Grace: A Platform for Optimized 1:1 Marketing at Scale

Better marketing through decision trees of multi-armed bandits

Jesse Hersch
Globys Inc., Seattle
jhersch@globys.com

Scott Miller
Globys Inc., Seattle
smiller@globys.com

Luca Cazzanti
NATO CMRE, La Spezia, Italy
luca.cazzanti@cmre.nato.int

Brian Flynn
Globys Inc., Seattle
bflynn@globys.com

Oliver Downs
Globys Inc., Seattle
odowns@globys.com

ABSTRACT

Grace is a software platform for digital marketing that automates the process of empirical evaluation and optimization of marketing campaigns targeting hundreds of millions of customers with thousands of messages. It measures the lift in performance produced by various messages using controlled experiments, and it uses a novel combination of learned decision trees and multi-armed bandits to target each customer with the right message to maximize lift. The exploration/exploitation trade-off is managed automatically by employing a Bayesian approach, Thompson sampling. Performance of the platform is demonstrated with a simulated example, and with a deployed implementation within a prepaid telecom company with millions of subscribers, where Grace generated tangible business impact by increasing revenue compared to business as usual.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; G.3 [**Probability and Statistics**]: Probabilistic algorithms

Keywords

multi-armed bandit, Thompson sampling, decision tree, contextual marketing, experimental design

1. INTRODUCTION

1.1 Overview

In the world of marketing, it is well known that different people respond differently to different advertisements. Obviously, showing the same ad to every target will not produce the best response.

Ideally, marketers would like to craft the optimal message for every target where possible. In theory, there are a number of platforms where this is possible, such as mobile phones and websites. However, there are two fundamental challenges that make this difficult in practice.

First, a metric needs to be selected that defines what it means to be "optimal," and that metric needs to be measured in a reliable way. Different marketing campaigns may have different goals, leading to different choices for the key performance indicator (KPI) to be optimized. For example, one could optimize for short-term revenue, full customer life-cycle revenue, or customer loyalty. Some KPIs are more easily measured than others, but in any case in order to measure the effect of a given campaign on the KPI we need to compare with a control group. Without control groups, it is impossible to know if a given campaign is better or worse than doing nothing.

Once sound metrics and controlled trials have been set up, the second challenge is to select messages for each target to optimize the KPI. In an age of telecoms and websites with hundreds of millions of subscribers, such a task is daunting. On the other hand, with such large subscriber bases, even a small improvement can add up to substantial revenue.

In this paper we will describe *Grace*: a software platform that addresses these two challenges in a scalable way, so that customer bases in the hundreds of millions can be served with marketing in a way that is optimal for each individual. Grace automates the design and execution of experiments over thousands of market segments and messaging variations, to evaluate the effectiveness of all these combinations at a much faster pace than previously possible. Grace discovers the most influential attributes of subscribers and messages that drive performance, and finds pockets where some types of messages are particularly effective or ineffective against specific types of individuals. Finally, Grace actively adjusts which messages are sent to whom based on these discoveries, to maximize the KPI over the whole target population. It automatically balances the competing needs to *explore* the space of responses for improved learning, and to *exploit* that knowledge to increase performance.

In this way, Grace relieves much of the burden from the marketer to evaluate and optimize campaigns. Grace allows marketers to spend their time working on higher-level activities, finding creative ways to take advantage of the intelligence that Grace acquires for them.

The methodology behind Grace — including the experimental framework, the message selection procedure, and the learning algorithm — is detailed in Section 2. Then in Section 3 we present simulation results on a small example for illustration of how the system works, and in Section 4 we report actual results from deploying Grace with a mobile phone operator in the EMEA region. We close with concluding remarks in Section 5.

To be concrete, our discussions will focus on the application of Grace in the context of a mobile telecom operating in the pre-paid space. Grace could be used in other contexts of course. It is ideally suited to a situation where the relationship between marketer and target is individual in nature, but there are a huge number of individuals, a rich source of data about them, and frequent interaction with them, such as with mobile operators, financial services, or e-commerce.

1.2 Related work

The core technology of Grace is a decision tree of multi-armed bandits. The multi-armed bandit [14] is a commonly used model for decision-making under uncertainty, analogous to a bank of slot machines. Each of a finite number of actions (pulling the arm of a slot machine) produces a separate random reward according to an unknown distribution, and the objective is to maximize a cumulative reward over some time horizon.

In the context of Grace, the "arms" of the multi-armed bandit are the messages that may be sent to a collection of users that are expected to have similar behavior. Thus, each message may have a different reward distribution depending on which type of user receives the message. To organize this vast array of arms, Grace uses a decision tree that branches over user properties and message properties.

Under some technical conditions, an optimal policy for selecting actions for the multi-armed bandit is given by the *Gittins index* [8, 21]. However, the Gittins index solution can be difficult to compute and it can suffer from incomplete learning, continuing to select suboptimal actions indefinitely with positive probability [4]. A simple and popular heuristic solution to multi-armed bandits is *Thompson sampling* [18], also called probability matching; see [16] for a survey of this approach. This technique chooses actions by randomly sampling the posterior distributions of unknown parameters of the reward functions and choosing the winning sample. The posteriors are then updated using Bayes' rule as new reward observations are collected. For this reason, multi-armed bandits played according to Thompson sampling are sometimes called "Bayesian bandits."

Thompson sampling is not only a simple heuristic to implement; it has also performed well in experiments [6]. Recent theoretical work on optimal regret bounds [1, 2, 5] may justify its robust performance, and there is an argument for Thompson sampling as a sound approach to more general decision and inference problems [12]. It has been applied to search query recommendation [9] and online advertising [15], and Google Analytics Content Experiments uses multi-armed bandits with Thompson sampling to test alternatives for web content [17]. Thompson sampling scales reasonably well with the number of arms since the sampling and updates of the reward distributions can be done independently, and the coordination procedure amounts to a simple max function. Still, applications of multi-armed bandits tend to be limited to determining optimal choices over the whole population, not tailoring the choice to each user as desired for 1:1 marketing.

Contextual bandits allow the reward to depend on exogenous variables such as user features, but algorithms seem to be limited to models where the dependence is linear [3, 11] or smooth enough to be represented by kernels [10, 20]. Grace models an arbitrary dependence of reward on context without any smoothness assumptions, through the use of decision trees.

Decision trees have also been used in marketing applications [13]. Standard machine learning techniques can generate decision trees in an automated and scalable way; however, decisions made with them are deterministic, and there is no accommodation of the exploration/exploitation tradeoff inherent in decision problems with incomplete information.

From an algorithmic perspective, the novel contribution of Grace is the combination of decision trees and multi-armed bandits to provide a scalable framework for decision-making under uncertainty.

2. METHODOLOGY

2.1 Users and messages

Consider a service provider that has many millions of users, and provides a pay-as-you-go type service where the more users consume the service, the more they pay. A typical example of such a service would be a mobile telecom carrier in the pre-paid space.

