On targeting Markov segments

Moses Charikar* Ravi Kumar† Prabhakar Raghavan† Sridhar Rajagopalan† Andrew Tomkins†

Abstract

Consider two user populations, of which one is *targeted* and the other is not. Users in the targeted population follow a Markov chain on a space of n states. The untargeted population follows another Markov chain, also defined on the same set of n states. Each time a user arrives at a state, he/she is presented with information appropriate for the targeted population (an advertisement, or a recommendation) with some probability. Presenting the advertisement incurs a cost. Notice that while the revenue grows in proportion to the flow of targeted users through the state, the cost grows in proportion to the total flow (targeted and untargeted) through the state. How can we compute the best advertisement policy?

The world-wide web is a natural setting for such a problem. Internet service providers have trail information for building such Markovian user models where states correspond to pages on the web. In this paper we study the simple problem above, as well as the variants with multiple targetable segments. In some settings the policy need not be a *static* probability distribution on states. Instead, we can *dynamically* vary the policy based on the user's path through the states.

We provide characterizations which reveal interesting insights into the nature of optimal policies, and then, use these insights for algorithm design. Targeting problems do not seem amenable to solutions using methods from familiar fields such as Markov decision processes.

*Computer Science Department, Stanford University, CA 94305. Supported by the Pierre and Christine Lamond Fellowship, NSF Grant IIS-9811904 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: `moses@theory.stanford.edu`. This work was done while the author was visiting IBM Almaden Research Center.

†IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. Email: `{ravi, pragh, sridhar, tomkins}@almaden.ibm.com`

1 Introduction

We study stochastic optimization problems of the following genre. Consider two user populations, of which one is *targeted* and the other is not. Users in the targeted population follow a Markov chain on a space of n states. The untargeted population follows another Markov chain, also defined on the same set of n states. Each time a user arrives at a state, information appropriate for the target population – say, a recommendation or an advertisement (henceforth an "ad") – is pitched to the user with some probability. If the ad is pitched to a targeted user, then a revenue is obtained (and that user disappears from the system). The act of pitching an ad incurs a cost. Therefore the cost of a policy for pitching ads at a state depends on sum of the targetable *and* non-targetable traffic through that state, while the revenue depends *only* on the targetable traffic. More generally, we may have k Markov chains, each corresponding to a different targetable segment of users (the home-buyers, the students preparing for the SAT's, etc.). We have available ads directed towards each segment (pitching a realtor, or an SAT vocabulary list, perhaps), and revenue is obtained only when an ad is pitched to a user from the appropriate segment.

We study the *static* problem, in which an ad is chosen with some probability that depends only on the state. We also study the *dynamic* policies, where the chosen ad can additionally depend on the path taken by the user to the current state. We consider each version of the problem under two metrics. In the *budgeted* metric, we seek to maximize revenue subject to a prescribed ad budget (i.e., an upper bound on overall cost). In the *unbudgeted* metric, we seek to maximize the excess of revenue over ad cost, without a fixed upper limit on the cost.

CONTEXT. The world-wide web is a natural setting for such problems. Currently, targeted advertisements on the web yield a large multiple of the revenue of untargeted advertisements. One of our motivations is to develop a mathematical basis for studying such issues. The states in our Markov chains correspond naturally to web pages; transition probabilities within a particular chain correspond to the browsing patterns of a particular segment of users. For instance, a chain modeling users interested in buying a car might show that 40% (say) of such users proceed from `Yahoo!`'s Auto shopping page to `autoweb.com`, while 30% proceed to `Carpoint.com`. While we cannot hope that every surfer action is best modeled as Markovian, we believe this first approximation yields valuable insights.

In this work we do not seek to infer the Markov chains; rather, we view them as having been "learned" from the surfing trails of prior surfers as a prelude to tackling the optimization problems we study. Internet service providers (ISP's) have access to such trail information, and can thus infer such models; more on this below. With multiple targetable segments, a large number of sample trails (some of which may be tagged with downstream transaction information of the form "this surfer went on to buy a car" or simply "this surfer clicked on an ad pitching SAT preparation materials") may be partitioned into segments (implicitly inducing a set of Markov chains). One approach to this would be to find a solution to a generalization of a *segmentation problem* [KPR98]. However, our emphasis here is not on inferring or segmenting these Markov models from trail data; we are interested in what we can do given these models. An interesting direction for further work would be the integration of segmented Markov chain inference and ad policy construction.

SUMMARY OF CONTRIBUTIONS. Section 3 gives a more complete description of our results; we give a quick overview here. First, unlike Markov decision processes (MDP's) [Whi92], there need not exist finitely-specified optimal strategies for the dynamic version of our problem. Nonetheless, the optimal strategy for the dynamic version can be reduced to a point-location problem in k-dimensional space, where k is the number of segments. While there are at most $k+1$ regions, and k of them are convex, the boundaries between the regions may be arbitrarily complex and may have no simple specification. Even so, at the loss of an amount ϵ of the revenue, it is possible to approximate the regions and the boundaries between them in polynomial size; this allows us to devise an algorithm for this problem.

Even if the user models are more general (not necessarily Markovian), the optimal dynamic strategy is deterministic. In contrast, the optimal static policy (which knows only the state a user is in, not the history of the user) is, in general, randomized. In fact, the gap in revenue between randomized and deterministic static policies can be arbitrarily large. For this case we provide approximation algorithms that guarantee a revenue within $(1-1/e)$ of the optimum.

RELATED WORK. Project 2000 [P00] at Vanderbilt University maintains an extensive collection of resources pertaining to marketing on the web. There has been much work on characterizing users in order to perform more effective targeted marketing [CN].

