Adaptive Modeling for Real Time Analytics: The Case of “Big Data” in Mobile Advertising

Donald Kridel
Univ. of St. Louis, Missouri
dkridel@gmail.com

Daniel Dolk
Naval Postgraduate School
drdolk@nps.edu

David Castillo
Voltari.com
David.Castillo@voltari.com

Abstract

Mobile marketing campaigns are now largely deployed through the intermediaries of demand side platforms (DSPs) who provide a performance-intensive real-time bidding (RTB) version of predictive analytics as a service. Performance thresholds are roughly 100ms for DSPs to decide whether and how much to bid for a potential client to receive a particular advertisement via their mobile device. This decision requires simultaneous access to multiple very large databases with typically millions of rows and the ability to execute multiple predictive models (e.g., logistic regression) to gauge the customer's propensity to engage. In this environment, analytic modeling must be automated via model feedback loops which adjust the models dynamically as real-time data streams in. We call this mode of analytics adaptive modeling. We detail the process of adaptive modeling from the perspective of a DSP and describe the corresponding model management environment necessary to plan, execute, and evaluate RTB campaigns.

1. Introduction

Programmatic marketing has become the predominant method of conducting advertising campaigns for mobile media. The traditional marketing relationship between publishers and advertisers has given way to a dynamic intermediary for on-line advertising in the form of ad exchanges which establish real-time markets for connecting advertisers with customers. In this on-line world, customers are not identified until they click on a publisher website which sets in motion a request for bids from current ad campaigns to present their ad to this particular individual. This is typically done through demand side platform (DSP) intermediaries who compete for the right to present the ad by submitting real-time bids to a mediating ad exchange. Ad exchanges therefore operate much the same as stock exchanges, and indeed computational advertising in this context, has much in common with high frequency stock trading because of the associated demanding performance requirements involved. The need to evaluate customer propensity via customer scoring coupled with real-time bid optimization must be done in less than half a second of elapsed time. This requires advanced “big data” technology to manage the very large data sets involved as well as the associated propensity models and bid optimization algorithms.

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. The propensity models themselves change many times during a campaign as the model adjusts to the data streaming in about who responds to an ad and who does not. Each propensity model may undergo roughly a hundred different instantiations over the lifetime of a single campaign (~30 days). This means there may be in the neighborhood of 10,000 active models in play during this overall time period.

Not only is the proliferation of models characteristic of real-time “big data” analytics, but so are the volatile datasets which feed them. In the most general setting, there may be millions of customers arriving in ad hoc fashion on hundreds of publisher websites across hundreds of advertising campaigns running in parallel. It is no longer feasible to design, build and test models using traditional lifecycle methodologies in this environment. We describe below an alternative scenario for “on the fly”, dynamic real-time modeling based upon automated models, which adapt to real-time customer behavior via model feedback loops. We term this process adaptive modeling, and we describe how this works from the perspective of a simplified rendition of a demand-side platform player in a real-time bidding setting for mobile advertising. Nevertheless, even this somewhat stripped down version of a DSP operation is sufficient to illuminate “big data” requirements for implementing dynamic, adaptive modeling and how this redefines the model management landscape for extreme computing environments.
2. Background: Mobile RTB Landscape

Figure 1 provides a simplified illustration of the structure of display online and mobile advertising. The dotted line between Publisher and Advertiser shows the historical direct sales relationship (note the Publisher and Advertiser may be the same for companies which run their marketing “in-house”). Programmatic marketing “intermediates” between the two parties via ad networks which aggregate advertisers and publishers, and ad exchanges which balance demand and supply in ad networks. On-line and mobile advertisers and publishers now work extensively, although not exclusively*, with ad exchanges through demand-side platforms (DSP) and supply-side platforms (SSP), facilitated by real-time bidding (RTB) as follows:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>A request for an ad is created when a user accesses a publisher’s web page (e.g., an airline site).</td>
</tr>
<tr>
<td>Step 2</td>
<td>The bid request is then passed to ad exchanges using the SSP software with information about the request such as the mobile device used (e.g., iPhone 5S), the time-of-day (TOD), day-of-the-week (DOW) and location (e.g., zip code = 63122).</td>
</tr>
<tr>
<td>Step 3</td>
<td>The ad exchange contacts DSPs and asks, “How much are you willing to bid for this request?”</td>
</tr>
<tr>
<td>Step 4</td>
<td>The DSPs each compute a bid (or not, if the ad request is not deemed sufficiently promising) which is submitted to the ad exchange which in turn determines the winning bid (usually the second highest bid).</td>
</tr>
<tr>
<td>Step 5</td>
<td>The winning bid is returned to the Publisher via the exchange.</td>
</tr>
<tr>
<td>Step 6</td>
<td>The winner's ad is displayed to the user on the Publisher's website in the form of an impression.</td>
</tr>
</tbody>
</table>

Response times for this process are very stringent with Steps 3 and 4 typically occurring in the range of 40 to 100-ms and the overall elapsed cycle time less than 100ms so that the ad is presented seamlessly to the user.