The carrier would like to maximize revenue. In this example, revenue comes from users consuming the service. So, in order to increase revenue, the carrier must in some way convince their users to consume more of the service. One way to attempt this is for the carrier to conduct marketing on their user base. The carrier has detailed knowledge of each user, so this marketing can be very fine-grained. For instance, the carrier has access to the user's social graph within the network, how the graph changes in time, billing and usage time series, and so on. If one could make use of this data in a scientifically sound way, the gains can be significant.

We define a *User* as a collection of scalar attributes. The attribute values maybe continuous (e.g., age or credit score), discrete (e.g., number of calls in a day), or categorical (e.g., gender or preferred language). We treat discrete-valued attributes the same as continuous-valued attributes in the training and use of decision trees.

Age and gender are examples of simple attributes. A user may have other scalar attributes which are not simple in nature, such as one indicating membership in a cluster. Say we cluster users on some number of attributes, and then store the resulting cluster membership of each user in a new categorical attribute. We call such attributes *derived* since they are derivative of other attributes. User attribute values, whether simple or derived, are entirely determined by the behavior and properties of a user (or the user's connections, in the case of attributes that are "socially aware" such as the degree of an ego network or PageRank).

Now define a *Message* as a collection of scalar attributes that define marketing message one could present to a user. For example, "Buy X within Y days and we'll give you Z." Here X, Y, and Z are all scalar attributes of the message, which could be either continuous or categorical. Further the message could be worded in different ways but with the same content, producing another categorical attribute, Message-Tone say. Other messages may have no tit-for-tat aspect at all, rather they are purely informative or educational: "Did you know that you can save on data with one of our Data Packages?" All the types of messages and their attributes may combine to produce many thousands of variations.

Grace allows the marketer to define eligibility rules predicated on user attributes. For example, marketers may want to allow the Data Package message above only for users with age greater than 18, say. The marketer will typically define a set of alternative messages or parameterized messages, along with eligibility rules shared by all messages in the set. This combination is called a *targeting group*, and the definition of targeting groups is the primary mechanism for the marketer to influence the operation of Grace.

The task then, is this: given one user out of many millions, pick the one message out of many thousands that maximizes a desired metric, revenue say. Further, the message must be picked so that performance against the metric can be measured in a scientifically sound way. This leads us to the topic of targets and controls.

2.2 Targets, controls, and experimental design

When one does experiments on a population of subjects, it is standard practice to use control groups. A control group is what allows the experimenter to draw causal inferences between treatments and outcomes. In this regard, marketing is no different than a clinical drug trial.

Grace uses the targeting groups created by the marketer as a starting point for defining the experimental units for assembling target and control groups. Users within the same targeting group may be meaningfully compared with one another since they are eligible for the same messages. Grace refines the targeting groups further into subgroups that have significantly different responses, as identified by the decision tree generated from the previous marketing iteration. These subgroups, called *contexts*, form the experimental units within which targets and controls are compared to produce estimates of the average *lift* in KPI the targets exhibit over controls. This subdivision of targeting groups into contexts is a way of reducing the variance of the lift estimates.

Targeting groups are also used to separate effects of different, simultaneous experiments. For example, if a user receives a message from one targeting group, they would be ineligible to receive messages from other targeting groups, until a sufficient lockout period has elapsed. Control groups are kept segregated from each other in the same way: if a user is a control in one targeting group today, they will be prevented from being a control in any other targeting group until a lockout period has elapsed. Also, switching between target and control from one day to the next is prevented by Grace, for obvious reasons.

In any experiment, the marketer can choose the proportion of targets to controls. For example, in the early stages of a campaign, marketers may choose to be conservative with a 50/50 or even lower split. As confidence grows, this ratio can be dialed up. As Grace is assigning users to target and control groups, it keeps track of this ratio, making assignments as needed so that the actual ratio tracks the desired ratio.

The marketer can also specify contact limits for each experiment, and for the system as a whole, e.g., no more than 3 messages per week per user.

Besides using control groups to measure the effect of marketing experiments, Grace uses controls in the performance metric to be optimized when choosing a message for each user. Given a specific KPI chosen by the marketer, such as 14-day revenue, Grace aims to optimize *total expected lift*, i.e., the expected gain in KPI in the target population compared to the control population. The model we use to select messages to maximize lift is a decision tree of Bayesian bandits, as described in the next few sections.

2.3 Bayesian bandits for message selection

Call the set of users U and the set of messages that can be sent to them M. If we know the expected reward $\mathrm{E}\{r \mid u, m\}$ for sending message $m \in M$ to user $u \in U$, then the optimal policy is simply to send the one with the highest expected reward:

$$\operatorname*{argmax}_{m \in M} \mathrm{E}\{r \mid u, m\}.$$

However, since we do not know the expected reward, we have a multi-armed bandit problem. In that problem, there are a number of "slot machines" with unknown payoff rates, and the task is to find a strategy for playing the machines that maximizes a (possibly discounted) expected reward over a future horizon. There is an inherent exploration/exploitation trade-off typical of decision problems with incomplete information.

As discussed in Section 1.2, optimal policies for multi-armed bandits are impractical to compute, but a Bayesian approach that naturally addresses the exploration/exploitation trade-off is Thompson sampling, a.k.a. randomized probability matching. In this framework, the uncertainty in our estimates of expected rewards is quantified by posterior distributions of mean rewards given the observed data. That is, $\mathrm{E}\{r \mid u, m\}$ is replaced by a random variable $\bar{r}(u, m)$ with known posterior distribution. The selected message is then the random variable

$$\operatorname*{argmax}_{m \in M} \bar{r}(u, m)$$

which is easily sampled by drawing samples of mean rewards $\bar{r}(u, m)$ for each m and picking the m that provides the highest mean reward. When enough data is collected that there is a clear "winner" among the mean reward distributions, then this selection rule chooses the optimal message; otherwise, other messages are sometimes selected, allowing the mean reward distributions to be refined with more data.

In our marketing application, however, it is impractical to maintain separate distributions of $\bar{r}(u, m)$ for each u and m, as there may be on the order of 10^8 users and 10^4 messages. Instead, we structure the space $U \times M$ with a partition $\Pi \subset 2^{U \times M}$ defined by a binary decision tree, and approximate the mean reward as identically distributed within each subset. Each $\pi \in \Pi$ has a single average reward $\bar{r}_\pi$, which in the Bayesian bandit framework is a random variable.

Effectively, Π defines a smaller multi-armed bandit, with each subset π corresponding to a single arm, which is related to the original multi-armed bandit as follows. When arm π is "played" for a given user u, a message is selected at random from

$$M_\pi(u) := \{m \mid (u, m) \in \pi\}$$

with uniform probability, and sent. The average payoff for u is therefore

$$\bar{r}_\pi(u) := \frac{1}{|M_\pi(u)|} \sum_{m \in M_\pi(u)} \bar{r}(u, m). \qquad (1)$$

We cannot maintain separate distributions for $\bar{r}_\pi(u)$ for each u, though, so we need to average over U in some way. We select $p(U)$, a probability measure on U which reflects the distribution of user attributes in the population, so that $\int \bar{r}_\pi(u)\,dp(U)$ reflects the average per-user reward for playing bandit arm π over the population. By playing the multi-arm bandit to maximize this average, we aim to maximize the total KPI over the whole population. In practice, $p(U)$ is derived from the emprical distribution of available samples of user attributes in π.