Berman, Krass, and Xu [BKX95, BKX96, BKX97] and Hodgson [Hod90] study *flow-intercepting facility location:* the placement of billboards or gas-stations at the nodes of a flow network (modeling traffic on a road system) so as to maximize the amount of flow that passes through at least one of the facilities. There is a cost given for locating a facility at each node. They show that given a budget, a simple greedy heuristic gives a solution intercepting at least $(1-1/e)$ times the flow in the optimal solution. Our work differs from theirs due to the added flexibility the web offers: (1) Whereas their facilities are located immutably at a set of nodes, even our static problem allows for probabilities with which ads are pitched. Certainly, one cannot imagine a gas station appearing probabilistically at a traffic intersection; but on the web, mixed advertisement policies are in fact the established norm. (2) The ability of an ISP to easily monitor trails gives rise to our dynamic problem; this has no analog in the world of flow interception.

The theory of Markov decision processes [Whi92] gives one way of viewing our setting. Unfortunately, it offers very few computational cues for us. Indeed, as we point out in the next section, our dynamic problem cannot be modeled by any finite-state Markov decision process.

Kumar *et. al.* [KRRT98] study *recommendation systems*, in which the system uses samples from the past behavior of users to give them recommendations. In their setting a user is a probability distribution on segments, and the challenge is to infer this competitively from the sample. In contrast, our users are engaged in a single targetable activity (segment) in any surfing session; more importantly, this activity is to be competitively inferred (for pitching) from the *sequence* of user actions (rather than a set of transactions as in [KRRT98]).

2 The model

We are given k Markov chains on the same set of n states. The states are represent the opportunities to advertise to users. The transitions represent observable user behavior. We assume that state 1 is the start state for all k processes, and n is the finish state (in the web context, this is the state corresponding to a user exiting from a browser session). We also have an initial mix $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ of users belonging to each segment; $\sum_{i=1}^{k} \alpha_i = 1$. Each arriving user independently chooses segment i (and thus follows the i-th Markov chain) with probability α_i.

COST AND REVENUE. Our model has two sets of parameters: a cost function $cost(i, v)$ which is the per-pitch cost of showing ad i (that is, an ad designed for a user from segment i) to a user in state v, and a revenue function $rev(i, v)$ which is a per-user revenue obtained on correctly pitching ad i to a user in segment i, while at v. Since both cost and revenue are specified on a per-user basis, the net cost or revenue from placing an ad at state u depends on the flow at (i.e., the number of users who pass through) a node.

This provides a framework for studying two phenomena that are well-known on the web: first, that advertising on a site with large traffic (like `yahoo`) is costlier; second, that advertising on a site with less (but better segmented) traffic (like `medweb`) is more expensive on a per user basis.

Our model allows non-targeted segments, i.e., segments which are not targeted by any advertiser. These are amongst our k segments and modeled simply by setting the associated revenue values to 0. We wish to point out the following important distinction: an untargeted segment simply adds to the flow at some (or all) vertices, but never generates revenue. Thus, it is desirable to not pitch in situations where there is a large amount of traffic belonging to untargeted segments. On the other hand, even if all segments were targeted, it might be reasonable to avoid pitching at a particular state. This depends on whether the mix of users visiting the state and the cost of advertising at the state make the net profit worthwhile.

We make one other assumption for definiteness: when a user is correctly targeted, the user immediately exits the system. We obtain a unit of revenue in this case, which should be viewed as an expectation over users who go on to make a purchase as well as those who do not. In other words, repeatedly targeting a prospective car-buyer with car ads does not increase the chance of a purchase. The effect of repeated advertisements on the probability of yielding revenue has been the subject of considerable study in conventional media and direct marketing [BH96], but has not been studied carefully for the web. On the other hand, it is the web that offers the facility for simply (and measurably) connecting a user to a service or product via a targeted advertisement.

ADVERTISEMENT POLICIES. A *static advertising policy* is defined by values $\sigma_{i,v}$ which denote the probability of pitching ad i at state v. Clearly $\forall v, \sum_i \sigma_{i,v} \leq 1$. In the *static ad problem* (SAP), $\sigma_{i,v}$ must be some fixed constant probability.

In the *dynamic ad problem* (DAP), a policy is defined by functions $\sigma_{i,v}(x)$ which depend on the user's history x (i.e., the user's path from the start state to state v) and return the probability of pitching ad i at state v to a user with history x. Again, $\forall v, \forall x, \sum_i \sigma_{i,v}(x) \leq 1$. This setting applies to ISP's who serve every browser click, and thus know the instantaneous surfing trajectory for each client. Note that in the dynamic setting if we unsuccessfully pitch an ad for a segment j, we can thereafter concentrate our policy on segments other than j, for that user.

For any advertisement policy (static and dynamic), we can compute the expected cost and the expected revenue for the given Markovian user model. In the budgeted problem, given a bound B on the expected cost, the objective is to obtain a policy that maximizes expected revenue. In the unbudgeted problem, the objective is to obtain a policy that maximizes the expected profit, i.e., the expected revenue minus the expected cost; here there is no upper bound on the cost.

For an algorithm ALG, we define $rev(\text{ALG})$ to be the total expected revenue of the policy produced by the algorithm. Similarly, $cost(\text{ALG})$ is defined to be the expected cost and $prof(\text{ALG})$, the expected profit, i.e., $rev(\text{ALG}) - cost(\text{ALG})$.

3 Overview of results

The two technical sections of the paper discuss the static (Section 4) and dynamic (Section 5) versions of the problem. In each section, we discuss both the budgeted and unbudgeted versions of the problem.

