View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Universidade do Minho: RepositoriUM

Using Deep Learning for Ordinal Classification of Mobile Marketing User Conversion

Luís Miguel Matos[1], Paulo Cortez[1], Rui Castro Mendes[1], and Antoine Moreau[2]

[1] ALGORITMI Centre, University of Minho, 4804-533 Guimarães, Portugal
id6929@alunos.uminho.pt, pcortez@dsi.uminho.pt, azuki@di.uminho.pt
[2] OLAmobile, Spinpark, 4805-017 Guimarães, Portugal
antoine.moreau@olamobile.com

Abstract. In this paper, we explore Deep Multilayer Perceptrons (MLP) to perform an ordinal classification of mobile marketing conversion rate (CVR), allowing to measure the value of product sales when an user clicks an ad. As a case study, we consider big data provided by a global mobile marketing company. Several experiments were held, considering a rolling window validation, different datasets, learning methods and performance measures. Overall, competitive results were achieved by an online deep learning model, which is capable of producing real-time predictions.

Keywords: Mobile Performance Marketing · Multilayer Perceptron · Ordinal Classification.

1 Introduction

The massive adoption of smartphones has increased of value of mobile performance marketing. These markets are implemented using a Demand Side Platform (DSP), which matches users to ads and is used by publishers and advertisers [9, 11]. A publisher is a web content owner (e.g., games) that attracts users. The web content is funded by DSP dynamic ads, which when clicked redirects the user to an advertiser site. When there is a conversion, the DSP returns a portion of the sale value to the publishers. Under this market, a vital element is the prediction of the user conversion rate (CVR), i.e., if there will be a conversion when a user sees an ad [4]. Typically, CVR prediction is modeled as a binary classification task ("no sale", "sale") by using offline learning Machine Learning methods, namely Logistic Regression (LR) [4], Gradient Boosting Decision Trees (GBDT) [8], Random Forests (RF) [4] and Deep Learning Multilayer Perceptrons (MLP) [9].

In contrast with the binary CVR studies (e.g., [4, 8]), in this paper we propose a novel ordinal classification of mobile CVR, which assumes five classes: "no sale", "very low", "low", "medium" and "high". This approach provides a good proxy to the client Lifetime Value (LTV) [11]. Thus, using such ordinal classifier, a DSP can better select the best ad campaign for a particular user by maximizing the expected conversion value. Following a recently proposed binary

deep learning approach [9], we explore three main MLP strategies to handle ordinal classification: pure classification, regression and ordinal classification. These approaches are tested using two learning models, offline and online, using a realistic rolling window validation and real-world big data from a global DSP. Also, the deep learning models are compared with a LR method.

2 Materials and Methods

2.1 Data

We collected the data from a worldwide marketing company (OLAmobile). The DSP generates two main events: redirects – the user ad clicks; and sales – when there is a conversion. All redirects and sales are stored at the DSP data center, being associated with a timestamp when they arrive. The DSP managed by the company generates millions of redirects and thousands of sales per hour.

We had access to a secure web service that allowed us to retrieve NR redirects and NS sales from the data center. Our computing server, an Intel Xeon 1.70GHz wth 56 cores and 2TB of disk space, is limited when compared with the data center and thus we work with sampled data. The data was collected during a two week period, starting at 30th May of 2019, via a stream engine that uses K computing cores to continuously retrieve redirects and sales, 'sleeping" every SR and SS seconds [9] (Table 1). The analyzed DSP contains two traffic modes: TEST – used to measure the performance of new campaigns; and BEST – with 80% of the traffic and including only the best TEST performing ads. The Y_{no} and Y_{yes} columns denote the number of collect "no sale" and "sale" events. Also, the R_Y column represents the sales ratio $R_Y = Y_{\text{yes}}/(Y_{\text{no}} + Y_{\text{yes}})$. Since we worked with sample data, the collected data ratio is higher than the expected real DSP one. To get a more realistic dataset, we randomly undersample [1] the number of sales (Y'_{yes}) such that a more realistic ratio ($R_{Target'}$) is obtained, which in this work was fixed to 5% for both BEST and TEST traffic. Thus, for each traffic mode there are two datasets: collected (C) and realistic (R). Table 2 presents a summary of the collected attributes. Most attributes are categorical and some present a high cardinality (e.g., city).