The construction of the binary decision tree defining Π is discussed in Section 2.5. Each interior node of the tree splits on some user attribute or message attribute, and each $\pi \in \Pi$ is identified with a leaf of the tree. The unique subset $\pi(u,m) \in \Pi$ containing a given pair (u,m) is obtained by walking the pair down the tree, taking whichever branch applies to u or m at each node, until a leaf is reached.

To summarize,

$$\bar{r}_\pi := \int \frac{1}{|M_\pi(u)|} \sum_{m \in M_\pi(u)} \bar{r}(u,m)\,dp(U)$$

is the random mean reward associated with each subset $\pi \in \Pi$ acting as an arm in a multi-armed bandit, and

$$m^*(u) = \operatorname*{argmax}_{m \in M} \bar{r}_{\pi(u,m)} \tag{2}$$

is the randomized probability matching selection rule. Algorithmically, the procedure for selecting messages using the decision tree of multi-armed bandits executes the following steps for each user u:

1. For each $m \in M$ for which u is eligible, walk (u,m) down the decision tree to find π and sample $\bar{r}_\pi$ to get reward $\hat{\bar{r}}(u,m)$.
2. If u is a target (not control), send message $m^* = \operatorname{argmax}_m \hat{\bar{r}}(u,m)$.
3. Update the distribution for $\bar{r}_{\pi^*}$ with observed reward, where $(u,m^*) \in \pi^*$.

In our marketing application, we do not observe the reward (lift in the chosen KPI) immediately; typically lift would be measured over days or weeks. So instead of incremental Bayesian updates in step 3 we re-estimate distributions in periodic batches, with data history windows much longer than the delay required to measure the reward. The computation of mean lift distributions is the subject of the next section.

2.4 Lift computation

Because mean *lift* is the reward we maximize with the multi-arm bandit, each subset $\pi \in \Pi$ needs to contain samples from the target population *and* the control population. Moreover, lift cannot be measured directly for any (u,m) pair. Instead, the samples from $U \times M$ are divided randomly into targets and controls, and what is observed for each (u,m) pair is either the target KPI ρ^t or the control KPI ρ^c. Furthermore, as discussed in Section 2.2, each sample is designated as belong to a particular *context*, which is a subset of U defined with the intention that targets within a context should produce similar responses to messages, and likewise for controls. This allows us to estimate mean KPIs reliably for targets and controls within a context with a reasonable number of samples.

For a given subset π, let $\Gamma_\pi \subset 2^U$ be the set of contexts represented by samples in π. Let $\mathcal{R}^t_{\pi,\gamma}$ be the set of samples of KPIs for (u,m) pairs designated as targets in $\pi \cap (\gamma \times M)$ for a context $\gamma \in \Gamma_\pi$, similarly define $\mathcal{R}^c_{\pi,\gamma}$ for control samples, and denote their sizes as $N^t_{\pi,\gamma} = |\mathcal{R}^t_{\pi,\gamma}|$ and $N^c_{\pi,\gamma} = |\mathcal{R}^c_{\pi,\gamma}|$. Given enough independent samples of targets and controls in a context, we model our uncertainty about mean KPIs in a context as normal random variables centered at sample means:

$$\bar{\rho}^x_{\pi,\gamma} \sim \mathcal{N}(\mu^x_{\pi,\gamma},\, (\sigma^x_{\pi,\gamma})^2)$$

$$\mu^x_{\pi,\gamma} = \frac{1}{N^x_{\pi,\gamma}} \sum_{\rho \in \mathcal{R}^x_{\pi,\gamma}} \rho \tag{3}$$

$$(\sigma^x_\pi)^2 = \frac{1}{N^x_{\pi,\gamma}(N^x_{\pi,\gamma} - 1)} \sum_{\rho \in \mathcal{R}^x_{\pi,\gamma}} (\rho - \mu^x_{\pi,\gamma})^2 \tag{4}$$

where x is either "t" or "c" in the above expressions. The mean lift for a context is then

$$\bar{r}_{\pi,\gamma} = \bar{\rho}^t_{\pi,\gamma} - \bar{\rho}^c_{\pi,\gamma},$$

whose uncertainty is normally distributed:

$$\bar{r}_{\pi,\gamma} \sim \mathcal{N}(\mu_{\pi,\gamma},\, \sigma^2_{\pi,\gamma})$$

$$\mu_{\pi,\gamma} = \mu^t_{\pi,\gamma} - \mu^c_{\pi,\gamma} \tag{5}$$

$$\sigma^2_{\pi,\gamma} = (\sigma^t_{\pi,\gamma})^2 + (\sigma^c_{\pi,\gamma})^2, \tag{6}$$

assuming target and control samples are drawn independently.

Since the per-context mean lift $\bar{r}_{\pi,\gamma}$ models a conditional expectation $\mathrm{E}[r \,|\, (u,m) \in \pi,\, u \in \gamma]$, the total mean lift in the subset π is a weighted sum of $\bar{r}_{\pi,\gamma}$ over $\gamma \in \Gamma_\pi$. If the contexts were disjoint subsets, the weights would be probabilities of the contexts under some distribution; however, the contexts are *not* disjoint. Nevertheless, for the purpose of assessing the lift that has already occurred within the target population, a natural choice of weight for a context is the fraction of the target population contained in that context:

$$\bar{r}_\pi = \sum_{\gamma \in \Gamma_\pi} \left(\frac{N^t_{\pi,\gamma}}{N^t_\pi} \right) \bar{r}_{\pi,\gamma}, \tag{7}$$

where $N^t_\pi = \sum_{\gamma \in \Gamma_\pi} N^t_{\pi,\gamma}$ is the number of target samples in π. We use the *target* population fraction instead of the total population fraction (including controls) because lift is attributed to targets only. With this weighting, $N^t_\pi \bar{r}_\pi$ produces the correct value for total lift over the subset π. Moreover, this is consistent with computing mean KPIs separately for target and control over all contexts then taking the difference, if the control population fractions are close to the target population fractions, i.e.,

$$\frac{N^c_{\pi,\gamma}}{N^c_\pi} \approx \frac{N^t_{\pi,\gamma}}{N^t_\pi}$$

which they should be if the target/control assignment for a user is chosen randomly and independent of the user.

For the purpose of predicting mean lift from future messaging, it is possible that the context weights should be different. It depends on whether one expects the distribution of users in the future to be much different from the past. As mentioned in Section 2.3, in practice we take the probability measure $p(U)$ to be the empirical distribution of observed samples, so the target population fraction remains the natural weighting. Consequently, the uncertain mean lift over a subset π is described by a normal distribution:

$$\begin{aligned}\bar{r}_\pi &\sim \mathcal{N}(\mu_\pi,\, \sigma^2_\pi)\\ \mu_\pi &= \sum_{\gamma\in\Gamma_\pi}\left(\frac{N^t_{\pi,\gamma}}{N^t_\pi}\right)\mu_{\pi,\gamma}\\ \sigma^2_\pi &= \sum_{\gamma\in\Gamma_\pi}\left(\frac{N^t_{\pi,\gamma}}{N^t_\pi}\right)^2\sigma^2_{\pi,\gamma}.\end{aligned}$$

As we will see in the next section, the procedure for generating the decision tree ensures that there are sufficient target and control samples in each branch to support this approximation.

