KASANDR: A Large-Scale Dataset with Implicit Feedback for Recommendation

Sumit Sidana
Univ. Grenoble Alpes, CNRS/LIG
sumit.sidana@imag.fr

Charlotte Laclau
Univ. Grenoble Alpes, CNRS/LIG
charlotte.laclau@
univ-grenoble-alpes.fr

Massih R. Amini
Univ. Grenoble Alpes, CNRS/LIG
massih-reza.amini@imag.fr

Gilles Vandelle
Kelkoo, France
gilles.vandelle@kelkoo.com

André Bois-Crettez
Kelkoo, France
andre.bois@kelkoo.com

ABSTRACT

In this paper, we describe a novel, publicly available collection for recommendation systems that records the behavior of customers of the European leader in eCommerce advertising, Kelkoo[1], during one month. This dataset gathers implicit feedback, in form of clicks, of users that have interacted with over 56 million offers displayed by Kelkoo, along with a rich set of contextual features regarding both customers and offers. In conjunction with a detailed description of the dataset, we show the performance of six state-of-the-art recommender models and raise some questions on how to encompass the existing contextual information in the system.

CCS CONCEPTS

•Information systems →On-line advertising; Recommender systems;

KEYWORDS

Collection; E-advertising; Implicit feedback; Recommender systems

1 INTRODUCTION

Given the increasing number of possible choices available for customers, especially for on-line shopping, the need for efficient recommender systems (RS) has become essential. RS aim to capture users' (i.e. customers') personalized preferences by suggesting them a list of items (i.e. products) that might be of their interest. From this suggested list, the users provide various types of feedback on specific items that have been presented to them, allowing the system to learn and improve the quality of future recommendations.

The feedback given by a user can be of different nature, and it has evolved over time from explicit feedback, given in the form of ratings on a numerical scale, to mostly implicit feedback inferred from user's behavior, such as clicking on items, bookmarking a page or listening to a song. Implicit feedback presents several challenging characteristics such as the scarcity of negative feedback, i.e., only positive observations, clicks for instance, are available. In addition, a user listening to a song, browsing through a web page, or clicking on a product does not necessarily mean that he or she likes the corresponding item, and it is therefore impossible to measure the degree of preference from such interactions.

This paper presents Kasandr (Kelkoo lArge ScAle juNe Data for Recommendation), a novel collection that gathers one month of Kelkoo's data collected from 20 European countries. This dataset contains 16 million clicks given by 123 million customers over 56 million offers that have been displayed to them during their surf sessions. These clicks come along with contextual information, such as the geographical location of users or the hierarchical taxonomy of offers, which make the collection challenging for the design of efficient recommender systems.

The number of research articles on implicit feedback has increased in very recent years, in particular due to collections that have been mainly shared across competitions like NetFlix[2], Kaggle[3] or RecSys[4]. As for other publicly available collections, the main purpose of the proposed dataset is to encourage research on RS algorithms that scale to commercial sizes and to provide a reference based on implicit feedback for evaluation. In addition, unlike other datasets,the fields describing the data are not blurred, giving the possibility to perform interpretable feature engineering. It also contains a rich set of contextual information on users, items and the search query. Finally, while challenge dataset are expected to disappear from the web once the challenge is over, we intend to maintain Kasandr and to enrich the collection by adding data in near future. In the following, we describe Kasandr and the collecting methodology in Section 2. Section 3, presents the performance of six state-of-the-art approaches for the task of click prediction on this collection. Finally, we conclude our presentation by summarizing the contributions and discussing possible further research that can be investigated in Section 4.

2 KASANDR DATASET

This section presents the data in details and provides descriptive statistics.

[1] https://www.kelkoo.com/

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080713

[2] http://www.kddcup2012.org/c/kddcup2012-track2
[3] https://www.kaggle.com/c/outbrain-click-prediction
[4] http://2015.recsyschallenge.com/challenge.html,http://2016.recsyschallenge.com

Table 1: Description of free-available files. train_set and test_set have been created from Click and Offers for training recommender algorithms and further details are in next section.