* Ads are still done via exclusive Advertiser-Publisher relationships as well as via SSPs only, however DSPs enjoy the advantage of reaching a potentially much wider audience.

![Diagram of ad network and ad exchanges for RTB](image)

Figure 1. Simplified structure of ad network and ad exchanges for RTB (adapted from [11])

Terminology adopted by the computational and mobile advertising community is summarized in Table 1.

<table>
<thead>
<tr>
<th>ENTITY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertiser</td>
<td>Owner of advertising campaigns</td>
</tr>
<tr>
<td>Publisher</td>
<td>Owner of medium (e.g., website) for online ad displays</td>
</tr>
<tr>
<td>Inventory</td>
<td>Set of publisher on-line sites where ads are displayed</td>
</tr>
<tr>
<td>Ad Network</td>
<td>Technology platforms connecting advertisers with publisher inventory</td>
</tr>
<tr>
<td>Ad Exchange</td>
<td>Technology platforms that facilitate buying and selling of mobile advertising inventory from multiple ad networks via real time bidding (ex: Google’s Double Click™).</td>
</tr>
<tr>
<td>DSP: Demand Side Platform</td>
<td>Companies which essentially provide “big data” predictive analytics as a service primarily to the “demand-side” of the industry, i.e., the agencies and advertisers (ex: AppNexus™, Voltari™).</td>
</tr>
<tr>
<td>SSP: Supply Side Platform</td>
<td>The supply side counterpart to DSP consisting of software platforms to sell display, video and mobile ads in an automated fashion via RTB (ex: OpenX™, RightMedia™)</td>
</tr>
<tr>
<td>Ad Request</td>
<td>Generated when mobile device user accesses Publisher's website.</td>
</tr>
<tr>
<td>Impression</td>
<td>Created when an ad is displayed to the user's mobile device on the Publisher's website.</td>
</tr>
</tbody>
</table>
The focus of this paper is on Steps 3 and 4 of Figure 1, namely bid computation for a DSP under very stringent time constraints, in the context of mobile advertising. We further restrict the domain to cookie-less computations. Although cookies potentially provide broader insight into past user purchase behavior and follow-on actions from impressions which can be useful in scoring customers, there are substantial issues associated with leveraging cookies in a mobile environment. Cookie-based approaches have been replaced with probabilistic mobile device fingerprints and are an important piece of the mobile ad landscape (see [8], e.g.) and a research area of continuing interest.

There are two separate but highly related dimensions to bid computation: customer scoring and bid strategies. Once an bid request has been received and broadcast, a DSP must evaluate the quality of the request, that is likelihood of that user responding to an ad (note there will typically be dozens or more ad campaigns underway at any point in time). This requires sophisticated real-time propensity modeling techniques that perform some version of customer scoring. Once scoring has been done, then the issue arises of how much to bid on that impression and this, in part, is dependent upon the bidding history up to that point in the campaign. Recalling that there will be multiple DSPs bidding at once, there are nevertheless significant opportunity costs associated with bidding too low or too high over the course of a campaign. We deal with each of these dimensions below, both of which are demanding “big data” challenges and radically accelerate the requirements for analytic modeling.

3. Propensity Modeling and Real-Time Customer Scoring

In the mobile advertising arena, propensity modeling is the “flip side” of conventional customer targeting as implemented in direct mail campaigns for example, wherein a list of prospects is provided a priori and prospects are ranked according to various “propensity to buy” criteria (see [5], e.g.). In the RTB environment by contrast, customers show up in an ad hoc manner whenever they access a publisher website. This requires the DSP to search large databases in highly constrained real-time for relevant information about a specific customer in order to determine their propensity to respond to any number of ongoing advertising campaigns. This, in turn, requires real-time access to a large number of probability models in the DSP’s model repository. In effect, it is necessary to fully automate the targeting and associated propensity modeling process.