There is a subtle technical issue with the normal approximation of mean KPIs from sample statistics (3) and (4): for the statistics to be unbiased, the samples should be drawn from the same distribution used to define the mean and variance. There is no concern in the space of users because the chosen probability measure $p(U)$ reflects the sample distribution of users (or user attributes). However, in the space of messages there is a potential discrepancy between the distribution of messages in the samples of π and the uniform distribution used in (1) that arises from how the bandit is played. It may happen that because of selections based on previous decision trees one message is under-represented in π, and the predicted mean KPI for that message will be governed mostly by the observed KPIs of other messages in π. This bias can be corrected by using weighted averages in (3) and (4), with weights inversely proportional to the prevalence of the respective messages in the samples of π.

2.5 Decision tree generation

Given a partition Π, the mean reward for a given u under our Bayesian bandit selection rule (2) is

$$\begin{aligned}\bar{r}_\Pi(u) &:= \mathrm{E}\big[\max_{\pi\in\Pi}\bar{r}_\pi\big] && (8)\\ &= \sum_{\pi\in\Pi}\mathrm{E}\big[\bar{r}_\pi \,|\, \bar{r}_\pi = \max_{\pi'\in\Pi}\bar{r}_{\pi'}\big]\,\Pr\big\{\bar{r}_\pi = \max_{\pi'\in\Pi}\bar{r}_{\pi'}\big\} && (9)\end{aligned}$$

Ideally, the decision tree should be generated in a way that tries to define a Π that maximizes the total mean reward $\int \bar{r}_\Pi(u)\,dp(U)$ while limiting the size of Π. Globally optimizing this metric is obviously difficult, so we use a heuristic branching scheme to generate the tree that tries to maximize an upper bound for the effect of a local split on the global reward (9).

The tree is generated recursively, starting with a single-node tree representing the trivial partition $\Pi = \{U \times M\}$. At each step, a leaf node of the tree, representing a subset in the current partition, is split according to some predicate on users or messages (e.g., u.Age>35 or m.Tone = ‘Urgent’), creating a refined partition that splits subset π into π_1 and π_2. In a fashion similar to the composition across contexts (7), the mean lift for π (before it is split) can be written in terms of the mean lifts for π_1 and π_2 as follows:

$$\bar{r}_\pi = \left(\frac{N^t_{\pi_1}}{N^t_\pi}\right)\bar{r}_{\pi_1} + \left(\frac{N^t_{\pi_2}}{N^t_\pi}\right)\bar{r}_{\pi_2}. \qquad (10)$$

The expression (10) will be compared to estimates of the mean lift achieved after a split on user or message attributes, to decide which split should be performed on a given leaf node of the decision tree. The effect on mean lift depends on the type of split, so they are considered separately below.

Split on messages.

If the split is on messages, the mean reward $\bar{r}_\pi$ is replaced by

$$\bar{r}'_\pi = \max\{\bar{r}_{\pi_1}, \bar{r}_{\pi_2}\}$$

in (8). The local gain in mean reward,

$$\Delta_{\pi_1,\pi_2} := \bar{r}'_\pi - \bar{r}_\pi,$$

is an upper bound on the achieved gain in $\bar{r}_\Pi$, since we would have to have $\bar{r}_\pi = \bar{r}_\Pi$ for the local gain Δ_{π_1,π_2} to be fully realized. Nevertheless, we will use Δ_{π_1,π_2} as a way to compare the merits of different splits.

Setting $\alpha = N^t_{\pi_1}/N^t_\pi$ for convenience, we have

$$\begin{aligned}\Delta_{\pi_1,\pi_2} &= \max\{\bar{r}_{\pi_1}, \bar{r}_{\pi_2}\} - [\alpha\,\bar{r}_{\pi_1} + (1-\alpha)\bar{r}_{\pi_2}]\\ &= \max\{(1-\alpha)(\bar{r}_{\pi_1} - \bar{r}_{\pi_2}),\ \alpha(\bar{r}_{\pi_2} - \bar{r}_{\pi_1})\}.\end{aligned}$$

Define the random variable

$$\varepsilon := \bar{r}_{\pi_2} - \bar{r}_{\pi_1} \sim \mathcal{N}(\mu_{\pi_2} - \mu_{\pi_1},\ \sigma^2_{\pi_1} + \sigma^2_{\pi_2}),$$

and the truncated expectations

$$\begin{aligned}\bar{\varepsilon}_+ &:= \mathrm{E}[\max\{\varepsilon, 0\}]\\ \bar{\varepsilon}_- &:= \mathrm{E}[\min\{\varepsilon, 0\}] = \mu_{\pi_2} - \mu_{\pi_1} - \bar{\varepsilon}_+.\end{aligned}$$

Then we have

$$\begin{aligned}\Delta_{\pi_1,\pi_2} &= \max\{-(1-\alpha)\varepsilon,\ \alpha\,\varepsilon\}\\ &= \max\{-(1-\alpha)\varepsilon,\ 0\} + \max\{\alpha\,\varepsilon,\ 0\},\end{aligned}$$

because the terms in the first max have opposite signs. The expected gain in mean lift is therefore

$$\begin{aligned}\mathrm{E}[\Delta_{\pi_1,\pi_2}] &= -(1-\alpha)\bar{\varepsilon}_- + \alpha\,\bar{\varepsilon}_+\\ &= \bar{\varepsilon}_+ + (1-\alpha)(\mu_{\pi_1} - \mu_{\pi_2}). \qquad (11)\end{aligned}$$

Split on users.

If the split is on users according to predicate $P(u)$, such that

$$(u,m) \in \pi_1 \quad \Leftrightarrow \quad (u,m) \in \pi \text{ and } P(u),$$

then the new mean lift over π is distributed according to

$$\begin{aligned}\bar{r}'_\pi \sim{}& \mathcal{N}(\mu_{\pi_1},\, \sigma^2_{\pi_1})\,\Pr\{P(u)\,|\,\exists m : (u,m) \in \pi\} +\\ & \mathcal{N}(\mu_{\pi_2},\, \sigma^2_{\pi_2})\,\Pr\{\neg P(u)\,|\,\exists m : (u,m) \in \pi\},\end{aligned}$$

where the probability is defined using the empirical measure $p(U)$. This is a simple Gaussian sum, since the predicate $P(u)$ is independent of the reward. The interpretation is that when a given $(u,m) \in \pi$ is evaluated for the multi-arm bandit, the mean lift is assigned as a sample of either $\bar{r}_{\pi_1}$ or $\bar{r}_{\pi_2}$ as appropriate, instead of a sample of $\bar{r}_\pi$ which is a blend of the two. The expected lift over *all* (u,m) pairs within π does not change, since $\alpha = \Pr\{P(u)\,|\,\exists m : (u,m) \in \pi\}$. However, there can be an effect on the global mean lift $\bar{r}_\Pi$ through interactions with mean lifts from other subsets.

