2018-04

An Operational Consumer Profile for Real-time Scoring

Kridel, Donald; Dolk, Dan

WDSI

AN OPERATIONAL CONSUMER PROFILE FOR REAL-TIME SCORING

Donald Kridel, Department of Economics, Univerisity of Missouri, St Louis USA, 314-516-5553
kridel@umsl.edu
Daniel Dolk, Naval Postgraduate School USA, drdolk@nps.edu

ABSTRACT

Recommender systems (ReCo's) have become a familiar artifact in cyberspace as a vehicle for increasing revenues while deepening customer loyalty and satisfaction. To facilitate real-time scoring required by the ReCo, a consumer profile (CP) must be accessible via a performance store to satisfy the service level agreements (SLAs) associated with real-time scoring. There are many applications that require real-time scoring, e.g. fraud detection in payment systems and ad placement in mobile advertising. In these diverse cases, the only real distinction is the notion of the "consumer" and associated features in the CP; for example, in payment systems the CP will likely be an "account profile", while in mobile advertising the CP would be a "device profile". In this paper, we describe how an operational CP functions in mobile ad placement.

INTRODUCTION

Customer targeting in the most general sense is the process employed to determine who to "contact". In a marketing campaign, this notion of who to contact is literal, e.g., which mobile ad requests do I service for specific advertising campaigns (see e.g. [1] [2], [3])? In other applications (payment systems), the "contact" may mean: do not service this request (as it is likely fraudulent in some respect) or service this request quickly (as this is a trusted customer). In either case (or several other similar cases), real-time scoring of a request is required. To meet the associated SLAs, an operational consumer profile (CP) is required. We have described this process in detail in [4]; here we expand on that treatment by describing the process employed to make the CP operational (up to date and available). As we show below, mobile-based ReCo goes well beyond the conventional notion of cross-sell recommender systems as embodied, for example, in Amazon™ and Netflix™ ("people who bought this book/movie also bought the following books/movies") to a much more dynamic, real-time, and "big data" landscape.

The customer targeting process for mobile ads involves real-time scoring for deployed click-propensity models. In this and many applications, this requires that (parts of) the CP be "near real-time". For example, in payment systems many recent transactions (or transaction attempts) could signal a fraud attack vector. Likewise, in the current ad-tech situation, recent behavior (in terms of physical presence or digital visitation) of the device could be key in identifying potential responders or offering particular ads/offers to a specific device.

We have described in detail the process of generating the device profile (and estimating the logistic models that will be used to score requests) in [5]. The issue here is that the profiling process takes significant time (several hours) as large amounts of data are processed. In the mobile ad case, over 1 billion signals per day are processed—more than 11,000 per second. The CP is refreshed nightly. As a result, parts of the CP loaded into the performance store the previous night can quickly

become stale or out of date. In particular, the device profile is comprised broadly of the following categories: (1) device attributes (e.g., iOS or Android); (2) demographics (e.g., income and age of the device owner); and (3) device histories or pathways (e.g., physical and digital points-of-interest visited). The first two of these categories change quite slowly (nightly updates are easily frequent enough); while the third category changes very quickly throughout the day. These portions of the CP must be updated on a near real-time basis to account for these changes. This requires updating the performance cache with a separate process (Spark or Scala); during the period of cache updates, the performance cache will be "more recent" than the CP stored in the data-center.

BACKGROUND

Mobile advertising faces the same customer targeting challenges mentioned above but in a much more demanding computational environment. Mobile advertising relies heavily upon what is called *programmatic marketing* or *computational advertising* [6, 7] where the majority of campaign management processes are conducted via computing intermediaries such as ad networks, ad exchanges, supply side platforms, and demand-side platforms wherein advertisements are delivered on an individual-by-individual basis, often using auction mechanisms to balance demand and supply. In programmatic marketing situations, the problem of mobile device recognition becomes a paramount factor to consider as does the additional constraint that the ReCo must operate in real-time, and in real-time bidding (RTB) situations, extremely constrained real-time. In general, ReCo must determine whether to serve an ad to an incoming request from a mobile device and its associated user, and if so, recommend which ad from the portfolio of current campaigns should be served. Further, this requires a response time on the order of 50-100ms.