Table 1. Summary of the collected DSP datasets

Traffic	NR	NS	SR	SS	K	Y_{no}	Y_{yes}	Y'_{yes}	R_Y	$R_{Y'}$
TEST	100	100	300	10	2	290,279	29,596	15,393	9.3%	5.04%
BEST	100	100	300	10	2	328,028	156,637	17,389	32.2%	5.03%

After consulting the DSP experts, we created the ordinal Y target by grouping the value into $k = 5$ classes (Fig. 1). The grouping was achieved by using rounded EUR values after using quantiles over the collected sales values (when >0): "none" – equal to 0; "very low" – > 0 and < 0.03; "low" – $>= 0.03$ and

Table 2. Summary of the DSP data attributes

Context	**Attribute**	**Description** (a – TEST traffic, b – BEST traffic)
user	country	user country: 198^b or 225^a levels (e.g., Russia, Spain, Brazil)
	city	user city: 10690^a or 13423^b levels (e.g., Lisbon, Paris)
	region	region of the country: $23^{a\,b}$ levels (e.g., Asia, Europe)
	browser	browser name: $14^{a\,b}$ levels (e.g., Chrome, Safari)
	operator	mobile carrier or WiFi: 404^b or 448^a levels (e.g., Vodafone)
advertiser	vertical	ad type: 4^a or 5^b levels (e.g., video, mainstream, dating)
	campaign	ad product identification: 1389^b or 1741^a categorical levels
	special	smart link or special offer: 1018^b or 1101^b levels
publisher	account	publisher type: 8^b or 9^a levels (e.g, app developer, webmaster)
	manager	publisher account manager: 19^b or 34^a categorical levels
sale	value	value of the conversion in EUR: numeric (e.g., 0.00, 0.01, 69.34)

< 0.10; "medium" – $>= 0.10$ and < 0.30; "high" – $>= 0.30$. As shown in Fig. 1, when there is a sale, the ordinal classes are relatively balanced. Also, TEST traffic presents a lower number of sales when compared with BEST traffic.

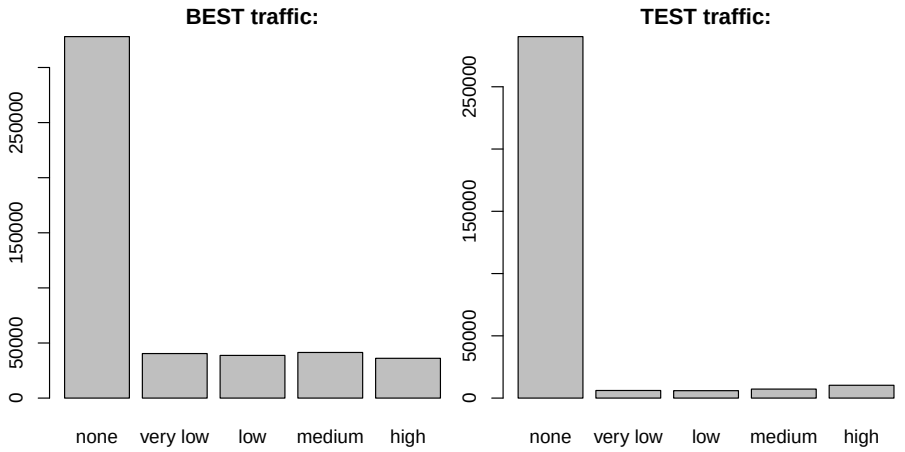

Fig. 1. Histograms for the ordinal sale classes for collected BEST and TEST datasets

2.2 Data Preprocessing and Ordinal Approaches

Since several attributes are sparse and present a high cardinality (Table 2), we applied the Percentage Categorical Pruning (PCP) transformation to all input attributes [9]. The transform works by merging the least 10% frequent levels in the training data into a "others" category. Then, the resulting values are

preprocessed using the known one-hot coding, which assigns one binary input per level. In [9], the PCP transform allowed to substantially reduce the input memory (e.g., reduction of 94% for the city attribute) and processing requirements.

