Samuel Dobesh

Maximum Flow Advertisements

1. Facebook has all sorts of personal information and thus, can entice advertisers by promising they can deliver ads targeted to specific demographic groups. Advertisers are drooling and ready to fork over some big bucks. Now Facebook is trying to figure out an efficient method to deliver targeted ads to as many users as possible.

   The first step in this process is to classify people into demographic groups. On Facebook these classifications might be based on things such as geographic location, age, interest in a type of music, college major. We'll assume this step is taken care of by the data mining group and they have identified k demographic groups $G_1, G_2, \ldots, G_k$. Note that these groups can overlap; G_1 could be people who live in Bellingham, G_2 could be people who like pizza, G_3 could be students majoring in computer science.

   The marketing department at Facebook is forging contracts with m different advertisers to show a certain number of ads to users of the site. Here's what a contract with the ith advertiser looks like:

   - For a subset of the demographic groups, $X_i \subseteq \{G_1, G_2, \ldots, G_k\}$, advertiser i will have its ads shown only to users who belong to at least one of the demographic groups in X_i
   - For a number r_i, the advertiser will have its ads shown to at least r_i users each minute

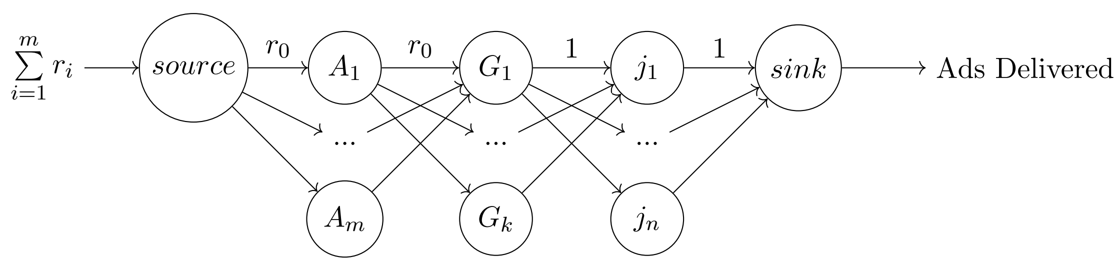
   Now consider the problem of showing ads for a single minute. Specifically, you want to create a way to show no more than one ad to each user at this minute to satisfy all of the advertising contracts. Suppose at a given minute there are n users on the site and we we know that user j belongs to a particular subset of the demographic groups.

   Show how to set up this problem as a maximum flow problem and explain how to use the result of max flow to determine if it's possible to show a single ad to each user at this minute so that the Facebook's advertising contracts with each of the m advertisers are satisfied.

Answer

To put this problem in terms of a max-flow problem, let us first determine the input to our source and the output of our sink. The max flow network will determine the routing of all advertisements, matching them up to possible candidates, each of them accepting at most a single ad. We will have the maximum possible ads flowing in, which would be the sum of all r_i. We can then say that the sink takes every ad that we were able to deliver.

Our first layer will represent the advertisers themselves, which we will label as A_i, from A_1 to A_m. Each advertiser, A_i, wants to send out r_i ads, so this will be the maximum inflow to each advertiser node. Each advertiser has their set of demographics they'd like to reach, but there is no guarunteee that the ads will visit all groups, they could go to just one. For that reason, we will hook up each A_i, to nodes representing the target groups, and each A_i is connected to every $G \in X_i$ by r_i. Each of these groups is then in turn connected to each of it's members, each represented by their own node. Each user will be indicated with subscrippted j, from j_1 up to j_n. Each user will have as many in going edges as groups they belong to, but only one outgoing edge, all with weights of one. This represents that although the user could recieve ads for any group, they can only recieve one at a time. We can than evaluate the graph with any well defined max-flow algorithm, and if the max flow through the system is less than the sum of r_i, then we can't deliver every ad. Otherwise, if they are equal, then we can fulfill every contract.



The runtime of this algorithm will be the cost of whichever max flow algorithm we select, plus the time of constructing the graph. To create our graph we'll need to set up edges from the source to the advertisers. Each advertiser will have one incoming edge, and $|X_i|$ outgoing edges to their target demographic groups. Each of these groups will then have as many outgoing edges as members, $|G_i|$. Finally each user node has one input and one output. If we add all that together we get $m + \sum_{i=1}^{m} |X_i| + \sum_{i=1}^{k} |G_i| + n$ edges. If we were to use a max flow algorithm like Edmonds-Karp, which is $O(|V||E|^2)$, we would incur the cost of all our edges again from making them, so our algorithm would run in $|E| + |V||E|^2$. The cost of constructing the edges is a lower order of growth, so the algorithms is all together still $O(|V||E|^2)$.