Automating the propensity modeling process requires a more flexible approach to model building than has typically been deployed in conventional analytical modeling and predictive analytics. Specifically, we need to be able to rapidly specify and solve on-demand, context-sensitive models using smart model formulation techniques that identify data variables, relationships and functional form. The system challenge is to generate models with statistical integrity for datasets that are unknown a priori. This, in effect, requires an automated modeling functionality, that in turn requires the ability to generate statistically robust and valid models “on the fly”.

Automated modeling is achieved via model templates designed to generate model instances, and model feedback loops employing dynamic model recalibration to ensure that each propensity modeling is tracking the “real world in real time” as closely as possible. We term this overall process adaptive modeling and describe how this works below. As we shall see, adaptive propensity modeling in the mobile advertising arena requires extreme “big data” functionality.

The propensity models allow the DSP to rank (score) each bid request coming from mobile advertising exchanges against the engagement objectives set by the clients. The compelling unique value proposition is that the propensity models are able to calculate the predicted behavior or response a user will have, at a specific moment in time, to a specific offer. All environmental variables, including behavior, location, content, device type, and time of day are considered and evaluated for influence and inclusion within the model. Data utilized in the modeling process is described below.

a. Data Dimension: Data Ingestion

Data ingestion is the process of obtaining, processing, integrating and importing data into one or
more databases (often data warehouses). This process involves extensive extraction, transformation and loading (ETL) procedures which attempt to integrate disparate data sources into a cohesive whole. Data ingestion typically requires “big data” technology to handle the extremely large data volumes that are involved. This is especially true in the mobile advertising RTB environment with its stringent real-time processing constraints. Four main sources of data are relevant:

- **Bid request data**: When an individual accesses a publisher website, the following basic information is generated: TOD/DOW (time of day/day of week), mobile device, website accessed and high level geo-location. This is extremely volatile data with typically millions of bid requests received on a daily basis.

- **Tier 1, or 1st Party**: customer data which the advertiser or publisher owns. This may contain critical information about customer purchase history and browsing behavior as well as customer segments, usage, and expenditures. Although such data can provide very valuable insight into an individual’s propensity to buy a particular product or service, availability is often restricted as advertisers are understandably reluctant to divulge such information for proprietary and privacy reasons. Again, there may be millions of customers alone, depending upon the size of the advertiser, not to mention all the various transaction data which is generated every day (think of corporations such as AT&T, American Express, and Amazon, for example).

- **Tier 2, or 2nd Party**: data in the DSP eco-system collected about customers over multiple campaigns as well as bidding histories. This data may include impressions, customer click-through history, possible engagement activity (follow-ups to click throughs), and previous campaign involvement as well as bidding streams for multiple campaigns detailing bids submitted, won or lost, and the winning bid if it was the DSP’s (the winning bid is not always made available to the DSPs who did not win the bid). These are also very large and volatile databases which grow larger on a daily basis as more and more campaigns are executed.

- **Tier 3, or 3rd Party**: data refers to external databases which a DSP may purchase to provide additional demographics (e.g., Axiom, Experian, Nielsen), points of interest (POIs), device information, industry-specific information (e.g., Polk database of automotive and marketing products), and social media. These databases tend to be a smaller and less volatile on average.

For example the Axiom database has 300M people in its file.

The ingestion process runs continuously as new data arrives. The ETL processes differ depending upon which data sources are involved. 2nd party data (and 1st party internal SDK data) are the most challenging since they result from near-real time processes that can generate hundreds of millions, or even billions, of transactions per day which is decidedly on a “big data” scale. 3rd party data on the other hand is considerably easier to manage because they are updated less frequently, and therefore the associated ETL procedures do not need to be run continuously.

b. **Model Dimension: Automated Propensity Modeling via Adaptive Model Feedback Loops**

Analytical models such as those encountered in traditional econometrics and optimization applications are inherently feedback-driven, however they are not always recognized as such. Figure 2a shows a version of the traditional “old school” modeling life cycle that implies a stepwise, iterative, “human in the loop” progression through several major phases of the process. Typically, for a model of any significant complexity whether in decision analytics or software engineering, each phase may consume a nontrivial amount of time often measured in weeks or months. Further there is an assumption of a high degree of human-model interaction required in the overall development. This time-honored method of building models which are reliable, robust, and thoroughly validated is no longer feasible in the current world of highly dynamic, “near real time” decision analytics. This process is too static, too time-consuming, and too slow to respond to a rapidly changing environment.

