UNSCRAMBL

Unscrambl Tech Report

ALGORITHMIC MARKETING USING qbo decisions™

Automated iterative mass experimentation & campaign optimization

"The true method of knowledge is experiment"
--- William Blake ---

Introduction

The key goal of any personalized marketing campaign is to target the right customer with the right offer using the right message on the right channel. However, figuring out this right combination of customer (or customer segment), offer, message and channel is not easy. While historical campaign responses and customer behavioral profiles can help us in seeing what worked and did not work in the past, they offer very limited insights to what might work today or in the near future. Customer preferences, competition movements, product specifications and offer types all keep changing rapidly, making it very difficult to extrapolate the future from the past.

The only way forward is gain knowledge through experimentation. We must be able to rapidly try out different combinations of segment definitions, offer conditions, messages and channels in order to figure out what works and what doesn't. Marketers often employ techniques like A-B testing to evaluate the performance of different offers.

However, current tools and methods for A-B testing are very manual and labor intensive. They rely heavily on the expertise and intuition of campaign managers. These campaign managers often select a target population for a campaign based on high-level business directives, and then define segments and offers in a rather static manner. Every segment is then sent an offer decided by these experts. If the campaign managers choose to do A-B testing to evaluate different offers for different segments, they are often forced to manually construct the experiments, monitor the runs of the experiments and then go back to do the analysis on the campaign responses. This whole process can take a while and requires several experts such as campaign managers, data scientists and IT teams that execute these campaigns. This process also doesn't scale as the number of segments, offers and channels increases.

In several enterprises, we have seen manual campaign configuration and experimentation approaches not scale. These approaches are all time consuming and have a lot of bias depending on the judgement of the decision maker. The exploration of the audience is limited to human manageable segments and often require a combination of experts.

At Unscrambl, we have built an automated marketing optimization technology that uses rapid, iterative, mass-scale experiments to determine the best combination of segments, offers, messages and channels. This is part of the Unscrambl qbo decisions product which supports real-time contextual AI-driven marketing. It can consider thousands of possible combinations simultaneously, which is not possible in a manual operation. It can be guided to optimize different metrics such as take-up rate, revenue and profit.

In this report, we shall describe some of the key components of the technology, the algorithms used, integration needs, operationalization steps and best practices.

Traditional Marketing

> *"Current tools and methods for A-B testing are very manual and labor intensive. They rely heavily on the expertise and intuition of campaign managers."*

Optimization & its Challenges

The key concept in marketing optimization is A/B testing. A/B testing is a randomized experiment with two variants, A and B, and it allows us to compare a subject's or a population's response to the two variants. It allows determining which of the two variants is more effective using statistical hypothesis testing or "two-sample hypothesis testing".

There are a number of challenges in practical usage of A-B testing.