STATIC ADVERTISEMENT PROBLEMS. The first observation for the static problem is that the optimal solution is not necessarily deterministic (i.e., where $\sigma_{i,v} \in \{0, 1\}$). Example A in Figure 1 shows a two state system where the mix of users in the system gets *refined* with time — waiting for a while results in a mix of users biased towards the targeted segment. The first segment in the example is targeted; the second is not; and the initial mix is uniform. A static policy that pitches at state u with some small probability is able to pitch to most of the targetable users, but avoids the cost of pitching to non-targetable users by probabilistically "waiting" until most of them have leaked away (exited). As the initial mixture becomes increasingly slanted towards targeted users, the benefits of a non-integral solution become arbitrarily large.

We give a greedy approximation algorithm to find such a non-integral set of pitching probabilities. This algorithm, called the SAND algorithm, generates revenue within $(1 - 1/e)$ of the optimal for any fixed budget.

If the budget is not fixed, but the goal is to maximize profit (revenue minus cost), the problem appears to be more difficult. It is related to the *prize-collecting set cover problem* on which there seems to be no prior work: given a collection of sets over $[n]$ and a revenue associated with each element and cost associated with each set, choose sets so that the revenue of the covered elements minus the cost of the chosen sets is maximized. We show that the natural greedy algorithm which repeatedly chooses the set that maximizes the ratio of the obtained new revenue to the added cost approximates the optimal to within $1 - \ln r/(r - 1)$ where r is the ratio of revenue of the optimal solution to its cost; this result is similar in spirit to the greedy algorithm for the variable catalog segmentation problem due to [KPR98].

DYNAMIC ADVERTISEMENT PROBLEMS. Unlike static policies, we show that there is always an optimal dynamic policies that is deterministic. A natural follow-on question is whether the (deterministic) decision about pitching an ad at a state can be made using only limited history. Unfortunately, this is not true. Example A in Figure 1 once again gives insight here. Imagine an initial mix of $(\alpha, 1 - \alpha)$ between the targeted and non-targeted segments. As $\alpha \to 0$, the number of iterations to wait before pitching goes to ∞. Thus, there is no finite bound on the size of the history required by the optimal deterministic policy. This is also the difference between our model and finite-state Markov decision processes.

Next, we show that for Markovian user models, computing the optimal policy is equivalent to a point location problem in a k dimensional simplex with at most $k + 1$ regions of which at most 1 may be non-convex. The i-th region corresponds to the (*a posteriori*) mixture densities that would result in segment i being pitched at state v. The only non-convex region corresponds to the mixtures where we prefer not to pitch any segment. Unfortunately, the boundary of this region can be arbitrarily complicated. Example B of Figure 1 shows how this region can be non-convex. There are three segments. For appropriate choices of α, and cost μ, if the initial mixture is $(\alpha, 0, 1 - \alpha)$ then the optimum policy pitches no ad at state u. Likewise for the initial mixture $(\alpha, 1 - \alpha, 0)$. But in the convex combination of those mixtures $(\alpha, (1 - \alpha)/2, (1 - \alpha)/2))$, the optimum policy pitches to segment 1.

If we were to assume a constant rate of leakage at each state (i.e., at each state every user has a constant probability of exiting), most users exit the system fairly quickly (within, say, $O(\log n)$ steps with high probability). This is not unreasonable on the web, and allows us to enumerate all possible histories of length $O(\log n)$ and, by using a simple dynamic program, compute an approximation optimum dynamic policy. If the leakage rate, however, were relatively small (say polynomially small), this approach fails.

Nonetheless we show that for every fixed ϵ, there is a polynomially-bounded approximation of each convex region which delivers a revenue within ϵ (additively) of the optimal. Further, this can be computed efficiently if the leakage rate is at least $1/\text{poly}(n)$. This final result is derived by constructing an appropriate linear program and showing that the optimum solution to the linear program approximates the regions of interest. We

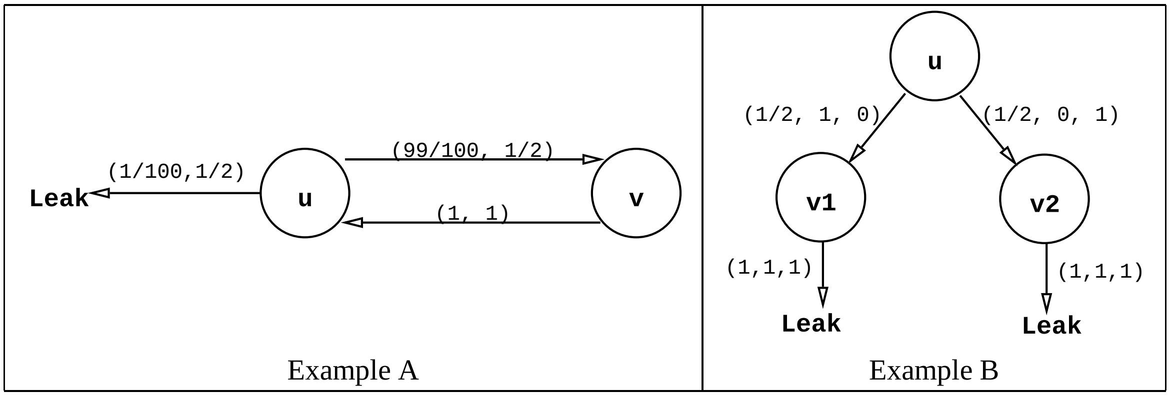

Figure 1: Example Markov chains: labels on the edges indicate the transition probabilities for the different Markov chains; a label $(p_1, \ldots, p_k)$ indicates a transition probability p_i for the i-th Markov chain.

conclude by mentioning a dynamic-programming approach to the budgeted DAP.

