LORE: A Large-Scale Offer Recommendation Engine through the lens of an Online Subscription Service

Rahul Makhijani[1], Shreya Chakrabarti[2], Dale Struble[2] and Yi Liu[2]
[1]Stanford University
[2]Amazon
rahulmj@stanford.edu

ABSTRACT

The majority of online market platforms and streaming platforms such as Amazon, Netflix, Spotify, etc. offer subscription based membership plans to access some/all of their products. In order to appeal to diverse customer groups, these services typically offer more than one type of plan. In this paper, we propose solutions to optimally recommend subscription plans to maximize user acquisition constrained by user eligibility and plan capacity (limited headcount per plan) simultaneously. We achieve this through a plan recommendation model based on Min-Cost Flow network optimization, which enables us to satisfy the constraints within the optimization itself and solve it in polynomial time. We present three approaches that can be used in various settings: a single period solution, sequential time period offering, and a clustering for large scale setting. We evaluate these approaches using offline policy evaluation methods and demonstrate their value. We also discuss some practical issues in the implementation and online performance.

1 INTRODUCTION

In the past few decades, there has been a rise in online market platforms for various services, such as selling goods (Amazon, Craigslist), dating (Tinder, OkCupid, etc.), ride-sharing (Uber, Lyft, etc.), and music (Spotify, Pandora) services. Many earn a significant portion of their revenue through subscription fees (Amazon Prime, Spotify, Youtube Red, Tinder, etc.). These revenues are either directly accretive or offset the cost of service and net positive downstream impacts. Marketing teams have various tools at their disposal to incentivize membership, however their use must be optimized relative to their potential gains.

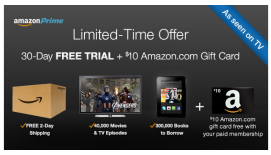
Service providers typically have multiple plans to offer a customer, however, they often present at most one (illustrated in figure **??**). Therefore, plan recommendation becomes an optimization problem to maximize user acquisition. Within this context there are typically user eligibility constraints and plan capacity constraints that must be incorporated. Eligibility constraints occur when there are users who are ineligible for some plans. For example, a user may not be eligible for a discounted student plan if not in school. Plan level capacity constraints arise for a multitude of reasons, including preservation of brand value, discouraging gaming behavior (e.g. limiting frequent free trials), contractual/legal constraints, and budget constraints (e.g. limited allocation of discounted plans). Combined, this leads to a constrained optimization problem where we determine when to present which plan to whom, and for how long with the goal being to maximize membership subscription rate for a given number of users with diverse eligibilities and plans with differing capacities.

Currently, there are a variety of approaches to optimization of recommendation with constraints [1, 6, 20]. However, existing approaches typically rank items ignoring capacity constraints during optimization but subsequently address them. In other words, they enforce strict plan capacity constraints downstream (post-optimization). For example propensity-based greedy approaches rank plans based on propensity scores and then allocate them according to available capacity downstream [22, 26]. This can lead to suboptimal solutions (illustrated in Sec. (4.1)). Directly addressing constraints within the optimization itself is usually avoided because constrained optimization with cost-based constraints is usually NP-Hard integer programs. The use of a constrained based optimization for targeting or recommending items has also been explored in Agarwal et al. [2] as a multiobjective optimization problem to balance promotions and engagements. However such a framework is not applicable to our domain since the targeting cannot be solved/done in a probabilistic sense. In order to avoid whiplash effects, it is necessary to show the same offer/plan to a given user atleast for a few times. In fact conversions usually happen after a few visits to the platform. Agarwal et al. [2] address this problem by building a cool-off system where the decision history is maintained but it defeats the purpose of having budget constraints in the first place since the constraint violations can be substantial and can be costlier in the 'offer' space as compared to the 'click' space.

In this paper we model all of the aforementioned constraints within the optimization problem by converting it to a Min-Cost Flow problem, which is commonly utilized in the operations research field to send flows/goods across a network at a minimum possible cost. We first deliver a static model to maximize plan subscription rate with all of the constraints modeled for a single point in time. We then expand it to address other needs usually raised in an online recommendation environment. Specifically, we build one model for optimal resource allocation across multiple time periods upfront and an idea to scale the optimization with clustering. We benchmark our approaches and compare their performance to existing approaches with offline policy evaluation. We also address shortcomings where the i.i.d assumption in offline policy estimators does not hold.

In the following sections we use the general terms: 'Item', 'Plan', and 'Offer' interchangeably to denote what the system recommends to users. It can be a membership subscription plan, offer, promotion, song, book, etc. While our solutions in section 4 accommodates general item recommendation (in which a user can be allocated more than one item), the analysis and discussion focus on recommendations similar to membership subscription offers where a single user can take at most one item.

(a) Amazon Prime subscription Offer 1



(b) Amazon Prime subscription Offer 2

This paper is structured as follows. Section 2 presents relevant literature, Section 3.2 discusses the offline policy evaluation methods for benchmarking our approaches and Section 4 formulates and solves the optimization problems. Section 5 shares all the experimental results, including the performance of our approaches and how they perform as opposed to competing approaches. Finally, Section 7 concludes the paper.

2 LITERATURE REVIEW

Offer recommendation approaches can be broadly classified into collaborative filtering (CF) and content-based filtering (CB) [9]. CF approaches started with user-based and then attention was paid to item-based to address the sparsity and scalability challenges in user-based CF [6]. CF techniques fall into three categories: memory-based, model-based and hybrid [36]. The most widely used evaluation metrics for prediction performance of CF are root-mean square error and mean absolute error. Parameswaran et. al. wrote the first paper (to our best knowledge) to solve for CF recommendation system with complex constraints (such as prerequisites) [1]. Christakopoulou et. al. solved for matrix factorization problem under capacity constraints[20]. They propose two solutions for satisfying capacity constraints. In the first solution, known as post-processing, they assign items according to learned ratings until capacity constraints starts to bind, leading to a sub-optimal solution. In the second, they solve the matrix factorization with a penalty for overuse of expected item capacity. Here, item capacity constraints are respected but reduce the ratings quality based on pair-wise ranking loss and square loss per their experiments.