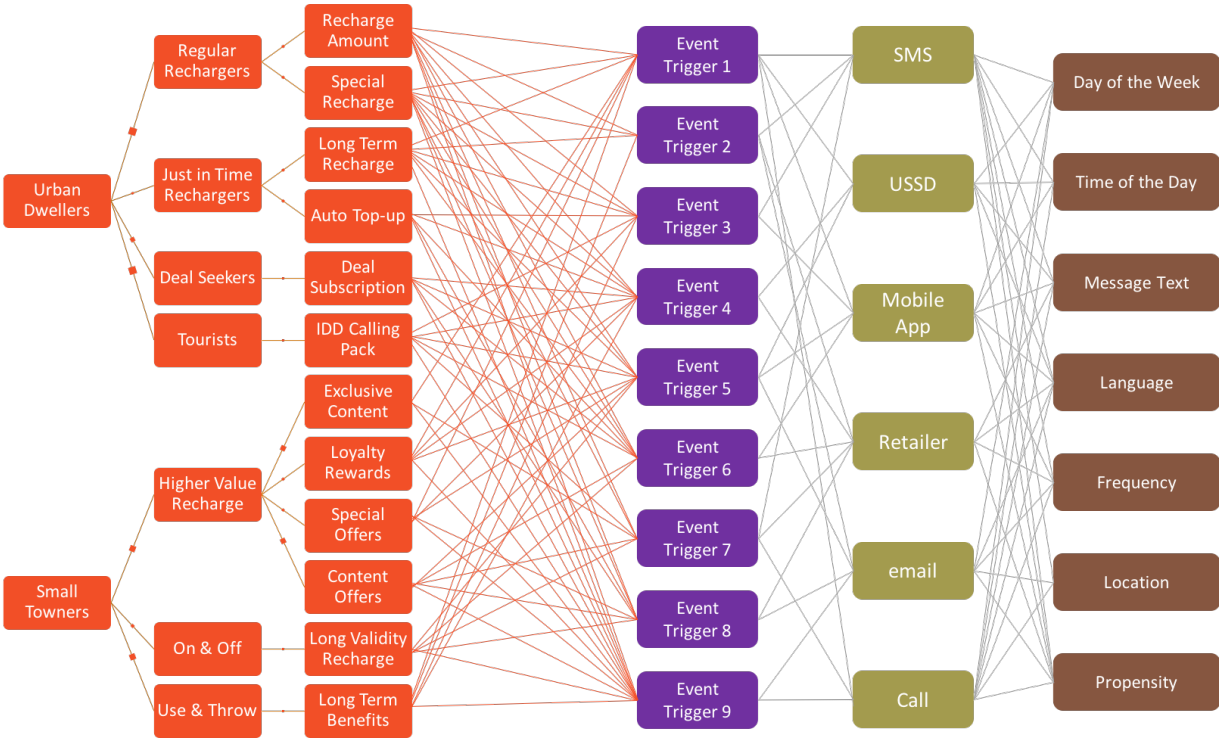
Doesn't scale to large number of variant combinations

A-B testing forces campaign managers to be very meticulous in designing the experiments and rely on their intuition in deciding what to test. This means that A-B testing is usually limited to testing a small set of segment-offer-channel combinations that a human can keep track of and analyze later.

Difficult to find optimal segmentation on a large number of variables

A-B testing techniques also don't do well in discovering the best segmentation among possibly hundreds of customer profile attributes. Most A-B testing analysis techniques rely on statistical segmentation that was constructed beforehand, and that may not be a good fit to explain offer take-up.

Fig.1: Evaluating millions of combinations to identify the best outcomes



Experiment construction and runs are time consuming

Constructing and running experiments often require data and integrations with a number of disparate enterprise systems. Customer profiles and historic campaign responses might be stored in data warehouses or data lakes. This data may need to be imported into data mining and analytics tools. A campaign manager and data science team would then need to get insights from this data to construct experiments. They may need to configure these campaigns manually in one or more marketing automation tools. In some cases, other tools might be needed to cover all the channels under consideration. Responses to these campaigns then need to be collected, stored back in the data warehouse or data lake, and then analyzed again. This whole process can be quite time consuming because of the number and variety of systems involved and skills needed.

Labor intensive and hence difficult to run on a continuous basis

Since experiment construction tends to be difficult, most enterprises do not have the wherewithal to continuously test their campaign definitions in a rapidly changing

world. This means that it is difficult to adapt to changing market conditions, customer behaviors and product offerings.

Automated Iterative Mass Experimentation

Unscrambl has pioneered the concept of Automated Iterative Mass Experimentation to overcome the challenges associated with traditional A-B testing. This AI-driven methodology aims to support the campaign manager in campaign optimization.

The channel is the delivery mechanism, e.g. SMS, in-app notification, chatbot, website, phones, etc. Again, we assume that the set of channels is fixed over the course of the experiment.

In our mass experimentation process (Figure 2), the system automates the creation of

First some definitions. We define a campaign as having 3 key elements: segment, offer and channel.

A segment can be defined either as an explicit list of customers, or a set of conditions on customer attributes. For example, a segment can be a list like [John, Alice, Mary] or a set of conditions like [age > 30 and number of transactions in last month < 5]. We assume that there is a fixed set of customer attributes available for us to construct segments.