4 Static advertisement problems

We consider static policies in which we seek to obtain the optimal values for $\sigma_{i,v}$ that maximize revenue. Section 4.1 deals with the budgeted case where we obtain a $(1 - 1/e)$-approximation algorithm for this problem. Section 4.2 deals with the unbudgeted version of this problem.

4.1 Budgeted SAP: The SAND algorithm

For a budget B, consider frst the case of one targeted population, and so the policy matrix $\vec{\sigma}$ can be treated as a vector. This problem can be seen to be NP-hard by reduction from knapsack.

Let $B' = B/T$, where $T = n^2$. We now define the *sand algorithm*, SAND. At each of the T rounds, SAND increases the probability at some node such that the total cost of the policy increases by at most B'. This node is chosen greedily so as to maximize the improvement in revenue of the new policy. More formally:

$\vec{\sigma} \leftarrow \vec{0}$;
for T steps, find u, δ such that increasing σ_u by δ maximizes $rev(\vec{\sigma'}) - rev(\vec{\sigma})$, and $cost(\vec{\sigma'}) - cost(\vec{\sigma}) \leq B'$, where
$\vec{\sigma'} = \vec{\sigma} + \delta\vec{e}_u$;
set $\vec{\sigma} \leftarrow \vec{\sigma} + \delta\vec{e}_u$.

Here $\vec{e}_u$ is the unit vector that has a 1 in the coordinate corresponding to vertex u and 0s everywhere else.

Theorem 1 *For any budgeted SAP,*

$$rev(\text{SAND}) \geq (1 - \exp(-1 + o(1)))\, rev(\text{OPT}).$$

Proof. After t rounds of greedy iteration, SAND will have incurred overall cost $cost_t$, and generated some revenue R'_t. Let $R^* = rev(\text{OPT})$. By the greedy choice of the node in the $(t+1)$-st iteration of SAND, the cost of the node went up by $\epsilon_t (\leq B')$, thereby resulting in at most $(1 - \epsilon_t/B)$ fraction of the difference between R^* and R'_t remaining, thereby

$$R^* - R'_{t+1} \leq (R^* - R_t) \cdot \left(1 - \frac{\epsilon_t}{B}\right). \qquad (1)$$

Now, the increase in cost at the t-th step is strictly less than B' only if this step increases the probability of pitching at a particular vertex to 1. If the probability of pitching at all vertices is increased to 1, then clearly, we obtain the optimal profit. If this is not the case, then $cost_t > B - nB' = B(1 - o(1))$ (because at most n of the cost increases are less than B'). Thus, $\sum_{i=1}^{t} \epsilon_i = cost_t \geq B(1 - o(1))$. Using (1) recursively, we get

$$\begin{aligned} R^* - R'_T &\leq R^* \cdot \prod_{i=1}^{T} \left(1 - \frac{\epsilon_i}{B}\right) \\ &\leq R^* \cdot \prod_{i=1}^{T} \exp\left(\frac{-\epsilon_i}{B}\right) \\ &\leq R^* \cdot \exp(-1 + o(1)), \end{aligned}$$

yielding the theorem. □

The running time of SAND is $T \cdot T(n)$ where $T(n)$ is the time to make the greedy choice. Since the cost at

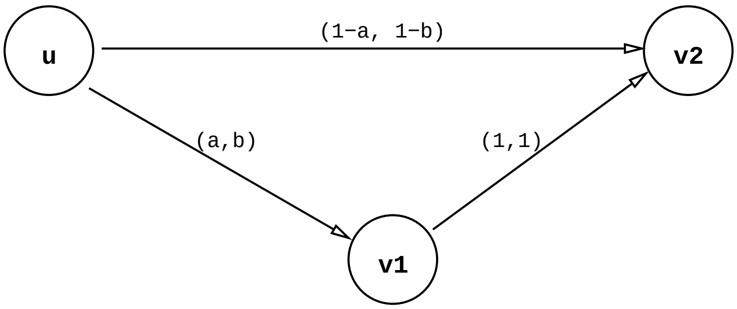

Figure 2: A Markov chain for which SAND is non-optimal.

a node is a convex and monotone increasing function of the probability of placing an ad at the node, a binary search will yield such a δ fairly quickly. The algorithm and analysis can be extended for $k > 1$, but to yield a guarantee that deteriorates as k increases.

To see that this algorithm need not find the optimal solution always, consider the example of Figure 2. If $a > b$, SAND will pitch segment 1 at state $v1$. However, if the cost is sufficiently low and the fixed budget sufficiently high, the optimal solution will pitch with probability 1 at $v2$, catching all segment 1 users. Any non-zero probability at $v1$ will increase the cost without changing the revenue.

4.2 Unbudgeted SAP: Prize-collecting set cover

We now consider the unbudgeted SAP. The exact complexity of this problem remains one of our major open problems. We consider a simpler version, a variation of the classical set cover problem, which we call the prize-collecting set cover problem. Let E be a set of elements $\{e_1, \ldots, e_n\}$, and $\mathcal{S}$ be a collection of subsets $\{S_1, \ldots, S_m\}$ of E. Every element $e \in E$ has an associated revenue $rev(e)$ and every set $S \in \mathcal{S}$ has an associated cost $cost(S)$. For a collection of sets $C \subseteq \mathcal{S}$ we define $rev(C) = \sum_{e \in \cup_{S \in C} S} rev(e)$ and $cost(C) = \sum_{S \in C} cost(S)$. We then define $prof(C) = rev(C) - cost(C)$. The objective is to pick a C so as to maximize $prof(C)$. The connection to the SAP should be clear: selecting a state at which to advertize is analogous to selecting a subset from S, with the interpretation that it "covers" the user trails that pass through it. Clearly this is also related to the unbudgeted flow-intercepting facility location problem.