For the mobile marketing data, the number of conversions (sales) is typically much lower than the number of ad clicks (redirects). While the target CVR data is unbalanced, in [9] we found that the binary deep learning MLP classifier was capable of high quality predictions even without any training data balancing method. However, when approaching the ordinal task and in particular for the realistic datasets (which present the lowest 5% conversion rate), the training data becomes extremely unbalanced. Thus, we opted to balance the realistic training data by using the SMOTE method [1], which creates new synthetic examples for the minority classes. This balancing method was only applied to training data and thus the test sets were kept with the original target values.

In this paper, we apply three approaches for the ordinal classification: Multi-class Classification (MC), regression and the k-1 ordinal approach (OA). The first approach discards the ordering and performs a simpler 5-class classification task. The second approach transforms each ordinal class into a numeric score y, where $y \in [0, 1, 2, 3, 4]$ (R1) or $y \in [0, 1, 2, 4, 8]$ (R2). The first regression scale (R1) uses equal spaced values, while the second one (R2) assigns larger distances to the highest sales ("medium" and "high"), in an attempt to favor such classes due to the minimization of squared errors. In both scales, the ordinal class is associated with the nearest scale value to the prediction (e.g., in R2 a prediction of 3.1 is assumed as the "medium" class). The third approach transforms the ordinal target, with the classes $V1 < V2 < ... < V_k$, into k-1 binary tasks [5]. Each classifier learns the probability of $P(Y > V_k)$, $k \in \{1, ..., k-1\}$. Then:

$$P(V_1) = 1 - P(Y > V_1) \tag{1}$$

$$P(V_i) = \max(0,\ P(Y > V_{i-1}) - P(Y > V_i)\ ), 1 < i < k \tag{2}$$

$$P(V_k) = P(Y > V_{k-1}) \tag{3}$$

The classifiers are independent and in a few cases we experimentally found it could occur that $P(Y > V_i) > P(Y > V_{i-1})$. Thus, we added the $\max(0, ...)$ function in Eq. (2) to avoid the computation of negative probabilities.

2.3 Deep Learning Methods

In previous work, a very competitive deep learning MLP model was proposed for binary CVR prediction, outperforming a convolution neural network and a logistic regression [9]. We adapt the same MLP, also known as Deep Feedforward Neural Network (DFFN) [6]. Let $(L_0, L_1, ..., L_H, L_O)$ denote a vector with the layer sizes with $m \in \{1, ..., H\}$ hidden layers, where $L_0 = I$ is the input layer size (I is the total number of input levels after the PCP transform) and L_O is number of output nodes. The adopted model consists of a trapezoidal shaped MLP, with $H = 8$ hidden layers of decreasing size: $(I, 1024, 512, 256, 128, 64, 32, 16, 8, L_O)$. In all hidden layers ($\{1, ..., 8\}$) we used the popular ReLU activation function, due to its fast training and good convergence properties. For multi-class classification,

the output layer contains L_O=5 nodes and the softmax function is used to output class probabilities $P(V_k) \in [0, 1]$. For the regression models, only $L_O = 1$ linear output node is used. Finally, for the $k - 1$ ordinal classification, one logistic output node ($L_O = 1$) is used, trained such that $P(Y > V_k) \in [0, 1]$.

During the training phase, we used the AdaDelta gradient function [6], which is based on a stochastic gradient decent method. Following our previous work [9], we used two approaches to avoid overfitting: dropout, which randomly ignores neural weights (dropout values of 0.5 and 0.2 for hidden layers $m = 4$ and $m = 6$); and early stopping, which stops the training when the validation error does not improve 1% within 3 epoch runs of after a maximum of 100 epochs.