Our approach to mobile media targeting is decidedly more dynamic than the collaborative filtering (CF)-based, cross-sell recommendation engine (Carrier ReCo) we have described in [8]. With the Carrier ReCo example, we assume access to individual user data and their subsequent purchase and browsing behaviors (first party data). Subsequently, the system makes automated recommendations, identifying products and services we predict the consumer may want to buy. Dynamic targeting, as described here, considers the inverse of this situation and asks, "for specific products or services (e.g., mobile ads), which customers are most likely to respond and potentially purchase them?" In this context, the system provides a market service recommending who to target and with what ad for mobile media advertisements. This is a considerably more difficult challenge since we are looking beyond just individual purchasing behavior to include additional consumer attributes that try to capture proxies for individual utility functions.

Mobile targeting requires that we adjust the targeting set dynamically as an advertising campaign unfolds allowing us to monitor who is responding to the ads in real time, and making changes to the targeting model "on the fly" if and when required. This application requires a more sophisticated recommendation engine including a transition from traditional dimensional data management to advanced "big data" techniques. Specifically, we implement discrete choice econometric models using both logistic and multinomial logistic regressions as the automated modeling component for initially identifying customers, coupled with a near real time, data-driven *model feedback* and *model balancing* loop which monitors the ongoing progress of a campaign and periodically updates the initial model to reflect actual "in-line" consumption.

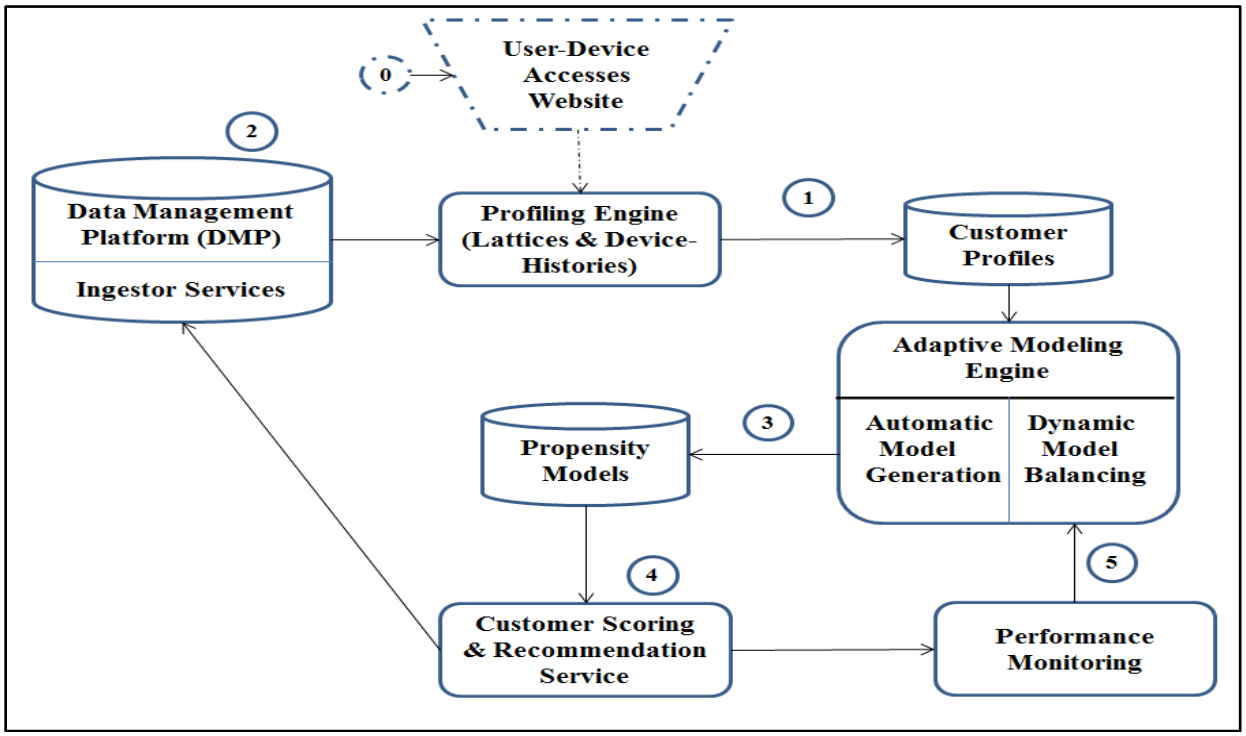
The Mobile ReCo system has extenuating data and modeling requirements. In the data domain, one of the defining characteristics of the Mobile ReCo we are describing is the immense size (millions of devices and over a billion signals) of the underlying databases coupled with the very high degree of volatility these databases undergo. Traditional dimensional data management models are ill-equipped to store and manage the large volume of data directed to the predictive models described above. Rather it is necessary to resort to highly parallel and distributed computing techniques. In addition to logging the customer activity occurring during a campaign, there will typically be a large portfolio of supporting databases that help define the CP (device profile) subsequently used in developing the propensity models described below.