An offer is a specific message that is sent or presented to a customer. Offers can just be a simple notification or can be

individual candidate campaigns. It can create a campaign by first defining a segment from an arbitrary set of customer attribute conditions and then associating the campaign with an offer and channel from the set of available offers and channels. The system first analyzes the customer profile and historical campaign performance

Fig.2: Key Steps in Marketing Experimentation

data to form some initial hypotheses. It then constructs experiments to help decide which campaigns work better. It executes these campaigns with optional integration to a campaign management system. It collects the results and analyzes the performance of these different campaigns. Based on this analysis, it decides on the next set of experiments, and keeps going until it can no longer significantly improve on the campaign performance. During this process, the system can explore a potentially large space of customer profile variables to come up with a unique hierarchical segmentation tailored to this campaign.

The AI uses a combination of Bayesian and traditional statistical techniques to predict the performance of the winning designs. It utilizes "weak" signals (compared to A/B testing's 95% or 99% confidence signals) and refines them across multiple iterations to produce a strong signal by the end of the experiment.

Typically, at the end of six to eight iterations, the system converges to a set of campaigns that maximizes a certain metric. The metric used can be offer acceptance rate, revenue or profit. Note that the system can also recommend not to send an offer to a certain segment.

One possible sequence of campaign explorations, starting from 3 offers and 3 segments (or 9 campaigns) to 14 offers and 29 segments (or 406 campaigns) in 3 iterations is shown above in Figure 3.

Fig.3: Experimentation Combinations

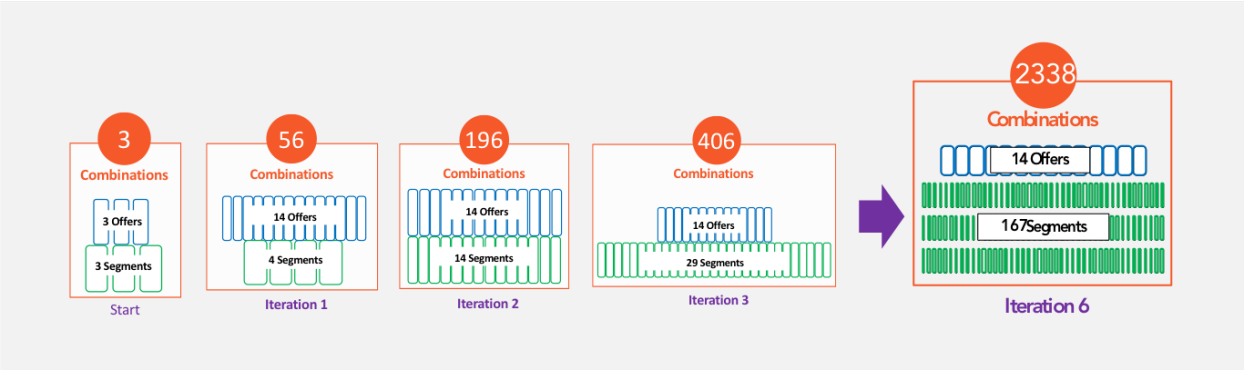
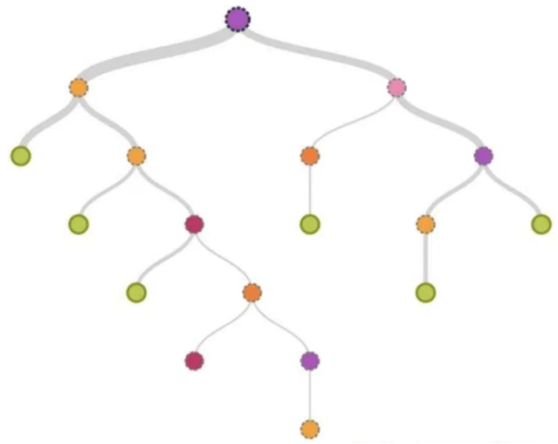


Fig 4. Resulting audience hierarchical segmentation, where different segments in the hierarchy are associated with different offers and channels



Key Steps in Automated Iterative Mass Experimentation

There are several key steps involved in running the automated experiments:

Experiment Definition

The experiment definition starts with the selection of the base audience that we want to run the campaigns on. In practice, the size of this base should be at least 50,000 so as to have statistically significant results. Also, we need to define the set of offers and channels. Usually, the number of offers varies from 2 - 20, and the number of channels is 1 - 5. We also define the space of customer profile attributes we would like to explore to create the segments. The number of attributes can be 1 - 500.