File name	Format	Features
Page_View	csv	UserId, CountryCode, Timestamp, Url
Search	csv	SearchId, UserId, CountryCode, isPrompt, Timestamp, QueryString
Offers	csv	OfferId, OfferViewId, UserId, OfferRank, Merchant, price, Timestamp, CountryCode
Click	csv	ClickId, UserId, OfferId, OfferViewId, CountryCode, Category, Source, Timestamp, Keywords, OfferTitle
Product_Cat	xml	id and labels of product category presented as a tree
train_set	csv	UserId, OfferId, Service Type, ProductCategory, Country, Merchant, Feedback (1 or -1)
test_set	csv	UserId, OfferId, Service Type, ProductCategory, Country, Merchant, Feedback (1 or -1)

2.1 Collection of the data

The dataset records interactions of Kelkoo's customers between June, 1^{st} 2016 and June, 30^{th} 2016. It is designed to provide useful information in order to create and develop effective algorithms for the recommendation task. Kelkoo's traffic can be broadly classified according to 4 service types: (1) Ads, (2) Kelkoo's Website, (3) Kelkoo's Partners, (4) Kelkoo Feed System (KFS) which are summarized in Table 2. Kelkoo has collaboration with around 1000 partners (publishers/affiliates) on which users are advertised with offers. Various scenarios in which database at Kelkoo gets populated can be broadly classified into 4 different types:

- User visits Kelkoo's website and enters a search keyword. In this case, 1 PageView, 1 SearchView (with unique SearchId), N OfferViews (all having unique OfferViewId, where OfferViewId is the concatenation of searchId and offerId) are generated. If the user does a click, 1 ClickView (with unique ClickId) is generated.
- User browsing through Kelkoo's or partner's website is shown an ad (either a standard ad, or the user is retargeted, or on the basis of user's context, for example, the content of the page user is browsing). In this case also, 1 PageView, 1 SearchView (with unique SearchId - search keywords generated based on the ad content) and N OfferViews (1 per offer) are generated.
- User enters search keywords in Kelkoo's partner's website which does not cache offers. For each such search, a new Search_Id is generated and hence new OfferViewId is generated (as OfferViewId is concatenation of Search_Id and Offer_Id). In this case, there is no way to confirm that offer was displayed to the user.
- User enters search keywords in Kelkoo's partner's website on which offers are cached. In this case several users can see the same set of offers cached by the partner, hence, generating the same OfferViewId. In this case also, it can not be said for sure that the offer is displayed to the user.

Table 2: Counts of the number of clicks done for each service type.

Type	Ads	Kelkoo site	Partners'Api	Kelkoo Feed System
Count	597,513	1,320,958	10,396,319	2,650,391

In addition, these data present a specificity that should be taken into account while developing a recommender model: if a click is made via KFS, while the click is stored in clicks, no record gets stored in offers, thus proving them to be useless for recommender algorithms.

2.2 Structure of the data

The dataset is divided into four main databases that contain implicit feedback (offers views, clicks) of the users that have interacted with Kelkoo ads as well as a lot of contextual information (for full details, see Table 1). For privacy reasons, the UserID, name of the merchant and source were anonymized. In terms of contextual features, we can mention, the followings:

- All four main files contain information about the geographic location of the user and the timestamp of each interaction. As mentioned previously, the data were collected across 20 countries and we provide the country code associated with each user.
- The click file contains the category of clicked products. There are more than 650 categories, provided by Kelkoo, organized hierarchically (according to two levels). We provide an XML file that describes this hierarchy and contains categories' ID and label.
- The search table contains details about the users query: the string used to retrieve offers (QueryString), the list of filters apply to some of the queries to refine the search and a Boolean feature that indicates whether or not the query is filled by the user in the search box (isPrompt).

Finally, we also provide the train set and the test set used in the next section. All these files and additional details about the features can be found on-line[5].

2.3 Basic statistics

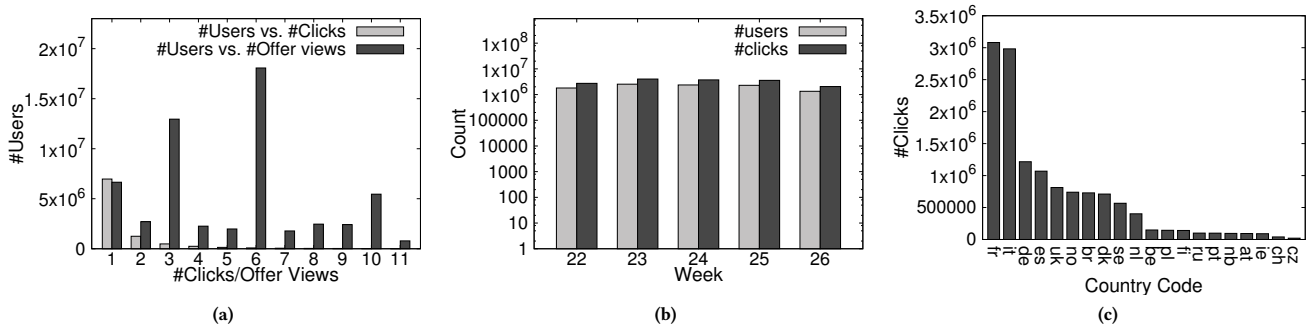
Table 3 and 4 report some basic descriptive statistics of the whole data. As outlined in these tables, we gather actions made by 123 million users over 56 million offers. In total, over the 3 billion offers displayed to those users, only 16 million were clicked resulting in the mega-sparsity of Kasandr.