For the purpose of obtaining an upper bound on the effect of the split, suppose $\bar{r}_\pi = \bar{r}_\Pi$ (as in the discussion of splits on messages), and further suppose there is another subset $\pi' \neq \pi$ from the partition such that $\bar{r}_{\pi'} = \bar{r}_\pi$. Then any increase in $\bar{r}_{\pi_1}$ or $\bar{r}_{\pi_2}$ relative to $\bar{r}_\pi$ will appear as an increase in $\bar{r}_\Pi$, but decreases will not affect $\bar{r}_\Pi$. The gain in mean lift Δ_{π_1,π_2} under these conditions is therefore positive.

The expected gain is given by

$$\mathrm{E}[\Delta_{\pi_1,\pi_2}] = \alpha\,\mathrm{E}[\delta_{\pi_1}] + (1-\alpha)\,\mathrm{E}[\delta_{\pi_2}]$$

where

$$\delta_{\pi_1} := \max\{\bar{r}_{\pi_1} - \bar{r}_\pi,\ 0\}, \quad \delta_{\pi_2} := \max\{\bar{r}_{\pi_2} - \bar{r}_\pi,\ 0\}.$$

Substituting $\bar{r}_\pi = \alpha\,\bar{r}_{\pi_1} + (1-\alpha)\bar{r}_{\pi_2}$ and $\varepsilon = \bar{r}_{\pi_2} - \bar{r}_{\pi_1}$, we have

$$\delta_{\pi_1} = \max\{-(1-\alpha)\varepsilon,\ 0\}, \quad \delta_{\pi_2} = \max\{\alpha\varepsilon,\ 0\}$$

which implies

$$\mathrm{E}[\delta_{\pi_1}] = (1-\alpha)(\bar{\varepsilon}_+ + \mu_{\pi_1} - \mu_{\pi_2}), \quad \mathrm{E}[\delta_{\pi_2}] = \alpha\,\bar{\varepsilon}_+.$$

Therefore, the expected gain from the split is

$$\mathrm{E}[\Delta_{\pi_1,\pi_2}] = \alpha(1-\alpha)(2\bar{\varepsilon}_+ + \mu_{\pi_1} - \mu_{\pi_2}). \tag{12}$$

Selecting the split.

To summarize, when splitting a leaf node π, candidate splits (π_1, π_2) are compared according to $\mathrm{E}[\Delta_{\pi_1,\pi_2}]$, an upper bound on the increase in mean lift obtained by the splits, defined by (11) or (12) depending on whether a given split is on message attributes or user attributes. The split that produces the greatest value is applied, producing two new leaf nodes and a refined partition. The algorithm continues splitting leaf nodes until there are no nodes with sufficient target and control samples to justify splitting.

Both expressions for $\mathrm{E}[\Delta_{\pi_1,\pi_2}]$ depend on the quantity $\bar{\varepsilon}_+ = \mathrm{E}[\max\{\varepsilon, 0\}]$ which indicates the degree to which $\bar{r}_{\pi_2}$ exceeds $\bar{r}_{\pi_1}$. It may be computed from the following formula. If $x \sim \mathcal{N}(\mu, \sigma^2)$ then

$$\mathrm{E}[\max\{x, 0\}] = \mu\,\Phi\Big(\frac{\mu}{\sigma}\Big) + \sigma\,\phi\Big(\frac{\mu}{\sigma}\Big),$$

where ϕ and Φ are the pdf and cdf of the standard normal distribution $\mathcal{N}(0,1)$. Despite the asymmetry in $\bar{\varepsilon}_+$, the expressions for $\mathrm{E}[\Delta_{\pi_1,\pi_2}]$ are symmetric in the sense that they are invariant to exchanging $(\pi_1,\ \alpha)$ with $(\pi_2,\ 1-\alpha)$. Hence, $\mathrm{E}[\Delta_{\pi_1,\pi_2}]$ really depends on the absolute separation of the distributions of $\bar{r}_{\pi_1}$ and $\bar{r}_{\pi_2}$.

For the results presented in the next two sections, we used a different measure of absolute separation to rank candidate splits:

$$d(\pi_1, \pi_2) := \frac{|\mu_{\pi_1} - \mu_{\pi_2}|}{\sqrt{\sigma^2_{\pi_1} + \sigma^2_{\pi_2}}}. \tag{13}$$

This metric may be seen as a proxy for $\mathrm{E}[\Delta_{\pi_1,\pi_2}]$, given the symmetry of Δ_{π_1,π_2} and the scaling of the mean differences by the standard deviation inherent in the computation of $\bar{\varepsilon}_+$. However, $d(\pi_1, \pi_2)$ does not explicitly depend on the population fraction α, so splits based on $d(\pi_1, \pi_2)$ sometimes have a tendency to isolate small subpopulations from the rest, resulting in smaller aggregate effects on the total mean lift $\bar{r}_\Pi$. Despite this flaw, Sections 3 and 4 show that trees generated using $d(\pi_1, \pi_2)$ have demonstrably improved the mean lift when used for messaging decisions.

The decision tree generation algorithm is summarized in Algorithm 1. In this description, categorical attributes of users or messages are separated into a set of boolean attributes, while continuous attributes are split along a set of split points determined by Chickering's K-tile method [7]. To handle missing data in either continuous or categorical attributes, we use the MIA method approach discussed by Twala [19].

Algorithm 1 Decision Tree Generation

```
1: Set $\Pi = \{U \times M\}$.
2: Choose a leaf node $\pi \in \Pi$ with a sufficient number of control and target samples for splitting. Our current threshold is 500 samples each. If there is no such node, stop.
3: for each user or message attribute $a$ do
4:     if $a$ is boolean then
5:         Split $\pi$ into $\pi_1^a$ and $\pi_2^a$ according to $a$ and $\neg a$ (which includes samples with no value for $a$), and set $d_a = d(\pi_1^a, \pi_2^a)$.
6:     else                                  ▷ $a$ is continuous
7:         for each split point $a_k$, $k = 1, \ldots, N_a$ do
8:             Split $\pi$ into $\pi_1^{a_k}$ and $\pi_2^{a_k}$ according to $a > a_k$ and $\neg(a > a_k)$ (which includes samples with no value for $a$), and set $d_{a_k} = d(\pi_1^{a_k}, \pi_2^{a_k})$.
9:             Split $\pi$ into $\pi_1^{a_k}{}'$ and $\pi_2^{a_k}{}'$ according to $a < a_k$ and $\neg(a < a_k)$ (which includes samples with no value for $a$), and set $d'_{a_k} = d(\pi_1^{a_k}{}', \pi_2^{a_k}{}')$.
10:        end for
11:        Set $d_a = \max\{d_{a_k}, d'_{a_k} \,|\, k = 1, \ldots, N_a\}$ and set $\pi_1^a$ and $\pi_2^a$ corresponding to maximizer.
12:    end if
13: end for
14: Choose $a$ that maximizes $d_a$, and replace $\pi$ with $\pi_1^a$ and $\pi_2^a$ in $\Pi$.
15: Go to step 2.
```

3. SIMULATED EXAMPLE

3.1 Problem description

It is useful to run Grace on a simulated data set, both for illustration purposes and for testing the platform. To this end we designed a *user simulator* that generates simulated user responses to marketing messages in a controlled way. The simulator allows the creation of user cohorts that have various pre-defined properties (such as Age and Gender). Each cohort may have different baseline phone usage characteristics (cohort 1 talks on the phone more than cohort 2, say), as well as different responses to marketing messages (cohort 1 prefers message X over message Y, say). When simulated users receive a message that they like, they use their phone more than their cohort's baseline usage amount. Likewise, when users receive messages they dislike, they use their phone less than the baseline. Using the phone more (less) results in more (less) revenue generated by that user.