The problem is even more acute in the “big data” world of real time decision analytics, especially in real-time bidding marketing campaign applications. Cycles must be tightened dramatically to the point that the “human-in-the-loop” is no longer a possibility on a continuous basis. With millions of new data points streaming in over very short time intervals, models must not only be running continuously but they must also be able to adjust to changing conditions “on the fly”. The “old school” lifecycle now takes place in a few milliseconds compared to days or weeks.

Another way of characterizing this contrast in modeling dynamics is in terms of system “deep structure” [10], where “deep structure” refers to relatively stable, non-volatile components of a system (such as database schemas and “surface structure” to volatile components such as rows of a database
which frequently get updated. In “old school” scenarios, models typically comprise the “deep structure” and data the “surface structure”. In “new school” applications, however, models become “surface structure” in addition to data. The “deep structure” component is a rule-driven knowledge base which generates model templates from which model instances are created. The rules in the knowledge base remain very stable but the models to which they give rise are constantly changing in accord with the constantly changing real-time datasets.

Automated models in this setting, i.e. the model templates and model instances which they subsequently generate, are necessary to accommodate the demanding response time requirements, and thus are better viewed as signal processors in a closed cybernetic control system. Figure 2b shows the real-time analytics modeling life cycle which involves tight model feedback loops and real time adaptation. This is necessary because the model for scoring impressions (customers) as they arrive, will eventually decay and must be recalibrated periodically to adapt to real-time circumstances.

Figure 2. Conventional vs. real-time model analytics lifecycles

Our approach to dynamic statistical modeling for mobile media applications is a blend of “old school” and “new school” as follows:

1. Use traditional rules of model-building practices (as shown in Figure 2a) to develop reliable and robust model templates for customer scoring (in particular, a discrete choice logistic regression for determining the likelihood of a specific impression to respond to a specific ad). Each model template will be used to generate model instances via feature selection from incoming datasets.

2. As an advertising campaign unfolds and data are collected in real time, recalibrate the model instances automatically “on the fly” to reflect who is responding to the advertisement and who is not. The concept is that this feedback-driven adaptive modeling approach will refine the initial model instances iteratively by absorbing real time events, recalibrating the current model instance at specified intervals, and dynamically redefining the resultant targeting market parameters as the campaign proceeds (Figure 2b).

- **Model template**

The model template is built from a knowledge base which contains sets of “rules” that are used to guide the automated development of the propensity model(s). These rules have been developed as the result of extensive applied research we’ve conducted on automated econometric modeling [3, 5, 6]. The knowledge base is used to build executable model templates that are instantiated dynamically. The template contains rules for data-filling and transformations, “include” rules for deciding which variables to test during model development, and “keep” rules for determining whether the tested variables remain in the final model [6]. Typically, the dependent variable in the model template is binary indicating whether the customer clicked, or 2nd clicked, on the campaign ad. Independent variables (IDVs) are identified from a large set of potential variables which are present in all three of the major data sources. Model templates may contain as many steps as desired and each step may contain a virtually unlimited number of variables. There may be hundreds or even thousands of these variables initially depending upon the nature of the campaign which are subsequently culled to approximately 100-150 final IDVs through an iterative winnowing process called feature selection. This winnowing process entails the execution of hundreds of different logit regression models that are dynamically refactored until all template rules have been satisfied until a final IDV set is converged upon. Because of the very large number of impressions usually encountered, a sample is drawn from the set of impressions.

“Standard” modeling templates (e.g., retail acquisition) are available by default; custom templates may be built, either because of statistical “preferences” of a client or because the situation is unique. The modeling template “fully controls” the development of the automated dynamic model-
building process. The model that is finally deployed is a result of many (potentially hundreds of) different model-tests employed during the model development process as controlled by the model templates.

Once propensity models have been developed, they must be activated for media buying. Our methodology uses real-time scoring to determine the audience fit to a combination of dimensions representing that unique mobile moment. These dimensions include device, publisher, ad content, time of day, and location. When the bid request is received, real-time scoring computes a set of probabilities that the user will engage in campaign content (ads). Typically there may be 100 active campaigns underway at any point in time. Each campaign has its own unique set of propensity models reflecting the most important predictors (and their associated elasticity’s) of the audience that is engaging (or not engaging) with the campaign content. The scoring engine ranks the scorecards for all active campaigns and selects the highest ranking campaign scorecard (if and only if, the engagement propensity is above a specific threshold) as the one upon which to bid to serve to the user (Table 3). Wall clock scoring time averages 10 milliseconds per bid request. Note that there will be multiple DSPs bidding on a single request and there will only be a single winner, typically the DSP which submits the 2nd highest bid (see the section below which describes the bid determination process).