CB systems make recommendations by analyzing the items and user profiles [34]. These systems use a rich set of techniques including supervised machine learning methods such as logistic regression and deep neural networks [33] and reinforcement learning methods such as Multi-armed Bandit (MAB) algorithms [22]. A typical MAB algorithm does not have budget or capacity constraints. Badanidiyuru et. al. was the first paper to model constraints in an MAB problem [5]. The bandits with Knapsack model as described in their paper, solves for reward maximization under stochastic integer constraints. The process stops the first time the total consumption of some resource exceeds its budget. Agarwal et. al. solved for the general case of this problem by suggesting fast stochastic convex algorithms [3]. However neither approach incorporate delayed rewards. In fact online learning frameworks have a large regret due to delay [19]. Two other problems that plague these frameworks and deteriorate performance are high correlation in the reward between arms (offers) and the warm - start problem (initial decisions have a higher weight).

In terms of objective function used in CB recommendation systems, early efforts were paid to optimize the target metrics as they are. Radcliffe et. al. and Lo et. al. [27, 31] then introduced the idea of modeling incremental lift in the targeting metric. Lo et. al. also introduced an integer optimization approach to the problem.[26]. The approaches do not tackle item budget constraints or effectiveness of propensity estimation with respect to counterfactuals.

While the research problem in our paper is a recommendation problem, it overlaps with online resource allocation problems given the capacity constraints. In the allocation problem, the goal is to maximize revenue where users arrive in an online manner and the decisions are irrevocable and instantaneous [4, 11]. Exploration is done to continuously learn the prices of items and if the users' bid is higher than the prices, the demands are met. Agarwal et. al. proposes a model of online linear program where columns are revealed one by one and although the problem is near optimal for the random permutation model, it requires large budgets to achieve near optimal competitive ratios and capacity constraints are met only in expectation [4]. Golrezaei et. al. also solve for the constrained assortments in real time with the use of inventory budget balanced algorithms and dual based algorithms which have a good competitive ratio under the adversarial case and stochastic i.i.d. case respectively[12]. However, LP-based and greedy algorithm tend to beat theses approaches in experimental settings as demonstrated in their paper as most settings in practice do not arise from the adversarial or stochastic iid settings. Cohen et. al. also use the min-cost flow to solve some relaxed versions of their integer programs arising in supermarket promotion problems. Their paper is focused on revenue maximization post demand estimation which isn't within in our scope (as we not focused on demand estimation) [35].

3 PRELIMINARIES

3.1 Min-Cost Flow Network

A traditional linear integer program (IP) in matrix form is formulated as

$$\begin{aligned} \max_{x} \quad & c^T x \\ s.t. \quad & Ax \leq b \\ & x \geq 0, \quad x \in Z_+^d. \end{aligned} \tag{IP}$$

This can be relaxed to a linear program by dropping the integral constraints (setting $x \in R_+^d$). The 'integrality gap' of an integer program is defined as the difference between the optimal values of the integer program in (IP) and its relaxed linear program. When the vector b is integral and the matrix A is unimodular (all entries are 1, 0, or -1 and every sub-minor has determinant of +1 or -1) then the 'integrality gap' is zero and the solution of the relaxed linear program is integer valued [7]. Hence, we can solve IP by instead solving the relaxed linear program [7].

A prominent example of a integer program with a unimodular constraint matrix is the min-cost flow network problem. [7]. This is the problem of minimizing the cost of sending flow through a network graph $G = (V, E)$ where V and E denote the set of nodes and edges respectively

$$\begin{aligned} \min_{(f_{i,j})} \quad & \sum_{\{i,j\}\in E} c_{i,j} f_{i,j} \\ s.t. \quad & \sum_j f_{i,j} - \sum_i f_{i,j} = s_i \quad \forall i \in N \\ & 0 \le f_{i,j} \le b_{i,j} \quad \forall (i,j) \in E. \end{aligned} \tag{1}$$

Above, $c_{i,j}$ and $f_{i,j}$ denote the unit shipping cost and the flow across edge $(i,j) \in E$ respectively and the constraints in (1) refer to supply-demand balance. A computational advantage of min-cost flow problems over standard linear programming problems is the use of network simplex algorithms which run an order of magnitude faster than linear programming solvers [7].

3.2 Offline Policy Evaluation

Usually, new policies are tested by performing an A/B test. However such an evaluation might require deploying new systems which can be quite costly. An alternative to A/B testing is offline policy evaluation that can serve as a proof of concept test before deployment [18]. In general, offline policy evaluation can be summarized in three steps:

(1) Training data collection - For each sample i having context x_i in the exploration phase; an action a_i is taken based on a logging policy $\mu(x_i)$ and reward $r_i(x_i, a_i)$ is logged.
(2) New policy definition - A policy $\nu(x_i)$ which defines action $\tilde{a}_i$ for sample i.
(3) Offline policy evaluation - Estimating the value V of policy ν if it was deployed in the place of policy μ[23].

A survey of offline policy evaluation methods can be found in [21]. One of the challenges in evaluating policies offline is how to estimate the counterfactuals [18]. The two most well known methods in literature are:-

(1) **Direct Method (DM) Estimator** - Here, we estimate the reward $\tilde{r}(x_i, \tilde{a}_i)$ for context x_i and action $\tilde{a}_i$; giving the offline policy estimate as

$$V_{DM}(\nu) = \frac{1}{n} \sum_i \sum_{\tilde{a}_i} \nu(x_i, \tilde{a}_i) \tilde{r}(x_i, \tilde{a}_i).$$

(2) **Inverse propensity Score (IPS) Estimator** - Instead of estimating the reward, the IPS estimator corrects for shift in policy action proportion whenever the new and old policy actions match akin to importance sampling.

$$V_{IPS}(\nu) = \frac{1}{n} \sum_i \frac{\nu(x_i, a_i)}{\mu(x_i, a_i)} r(x_i, a_i).$$

Here $\nu(x_i, a_i)$ indicates the probability of choosing action a_i for sample i.