The workflow of running the user simulator is as follows:

1. Define messages. In this example, we created four messages: M_1 through M_4
2. Define cohorts. In this example, we simulated 50,000 users, each belonging to one of three cohorts:
   (a) 40% females who like M_1 and dislike M_2.
   (b) 20% females who like M_2 and dislike M_1.
   (c) 40% males who like M_2 and dislike M_1.

   All cohorts are indifferent to M_3 and M_4.

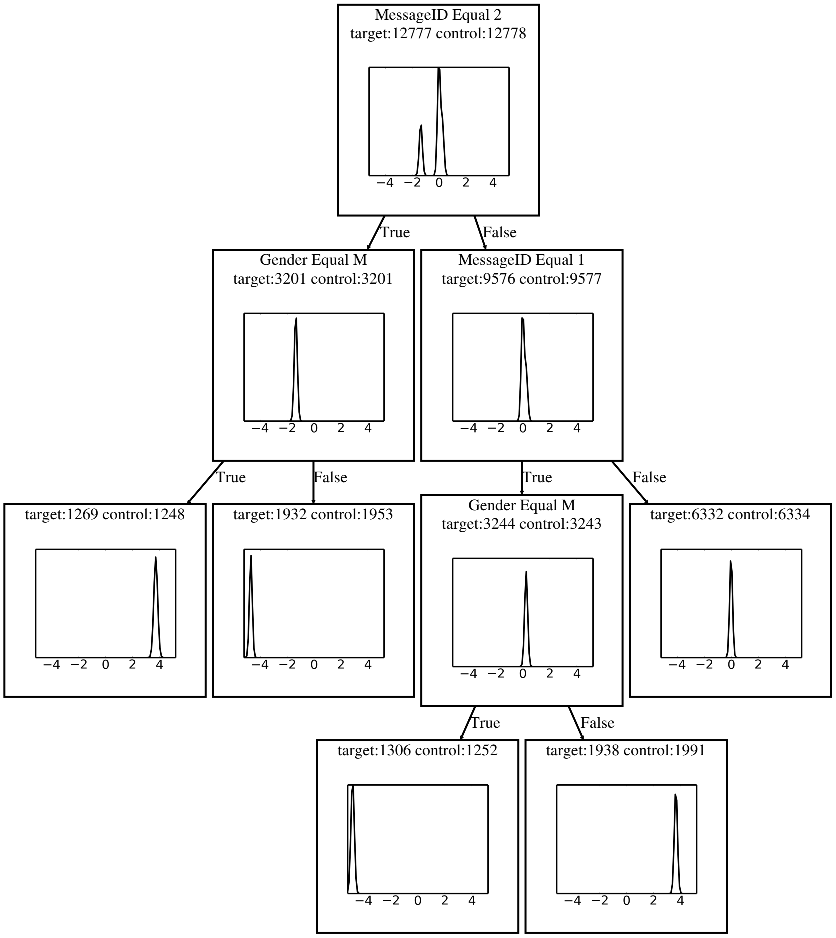

Figure 1: Decision tree trained on simulated user responses

3. Half of the 50k users are assigned to a "business as usual" (BAU) group which are completely off-limits to Grace. Users that are BAU members are not allowed to be targets or controls. The BAU group is used to evaluate global performance of the platform.
4. Choose the KPI to optimize. In this case we used 7 day revenue, which is the revenue generated by a user in the 7 days following message delivery.
5. Configure experimental design. In this case we used a 7 day lockout, which means any target that received a message in the last 7 days is ineligible for messages today. Controls are subject to the same lockout period.
6. Run Grace for (at least) 7 simulated days where messages are picked randomly. This is to create training data for the decision tree. Because of the lockout, messages will be "sent" only on the first of these 7 days. It is necessary to run for at least 7 days so that the 7-day KPI can be computed. In this case the training period lasted 14 days.
7. Train a decision tree on the data generated in step 6, using the algorithm described in Section 2.5.
8. Run Grace for a further 7 simulated days, this time using the decision tree trained in step 7 to pick messages.

3.2 Decision tree training

Figure 1 shows a truncated rendering of the decision tree that was trained in Step 7 above (we truncated the tree for display purposes in this article). For each node of the tree, the following information is shown:

- Split predicate (for non-leaf nodes). For example the split at the root node is on MessageID=2. True branches are on the left, false branches on the right.
- Target and control counts supporting the node.
- A graph of distributions of mean lift across all contexts in the node.

The distribution shown in a node is not quite the distribution of $\bar{r}_\pi$, which is always Gaussian. Instead, it is a Gaussian mixture, with one mixture component per context and targeting group, weighted by the population fraction. This is a more informative way to visualize when significant differences in mean lift are present across contexts or targeting groups.

In this case, the root node is split by MessageID, and the reason is clear from the bimodal shape of the distribution: M_2 performs more poorly on average than other messages. But the rest of the nodes show no bimodal nature, meaning either there is only one context contained in them or there are no obvious differences in mean lift among the multiple contexts. Nevertheless, the tree reveals large differences in mean lift among subpopulations — in this case, gender. Also note that the split on MessageID=1 at the second level revealed a slight (but statistically significant) mean lift difference, but its main effect is to enable the much larger split by gender at the next level of the tree. The tree training has effectively picked out the cohorts that prefer one message over another, allowing the multi-arm bandit to select more effective messages for each user.

3.3 Performance results

Figures 2 and 3 show the action of running Grace on the simulated data. In Figure 2, counts for each message M_1 through M_4 are shown. The dark bars show the message counts *before* training a model when messages are sent to targets at random, showing no preference of one message over another. The light bars show message counts *after* training, showing that the model strongly prefers to send M_1 and M_2 over M_3 and M_4.

Figure 3 shows the percent revenue lift before and after training a model, along with error bars reflecting 95% confidence. Before training a model, M_1 shows a slight positive lift, and M_2 shows a strong negative lift. After training however, both these messages show strong positive lift. The dramatic improvement of M_2 is a demonstration of "getting the right message to the right user": M_2 is in fact an overall loser when sent to the entire population, but is a strong winner when sent to the right sub-population. The ability to send the right message to the right user is one of the key benefits of Grace.