Table 3: Overall Dataset Statistics: 2016-06-01 to 2016-06-30.

# of users	# of unique offers	# of offers shown	# of clicks
123,529,420	56,667,919	3,210,050,267	16,107,227

Figure 1(a) shows that the number of users falls sharply as the number of clicks rises, and most of the times either 3 or 6 offers are shown to the users. Figure 1(b) depicts how the number of users and the number of clicks vary during the month. We can see that both numbers remain stable over the weeks. In addition, as previously mentioned, the data is collected across 20 countries and

[5]http://ama.liglab.fr/kasandr/, http://archive.ics.uci.edu/ml/datasets/KASANDR



(a) (b) (c)

Figure 1: (a) Number of clicks and number of offer views vs. number of users; (b) Number of clicks and number of users who did at least one click per week; (c) Number of clicks per country.

most of the clicks are generated by France and Italy, followed by Germany (see Figure 1(c)).

Table 4: Overall Dataset Aggregate Statistics.

Sparsity	99.9999997848%
Average # of Offers Shown to 1 user	26
Maximum # of clicks done by 1 user	3,722
Minimum # of clicks done by 1 user	0
Average # of clicks done by 1 user	0.13
Average # of clicks done by 1 user (if user did at least one click)	1.71

From Table 5, one can observe that, over a month of data, very few number of users actually return to the system, when compared to the number of new users that emerge every week. This observation indicates that the time-window considered for making recommendation is important and gives information on how often a recommender model should be trained (offline) in order to provide relevant recommendations.

Table 5: Number of new users and returning users per week.

Week Number	# New Users	# Returning Users
23	36,932,009	165,951
24	26,736,201	199,467
25	22,358,876	185,749
26	13,908,242	135,303

Next, we compare several baseline methods on Kasandr. For computational reasons and as each country has a different set of offers for the customers, the investigated methods are run per country and the results are then aggregated (both in micro and macro way).

3 STATE-OF-THE-ART PERFORMANCES

Hereafter, we provide results obtained from baseline methods including non-machine learning approaches and three algorithms that have proven efficient for the recommendation task based on implicit feedback.

3.1 Compared methods

We choose three non-machine learning approaches: the random rule (Rand), that consists in recommending random items to the user, the popularity rule (Pop), that consists in recommending items with the best degree of success among all users and the past interaction technique (PastI), that consists in recommending items that the user has already interacted with. We also train 3 state-of-the-art recommender models: Matrix Factorization (MF) [3], Factorization Machines (FM) [4] and Field-Aware Factorization Machines (FFM) [2]. FFM has won two recent world-wide click-through rate prediction competitions (hosted by Criteo and Avazu). In terms of implementation, we use LIBFM and LIBFFM for FM and FFM, respectively. For MF, we use built-in implementation of Spark which is based on [1]. We implement our version of Pop and PastI. We also perform parameter tuning for the aforementioned machine learning algorithms on a different validation set and report the optimum ones in Table 6.

Table 6: Parameters used for compared approaches.

Algorithm	Optimization	#Iterations	#Latent Factors	Learning Rate	Reg Param
MF	ALS	20	50	N.A.	0.01
FM	SGD	10	1,1,10	0.001	0.01
FFM	SGD	15	8	0.2	0.001

3.2 Experimental setting

The recommendation performance of all methods is evaluated on the test set. For each user in the test set, a ranking of items (only the items that the user interacted with) is generated and the mean average precision (MAP) is computed with a cut-off of $k = 5, 30$ and 100. We recall that the average precision@k (AP) is defined as the precision (i.e. the percentage of correct items among the first k recommendations) at the position of every correct item in the ranked results:

$$AP = \frac{1}{|\mathcal{I}_{rs}|} \sum_{k \in \mathcal{I}_{rs}} P(k),$$

where $\mathcal{I}_{rs}$ is the set of relevant items selected by the algorithms. Then, the mean of these AP's across all relevant queries is the MAP. Furthermore, because we run the tested approaches per country,

Table 7: Comparison between all tested methods in terms of Micro and Macro MAP. The best results are in bold.