We analyze the performance of a natural greedy algorithm for this problem. Let $C = \emptyset$. For the current collection C and all sets $S \in \mathcal{S}$, compute $\gamma(S, C) = (rev(C \cup S) - rev(C))/(cost(C \cup S) - cost(C))$. Let $S_{\max}$ be the set that maximizes $\gamma(S, C)$. If $\gamma(S_{\max}, C) \leq 1$, the algorithm stops and outputs C. Otherwise, the algorithm sets $C = C \bigcup \{S_{max}\}$ and repeats.

We now analyze the resulting profit. Let C^* be an optimal solution, and let $c^* = cost(C^*)$, $r^* = rev(C^*)$, and $p^* = r^*/c^*$. We will bound the approximation ratio of the greedy algorithm in terms of p^*.

Consider the collection C_t maintained by the greedy algorithm at stage t of its execution. Let $r_t = rev(C_t)$ and $c_t = cost(C_t)$. By adding all of C^* to C_t, we can increase revenue by at least $r^* - r_t$ while increasing cost by at most c^*. So there is an $S \in \mathcal{S}$ with $\gamma(S, C_t) \geq (r^* - r_t)/c^*$. So long as this is at least 1, the greedy algorithm will continue. We will analyze the revenue of the solution obtained by the greedy algorithm until it stops. Let ∂r be the change in revenue when the new set is added to the collection and ∂c be the change in cost. Then, the above analysis shows that: $\partial r \geq \partial c(r^* - r_t)/c^*$. Let us integrate this expression from the initial value $r_0 = 0, c_0 = 0$, to the final value r_f, c_f, when the algorithm stops. This gives us

$$\ln\left(\frac{r^*}{r^* - r_f}\right) \geq \frac{c_f}{c^*}, \text{ or } c_f \leq c^* \ln\left(\frac{r^*}{r^* - r_f}\right).$$

Since the greedy algorithm terminates now, $(r^* - r_f)/c^* = 1$. (The equality assumption is without loss of generality.) Using this, the value of the final solution is at least $r_f - c_f \geq c^*(p^* - 1 - \ln p^*)$. Taking $x = r^*/c^*$, we have:

Theorem 2 *The approximation ratio of the greedy algorithm is at least* $1 - \ln x/(x-1)$.

Unfortunately, this ratio goes to 0 as x goes to 1. The algorithm has an approximation ratio bounded away from zero if the ratio of the profit to the cost of the optimal solution is bounded away from one; this is similar in spirit to the variable catalog segmentation approximation in [KPR98].

Consider the natural linear programming relaxation for this problem. We can construct a family of instances such that integrality gap tends to $1 - \ln x/(x-1)$ where x here is the ratio of the profit of the optimal LP solution to the cost of the optimal LP solution.

The greedy algorithm and the analysis for the prize-collecting set cover in fact extend to the seemingly more general unbudgeted SAP. A greedy algorithm along the same lines as the algorithm SAND can be obtained for this problem and a modified form of the above analysis applies.

5 Dynamic advertisement problems

We now consider dynamic policies, which take into account the path x of a user from the start state to the current state. A dynamic policy is a collection of functions $\sigma_{i,v}(x)$ giving the probability of pitching segment i to a user in state v with history x. Clearly, $\forall x, v$, $\sum_i \sigma_{i,v}(x) \leq 1$ with $1 - \sum_i \sigma_{i,v}(x)$ being the probability of not pitching any ad to a user at state v with history x. We focus on the unbudgeted DAP; we will briefly talk about the budgeted problem at the end of this section.

We show the following: (1) The optimum dynamic policy is deterministic, i.e., $\sigma_{i,v}(x)$ is either 0 or 1. (2) The problem of computing $\sigma_{i,v}(x)$ reduces to a point-location problem in a k-dimensional simplex with $\leq k+1$ regions, of which at most one is non-convex. Unfortunately, the boundaries of the non-convex region can be arbitrarily complicated. (3) For every ϵ, there is a polynomially-bounded approximation of each convex region which delivers a revenue within ϵ (additively) of the optimal.

We begin in Section 5.1 by discussing the problem in the more general setting of arbitrary paths through an infinite tree, and then apply these results to Markov processes in Section 5.2 to provide a characterization of optimal solutions. In Section 5.3 we give approximation algorithms.

5.1 Dynamic policies on infinite trees

We begin by considering the infinite tree of all possible histories, and then apply the lessons learned here to the case of Markov chains. Let $\mathcal{T}$ be any (possibly infinite) tree rooted at r, with vertex set X. For each $x \in X$, let $D(x) \subseteq X$ be the children of v, and $\Pi(x) \subseteq X$ the ancestors of x. Note that, whereas x typically denotes the history of a user, in this section it denotes a node of the tree since this uniquely encodes the history.

We consider k segments $\{\mathcal{P}_i\}$, each running on the same (infinite) skeleton $\mathcal{T}$. Each process consists of a user who begins at the root and traces a random path $p = (r = x_0)x_1x_2\cdots x_t$ in the tree, $x_j \in D(x_{j-1})$. At each point x, process i terminates (i.e., the user vanishes) with probability $\rho_i(x)$.

Let $\mathcal{P}_i(x)$ denote the probability that a user from process i passes through x. Let $\vec{\alpha}(x)$ be the posteriori distribution of the processes at x, i.e., $\alpha_i(x) = \mathcal{P}_i(x)/(\sum_j \mathcal{P}_j(x))$. As always, $rev(x)$ denotes the revenue from correctly targeting at x, and $cost(x)$ the cost of advertising at x.