Figure 4 shows normalized revenue for two groups of users, BAU and Grace-addressable. Recall that the BAU group is off-limits to Grace. This means that users in BAU never receive any messages. Their behavior is determined entirely from their cohort's baseline behavior. The x-axis is a time axis, showing the 25 days of the simulation. The y-axis is revenue for a given day, normalized by the revenue produced on day zero. For the first 14 days of the simulation, messages were sent by Grace randomly for the purpose of gathering data. Then a decision tree was trained and first used on day 15, producing a pronounced spike in the Grace curve as users respond to the optimized messages by recharging. A second wave of revenue follows as users recharge after their balance is used up.

4. PRACTICAL APPLICATION

4.1 Application description

In October, 2014, Grace was deployed by a mobile telecom in EMEA, which we will call *Acme* to protect identity. Acme has a prepaid subscriber base of 2.2 million from a variety of international backgrounds. Grace was configured to send 450 different messages, at first just educational messages about existing products, then later with special offers. The messages had several properties that could be optimized by Grace, including reward/punishment phrasing, language, product format, and time of day and day of week for sending the message.

User attributes available for classifying users include:

- recharge amounts and timing;
- usage statistics for voice and SMS (separated by inbound vs. outbound, in-network vs. out-of-network,

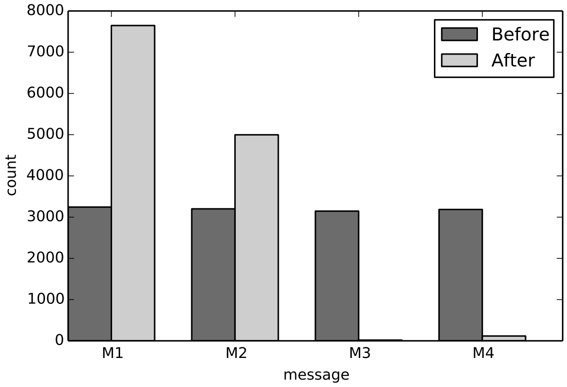

Figure 2: Message counts before and after model training. Before training, all four messages were sent with equal frequency. Afterward, M_1 and M_2 were strongly favored.

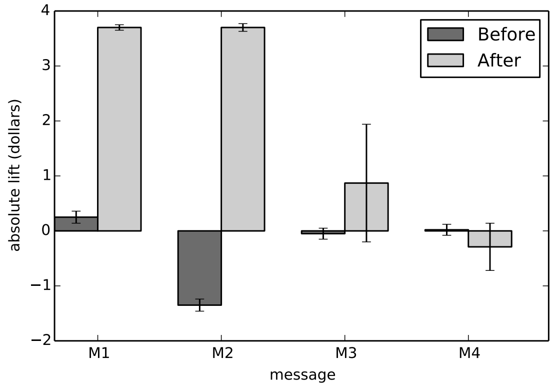

Figure 3: Revenue lift per target, before and after model training. The error bars for M_3 and M_4 are large due to the suppressed counts for these messages.

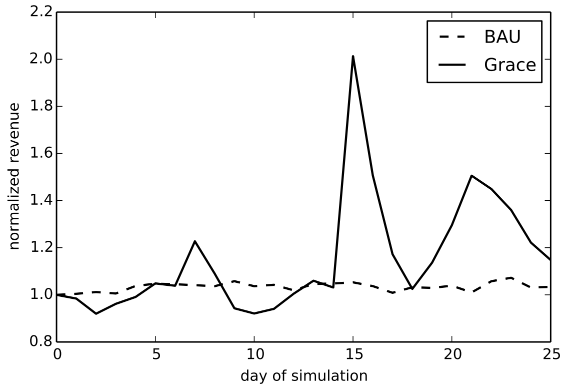

Figure 4: Simulated daily recharge revenue produced by Grace vs BAU. Model was enabled on day 15. Additional revenue is realized when users recharge on day 15 and in second wave of recharges around day 21.

local vs. international) and data;
- nationality and language;
- statistics derived from social graph, such as PageRank and revenue from ego network;
- derived properties such as cluster membership for various attributes.

Grace measured and attempted to optimize lift in recharge revenue over the 14 days after sending the message. Once a user is sent a message from Grace, that user is prevented from receiving another message from Grace for 14 days, and controls are locked out for the same period. The 14-day duration is chosen because customers typically recharge multiple times over this period, so we have a chance to determine the impact on revenue beyond the initial purchase.

4.2 Deployment statistics

Because of the large amount of data Grace operates on, it runs on a cluster of servers using Hadoop. In Acme's case, there are 3 Hadoop nodes comprising 24 cores, processing a compressed data stream of 5 GB/day containing 62 million records/day. Marketing decisions are made twice per day using the latest decision tree, with new trees being generated every few days.

4.3 Performance results

After one month of randomized messaging for training, Grace started optimizing marketing in November, 2014, and has been producing positive results for Acme since then. The positive signal is not as strong at Acme as in the simulated results presented in Section 3; however, when you multiply even a small lift by a large subscriber base, the results can be compelling.

In Figure 5, we show message frequency before and after model training, for four messages from the same targeting group, analogous to Figure 2. In Figure 6 we show the lift generated by these four messages before and after model training, analogous to Figure 3. Note that Grace converted the first three of the messages from neutral or negative lift to positive lift by refining the target audience, and increased the relative frequency of these winners within the targeting group. The fourth message, M_{254}, shows statistically insignificant lift and is therefore sent less often. The fact that it is still sent occasionally after training demonstrates Grace's exploration of messages with uncertain performance.

Figure 7 plots the cumulative increase in recharge revenue over time provided by Grace in comparison to a separate BAU group. The revenue is expressed as a percentage of an average monthly revenue baseline from before deployment. The effects of various events that occurred along the timeline from 10/2014 through 2/2015 are visible in the revenue curve. At first, the revenue lift is fairly flat (tracking BAU) while the system is training, until Grace starts optimized marketing on 11/24/2014. Soon afterward the lift rises dramatically, but the rise is interrupted on 12/15/2014 when the data feed is accidentally cut off and Grace is forced to use increasingly stale data in its marketing decisions. Marketing is turned off on 12/22/2014, and the revenue lift continues to drop. Data covering the period of interruption is received a week later, and used to train a new decision tree. When marketing is re-enabled on 12/31/2014, the lift increases again. This unintended sequence of events provides a sort of "natural experiment" that bolsters the claim that

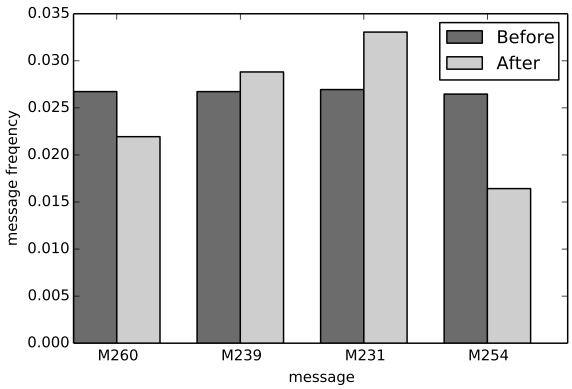

Figure 5: Acme message frequency before and after model training.

