setting K, we compared the clustering methods by executing the first experimental design phase. The COOLCAT and ROCK algorithms did not provide results in useful time. We executed these algorithms and waited during

3 days and, besides the time they took executing, they have shown to be unable to deal with this quantity of data, often returning processing errors. As such, both of these algorithms were excluded from the second phase. Thus, in Table 3 we report only the K-modes, COBWEB and CLOBE average rolling window results. Also, the similarity between the cluster methods (in terms of the rand-index) is presented in Table 4 Regarding the remaining algorithms, it is important to notice that only K-modes receives the number of clusters (K) as a parameter. Thus, we have adjusted the COBWEB (cutoff) and CLOPE (repulsion) parameters to indirectly control the number of clusters, such that on average $K \approx 20$. Although it was the faster algorithm, CLOPE presented low values for the silhouette and Dunn index metrics. The second best performing algorithm, COBWEB revealed a high similarity with K-Modes (rand-index of 0.9) and was selected for the second phase, since it achieved better internal metric values under a lower computational cost.

Table 3. Average first stage clustering results (best values in **bold**)

Metric	**K-modes**	**COBWEB**	**CLOPE**
Silhouette	0.17	**0.30** (0.27)	-0.10
Dunn Index	0.18	**0.20** (0.20)	0.06
Clusters	20	19.67	19.33
Execution time (s)	8133.19	6093.06	**2432.38**

Table 4. Rand-index results for cluster similarity comparison

Algorithms	**Rand-Index**
COBWEB and CLOPE	0.64
K-Modes and CLOPE	0.63
K-Modes and COBWEB	0.90

In the second phase, COBWEB provided 25 different clusters when trained with data from the first four days. The respective silhouette and Dunn index test set results (computed using the last three days) are shown in brackets in Table 3. The average rolling window training and holdout test clustering metrics are similar (e.g., silhouette of 0.30 and 0.27), showing that the clusters have stable quality values through time.

The left of Figure 2 presents the external evaluation results in terms of CVR (x-axis) versus LTV (y-axis). In the graph, each cluster label is set within a circle that is proportional to the cluster size (N). To simplify the analysis, the graph includes only the nine biggest clusters (such that $N > 300$). Using a multi-objective analysis to the business metrics, we selected four interesting clusters: 1 – with 4,179 redirects, highest CVR (1.4%) and LTV (2.1) values; 15 – largest cluster with 4,518 redirects, low LTV and CVR slighly above average (1.07%); 21 – second largest cluster with 4,419 redirects, good LTV (1.0) and low CVR (0.6%); and 25 – small cluster with 355 redirects, low CVR and LTV values. The right of Figure 2 shows the main differences between a high (1) and bad (25) performing clusters, in terms of the mode values for the six clustering attributes. The graph shows differences between the clusters in terms of the vertical and account attributes. The full mode values for the four selected clusters are shown in Table 5, which highlights the main cluster differences.

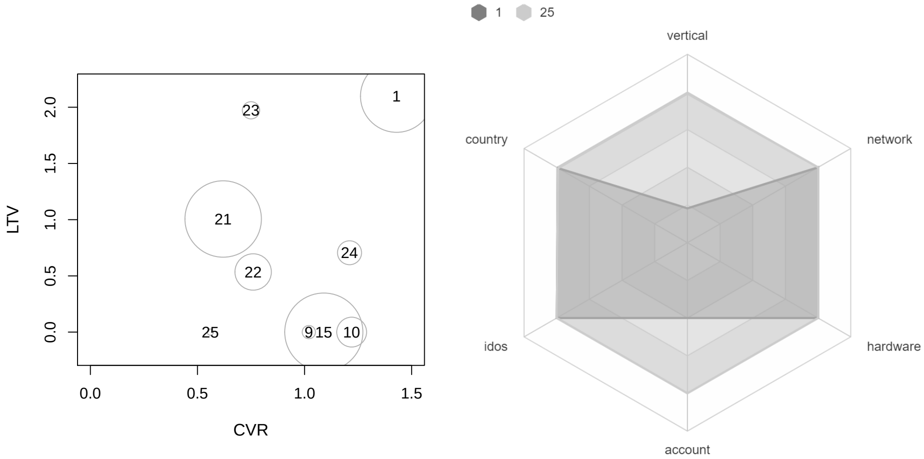

Fig. 2. Business metrics for the obtained clusters (left) and characterization of clusters 1 and 25 (right)

Table 5. Characterization of the selected clusters (mode values)

Cluster	vertical	country	idos	account	hardware	network
1	VOD	Russia	Android	advertiser	smartphone	mobile
15	mainstream	Russia	Android	advertiser	smartphone	mobile
21	mainstream	Russia	Android	advertiser	smartphone	WiFi
25	mainstream	Russia	Android	network	smartphone	mobile

These results were shown to OLAmobile experts and the obtained feedback was positive. In particular, the selected clusters are aligned with their business knowledge and were considered interesting, presenting a good potential for supporting publisher's decisions in terms of adhering to a DSP market or renting digital spaces.