Modeling in this extreme environment requires a radical departure from conventional analytic modeling and data mining approaches. For example, there will typically be on the order of a hundred active ad campaigns running simultaneously, each with its own associated propensity model. (There are also "channel" models that are used to bootstrap campaigns until there is enough campaign data to support its own propensity model.) The propensity models themselves change many times during a campaign as the model adjusts to the data streaming about who responds to an ad and who does not. The model management environment of Mobile ReCo then is a highly real-time predictive analytics-driven setting which in turn requires the automatic model generation and adaptive model balancing we have described earlier. (The modeling process is described in detail in [5].)

ARCHITECTURE

Figure 1 summarizes the structural components and the general work-flow of the demand-side platform (DSP) at a high level.

Figure 1. DSP Workflow Architecture for Mobile Advertising

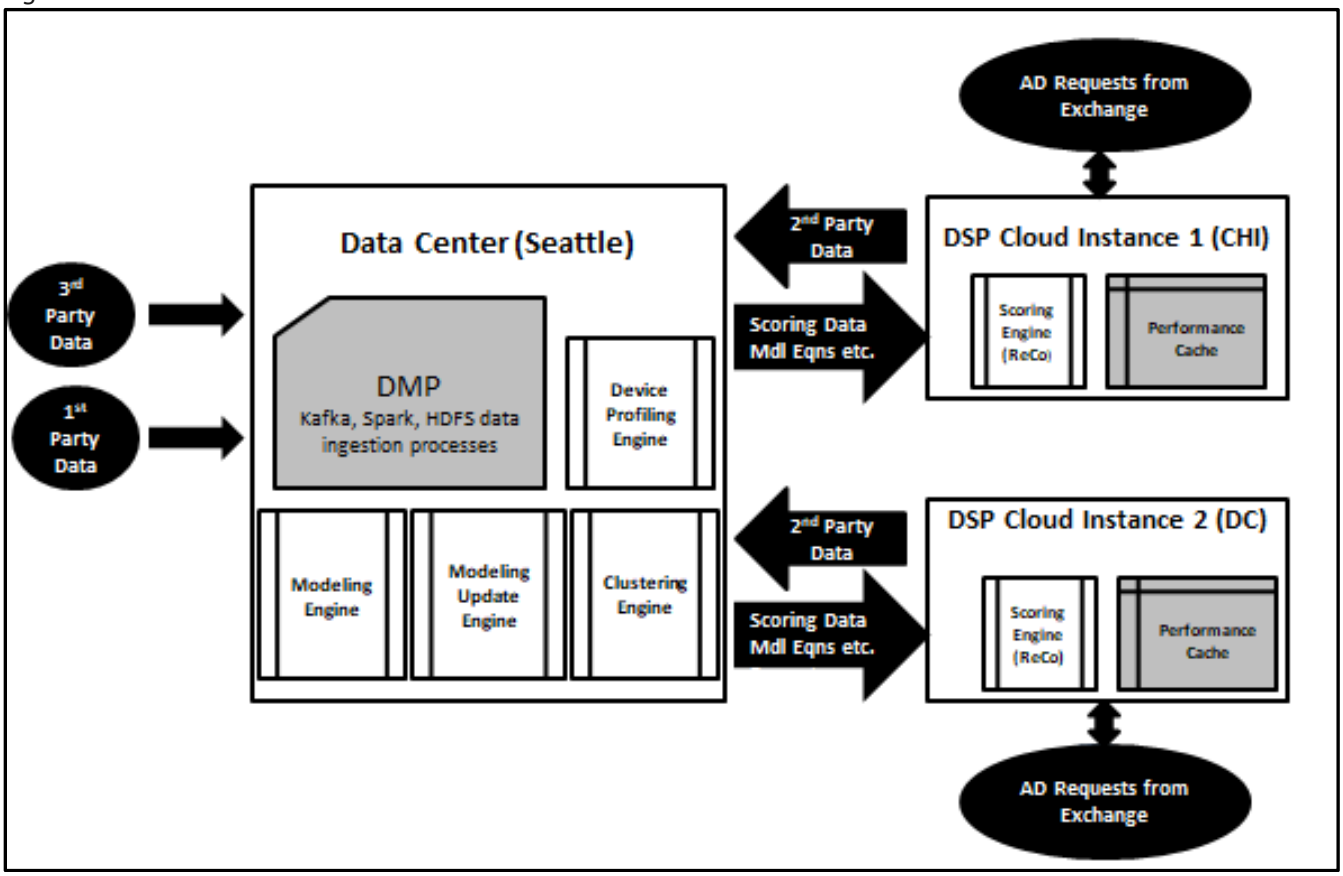


The architectural approach utilized to support our adaptive modeling approach for mobile advertising (Figure 2) implements the following general physical components:

1. Corporate Data Center
   - Data collection – 3rd party data, 2nd party data, and 1st party software development kit (SDK) data
   - Ingestion (Extract/Transform/Load (ETL) or data processing)
   - Predictive modeling (Logit and multinomial logit propensity models)
2. Cloud
   - Real-time scoring
   - Bid evaluation