Since we work with stream data, the learning models should be dynamic, assuming a continuous learning through time. We compare two MLP learning modes (proposed in [9]) for ordinal classification: reset – offline mode, when new training data is available (new rolling window iteration, Section 2.4), the whole neural weights are randomly set; and reuse – online learning, any new training starts with the previous fitted MLP weights (from previous rolling window training) and only the new input node (due to appearance of new input levels) to first hidden layer connections are randomly set.

2.4 Evaluation

The learning models are evaluated using robust rolling window validation [10], which simulates a classifier usage through time, with multiple training and test updates. In the first iteration ($u = 1$), the model is adjusted to a training window with the W oldest examples and then predicts T test predictions. In the next iteration ($u = 2$) , the training set is updated by discarding the oldest T records and adding T more recent ones. A new model is fit, producing T new predictions, and so on. In total, this produces $U = D - (W + T)$ model updates (training and test iterations), where D is the data length (number of examples). After consulting OLAmobile experts, we opted to use the realistic values of $W = 100,000$ and $H = 5,000$, which results in the model updates: $U = 43$ – collected TEST traffic; $U = 76$ – collected BEST traffic; $U = 41$ realistic TEST traffic; and $U = 49$ realistic BEST traffic.

For each rolling window iteration, we collect the test data measures and the computational effort (in seconds). We compute the F1-score, $\mathrm{F1}_k$ for each class V_k, which considers both precision and recall [13]. The global measure is obtained by using the Macro-averaging F1-score (MF1), which weights equally the F1-score for each class. We note that the ordinal classes are unbalanced, and thus other F-score averaging measures, such as micro or weight averaging, would favor mostly models that classify well the no conversion "none" class. As a secondary global measure, we adopt the Mean Absolute Error for Ordinal Classification (MAEO) [12], which computes how far (using absolute errors) are the predictions from the target (e.g., the error is 2 if the prediction is "low" and the target is "high"). This measure is often used in ordinal classification but, similarly to the micro and weight averaging F1-score measures, it tends to produce low values when the classifier is more biased to correctly predict the

"none" class. The rolling window results are averaged over all U iterations and the Wilcoxon test is used to check if paired MF1 differences are significant [7].

3 Results

The experiments were coded in Python using the Keras library [2]. We tested two types of datasets (C and R), two types of traffic (BEST and TEST), three ordinal methods (MC, R1/R2, OA), two MLP learning modes (reset, reuse) and a LR baseline model. Table 3 shows the obtained average rolling window classification results. Overall, the best learning algorithm is MLP reuse, which tends to produce the highest MF1 values, often with statistical significance when compared with MLP reset or LR model. When compared with LR, MLP reuse presents an MF1 improvement that ranges from 7 to 15 percentage points. The reuse learning always requires less computational effort when compared with the reset mode. As for the ordinal methods, the multi-class (MC) and $k-1$ OA achieve the best overall MLP reuse results, with slight F1-score differences. In general, the regression ordinal scales (R1 and R2) produce worst F1-score results. Only in two cases (R BEST and TEST), R2 obtained the best F1-score for the "high" conversion class (V_5). The $k-1$ OA method requires more computation than the MC approach. Yet, we note that in this work we used one processing core for each model, thus the OA effort could by substantially reduced if $k-1$ cores were used to fit each of its individual binary models.

4 Conclusions

In this paper, we used big data from a mobile marketing company. The goal was to predict the type of conversion rate (CVR) when an user clicks an ad, set in terms of five ordinal classes. Using a realistic rolling window validation, we compared three main ordinal methods (multi-class – MC, regression – R1/R2 and $k-1$ ordinal approach – OA) using two deep learning approaches (offline – MLP reset; and online – MLP reuse) and a logistic regression (LR) model.