4 Conclusions

Mobile performance marketing is a growing industry due to current ubiquitous usage of personal devices (e.g., smartphones, tablets). This industry includes several players: publishers, which attract users but need to be funded by ads; Demand-Side Platforms (DSP), which act as brokers, matching users to ads and managing payouts when there is a conversion (e.g., sell); and advertisers, which resort to DSPs for promoting their products and increase sales.

In this paper, we adopt a novel categorical clustering approach for this industry, which groups publishers into similar segments, in order to provide them with more informative expected revenue metrics. As a case study, we worked with data from OLAmobile, a global mobile performance marketing company that receives millions of ad clicks per hour. We collected a sample dataset with 48,900 records related with European ad clicks during one week. Then, we conducted a first set of experiments in order to compare five categorical clustering methods using a first internal evaluation that used objective clustering quality metrics. The best performing algorithm (COBWEB) was then analyzed under a second evaluation phase, based on business metrics, namely the Conversion rate (CVR) and Lifetime value (LTV). Such analysis allowed the particular identification of four interesting clusters, including a large cluster with high CVR and LTV values, and a smaller cluster with low CVR and LTV values. These clusters were characterized in terms of their mode values (considering six clustering attributes), resulting in a positive feedback from domain experts in terms of their validity and usefulness for publishers. In the future, we wish to collect more data (e.g., worldwide) and also study data scalability issues. For instance, by adapting the categorical clustering algorithms to the MapReduce framework [8].

Acknowledgements

This work was supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/-00319/2013, and project: NORTE-01-0247-FEDER-017497.

References

1. D. Agarwal, B. Long, and D. Xin. LASER : A Scalable Response Prediction Platform For. *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*, pages 173–182, 2014.

2. C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.
3. M. Alamuri, B. R. Surampudi, and A. Negi. A survey of distance/similarity measures for categorical data. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1907–1914, 2014.
4. A. Amini, T. Y. Wah, and H. Saboohi. On density-based data streams clustering algorithms: A survey, 2014.
5. G. Brock, V. Pihur, S. S. Datta, and S. S. Datta. clValid : An R Package for Cluster Validation. *Journal Of Statistical Software*, 25(March 2008):1–28, 2008.
6. M. Du, R. State, M. Brorsson, and T. Avenesov. Behavior profiling for mobile advertising. *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT '16*, pages 302–307, 2016.
7. G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.
8. K. D. Garcia and M. C. Naldi. Multiple parallel mapreduce k-means clustering with validation and selection. In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*, pages 432–437. IEEE, 2014.
9. J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.
10. Y. Hu, J. Shin, and Z. Tang. Pricing of online advertising: Cost-per-click-through vs. cost-per-action. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2010.
11. V. Melnykov. Model-based biclustering of clickstream data. *Computational Statistics and Data Analysis*, 93:31–45, 2016.
12. N. Oliveira, P. Cortez, and N. Areal. The impact of microblogging data for stock market prediction: using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144, 2017.
13. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.
14. M. Regelson and D. Fain. Predicting click-through rate using keyword clusters. *Proceedings of the second workshop on sponsored search auctions (Vol. 9623)*, pages 1–6, 2006.
15. J. Reps, U. Aickelin, J. Garibaldi, and C. Damski. Personalising mobile advertising based on users' installed apps. In *IEEE International Conference on Data Mining Workshops, ICDMW*, pages 338–345, 2015.
16. N. Sharma, A. Bajpai, and R. Litoriya. Comparison the various clustering algorithms of weka tools. *International Journal of Emerging Technology and Advanced Engineering*, 2(5):73–80, 2012.
17. M. Sora, S. Roy, and S. I. Singh. FLoMSqueezer : An Effective Approach For Clustering Categorical Data Stream. *International Journal of Computer Science Issues*, 8(6):284–291, 2011.
18. Q. Su and L. Chen. A method for discovering clusters of e-commerce interest patterns using click-stream data. *Electronic Commerce Research and Applications*, 14(1):1–13, 2015.
19. S. Yuan, A. Z. Abidin, M. Sloan, and J. Wang. Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users. *CoRR*, abs/1206.1754, 2012.
20. W. Zhang, T. Du, and J. Wang. Deep learning over multi-field categorical data. In *European conference on information retrieval*, pages 45–57. Springer, 2016.