In our implementation, the "heavy" ReCo lifting (i.e., DMP data ingestion, customer profiling, summarization processes, and propensity modeling) occurs in the corporate data center whereas customer scoring and bid evaluation take place in the cloud as shown in Figure 2. We utilize a lambda architecture ([9]) consisting of the batch process (occurring in the data-center) and the streaming process used to update the performance cache (for real-time scoring) as well as to feed the 1st and 2nd party data back into the batch process. As is common in these architectures, maintaining consistency across the two processes (batch and streaming) can be an intricate problem to solve.

Figure 2. DSP Architecture



To address the consistency issue, we maintain two versions in the performance cache: the batch (updated nightly from the previous day) and the 'daily' update (updated in near real-time via the streaming process). The necessary feature value (batch + stream) needed for scoring is calculated as the necessary variables are collected from the performance cache (in scoring preparation or feature preparation). The only intricacy in this particular case is maintaining the correct real-time value. For example, suppose the previous day's batch value was 10 visits to a particular POI, for example, and there were 4 additional visits during the current day. If the next batch value is 14, today's value would be reset to 0 (new visits would once again start at 1). On the other hand, if the new batch value is 12 (suggesting that 2 of yesterday's visits were so late-in-the-day that they were not included in the new batch value, then the today's value would be reset to 2. In this case (device visits to POIs), this process is straightforward enough. In cases where the streaming update contains more information about an event (in the case of payments, for example, an amount, a transaction ID, a payee), the update process needs to be more intricate to maintain all required data about the event.