The dilemma is the following: if we wait too long before advertising, we run the risk of the user vanishing and thereby lose potential revenue. On the other hand, waiting longer reveals more information about which segment the user is from.

A *strategy* σ for $\mathcal{T}$ is a probability distribution on the set $\{0, \cdots, k\}$. Here $\sigma_0(x)$ is the probability of pitching nothing at x, and $\sigma_i(x)$ is the probability of pitching i at x (if i is an untargeted segment, then we will pitch nothing). A *deterministic strategy* is a strategy in which $\sigma(x) \in \{\vec{e_0}, \vec{e_i}, \ldots, \vec{e_k}\}$ where $\vec{e_i}$ is the is the i-th unit vector. That is, at any node, a deterministic strategy pitches nothing or always pitches to the same segment. We give the following lemma without proof.

Lemma 3 *The optimal strategy is deterministic.*

The following is a description of the optimal strategy which, for finite trees, trivially induces a dynamic programming algorithm that is polynomial in the size of $\mathcal{T}$ (and exponential in the number of segments.) Let $J \subseteq \{0, 1, \ldots, k\}$. Then $\vec{\sigma}$ is a J-strategy if for $i \notin J$, $\sigma_i(x) = 0$. The optimal strategy is the optimal $K = \{0, 1, 2, \ldots, k\}$-strategy. Since, the optimal strategy is deterministic, we know that $\vec{\sigma}(x)$ is one of the unit vectors or 0. Consider an arbitrary J. The various options at x are either to pitch a particular i at x or to pitch nothing; recall that the optimal strategy may pitch nothing at some state x even if, at x, we can be certain that the user is from one of the targeted segments (if, for instance, at the next level there is little probability of the user escaping, and we will know exactly *which* targeted segment the user belongs to). Let $\phi_J(x, \vec{\alpha})$ denote the optimal profit possible for a user at x with a J-strategy, if $\vec{\alpha}$ gives the posteriori distribution of the user's segment (which is non-constant since it depends on ads pitched at ancestors of x). Let $\vec{z_i}(\alpha)$ denote the probability distribution $\vec{\alpha}$ conditioned on the user not belonging in segment i. Thus, $\vec{z_i}(\vec{\alpha})_i = 0$, and $\vec{z_i}(\vec{\alpha})_{j \neq i} = \alpha_j/(1 - \alpha_i)$. Let $p(y, x, \alpha)$ denote the probability that the user moves to y from x. Let $\vec{a}(y, x, \alpha)$ denote the new posteriori distribution on segments given that the user moves to y. Suppose we pitch an ad for segment $i \in J \backslash \{0\}$, the maximum profit we can make is given by

$$\phi_{J,i}(x,\vec{\alpha}) = \alpha_i rev(i,x) - cost(i,x)$$
$$+(1-\alpha_i)\sum_{y\in D(x)} p(y,x,\vec{z}_i(\vec{\alpha}))\phi_{J\setminus\{i\}}(y,\vec{a}(y,x,\vec{z}_i(\vec{\alpha})))$$

On the other hand, if we do not pitch any ad, the maximum profit we can make is given by

$$\phi_{J,0}(x,\vec{\alpha}) = \sum_{y\in D(x)} p(y,x,\vec{\alpha})\phi_J(y,\vec{a}(y,x,\vec{\alpha})).$$

Then we have,

$$\phi_J(x,\vec{\alpha}) \;=\; \max_i\{\phi_{J,i}(x,\vec{\alpha})\}$$

with base case, $\phi_\emptyset(x) = 0$.

5.2 From trees to Markov processes

We restrict our attention again to Markov chains. Notice that, for any dynamic strategy, the decision to pitch only depends on the current (posteriori) distribution on the segments (as determined by the history). Let us again denote this mix by $\vec{\alpha}$. Then, for any state v, let $G_{i,v}$ denote the set $\{\vec{\alpha} \mid i$ is the optimal pitch at v if the incoming mix is $\vec{\alpha}\}$. Let $G_{\phi,v}$ be $\{\vec{\alpha} \mid \vec{\alpha} \notin G_{i,v}$ for any $i\}$.

The sets $G_{\cdot,v}$ are a partition of the probability simplex representing all the possible mixtures that could enter state v, into regions corresponding to the possible actions (viz, pitching any segment i, and declining to pitch). $G_{\phi,v}$ represents the "no pitch" option.

Lemma 4 *For $1 \le i \le k$, $G_{i,v}$ is convex.*

As a simple corollary, note that if there is a single targetable segment then the optimal policy at a fixed state v is to pitch whenever the posteriori probability that the user is targetable (as determined by the history) exceeds some fixed threshold.

Observation 5 *$G_{\phi,v}$ is not necessarily convex, as shown in Example B of Figure 1.*

5.3 Approximating the unbudgeted DAP

In this section we will use a linear programming formulation to obtain an approximation for the unbudgeted DAP. For ease of notation, we will use α instead of $\vec{alpha}$ to denote the probability distribution on segments.