The DM estimator tends to have a high bias. The IPS estimator tends to be an unbiased estimator of $V(\nu)$. However it suffers from the problem of high variance specifically when the logging policy probability is close to 0 [32]. There are several thresholding schemes to correct for the high variance of the IPS estimator [16]. Swaminathan et.al. propose the Self Normalizing IPS (SNIPS) estimator for controlling the variance of the IPS by use of control variates[14, 37]. Yet another estimator that balances bias and variance is the doubly robust estimator (DR)[10, 25]. It uses the DM method as baseline and applies an IPS based correction whenever the reward data is available (when new and old policy actions match.)

4 MODEL

In this section, we state the problem along with the modeling assumptions and some solutions to address it. The problem of maximizing customer acquisition can be formally defined as

Given a finite set $S = \{S_1, \ldots, S_f\}$, with each element being a subset from the set of all potential offers $\{O_1, \ldots, O_m\}$, where each offer O_j has a corresponding budget b_j; what is the best way to allocate the subset of offers among S to n customers to maximize the expected settled/conversion yield s.t. customer i receives at most c_i offers ?

The budget b_j for offer O_j indicates that it can be shown to a maximum of b_j users. Such budgets constraints are made to preserve the brand value, avoiding gamification and to ensure that the discounts do not make it unfair to previous paying members. Another advantage of pre-splitting the offer budgets has modeling advantages where min-cost flow technique can be used that helps with scalability. This rules out schemes such as showing offers to everyone or a first come first serve process. The assumption of a finite sized subset of offers is necessary to avoid an exponential number of variables. It is also reasonable from a practical point of view as rarely do we have more than 5 different kind of offers and a bundle of more than 3 offers. Finally, We look at the expected conversion yield which incorporates the fact that the customer is a paying member after a certain time period has elapsed after the offer period (i.e. there is always a drop off after the offer period/trial period is over and after the first payment period). The exact time period can determined from customer survival curves.

We present three approaches: single period solution, sequential time period offering and clustering for large scale setting to solve the problem.

4.1 Single Period Solution

We start with the case where each set S_s has cardinality 1 and each customer can be allocated atmost one offer. Hence at a single point in time, we optimize the allocation of offers from a set of options $\{O_1, \ldots, O_m\}$ where each offer has a finite quantity b_j. This is mathematically formulated as

$$\begin{aligned} \max_{(X_{i,j})} \quad & \sum_{i\in N} \sum_{j\in J} P(Y_i|F_i, O_j) X_{i,j} \\ \text{s.t.} \quad & \sum_{j\in J} X_{i,j} = 1 \ \forall i \\ & \sum_{i\in N} X_{i,j} \le b_j \ \forall j \\ & X_{i,j} \in \{0,1\} \ \forall i,\ j \end{aligned} \tag{2}$$

In the above, $P(Y_i|F_i, O_j)$ denotes the propensity for user i to convert if given offer j and historical features F_i. The variable $X_{i,j}$ denotes the fact that user i has been allocated offer j. The user level constraint adds to 1 since we can also include no-offer as a kind of offer. As a byproduct, ineligibility of any user i for offer j can be modeled by adding a constraint $X_{i,j} = 0$. The next proposition

Segment	Segment Size	$Pr_N(Y)$	$Pr_A(Y)$	Greedy policy	Expected Conversions	Optimization policy	Expected Conversions
1	100	25%	50%	Offer N	25	Offer A	50
2	100	60%	70%	Offer A	70	Offer N	60
Total	200				95		110

Table 1: Comparison of the performances of constrained optimization vs Greedy policy on a contrived example.

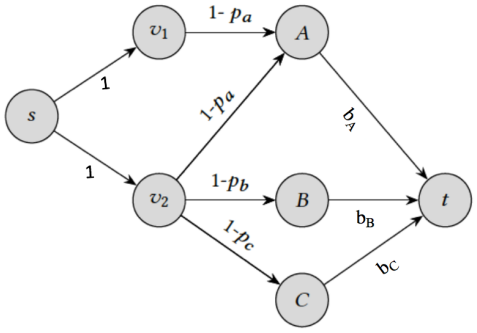


Figure 2: Reduction of Promotional Optimization Program to Min-Cost Network Flow Problem for two users and three offer types. v_i is the node for user i. User 1 is only eligible for offer A. The capacity/cost of transporting a unit flow through a pipe from source node s to user node is $1/0$ resp. The capacity/cost of transporting a unit flow through a pipe from a user node to an offer node O is $1/(1-p_O)$ resp. The capacity/cost of transporting a unit flow from an offer node O to terminal node t is $c_O/0$ resp. We send a flow of 2 through source node s.

proves that the optimization problem (2) can be solved by linear solvers and in particular, modeled as a Min-Cost Flow problem.

Proposition 1. *The integrality gap of Problem (2) is zero and the solution of the relaxed linear program is integral.*

Proof. We show how the problem can be reduced to the min cost flow. Introducing variables X_j as auxiliary variables corresponding to offer nodes in the network (See Fig. 2), equation 2 can be rewritten in terms of Min-Cost flow (with unimodular constraint matrix) as

$$\begin{array}{ll} -n + \min & \sum_{i\in N}\sum_j (1 - P(Y_i|F_i, O_j))X_{i,j} \\ s.t. & \sum_j X_{i,j} = 1 \ \ \forall i \\ & \sum_i X_{i,j} = X_j \\ & X_j \le b_j \ \ \forall j \\ & 0 \le X_{i,j} \le 1 \quad \forall\ i,\ j \end{array} \tag{3}$$

Hence the integrality gap of problem (2) is zero and the solution of the relaxed linear program is integral. □