The best results were achieved by the MLP reuse model and the MC and OA approaches. Such model is capable of real-time predictions. For instance, the 5,000 predictions for the C BEST MC setup require 42 s, which results in an average 8 ms per prediction. Interesting results were achieved for the collected (C) datasets, with most F1-scores above 50%, macro F1-scores of 64% and 54%, as well as a low MAEO error (lower than 0.5). As for the realistic (R) datasets (with lower amount of conversion cases), while low MAOE errors where obtained (e.g., lower than 0.30), the individual F1-scores are lower when compared with the collected datasets, resulting in an average macro F1-score of 38% and 45%.

Considering all obtained results, we recommend the MC MLP reuse model, which requires the training of just one classifier and is capable of real-time predictions. This model is potentially capable of providing value to the analyzed marketing company, since it currently does not have any method to estimate the value level of a conversion. In particular, for TEST traffic the company uses a

Table 3. Classification results (best values in **bold**; best models highlighted in gray).

Data	Traffic	Meth.	Model	Global		F1-score per class					Effort
				MAEO	MF1	$F1_1$	$F1_2$	$F1_3$	$F1_4$	$F1_5$	(s)
C	BEST (U=76)	MC	MLP reset	0.48	0.57	0.87	0.49	0.35	0.54	0.58	61
			MLP reuse	**0.45**	**0.64**[a]	**0.88**	**0.56**	**0.48**	**0.61**	0.66	42
			LR	0.51	0.57	0.86	0.51	0.38	0.51	0.59	**35**
		R1	MLP reset	0.63	0.22	0.85	0.22	0.03	0.01	0.00	46
			MLP reuse	0.61	0.24	0.84	0.30	0.04	0.00	0.00	44
		R2	MLP reset	0.77	0.28	0.83	0.00	0.00	0.31	0.24	49
			MLP reuse	1.02	0.28	0.77	0.00	0.00	0.26	0.39	47
		OA	MLP reset	0.47	0.59	0.87	0.51	0.43	0.54	0.61	117
			MLP reuse	**0.45**	**0.64**[b]	**0.88**	0.55	**0.48**	**0.61**	**0.67**	94
			LR	0.51	0.54	0.86	0.47	0.35	0.49	0.54	39
	TEST (U=43)	MC	MLP reset	0.25	0.19	0.95	0.00	0.00	0.00	0.00	46
			MLP reuse	0.19	0.51	**0.96**	**0.36**	0.22	0.45	0.55	45
			LR	0.21	0.43	0.96	0.26	0.14	0.36	0.45	**31**
		R1	MLP reset	0.22	0.22	**0.96**	0.08	0.02	0.01	0.01	51
			MLP reuse	0.22	0.23	**0.96**	0.15	0.03	0.00	0.00	45
		R2	MLP reset	0.28	0.30	0.95	0.00	0.00	0.24	0.33	50
			MLP reuse	0.27	0.31	0.95	0.00	0.00	0.24	0.34	47
		OA	MLP reset	0.21	0.31	**0.96**	0.10	0.09	0.20	0.19	118
			MLP reuse	**0.18**	**0.54**[b]	**0.96**	0.34	**0.31**	**0.50**	**0.57**	109
			LR	0.21	0.42	**0.96**	0.24	0.13	0.34	0.41	39
R	BEST (U=49)	MC	MLP reset	0.37	0.37	0.91	0.20	**0.19**	0.24	0.29	100
			MLP reuse	0.31	**0.38**[c]	**0.93**	**0.21**	**0.19**	**0.26**	0.30	**84**
			LR	0.93	0.26	0.76	0.13	0.12	0.13	0.15	120
		R1	MLP reset	0.32	0.21	0.88	0.10	0.07	0.01	0.00	106
			MLP reuse	**0.30**	0.21	0.89	0.10	0.07	0.00	0.00	87
		R2	MLP reset	0.68	0.25	0.87	0.00	0.00	0.08	0.28	104
			MLP reuse	0.61	0.26	0.89	0.00	0.00	0.08	**0.33**	88
		OA	MLP reset	0.34	0.37	0.92	0.19	**0.19**	0.24	0.30	278
			MLP reuse	0.37	0.36	0.92	0.20	**0.19**	0.22	0.25	219
			LR	0.89	0.25	0.73	0.10	0.10	0.14	0.19	109
	TEST (U=41)	MC	MLP reset	0.23	0.44	**0.96**	**0.27**	**0.26**	0.33	0.40	130
			MLP reuse	0.22	**0.45**[c]	**0.96**	**0.27**	**0.26**	**0.35**	0.41	99
			LR	0.69	0.30	0.83	0.13	0.11	0.17	0.23	156
		R1	MLP reset	0.19	0.25	0.95	0.18	0.10	0.03	0.00	132
			MLP reuse	**0.18**	0.23	0.95	0.17	0.04	0.00	0.00	103
		R2	MLP reset	0.30	0.28	0.95	0.00	0.00	0.14	0.33	130
			MLP reuse	0.29	0.31	0.95	0.00	0.00	0.15	**0.46**	**97**
		OA	MLP reset	0.21	0.44	**0.96**	0.24	0.24	0.34	0.42	303
			MLP reuse	0.21	0.44	**0.96**	0.25	**0.26**	0.33	0.42	240
			LR	0.65	0.29	0.81	0.09	0.11	0.18	0.28	123