Optimization Metrics

One can define various objectives for each campaign, namely Revenue, Profit or Conversion optimization.

Revenue Optimization

Revenue is defined in terms of the incremental monetary value that the enterprise gets if a customer accepts an offer. Choosing revenue as the optimizing criteria focuses on recommending offers which can maximize the revenue per user. A positive revenue per user value for any offer qualifies it for recommendation, while a negative revenue per user value forbids it from being considered again.

Profit Optimization

Profit is defined in terms of the incremental monetary value that the enterprise gets if a customer accepts an offer after subtracting the cost of the offer. Profit optimization results in recommending offers which can maximize the profit per user. Again, a positive profit per user value for any offer qualifies it for recommendation, while a negative profit per user value forbids it from being considered again.

Conversion Optimization

Choosing conversion ratio as optimizing criteria focuses on recommending offers which have the highest acceptance rate for the audience. We define a base conversion rate (minimum required conversion rate for any offer to be viable) and Offers showing statistically significant increment from the base conversion rate are qualified for recommendation. Offers which have conversion rate significantly lower than the base conversion rate are eliminated and not considered in further iterations.

Additional Constraints

There are several additional constraints that we can specify to guide the experiment execution. This includes:

- Time triggers when the offers should be sent to the customers. E.g. all offers are to be sent at 9 AM, or when a customer does a certain activity.
- Preferences of which offers should go to which customers
- Hard rules on which offers can go to which customers

Experiment Construction & Iteration

At the beginning of the process, let's assume we start with a base of 100,000 customers that we want to target with offers of a certain kind. For example, all these customers are in danger of churning and we want to know which churn-prevention offer to send to which customer. Let's assume there are 10 offers. For ease of explanation, let's assume there's only 1 channel, but the process can be extended to experiment with multiple channels.

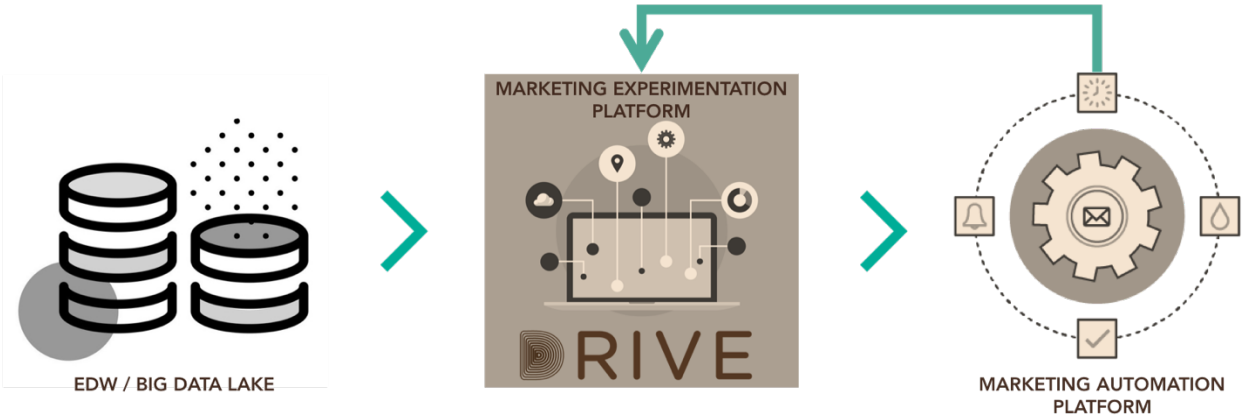
In the first iteration, we perform A/B tests for these 10 offers on 10 small samples of the base of customers. Another sample is assigned as the control group. The system automatically constructs the corresponding campaigns, sends the offers and analyzes the responses. It uses feature extraction methods to estimate the best splitting feature using the responses from these tests and on the basis of this feature, splits the base into 2 or more segments using a hierarchical clustering approach. We now construct and run additional campaigns on these smaller segments to test how the different offers perform on these segments.