Leveraging this approach to solve (2) provides benefits over a greedy approach. The greedy approach ranks customers according to $Pr_A(Y)$ (propensity to convert given offer A) and offer it to the top ranked customers till offer capacity is satisfied. It then proceeds to do the same for the remaining offers among the remaining customers. However, this is not optimal to maximize the expected conversion yield since it does not take into account the optimal incremental yield (yield from offer – yield from no offer) or stack ranks based on $Pr_A(Y)$ rather than $Pr_A(Y) - Pr_N(Y)$ ($Pr_N(Y)$ is propensity to convert without an offer). The following toy example illustrates this. Say we have 200 users and we can offer A to only 100 of them with users belonging to two different propensity segments as indicated in table 1. The greedy policy offers A to segment 2 since it has a higher propensity to convert given offer A (70%) while the optimization framework obtains an extra 7.5% increment in yield since it offers A to segment 1 which has higher incremental yield.

We now illustrate the mathematical formulation of general setting of the Problem (2) below:-

$$\begin{array}{ll} \max\limits_{(X_{i,j})} & \sum\limits_{i\in N}\sum\limits_{s\in S} P(Y_i|F_i, S_s)X_{i,s} \\ \text{s.t.} & \sum\limits_s X_{i,s} = 1 \ \forall i \\ & \sum\limits_s\sum\limits_i X_{i,s,j} \le b_j \ \forall j \\ & X_{i,s} = X_{i,s,j} \ \forall\ i,\ s\ \&\ j \in S_s \\ & X_{i,s,j} = 0 \ \forall j \notin S_s \\ & \sum\limits_s\sum\limits_j X_{i,s,j} \le c_i \ \forall i \\ & X_{i,s} \in \{0,1\} \ \forall i,\ s \\ & X_{i,s,j} \in \{0,1\} \ \forall i,\ s,\ j \end{array} \tag{4}$$

Here, variable $X_{i,s}$ denotes the fact that user i is offered subset $S_s \in S$ of offers and $P(Y_i|F_i, S_s)$ denotes the propensity for user i to convert if given the subset $S_s \in S$ and historical features F_i. The first constraint illustrates the fact The $X_{i,s,j}$ denotes the fact that offer j is shown to user i as a consequence of being shown subset S_s. The fact that offer j might not belong to S_s is denoted by setting $X_{i,j,s} = 0 \forall j \notin S_s$. The third and fourth constraints are to ensure consistency between the indicator variables. The fifth constraint refers to the fact that user i is limited to at most c_i offers.

We state without proof that the general setting (4) can also be reduced to min-cost and hence be solved by linear solvers. The proof follows exactly along the lines of proposition 1 by introducing auxilary variables.

Proposition 2. *The integrality gap of the generalized single stage program (4) is zero and the solution of the relaxed linear program is integral.*

4.2 Scaling

The current solution could be scaled for application to a large population by the idea of clustering specifically when all users want the same number of items/offers. Here we cluster the population into various clusters (based on feature similarity) and assign an offer to the entire cluster. We can use clustering on the population features into k clusters (say k =100) such that all clusters are of roughly the same size . We rerun the optimization on clusters (Clusters are now treated as individuals) and everyone in the cluster

is given the same offer. The above set of equations, (2), can be modified as follows:

$$\begin{aligned} \max \quad & \sum_{i \in K} \sum_{j \in J} \tilde{P}_{i,j} X_{i,j} \\ s.t. \quad & \sum_{j} X_{i,j} = 1 \quad \forall i \\ & \sum_{i} X_{i,j} \leq |K| f_j \quad \forall j \\ & X_{i,j} \in \{0,1\} \end{aligned} \tag{5}$$

where $X_{i,j}$ denotes the indicator variable where cluster i receives offer j, K denotes the set of all clusters $\{1, 2, \ldots, k\}$, f_j denotes the fraction of clusters that are offered offer j. The profit of assigning an offer j to cluster i is the sum of the profits of assigning an offer j to all people in the cluster. Hence, we have the equation $\tilde{P}_{i,j} = \sum_{q \in i} P(Y_q | F, O_j)$.

Clustering helps with scaling at a large with the tradeoff being loss of accuracy (due to every user in the cluster being allocated the same offer). However, it is is practically only useful when you are targeting over 20 million variables (since most commercial solvers can easily solve upto large scale linear optimization within 30 mins with a higher RAM). Also using traditional clustering algorithms(k-means and Hierarchical) can be quite slow and can create bottle necks in the system. It is imperative to use clustering algorithms that are fast and scale well for high dimensional spaces and large number of problems. Clustering done on a reduced subspace (e.g. just based on propensity scores itself) can be suboptimal as inter distance between points can decrease as points are mapped to a lower dimensional space.

4.3 Sequential Time Period Offering

In this section we extend the single point in time solution to a multi-period approach (solved up front at a single point in time). That is, we build a sequential optimization model where we make the offer decision for multiple time periods. We illustrate the equations for two time periods in equation (6). This method can be easily illustrated for multiple time periods. We again illustrate the mathematical for the simple setting when each user is given a single offer for each time period and each subset $S_s \in S$ has cardinality one.

$$\begin{aligned} \max \quad & \sum_{i \in N} \left(\sum_{j \in J} P(Y_i | F_i, O_j) X_{i,j} + \sum_{j,k \in J} \tilde{P}(Y_i | F_i, O_j, O_k) X_{i,j,k} \right) \\ s.t. \quad & \sum_{j} X_{i,j} = 1 \quad \forall i \\ & \sum_{j} \sum_{k} X_{i,j,k} = 1 \quad \forall j,k \\ & \sum_{k} X_{i,j,k} = X_{i,j} \quad \forall i,j \\ & \sum_{i} X_{i,j} + \sum_{i} \sum_{l} X_{i,l,j} \leq b_j \quad \forall j \\ & X_{i,j} \in \{0,1\} \quad \forall i,j \\ & X_{i,j,k} \in \{0,1\} \quad \forall i,j,k \end{aligned} \tag{6}$$

Here, indicator variable $X_{i,j}$ denotes the fact that user i is offered j in period one and indicator variable $X_{i,j,k}$ denotes the quantity of offer k given to user i given the fact that user i is offered offer j in period one. $\tilde{P}(Y_i | F_i, O_j, O_k)$ indicates the propensity to convert with offer k in the second period given that the user did not convert in the first period The first two constraints refer to total number of offers in all that we offer to user i. The third constraint ensures consistency between the indicator variables. The fourth constraint refers to the capacity of offer j that can be offered to users.