- **Recalibration**

Recalibration modifications include, in principle, both respecification of independent variables and computation of the associated regression coefficients. Altering the contents of the IDV set dynamically is a topic for further research.

Recalibration is workflow-driven in that campaign-specific propensity models are queued up for recalibration at set intervals, roughly in accord with how new the campaign is. Newer and thus smaller campaigns data-wise will be updated more frequently in the beginning (as often as four times a day) until enough impressions are received to stabilize the model whereas more mature and thus larger campaigns data-wise may only be recalibrated less frequently. As new impressions increase the number of observations up to a pre-specified maximum level, impressions arriving after that require one of three forms of sampling: fully randomized, retain only the most recent observations, or some hybrid thereof. The assumption is that more recent observations are the most relevant in recalibrating models. We are investigating the issue of optimal refresh times in order to determine the tradeoffs that accrue between scalability and the number of model refreshes.

- **Model evaluation**

The “gold standard” in measuring advertising campaign effectiveness is lift [1] which measures in one form or another the improvement in response resulting from the application of predictive analytics compared to random targeting in generating a mailing list, prospect database, or in our case, scoring customers as they appear. Figure 3 shows a representative simplified lift chart for a campaign. Model deciles are a measure of customer scores with respect to propensity to engage where the higher the decile, the more promising the prospect. In the diagram below, the lift rate is substantial (about a factor of 2-3) for the high decile prospects, and negative for prospects below the 3rd decile. This highlights the critical role of generating accurate propensity models. The most common metric currently used is CTR where the “average” industry response rate for random targeting is in the range of 0.4%. An active area of research is to develop richer criteria for engagement than just CTR; this is where the use of cookies is beneficial to track what level of engagement may have subsequently taken place (e.g., 2nd click, web page browsing, purchase, etc.)

Another important aspect of model evaluation is to determine overall campaign model effectiveness, given the fact that there will typically be dozens, if not hundreds of model versions over the lifetime of a campaign because of the recalibration process. Our default approach is to use the very last model used in
a campaign as the one that is stored in the model archives for future reference, based upon the assumption that the last model dealt with the most current impressions and thus best reflects “current reality”. However it is debatable whether this a reasonable assumption and it may be revealing to do further analyses to compare which model versions performed best for particular time periods as well as over the entire lifetime of the campaign. This could also potentially shed light on the marginal benefits of model recalibration and help determine whether there are optimal or near-optimal refresh times.

2. Adjust the initial bid price to reflect the customer score and the relative appeal of this impression. (Note that the initial bid will be 0, i.e. “no bid”, if the customer score is low and deemed unpromising.)

This overall process is usually called bid optimization [2, 4] although not all bidding models employ optimization in the operations research sense.

One important aspect of bid optimization is the existence of external campaign budget constraints which affect the bidding strategy. Campaigns typically run on the scale of 30-90 days and will have an allocated budget specified. Although the budget may change over the course of a campaign as advertisers revise their goals depending upon marketing success or lack thereof, a constant budget is usually assumed over the lifetime of a specific campaign. One of the main concerns is campaign pacing which is concerned with how rapidly the budget is spent as a campaign unfolds. Overpacing occurs when the budget is spent prematurely before the campaign’s end date; underpacing occurs when there is budget left over. In either case, there is an issue of opportunity cost. With overpacing, for example, there may be promising impressions near the end of the campaign which can’t be accessed for lack of budget but which may be more promising than those for which winning bids were placed. Conversely with underpacing, a conservative bidding strategy, for example, may leave quality impressions “on the table” so to speak, which otherwise would have been promising to pursue. Advertisers typically want some degree of pacing implemented in their campaigns to avoid spending all their budget in the first few days of a campaign.

Factors with an impact on pacing include time of day (TOD), day of week (DOW), and display frequency. TOD warrants consideration because there will be peak times during the day when mobile users access their devices and down times (mostly during the night and early morning) when they don’t. Allocating more budget to the former intervals increases the probability of receiving higher quality impressions. Similar arguments apply for DOW. Several models have been presented to enforce budget smoothing such that campaign spending aligns well with demand peaks [7, 12]. The goal is to reach the end of the campaign and the end of the budget more or less simultaneously while optimizing impression quality in the process. Typically, these approaches divide the campaign into distinct time intervals based on expected demand, and then specify a budget cap accordingly for each interval.