This process is repeated recursively (one cycle every few days), thus creating new branches and expanding the tree. For each node in the tree, statistical significance tests are performed to declare (Tag) the best offer for a particular node, or potentially even tag the node as no-offer; i.e. none of the offers work well for this node. We use a variety of statistical significance tests such as Bayesian hypothesis testing and frequentist testing for this purpose.

The process is terminated when either one offer is tagged per node or one reaches the pre-defined maximum depth of the tree or the number of customers under a certain node is below a pre-defined threshold. At the end of this process, we end up with a tree where each node in the tree corresponds to a segment that is tagged with exactly the best offer for that segment.

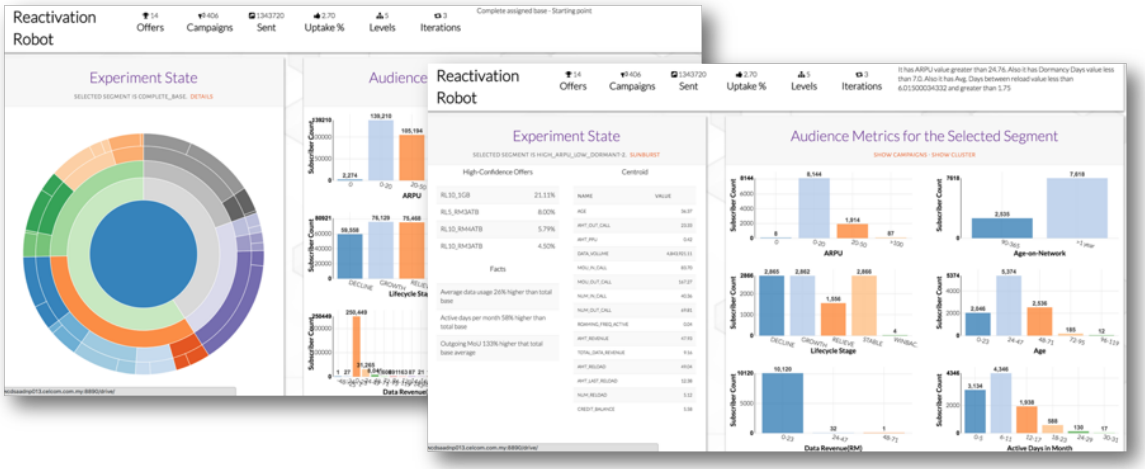
Operationalization

Fig.5: Key integrations with Other Enterprise Systems



There are two key integrations involved in the operationalization of Unscrambl qbo decisions Marketing Experimentation platform. The first is with a store containing recent customer profiles. Typically, this is stored in a data warehouse or big data lake. qbo decisions can query these data sources to fetch recent data periodically, or this data can also be pushed to qbo decisions in the form of flat files.

The second key integration is with one or more marketing automation platforms. The marketing automation platforms will actually execute the campaigns that are constructed by Unscrambl qbo decisions and can also collect and report on the responses, and send the responses back to qbo decisions. Depending on the APIs exposed by the marketing automation platform, the campaign definitions can be fed in an automated or a semi-automated fashion.



Screenshots (below) of Unscrambl qbo decisions platform for monitoring the progress of the experiments.

About Unscrambl

Founded in Atlanta, and now Headquartered in Singapore.

Unscrambl is one of the fastest growing enterprise software companies at the forefront of developing Decisioning AI & Conversational AI products.

We believe in the potential of **Human Intelligence** + **Machine Intelligence.**

COGNITIVE ASSISTANTS.
COGNITIVE ADVISORS.
COGNITIVE EXPERTS

Unscrambl has created AI-powered cognitive agents for the enterprise, disrupting traditional business processes – from how insights are accessed, to how decisions are made, to how actions are taken.

These cognitive agents are powered by our proprietary AI Technology platform Unscrambl BRAIN™

Unscrambl has been recognized by Gartner as a trailblazer in the fields of Conversational Analytics, Continuous Intelligence, and Contextual Insights for Hype Cycle 2018. Unscrambl is a preferred Data & AI partner for Microsoft.