Let $x^*_{v,\alpha}$ denote the optimal revenue that can be obtained from a user at a state v with mix α on the segments. Recall that $p(u,v,\alpha)$ is the probability that the user moves to u, $\vec{a}(u,v,\alpha)$ is the new posteriori distribution on segments given that the user moves to u and $\vec{z}_i(\alpha)$ is the probability distribution α conditioned on the user not belonging in segment i. Let $T_i(v,\alpha)$ denote the maximum possible revenue that can be obtained given that we pitch an ad for segment i at the user. Also let $T_0(v,\alpha)$ denote the maximum possible revenue that can be obtained given that we defer pitching an ad at the user. Then converting (1) from trees to Markov chains, we have:

$$\begin{aligned} T_i(v,\alpha) &= \alpha_i rev(i,v) - cost(i,v) \\ &\quad +(1-\alpha_i)\sum_u p(u,v,\vec{z}_i(\alpha))x^*_{u,\vec{a}(u,v,\vec{z}_i(\alpha))}, \\ &\quad \text{for } i>0 \text{ and} \\ T_0(v,\alpha) &= \sum_u p(u,v,\alpha)x^*_{u,\vec{a}(u,v,\alpha)}. \end{aligned}$$

We assume that at each step the user escapes with probability at least p_e. Recall that state n is taken to be the "escape" state, from which no further revenue is possible. Then for all $v, \alpha, p(n,v,\alpha) \ge p_e$.
Notice that $x^*_{v,\alpha} = \max_{i\ge 0}\{T_i(v,\alpha)\}$. The values $x^*_{v,\alpha}$ can be computed as the optimal solution to the following linear program (LP1):

$$\min : \sum_{v,\alpha} x_{v,\alpha}$$
$$x_{v,\alpha} \ge T_i(v,\alpha) \quad \forall v,\alpha,i$$
$$x_{v,\alpha} \ge 0 \quad \forall v,\alpha,$$

where $x_{v,\vec{\alpha}}$ denotes the profit starting at vertex v and initial state $\vec{\alpha}$.

Notice that LP1 has infinitely many variables (and constraints) as the parameter α varies over the probability simplex. We will show that for a suitable discretization of the probability simplex, we can come up with a linear program with polynomially many variables and constraints whose solution yields a strategy whose value is close to the optimal strategy.

Let us choose a $\delta > 0$ as the discretization parameter. For a probability distribution α over the segments, we define $\bar{\alpha}$, the discrete point corresponding to α, as follows: For $i > 0$, $\bar{\alpha}_i$ is the smallest multiple of δ less than α_i. Also $\bar{\alpha}_0 = 1 - \sum_{i=1}^k \bar{\alpha}_i$. Let P be the probability simplex consisting of all possible probability distributions over the segments. Let Γ denote the set $\{\bar{\alpha} \mid \alpha \in P\}$. We now write the following modified linear program (LP2):

$$\min : \sum_{v,\alpha \in \Gamma} y_{v,\alpha}$$
$$y_{v,\alpha} \geq T_i(v,\alpha) \quad \forall v,\alpha \in \Gamma, i$$
$$y_{v,\alpha} \geq 0 \quad \forall v,\alpha \in \Gamma,$$

where $y_{v,\alpha}$ is a discretized version of $x_{v,\alpha}$. As earlier, the terms $T_i(v,\alpha)$ can be written:

$$\begin{aligned} T_i(v,\alpha) &= \alpha_i rev_i(v) - cost_i(v) \\ &\quad +(1-\alpha_i)\sum_u p(u,v,\vec{z}_i(\alpha)) y_{u,\overline{\vec{a}(u,v,\vec{z}_i(\alpha))}}, \\ &\quad \text{for } i > 0 \text{ and} \\ T_0(v,\alpha) &= \sum_u p(u,v,\alpha) y_{u,\overline{\vec{a}(u,v,\alpha)}}. \end{aligned}$$

Let $y^*_{v,\alpha}$ denote the value of the variable $y_{v,\alpha}$ in the optimal solution to the above linear program. It can be shown that $x^*_{v,\alpha} - \epsilon \leq y^*_{v,\alpha} \leq x^*_{v,\alpha}$, for some ϵ whose value depends on the discretization parameter δ. The following technical lemmas are presented without proof:

Lemma 6 *For all $\alpha \in P$,*

$$x^*_{v,\bar{\alpha}} \geq x^*_{v,\alpha} - k\delta.$$

Notice that a basis for LP2 consists of one $y_{v,\alpha}$ constraint for each variable $y_{v,\alpha}$.

Claim 7 *Let $\mathcal{B}$ be a basis for LP2. Let $y^{\mathcal{B}}_{v,\alpha}$ be the value of variable $y_{v,\alpha}$ in the basis solution corresponding to $\mathcal{B}$. Then $y^{\mathcal{B}}_{v,\alpha} \leq y^*_{v,\alpha}$.*

Claim 8 *Let S be an assignment of values to the variables in LP2 such that $y_{v,\alpha}$ gets the value $y^S_{v,\alpha}$. If, for some basis $\mathcal{B}$, all the constraints in $\mathcal{B}$ are violated by the assignment S, then $y^S_{v,\alpha} \leq y^{\mathcal{B}}_{v,\alpha}$*

Proof. Let $z_{v,\alpha} = y_{v,\alpha} - y^S_{v,\alpha}$. Consider the system of linear equations given by $\mathcal{B}$. We will write the equations in terms of the variables $z_{v,\alpha}$ to obtain a new system of equations $\mathcal{B}'$. We then show that the values of $z_{v,\alpha}$ in the solution to $\mathcal{B}'$ are all non-negative. Consider any equation in the basis $\mathcal{B}$. It is of the form $y_{v,\alpha} = c_{v,\alpha} + \sum p_{u,\beta} y_{u,\beta}$ where $c_{v,\alpha} > 0$ and $\sum p_{u,\beta} \leq 1 - p_e$. Note that $y^S_{v,\alpha} \leq c_{v,\alpha} + \sum p_{u,\beta} y^S_{u,\beta}$. The corresponding equation in $\mathcal{B}'$ is $z_{v,\alpha} = \sum p_{u,\beta} z_{u,\beta} + c'_{v,\alpha}$, where $c'_{v,\alpha} = c_{v,\alpha} + \sum p_{u,\beta} y^S_{u,\beta} - y^S_{u,\beta} \geq 0$. Let $z^*_{v,\alpha}$ be the variable with the minimum value in the solution to $\mathcal{B}'$. Then $z^*_{v,\alpha} \leq z_{u,\beta}$ for all u, β. and $z^*_{v,\alpha} \geq 0$ since, $(1 - \sum p_{u,\beta}) z^*_{v,\alpha} \geq c'_{v,\alpha} \geq 0$. $\square$

Consider the optimal solution to LP1. For each $v, \alpha \in \Gamma$, there is some $x_{v,\alpha}$ constraint in LP1 which is tight. Consider the corresponding constraint in LP2, $y_{v,\alpha} \geq c_{v,\alpha} + \sum p_{u,\beta} y_{u,\bar{\beta}}$ The set of all such constraints in LP2 forms a basis. Let us call this $\mathcal{B}$.