	Rand		Pop		PastI		MF		FM		FFM		FFM-F	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
MAP@5	2.41E-6	1.54E-005	0.004	0.004	0.017	0.011	0.044	0.037	0.721	0.814	0.732	0.829	**0.760**	**0.861**
MAP@30	4.25E-6	2.33E-005	0.004	0.005	0.017	0.011	0.044	0.037	0.726	0.817	0.736	0.831	**0.764**	**0.862**
MAP@100	5.64E-6	2.996E-005	0.005	0.005	0.016	0.011	0.044	0.037	0.726	0.817	0.735	0.831	**0.763**	**0.862**

we define macro MAP as:

$$\text{Macro MAP@k} = \frac{1}{\mid c \mid} \sum_{c \in C} MAP@k(c)$$

and micro MAP as:

$$\text{Micro MAP@k} = \sum_{c=1}^{C} \frac{n_c}{N} MAP@k(c),$$

where c, n_c and N are the country, number of users in that country and total number of users, respectively. One can observe that Micro MAP takes into account the size of the traffic within each country and gives more weight to bigger countries while Macro MAP simply averages the results obtained for all countries.

Furthermore, we only keep the users who clicked at least once and the offers which were either shown or clicked by such users. For all interactions, we assigned +1 (positive feedback) if the user clicked on an offer that was shown to him, and -1 if the user did not click (negative feedback).

Finally, we sort the data w.r.t the timestamp and further divide it into 70% for training and 30% for testing, for all recommender algorithms. Such temporal split makes more sense than random split because the interest of users change over time and is also more realistic with respect to the on-line setting.

3.3 Results

Table 7 reports MAP@5, 30 and 100 of all compared methods. As expected, non-machine learning methods namely Rand, Pop and PastI do not perform well. Similarly, we observe that MF also performs poorly when compared to FM and FFM. This result can be attributed to the fact that the number of new users in the test set is larger than the number of returning ones, and MF is well-known to fail to learn any latent factors for such users.

However, FM and its extension FFM are designed in a way that allow them to overcome this drawback and to learn from a reduced amount of positive feedback. For FFM we include the userId, offerId, country code, offer category and merchant, as fields.

Then, we also propose to compute two supplementary count features from the raw data: the number of times the user clicked, regardless of the items, and the number of time an offer is clicked, regardless of the users. This version is referred to as FFM-F in the following. As shown in Table 7, FFM-F outperforms all the other models. We believe there is still room for improvement of FFM by doing such feature engineering; for instance by including the same count but computed on different time-windows, such as per week, as for now we consider the whole month.

One can also observe that results in terms of Macro MAP for FM and all its derivatives are usually higher than the results in terms of Micro MAP. A very simple explanation comes from the fact that the latter takes into account the size of the traffic of each country, and for instance, FFM-F obtains a MAP of 0.6397 for France versus a MAP of 0.9787 for Ireland which generates less traffic.

Finally, Table 8 reports the training and testing time for each approach on all countries. Not surprisingly, non-machine learning approaches are less computationally demanding. We can also see that FFM-F is only slightly slower than FFM, as it includes the two extra quantitative features but still much more faster than MF.

Table 8: Training and testing time (in seconds).

	Rand	Pop	PastI	MF	FM	FFM	FFM-F
Train	341.759	630.112	139.409	36067.117	1142.096	1804.565	2179.745
Test	0	0	0	10259.487	444.924	462.800	490.498

4 DISCUSSION

In this paper, we presented a novel dataset in order to encourage future research on recommendation systems using implicit feedback. It is designed to investigate a wide range of recommendation algorithms as it includes many contextual features about both customers and proposed offers. For comprehensiveness, a description of side information and statistics are presented. We also conducted experiments and compared strong baselines approaches, where we observed that, FFM was the best approach for this problem. We also demonstrated that feature engineering can greatly improve the results and should be more investigated on Kasandr.

Another interesting perspective include the integration of textual information available in Kasandr using the URL to retrieve the content of the page on which the item is presented, the tag associated to it, or the query string entered by the user for his search. For this purpose, models based on text mining, semantic analysis or natural language processing can be investigated. We also left aside other features in the experimentation such as the consumer's behavior w.r.t. the type of device that s/he is using or the price of the items which we believe that they can greatly impact the performance of RS.

ACKNOWLEDGEMENT

We thank FEDER for having financed in part of the Calypso project that helped the creation of this dataset.

REFERENCES

[1] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of ICDM'08 , December 15-19, 2008, Pisa, Italy*. 263–272.

[2] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of RecSys'16, Boston, MA, USA, September 15-19, 2016*. 43–50.

[3] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[4] Steffen Rendle. 2010. Factorization Machines. In *Proceedings of ICDM'10, Sydney, Australia, 14-17 December 2010*. 995–1000.