We state without proof that the sequential setting (6) can also be solved by linear solvers. The proof follows exactly along the lines of proposition 1 after we introduce auxiliary variables X_j and $X_{j,k}$ for the first stage and second stage offers respectively).

Proposition 3. *The integrality gap of the sequential program (6) is zero and the solution of the relaxed linear program is integral.*

This idea could easily be extended to k time periods by creating m^k variables (all possible offer sequences) and the min-cost formulation would still hold in its current form. However, sequential formulation for more three time periods from a modeling and practical standpoint has some problems. First one, the estimation of propensities over multiple time periods is a hard problem as a lot of temporal effects come into play Preallocation of offers for later time period might lead to suboptimal solutions in the earlier time period if the propensity models are incorrectly built without temporal effects. Secondly, the sequential formulation also needs to rebalanced after every time period to account for customers who are already converted.

5 EXPERIMENTAL RESULTS

5.1 Data

We use the email campaign dataset for an Internet-based Retailer that is provided by Hillstrom [15] which is well known in the uplift modeling literature[26]. We used this publicly available dataset for reproducibility and as it closely mimics multiple promotion offers as it has one control and two treatments. Note a key difference lies in the nature of marketing - emails may be targeted to customers who may be newbies or returning customers whereas an online subscription service usually targets people with promotion offers when they are using the platform. The email dataset contains information about 64,000 customers who were subjected to a test e-mail campaign for two weeks and had last purchased within at most twelve months. Unfortunately, we couldn't find any public datasets larger than 64000 which has multiple treatments. However we have used the same approach for 5 MM customers (15 MM variables and 5 MM constraints).

The email campaign was run as follows: -

- $\frac{1}{3}$ were randomly chosen to receive an e-mail featuring men's merchandise (ME),
- $\frac{1}{3}$ were randomly chosen to receive an e-mail featuring women's merchandise (WE),
- $\frac{1}{3}$ were randomly chosen to not receive an e-mail (NE).

Using this data, we solve the problem of maximizing the expected visit yield under an optimization framework where we determine which users are to be offered one of the three offers $\{ME, WE, NE\}$ depending on their propensity of visit and budget constraints. We

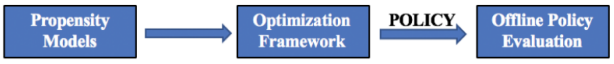


Figure 3: Task overflow

could target conversions as well however the signal to noise ratio (SNR) for conversions was quite low as compared to visits. This task has three stages as depicted in Fig. 3. The optimization engine requires the estimation of the propensity to visit. The engine allocates the offers and the performance is assessed using offline policy evaluation as described in Sec. (3.2) specifically when the optimization policy suggests targeting using a different email from the one used during the campaign.

5.2 Propensity Model Results

The data was split randomly into training, validation, and test sets in a 60:20:20 ratio. Success is defined if a visit occurred. We obtain the probability values $p_{i,j}$ using traditional supervised learning methods in Hastie et al. [13] as compared to MNL models since at each instance only one email is presented to each customer. We built one binary classification model per email type with user profile information as features. We used gradient boosting for building the classifier models using user profile information as features. However in order to handle large scale and imbalanced data (small percentage of positive class), balanced random forestsas described in [8] are a better alternative as they tend to be computationally and statistically efficient. We evaluated the models based on their accuracy of probability estimates and standard classification metrics such as F1- score, AUC-PRC, AUC-ROC. It is imperative to target metrics such as AUC-PRC and F1 score for binary classifiers for conversions since the data tends to imbalanced and AUC ROC might mask the performance of poor classifiers [17].

We finally used isotonic regression to calibrate the prediction probabilities in the validation set [30], leading to a decrease in the Brier score between predicted and observed probabilities. The resulting probability fits are good (how accurately the predicted scores align with the actual visit rates of users) as shown in Fig. 4.

5.3 Offline Evaluation

We compare the DM, DR, IPS and SNIPS estimators as described in Section 3.2 for single period optimization and compare it to competing policies. We target at most 10% of the population with ME and WE (atmost 6400 people are sent ME and atmost 6400 people are sent WE). We compare the optimization against the following competing policies :-

- **Randomized policy:** The randomized policy randomly samples 20% users (budget capacity of email ME +WE) and targets half of them randomly with ME and the other half with WE. The randomized policy estimator values as shown in Table 2 are obtained after averaging over 50 different random samples.
- **ME Based Ranking**: The ME based ranking ranks users based on Pr_{ME} (propensity to visit when targeted with ME) and selects the top ranked 10%(b_{ME}) (budget of ME) of the users and targets them ME and then among the remaining users; selects 10%(b_{WE}) (budget of WE) users having the highest Pr_{WE}.
- **WE Based Ranking**: The WE based ranking ranks users based on Pr_{WE} (propensity to visit when targeted with ME) and selects the top ranked 10%(b_{WE}) (budget of WE) of the users and targets them WE and then among the remaining users; selects 10%(b_{ME}) users having the highest Pr_{ME}.