4. Bid Optimization

Whereas customer scoring is calculated entirely from customer attributes, bid price computations must also take into account the bidding context of a campaign as it unfolds. Although propensity modeling constitutes the crux of bid determination, there will be many DSPs bidding on each impression and it’s possible that competing strategies can overwhelm the customer scoring dimension. As a campaign unfolds, it is essential for a DSP to gauge how well it is doing in the overall bidding marketplace. Too aggressive a bidding strategy may exhaust budget prematurely with the attendant opportunity cost of having to forego quality impressions which arrive later in the campaign, whereas too conservative an approach may also overlook quality impressions. Just as in the customer scoring scenario where it is imperative to track how well the models are performing, so it is in bidding strategies. Strategies need to be adaptive and responsive to real-time data streaming.

Thus there is effectively a related, but separate, two-step process of bid determination (Figure 5):

1. Compute an initial bid price depending upon campaign constraints (e.g., budget smoothing and pacing requirements) and real-time analytics of the bidding stream to detect possible patterns in the bidding time-series.

Figure 3. Generic Lift Diagram
Once bid optimization determines an initial bid price, an adjusted bid price is determined based upon the Publisher, TOD/DOW, device, and high-level geo-location data associated with the bid request. Using 3rd party, 2nd party (behavioral) and optionally 1st party data as described in the section above, a Customer Score is then determined as the highest scoring campaign of all the campaigns for which the ad can be served on the impression device at that time. If the highest score > minScore, then this campaign is forwarded along with its index value, or score. The initial bid is then adjusted depending on the robustness of the score. In general, if the score is high then the bid is adjusted higher, otherwise if the index is marginal, the initial bid may remain unchanged or even decremented. B1 may be adjusted even further (B2) to accommodate any pacing constraints in effect which are designed to facilitate budget smoothing over the duration of the campaign. Also, minimum and maximum allowable bids may be determined by the ad agency a priori to initiating a campaign and these may further constrain the final bid price.

Most of the literature on bid pricing assumes that customer scoring is the more important factor and bid optimization operates on the margin. However, recent research suggests that bidding strategies may be equally as important as customer scoring in successful campaigns, at least for sponsored search auctions [12]. This is a fruitful area for future research where we plan to explore to ascertain what actual impact of various bidding strategies has on the overall success of a campaign.

Figure 4 provides an overall schematic and summary of the data, models, and processes discussed above. The combination of customer scoring and bid optimization in extreme real-time presents very unique computational challenges. In the next section, we discuss some of the hardware and software suite currently implemented to address this “big data” application.

5. Architecture

The architectural approach utilized to support adaptive modeling for mobile advertising implements the following general components:

- Data collection – 2nd party data, and 1st party SDK data
- ETL or data processing
- Predictive modeling

- Real-time scoring

Data collection implements a cascading technology using Spark streaming and Kafka. Spark is similar to Hadoop but performs its tasks in memory. Kafka is an open source high throughput distributed “publish subscribe” messaging framework. The combined Kafka and Spark framework enable high performance data aggregation on large scale datasets. A number of ETL and data processing tasks are performed using Kafka backend services in conjunction with Spark. This enables large amounts of data to be processed in shorter amounts of time than with conventional Hadoop-based Map Reduce jobs.

The architecture for predictive modeling uses two forms of processing. Initial model training is performed using Hadoop Map Reduce services. Among these services are Feature Selection and Propensity Modeling. Feature selection employs a typical Map Reduce framework for pruning the training dataset into the set of most important variables. Propensity modeling accepts the output of Feature Selection and iterates until all model conditions have been satisfied and generates a scorecard that is used by the real-time scoring engine.

The second type of processing is used by propensity modeling and relies on a Kafka and Spark implementation. Kafka and Spark within this context of propensity modeling provides a “near real-time” approach for performing adaptive modeling. It fuels a process whereby the propensity model’s predictors are updated with new coefficients and elasticities. As noted in an earlier section, predictors are not dropped or added during this phase of the process. Rather the focus is to fine tune the scorecard until the next version of the training data scorecard arrives and begins the process all over again.