The data and model dimensions of Mobile ReCo clearly require a "big data" solution with respect to hardware and software implementation. Specifically, this application requires a high degree of parallelized computing using a variety of big data solutions. The data dimension of the system uses Spark streaming in combination with Kafka which enables large amounts of data to be processed in shorter amounts of time than with conventional Hadoop-based Map Reduce jobs. The final component of the architecture supports real-time scorecard generation which assesses the user's likelihood to engage with ad content. The architecture used to perform real-time scoring utilizes a Reactor pattern and is implemented in a compiled language to avoid performance problems associated with garbage collection [10].

The field of "big data" software tools is changing quickly as might be expected in such a fast-growing field. More recent deployments are experimenting with newer tools such as Scala and Scalding which perform ETL operations, data synthesis and feature selection on the very large numbers of data signals stored in log files on a Hadoop Distributed File System (HDFS). This approach preserves the integrity of the dimensional data model while minimizing the amount of rework necessary in other software components and services. We are also actively investigating the use of GPU's (graphical processing units) which show significant promise for dramatically parallelizing all dimensions of the Mobile ReCo.

CONCLUSIONS AND FUTURE RESEARCH

We've indicated how automated adaptive modeling improves ReCo capability. However, in many cases, this requires that significant portions of the CP be updated in near real-time and that the batch and streaming portions of the CP be maintained consistently. Meeting this requirement places the class of applications we deal with squarely in in the forefront of current high-performance, "big data" computing. To gain further insight into this process of synchronizing the performance cache with the CP, we are developing a discrete event simulation model to evaluate how we can make synchronization more efficient.

REFERENCES

[1] Rossi, P., McCulloch, R., Allenby, G.. The value of purchase history data in target marketing. *Marketing Science, Vol. 15, No. 4,* 1996, pp. 321-340.

[2] Shaffer, G. and Zhang, Z. Competitive coupon targeting. *Marketing Science, 14, 4,* 1995, 395-416.

[3] Kridel, D. and Dolk, D. A self-service automated targeting portal: an example of model as a service. *Proceedings of the 38th WDSI Conference,* Lihue, HI, April 2009.

[4] Kridel, D. and Dolk, D. Automated self-service modeling: Predictive Analytics As a Service, Information Systems and e-Business Management, March 2013, Volume 11, Issue 1, pp 119-140.

[5] Kridel, D., Dolk, D., Castillo, D. Adaptive Modeling and Dynamic Targeting for Real Time Analytics in Mobile Advertising , *International Journal of Systems and Service-Oriented Engineering (IJSSOE),* Volume 7, Issue 2 (Special Issue on Big Data Systems, Analytics, Techniques, and Services), 2017, 24-39.

[6] Broder, A.Z. Computational advertising and recommender systems. *Proceedings of the 2008 ACM Conference on Recommender Systems,* ACM, New York, NY, 2008, 1-2.

[7] Yuan, S.T. and Tsao, Y.W. A recommendation mechanism for contextualized mobile advertising. *Expert Systems with Applications, Vol. 24, No. 4,* 2003, pp. 399–414.

[8] Kridel, D., Dolk, D., Castillo, D. Recommender systems as a mobile marketing service, *JSSM, Vol 6, No. 5A,* December 2013, pp. 32-48.

[9] Marz, Nathan and James Warren, J., *Big Data: Principles and best practices of scalable realtime data systems,* Manning Publications, April 2015.

[10] Schmidt, D., Stal, M., Rohnert, H., Buschmann, F. *Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects*. September 2000, Wiley.