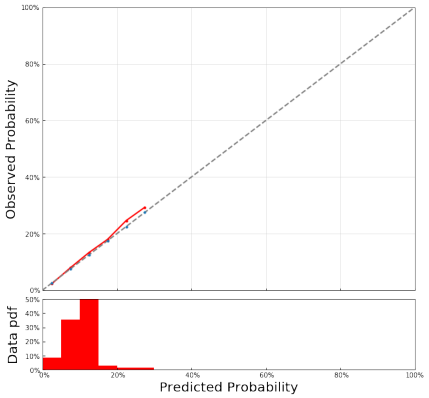


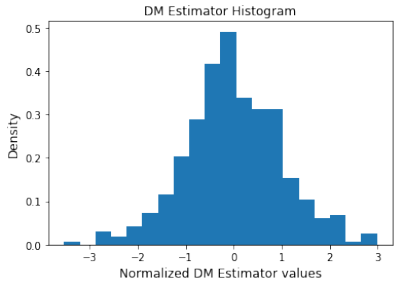
Figure 4: Probability Calibration: The top part of the figure illustrates the accuracy of the propensity scores by comparing the expected percentage of visitors as predicted by the model (blue line) against the actual percentage of visitors (red line). The alignment of the expected to actual illustrates the model accuracy as a probability estimate. The bottom part of the figures represents the distribution of scores observed in the population and help explain the fits.

The results are shown in Table 2. The maximum possible yield is obtained when the offer constraints are completely relaxed and every user is sent an email for which she has the highest propensity to visit . The minimum possible yield is obtained by obtaining the estimator rewards when no one is sent any email. As seen from Table 2, the mean reward from optimization outperforms ranking based (random) policy by atleast 8% (16%) across all estimators. The variances values are small and are not depicted in the table.

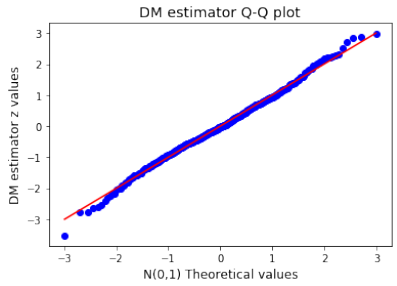
An important aspect to address here is in regards to the calculation of confidence intervals for the optimization policy. One of the assumptions made in offline policy generation is that the policies are stationary and the action for sample i is independent of the history of past actions for other samples. However, this assumption is invalid for the optimization policy since it solves the problem under the global capacity constraints making it dependent on past actions and other samples. Hence the confidence intervals for the estimators cannot be calculated under the i.i.d. assumption. We use the bootstrap method for calculating the confidence bounds. The verification of normality assumption for the bootstrapped values as shown is done empirically and the q-q plot for a 1000 samples is shown in Fig. (5). A similar approach was used in Li et al. [24].

5.4 Sequential Offers

Since we do not have any offer data for sequential offers, we assume an independence between the probability of conversion for offers



(a) DM bootstrap Histogram



(b) DM bootstrap values QQ plot

Figure 5: QQ Plot for DM reward values indicates that the DM bootstraped values are "close" to a normal distribution.

	DM	DR	IPS	SNIPS
Randomized	11.75 %	11.75 %	11.87 %	11.76 %
ME Based Ranking	12.98 %	13.45 %	13.64 %	13.47 %
WE Based Ranking	12.43 %	12.70 %	12.93 %	12.76 %
Optimization	**13.55** %	**14.07** %	**14.10** %	**14.05** %
Minimum Yield [1]	10.51 %	10.60 %	10.71 %	10.61 %
Maximum Yield [2]	19.66 %	19.75 %	19.93 %	19.75 %

Table 2: The table compares the mean of the expected settled yield for different policies using the different offline estimators. The estimator reward values are calculated over 100 bootstrap samples.

for two sequential time periods. Hence in Eq. (6), we assume

$$\tilde{P}(Y|F, O_j, O_k) = (1 - P(Y|F, O_j))P(Y|F, O_k).$$

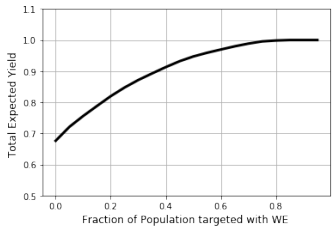
. We only compared the DM estimator (since no logging probability is known for sequential offering) for sequential offering as compared to myopic optimization over two periods. The sequential offering has a lift of 12% over the myopic policy over two time periods when the ME and WE were again targeted to 10% of the population when the budget capacity of the offers is split equally among the two myopic periods. It is also possible to split the budgets smartly and do well in single period solutions but we did not explore this in detail.

6 PRACTICAL CONSIDERATIONS

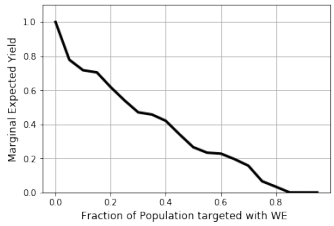
6.1 Optimal Budget Determination

An important practical consideration is setting the budgets for the offers in Prob (2). As we relax constraints for the offers (e.g. offering more people a particular offer), it is possible to get higher expected settled yield. Figure (6) indicates the total expected yield and problem of decreasing returns to scales for the email dataset as more and more people are targeted with WE. The marginal settled yield values are similar to the dual value of the offer. We can create an efficient frontier surface to determine the optimal offer constraints analogous to the frontier curves in portfolio selection [28]. However this surface can be non-convex. An approximately optimal solution is to determine the optimal budget for one offer at a time when the marginal expected yield goes below a predetermined threshold. For the email targets dataset we have two different order sequences {ME, WE} (first determine the budget for ME and then WE) and {WE, ME}. For m offers, this may lead to $m!$ possible orderings to consider. However, we can determine the offer sequence greedily based on business constraints. For example if we need to be judicious with a particular offer, we would determine its budget last. As shown in Table 3, we can choose the sequence {WE, ME } if the ME email is costlier and both sequences have nearly the same total expected yield.

It is imperative to keep in mind that the lift of optimization over greedy only when the budget fractions (relative to population size) are small (when only a small fraction of people are targeted). If budget fractions are large, the low lift might not justify the engineering expense of building and deploying such a system.