a – Statistically significant when compared with MC reset and LR.

b – Statistically significant when compared with OC reset and LR.

c – Statistically significant when compared with MC LR.

random selection of ads, which produces much lower macro F1-scores. For instance, for the realistic TEST dataset, the random class assignment results in an average macro F1-score that is around 10%. In future work, we intend to improve the realistic ordinal results by adopting a multi-objective (F1-score for each class) evolutionary learning to train the MLP model [3].

Acknowledgments

This article is a result of the project NORTE-01-0247-FEDER-017497, supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). This work was also supported by Fundação para a Ciência e Tecnologia (FCT) within the Project Scope: UID/CEC/00319/2019.

References

1. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter **6**(1), 20–29 (2004)
2. Chollet, F., et al.: Keras. `https://keras.io` (2015)
3. Cortez, P.: Modern optimization with R. Springer (2014)
4. Du, M., State, R., Brorsson, M., Avanesov, T.: Behavior profiling for mobile advertising. In: Anjum, A., Zhao, X. (eds.) Proceedings of the 3rd BDCAT 2016, Shanghai, China, December 6-9, 2016. pp. 302–307. ACM (2016)
5. Frank, E., Hall, M.A.: A simple approach to ordinal classification. In: Raedt, L.D., Flach, P.A. (eds.) Machine Learning: EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2167, pp. 145–156. Springer (2001)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
7. Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric statistical methods. John Wiley & Sons (2013)
8. Lu, Q., Pan, S., Wang, L., Pan, J., Wan, F., Yang, H.: A practical framework of conversion rate prediction for online display advertising. In: Proceedings of the ACM ADKDD. pp. 9:1–9:9. Halifax, Canada (Aug 2017)
9. Matos, L., Cortez, P., Mendes, R., Moreau, A.: Using deep learning for mobile marketing user conversion prediction. In: Proceedings of the IEEE Int. Joint Conference on Neural Networks (IJCNN 2019). IEEE, Budapest, Hungary (Jul 2019)
10. Oliveira, N., Cortez, P., Areal, N.: 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)
11. Silva, S., Cortez, P., Mendes, R., Pereira, P.J., Matos, L.M., Garcia, L.: A categorical clustering of publishers for mobile performance marketing. In: In proceedings of SOCO'18. vol. 771, pp. 145–154. Springer, San Sebastián, Spain (Jun 2018)
12. Sousa, R.G., Yevseyeva, I., da Costa, J.F.P., Cardoso, J.S.: Multicriteria models for learning ordinal data: A literature review. In: Artificial Intelligence, Evolutionary Computing and Metaheuristics - In the Footsteps of Alan Turing, pp. 109–138 (2013)
13. Witten, I.H., Frank, E., Hall, M.A.: Data mining: practical machine learning tools and techniques, 3rd Edition. Morgan Kaufmann, Elsevier (2011)