Choose $\epsilon > \max\{(1-p_e)(k\delta+\epsilon), \frac{1}{p_e} - 1)k\delta\}$. Consider the assignment S that assigns the value $y^S_{v,\alpha} = x^*_{v,\alpha} - \epsilon$ to the variable $y_{v,\alpha}$. We show that for such a value of ϵ, the assignment S violates all the constraints in $\mathcal{B}$. First, note that

$$y^S_{u,\bar{\beta}} = x^*u, \bar{\beta} - \epsilon \geq x^*u, \beta - k\delta - \epsilon.$$

Consider the constraint

$$y_{v,\alpha} \geq c_{v,\alpha} + \sum p_{u,\beta} y_{u,\bar{\beta}}.$$

Now,

$$\begin{aligned} & c_{v,\alpha} + \sum p_{u,\beta} y^S_{u,\bar{\beta}} \\ \geq \;& c_{v,\alpha} + \sum p_{u,\beta}(x^*u, \beta - k\delta - \epsilon) \\ = \;& c_{v,\alpha} + \sum p_{u,\beta} x^*u, \beta - \sum p_{u,\beta}(k\delta - \epsilon) \\ = \;& x^*_{v,\alpha} - \epsilon + \epsilon - \sum p_{u,\beta}(k\delta - \epsilon) \\ = \;& y^S_{v,\alpha} + \epsilon - \sum p_{u,\beta}(k\delta - \epsilon) \\ \geq \;& y_{v,\alpha} + \epsilon - (1 - p_e)(k\delta + \epsilon) \\ > \;& y_{v,\alpha}. \end{aligned}$$

Hence this constraint is violated. This is true of every constraint in $\mathcal{B}$.

By Claim 8, it follows that $y^{\mathcal{B}}_{v,\alpha} \geq y^S_{v,\alpha} = x^*_{v,\alpha} - \epsilon$. Also, by Claim 7, it follows that $y^*_{v,\alpha} \geq y^{\mathcal{B}}_{v,\alpha} \geq x^*_{v,\alpha} - \epsilon$. Hence, $y^*_{v,\bar{\alpha}} \geq y^*_{v,\alpha} - (k\delta + \epsilon)$.

5.4 The budgeted DAP

Dynamic policies that need to work under budget constraints seem hard to characterize. Even in case that the underlying graph is acyclic, the problem generalizes the *precedence-constrained knapsack* problem. As observed by Chekuri [Che98], this can be solved using dynamic programming. We can apply this insight to the budgeted DAP; however, the size of the dynamic program is exponential in the number of segments, and grows with the magnitude of the budget.

6 Further work

Our work raises a number of directions for further work; we now summarize the salient open problems. (1) What is the complexity of solving the unbudgeted SAP? (2) Our algorithm for the unbudgeted DAP makes use of the "leakage" assumption; while this is defensible for practical purposes, can we do without it? Dispensing with this assumption may directly yield a combinatorial algorithm. (3) In the budgeted DAP, can we allocate ad budget between multiple segments without recourse to dynamic programming? (4) Can we integrate the inference of the Markov model and the determination of the ad policy? Here is an intuitive algorithmic framework for this: as we watch a user trail, we see whether the trail resulted in a transaction in a particular segment (say, purchased a car). If so, we increase the likelihood of pitching a car ad at all nodes through which that user trail passed.

References

[BH96] E. N. Berkowitz and S. Hartley. *Marketing.* Richard D. Irwin, Inc., 1996.

[BHK95] O. Berman, J. Hodgson, and D. Krass. Flow-interception problems, *Facility Location*, pp. 389–425, 1995.

[BKX95] O. Berman, D. Krass, and C. W. Xu. Locating discretionary service facilities based on probabilistic customer flows, *Transportation Science*, 29(3):276–290, 1995.

[BKX96] O. Berman, D. Krass, and C. W. Xu. Location flow-intercepting facilities: New approaches and results, *Annals of Operations Research*, pp. 121–143, 1996.

[BKX97] O. Berman, D. Krass, and C. W. Xu. Generalized flow-interception facility location models with probabilistic customer flows, *Stochastic Models*, 13:1–25, 1997, pp 1-25

[Che98] C. Chekuri. *Personal communication*, 1998.

[CN] CommerceNet/Nielsen. The CommerceNet/Nielsen Internet Demographics Survey. `www.commerce.net/research/stats/stats.html`

[Hod90] J. Hodgson. A flow-capturing location-allocation model. *Geographic Analysis*, 22(3), 1990.

[KPR98] J. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Segmentation problems, *Proc. 30th Symposium on Theory of Computing*, pp. 473–482, 1998.

[KRRT98] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis, *Proc. 39th Foundations of Computer Science*, pp. 664–673, 1998.

[P00] Project 2000. `www2000.ogsm.vanderbilt.edu`

[Whi92] D. J. White. *Markov Decision Processes*. John Wiley & Sons, 1992.