(a) Total Expected Yield



(b) Marginal Expected Yield

Figure 6: The total (marginal) expected yield versus the fraction of population target with WE. Values are normalized.

Offer Sequence	WE fraction	ME fraction	Expected Yield
{WE, ME}	0.7	0.6	19.5 %
{ME, WE}	0.15	0.9	19.66 %

Table 3: The table compares the budget for the email dataset for different offer sequences. The email fraction is the maximum fraction of population that can be targeted with that email offer. The budget for an email is determined to be the population fraction at which the marginal expected yield falls down below 10%.

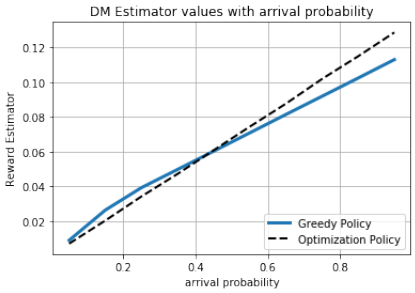


Figure 7: DM Estimator for Optimization and Greedy Policy for different arrival probabilities. The greedy policy has been evaluated over various permutations of user arrivals.

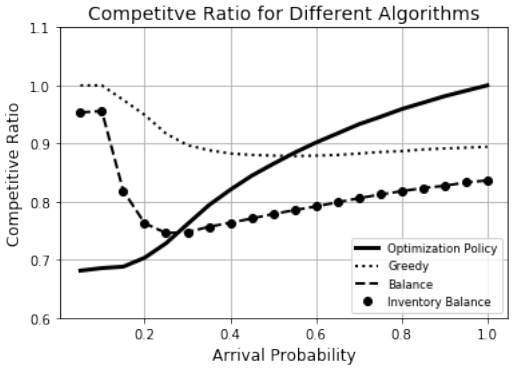


Figure 8: CR of different algorithms for different arrival probability. For each arrival probability, the minimum value of the CR over various arrival input instances is demonstrated.

6.2 Matching offline estimation with real-time traffic arrival

One of the challenges with preassigning offers offline is that the customers pre-assigned with a given offer might not arrive on the platform. Ideally this should not be case as the propensity models do involve historical features for customer and customers with high recency usually have a low propensity of conversion for all offers and the optimization framework would not assign any offer to such customers if the propensity estimates are correct. Nevertheless, it is imperative to test and understand the performance under the different arrival rate.

We first investigated the performance of the offline optimization policy vs greedy policy on the expected settled yield for real-time traffic by assuming only a percentage of the population arrives. It is empirically hard to accurately determine the arrival probability of an individual user. However, the arrival rate of the entire population can be better estimated with greater confidence. If on average only a fraction of the targeted population arrives, a greedy policy that assigns a user an offer with the highest propensity to convert (while maintaining the budget constraints) in a first come first serve manner tends to perform better than the optimization policy. However, as a larger fraction of the target population arrives the optimization policy wins over the greedy policy. These observations are demonstrated for the email dataset by Figure 7.

There exist other algorithms in practice that are robust to arrival sequence such as balance algorithm in Mehta et al. [29] and inventory balancing (IB) algorithm proposed in Golrezaei et al. [12]. These algorithms work in the following way: for each user arriving, give the offer with the highest penalized reward. The penalized reward in this case would be to multiply the propensities by a penalty which is a convex decreasing function of the remaining inventory of the offer. For instance, we provide the offer $j = \arg\max_O \{p_O f(x_O)$ where x_O is remaining budget fraction and f is the penalty function. The balance and the inventory use the penalty functions of $1-\exp(b-1)$ and $\frac{e}{e-1}(1-\exp(-b))$ respectively where b is the remaining inventory. We evaluate the performance of greedy, balance and IB and optimization policy for the email dataset as shown in Figure 8. We use the metric of competitive ratio (C.R.) to evaluate the online performance of all these algorithms. The C.R. for an algorithm ALG in the adversarial setting is defined as the minimum ratio of the reward obtained by the algorithm $ALG(I)$ and the optimal reward, $OPT(I)$ over all input instances I. Hence, $C.R. = \min_I \frac{ALG(I)}{OPT(I)}$. Both Balance and IB have a competitive ratio of $1 - \frac{1}{e}$ under the adversarial setting. For each instance I, $OPT(I)$ is determined by solving Eq. (2) for the sub-set of population that arrives. As demonstrated in Fig. (8), 'robust' solutions might not perform well in practice since most arrival sequences in practice aren't adversarial.

7 CONCLUSION

The work described in this article solves the problem of optimally recommending plans/offers/items to users to maximize a business objective (profit, revenue, number of subscriptions, etc.) in the presence of constraints (limited quantity of items, user eligibility constraints, etc.). We use propensities as surrogates for user rewards and apply a min-cost flow based approach to solve the optimization problem in the presence of business constraints. We demonstrate the efficiency of our policy over other policies in maximizing yield with the same constraints and cost to the business in an offline setting. We show how our method can be applied to multi-period and real-time settings. However, more work is required to propose solutions that are dynamic in nature. Although we proposed a method for sequential offering, a dynamic sequential offering tailored to each user is still open.

REFERENCES

[1] Parameswaran Aditya, Venetis Petros, and Hector Garcia-Molina. 2011. Recommendation Systems with Complex Constraints: A Course Recommendation Perspective. *ACM Trans. Inf. Syst.* 29, 4, Article 20 (Dec. 2011), 33 pages. https://doi.org/10.1145/2037661.2037665

[2] Deepak Agarwal, Shaunak Chatterjee, Yang Yang, and Liang Zhang. 2015. Constrained optimization for homepage relevance. (2015), 375–384.

[3] Shipra Agrawal and Nikhil R. Devanur. 2015. Fast Algorithms for Online Stochastic Convex Programming. (2015), 1405–1424.