The final component of the architecture supports real-time scoring which assesses the user’s likelihood to engage with ad content. Real-time scoring is better described as “near real-time” scoring and is responsible for identifying the device, collecting the device profile associated with the device, and scoring the user profile against potentially hundreds of campaign scorecards within a 40 to 100 millisecond timeframe. Moreover, real-time scoring also is responsible for computing the bid price once the user probabilities have been computed. 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 [9].

894
6. Conclusions

We’ve presented an exemplar of an extreme big data application for demand-side platform (DSP) processing in a real-time auction bidding scenario for mobile display advertising. The very large and volatile datasets coupled with an equally large portfolio of dynamic propensity models and bid optimization strategies in place at any point in time completely transforms the model management landscape. Conventional modeling lifecycles no longer apply and the new model dynamics rely heavily upon adaptive modeling with the following characteristics:

- Automated modeling: model generation must be done “on the fly” using deep structure templates developed from best modeling practices. Propensity models using logistic regressions with large numbers of observations and independent variables for predicting customer behavior are required for DSP operations.

- Continuous model feedback loops: model effectiveness decays over the lifetime of an advertising campaign and must be continuously monitored and recalibrated periodically to reflect current customer behaviors. This recasts the model lifecycle in the paradigm of system dynamics, cybernetics and control theory.

- Real-time, streaming analytics: Models are dynamically driven by real-time data streams requiring advanced analytics capabilities. Bid optimization, for example, requires sophisticated algorithms from the areas of signal processing, machine learning, control theory, and traditional operations research optimization modeling.

- Model storage and retrieval: the huge numbers of models and associated model instances which proliferate during even a single campaign must be stored in a well-organized model repository for future reference and must be retrieved using some form of case-based reasoning techniques to identify models relevant to the initiation of new campaigns.

We have complemented our elicitation of dynamic model requirements with specific approaches we have adopted for addressing some of them, particularly propensity modeling and customer scoring. There are many research challenges remaining in this field, a few of which we summarize in Table 3. Extreme big data applications such as the DSP environment we have described push analytics to the very edge and constitute an exciting new era for model and data management.
Table 3. Sample of research issues for real-time analytics in mobile ad displays.

<table>
<thead>
<tr>
<th>TOPIC</th>
<th>ISSUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model refresh strategies</td>
<td>• What is a “near optimal” refresh schedule by TOD, DOW, campaign phase, and other criteria?</td>
</tr>
<tr>
<td></td>
<td>• How to implement “in line” refreshes using suitably compressed model proxy which can be computed in near real-time?</td>
</tr>
<tr>
<td></td>
<td>• How to implement dynamic, in-line model decay detection (vs. current pre-specified time intervals)</td>
</tr>
<tr>
<td>Model evaluation</td>
<td>• Which model version(s) to save in repository?</td>
</tr>
<tr>
<td></td>
<td>• How well did each model version perform in its time slot?</td>
</tr>
<tr>
<td></td>
<td>• How effective are model recalibrations in general? What is the relationship between recalibrations and overall campaign performance?</td>
</tr>
<tr>
<td></td>
<td>• Post facto, can we use genetic algorithms to determine optimal or near-optimal regression models by time interval and for the overall campaign? How do these optima compare to actual in terms of campaign performance?</td>
</tr>
<tr>
<td>Model versioning</td>
<td>• Should we save all model versions generated during a campaign, only the most successful ones, or only the last one (current default)?</td>
</tr>
<tr>
<td></td>
<td>• How to represent models in the repository for efficient and effective retrieval in subsequent campaigns? Some form of case-based reasoning?</td>
</tr>
<tr>
<td>Bid optimization</td>
<td>• Can we leverage well-known signal processing algorithms such as fast Fourier transforms and or control theory techniques to predict bidding behavior, and if so, how?</td>
</tr>
<tr>
<td></td>
<td>• Bid smoothing pacing: do bid smoothing pacing strategies really matter? If so, how to implement intelligent bid smoothing algorithms?</td>
</tr>
<tr>
<td></td>
<td>• What are the relative impacts of bid optimization vs. customer scoring? Which is the more important factor to consider when determining a bid and to what extent?</td>
</tr>
<tr>
<td></td>
<td>• Comparison of bidding strategies: spend as fast as you can; spend only during most desirable windows; spend according to a smoothing schedule? Construct genetic algorithm to test hybrid strategies.</td>
</tr>
</tbody>
</table>

7. References