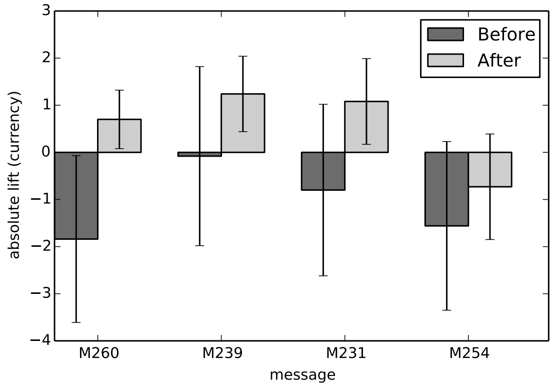

Figure 6: Acme revenue lift per target, before and after model training. The first three messages were turned from losers to winners. The last has statistically insignificant lift, and is "played" less often.

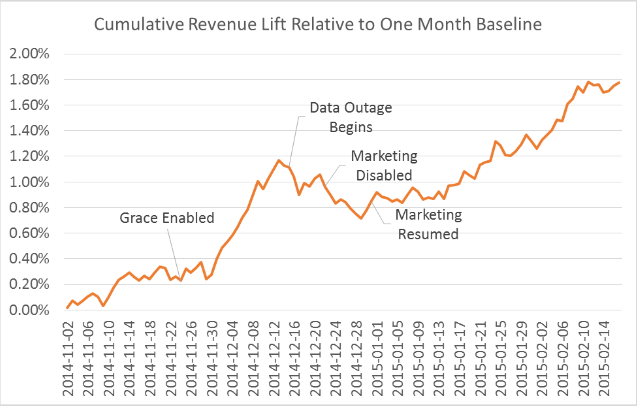

Figure 7: Cumulative recharge revenue lift for Acme

Grace's optimized messaging is the cause of the increased revenue.

5. CONCLUSIONS

In this paper we have described the software platform *Grace* for automatic optimization of marketing campaigns at the scale of hundreds of millions of targets. Grace measures the lift in a designated KPI using carefully controlled experiments, and it optimizes lift using a novel combination of multi-armed bandits and learned decision trees for segmenting the user and message space. It solves the explore/exploit trade-off by employing a Bayesian approach to decision-making, in the form of Thompson sampling.

We presented simulation results on a simple problem to show how Grace produces lift by learning to send the right message to the right user. Finally, we described the deployment of Grace in a prepaid telecom application with millions of users and hundreds of different messages, and we showed the platform creating real value over time.

Future plans for Grace include extensions to optimize multiple KPIs, enhancement of the architecture to accommodate real-time interactive decisioning (e.g., in customer call centers), and application to other domains including financial services, online gaming, and e-commerce.

6. ADDITIONAL AUTHORS

Phil Barker, Globys, Inc., pbarker@globys.com
Matt Danielson, Globys, Inc., mdanielson@globys.com
Julie Penzotti, Globys, Inc., jpenzotti@globys.com
Richard Sharp, Globys, Inc., rsharp@globys.com
Garrett Tenold, Globys, Inc., gtenold@globys.com
Artem Yankov, Univ. of Michigan, yankovai@umich.edu

7. REFERENCES

[1] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-arm bandit problem. In *Proc. 25th Ann. Conf. Learning Theory (COLT 2012)*, volume 23 of *JMLR Workshop and Conf. Proc.*, pages 39.1–39.26, 2012.

[2] S. Agrawal and N. Goyal. Further optimal regret bounds for Thompson sampling. In *Proc. 16th Int'l. Conf. Artificial Intelligence and Statistics (AISTATS)*, volume 31 of *JMLR Workshop and Conf. Proc.*, pages 99–107, 2013.

[3] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proc. 30th Int'l. Conf. Machine Learning*, volume 28 of *JMLR Workshop and Conf. Proc.*, pages 127–135, 2013.

[4] M. Brezzi and T. L. Lai. Incomplete learning from endogenous data in dynamic allocation. *Econometrica*, 68(6):1511–1516, Nov. 2000.

[5] S. Bubeck and C.-Y. Liu. Prior-free and prior-dependent regret bounds for Thompson sampling. In *Advances in Neural Info. Proc. Sys. 26 (NIPS 2013)*, 2013.

[6] O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Info. Proc. Sys. 24 (NIPS 2011)*, Granada, Spain, Dec. 2011.

[7] D. M. Chickering, C. Meek, and R. Rounthwaite. Efficient determination of dynamic split points in a decision tree. In *Proc. 2001 IEEE Intl. Conf. on Data Mining*, San Jose, CA, USA, 2001.

[8] J. C. Gittins. Bandit processes and dynamic allocation indices. *J. Royal Stat. Society, Ser. B: Methodological*, 41:148–177, 1979.

[9] C.-C. Hsieh, J. Neufeld, T. King, and J. Cho. Efficient approximate Thompson sampling for search query recommendation. In *Proc. 30th ACM/SIGAPP Symposium On Applied Computing (SAC'15)*, 2015. Available at http://oak.cs.ucla.edu/~chucheng/publication/sac15.pdf.

[10] A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Info. Proc. Sys. 24 (NIPS 2011)*, pages 2447–2455, 2011.

[11] L. Li, W. Chu, J. Langford, and R. E. Shapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. 19th Int'l Conf. World Wide Web (WWW2010)*, pages 661–670, Raleigh, NC, Apr. 26–30, 2010.

[12] P. A. Ortega and D. A. Braun. Generalized Thompson sampling for sequential decision-making and causal inference. *Complex Adaptive Systems Modeling*, 2(2), 2014. Available at http://www.casmodeling.com/content/2/1/2.

[13] N. J. Radcliffe and P. D. Surry. Real-world uplift modelling with significance-based uplift trees. Stochastic Solutions white paper, available at http://stochasticsolutions.com/pdf/sig-based-up-trees.pdf, 2011.

[14] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

[15] E. M. Schwartz, E. Bradlow, and P. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. Univ. of Mich. Ross School of Business, Working Paper No. 1217, Dec. 2013.

[16] S. L. Scott. A modern Bayesian look at the multi-armed bandit. *Appl. Stoch. Model. Bus. Ind.*, 26(6):639–658, Nov. 2010.

[17] S. L. Scott. Multi-armed bandit experiments. Google Analytics Blog, http://analytics.blogspot.com/2013/01/multi-armed-bandit-experiments.html, Jan. 23, 2013.

[18] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933.

[19] B. E. T. H. Twala, M. C. Jones, and D. J. Hand. Good methods for coping with missing data in decision trees. *Pattern Recogn. Lett.*, 29(7):950–956, May 2008.

[20] M. Valko, N. Korda, R. Munos, I. N. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proc. 29th Conf. Uncertainty in Artif. Intel. (UAI)*, pages 654–663, July 2013.

[21] R. Weber. On the Gittins index for multiarmed bandits. *Ann. Applied Prob.*, 2(4):1024–1033, Nov. 1992.

PATENTS PENDING

This work is the subject of multiple US and International patents pending.