[4] Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. 2014. A Dynamic Near-Optimal Algorithm for Online Linear Programming. *Oper. Res.* 62, 4 (Aug. 2014), 876–890. https://doi.org/10.1287/opre.2014.1289

[5] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2018. Bandits with Knapsacks. *J. ACM* 65, 3, Article 13 (March 2018), 55 pages. https://doi.org/10.1145/3164539

[6] Joseph Konstan Badrul Sarwar, George Karypis and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th*

international conference on World Wide Web (2001), 285–295.
[7] D.P. Bertsekas. 1991. Linear Network Optimization. *MIT Press, Cambridge, MA* (1991).
[8] Andy Liaw Chao Chen and Leo Breiman. 2004. Using Random Forest to Learn Imbalanced Data. *https://statistics.berkeley.edu/tech-reports/666* (2004).
[9] Young Choi Deuk H. Park, Hyea K. Kim and Jae K. Kim. 2012. A literature review and classification of recommender systems research. *Expert Systems with Applications* (2012), 10059–10072.
[10] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. (2011), 1097–1104. http://dl.acm.org/citation.cfm?id=3104482.3104620
[11] Guillermo Gallego and Garrett van Ryzin. 1994. Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons. *Manage. Sci.* 40, 8 (Aug. 1994), 999–1020. https://doi.org/10.1287/mnsc.40.8.999
[12] Negin Golrezaei, Hamid Nazerzadeh, and Paat Rusmevichientong. 2014. Real-Time Optimization of Personalized Assortments. *Manage. Sci.* 60, 6 (June 2014), 1532–1551. https://doi.org/10.1287/mnsc.2014.1939
[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. The Elements of Statistical Learning. (2001).
[14] T. Hesterberg. 1995. Weighted average importance sampling and defensive mixture distributions,. *Technometrics, pp. 185–194,* (1995).
[15] Kevin Hillstrom. 2008. The MineThatData e-mail analytics and data mining challenge. *http://blog.minethatdata.com/2008/03/minethatdata-e-mailanalytics-and-data.html* (2008).
[16] Edward L. Ionides. 2008. Truncated importance sampling. *Journal of Computational and Graphical Statistics, 17(2):295–311* (2008).
[17] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. 2013. Facing Imbalanced Data–Recommendations for the Use of Performance Metrics. (2013), 245–251.
[18] Thorsten Joachims and Adith Swaminathan. 2016. Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement. (2016), 1199–1201. https://doi.org/10.1145/2911451.2914803
[19] Pooria Joulani, Andras Gyorgy, and Csaba Szepesvári. 2013. Online learning under delayed feedback. (2013), 1453–1461.
[20] Jaya Kawale Konstantina Christakopoulou and Arindam Banerjee. 2017. Recommendation with Capacity Constraints. *ACM on Conference on Information and Knowledge Management* (2017), 1439–1448.
[21] Diane Lambert and Daryl Pregibon. 2007. More Bang for Their Bucks: Assessing New Features for Online Advertisers. (2007), 7–15. https://doi.org/10.1145/1348599.1348601
[22] Tor Lattimore and Csaba Szepesvari. 2018. Bandit Algorithms. (2018).
[23] Damien Lefortier, Adith Swaminathan, Xiaotao Gu, Thorsten Joachims, and Maarten de Rijke. 2016. Large-scale Validation of Counterfactual Learning Methods: A Test-Bed. *CoRR* abs/1612.00367 (2016). arXiv:1612.00367 http://arxiv.org/abs/1612.00367
[24] Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. 2015. Counterfactual Estimation and Optimization of Click Metrics in Search Engines: A Case Study. (2015), 929–934. https://doi.org/10.1145/2740908.2742562
[25] Dudík Miroslav; Erhan Dumitru; Langford John; Li Lihong. 2014. Doubly Robust Policy Evaluation and Optimization. *Statistical Science, Vol. 29, No. 4, 485-511; doi:10.1214/14-STS500* (2014).
[26] D. Lo, V.S.Y.; Pachamanova. 2015. From Predictive Uplift Modeling to Prescriptive Uplift Analytics: A Practical Approach to Treatment Optimization While Accounting for Estimation Risk. *Journal of Marketing Analytics, 3 (2): 95* (2015).
[27] Victor S. Y. Lo. 2002. The True Lift Model: A Novel Data Mining Approach to Response Modeling in Database Marketing. *SIGKDD Explor. Newsl.* 4, 2 (Dec. 2002), 78–86. https://doi.org/10.1145/772862.772872
[28] Harry Markowitz. 1952. Portfolio Selection. *The Journal of Finance, vol. 7, no. 1, pp. 77–91* (1952).
[29] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM)* 54, 5 (2007), 22.
[30] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting Good Probabilities with Supervised Learning. (2005), 625–632. https://doi.org/10.1145/1102351.1102430
[31] N. J.; Radcliffe and P. D. Surry. 1999. Differential response analysis: Modelling true response by isolating the effect of a single action,. *in Proceedings of Credit Scoring and Credit Control VI, Credit Research Centre, University of Edinburgh Management School* (1999).
[32] P. Rosenbaum and D. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika, 70(1):41–55,* (1983).
[33] Aixin Sun Shuai Zhang, Lina Yao and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM computer survey 1* (2018), Article 1, 35 pages.
[34] L. Si and R. Jin. 2003. Flexible mixture model for collaborative filtering. *Proceedings of the 20th International Conference on Machine Learning (ICML '03)* (2003), 704–711.
[35] Cohen Maxime C; Ngai-Hang Z Leung; Kiran Panchamgam; Georgia Perakis; Anthony Smith. 2017. The impact of linear optimization on promotion planning. *Operations Research 65(2)* (2017), 446–448.
[36] Xiaoyuan Su and Taghi Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* (2009), Article ID 421425, 19 pages.
[37] Adith Swaminathan and Thorsten Joachims. 2015. The Self-normalized Estimator for Counterfactual Learning. (2015), 3231–3239. http://dl.acm.org/citation.cfm?id=2969442.2969600