Topics in Game Theory
Spring 2009
Lecture 6: Pricing in Social Networks

Lecturer: Hamid Mahini
Scribe: Mudassir Shabbir

March 23, 2009

1 Overview

In this lecture we discuss optimal pricing for revenue maximization in presence of positive network externalities. Decision whether a user is going to buy a good or not, also depends upon set of other buyers who have already owned that good, along with the price of good. We consider this problem without price discrimination that is a seller iteratively anounces a price for the good to all buyers and any interested buyer can buy it at the anounced price.

2 Monetizing Social Network

There are many ways in much social networks (Facebook, Orkut, MySpace etc.) can influence the price and revenue of a good. Contextual display advertising uses information and Viral marketing uses postive influence of a set of buyers on a perspective buyer. For example if a friend of mine has bought a Sony laptop, it may positively affect my decision to buy a Sony laptop. There can many reasons for that. Few of them are:

1. Complatibility: If a buyer X frequently shares documents, photographs, and other multimedia with her friends, and a majority of her friends have MAC, she may be motivated to buy MAC due to software compatibility reasons.

2. Confidence: Before buying a good G, if a buyer X knows that many people have already using G, he may feel more confident about the quality of G because now X knows more about it.

3. Discount: For a buyer X, value of cell phone service that offers discount for calls among people using their service increases more friends and family buy the same service.

4. Emotional Effects: Buyer may be emotionally motivated to buy G if it has already being bought by some “special” friend.

3 Public Pricing and Private Pricing

Public Pricing (or Fixed Pricing) is defined as pricing scheme in which any anounced price is visible to all buyers and any interested buyer can buy the good at that price.

On the other hand, in Private Pricing seller can offer a private price for each buyer and good is not available to buyer at that price to any other buyer whose not be offered that price.

We assume an iterative posted price setting in which we post a public price at each step.

4 $\mathsf{Stable}(k)$ Problem

4.1 Definition

There are n buyers in the market and our goal is to find a sequence $p_1, \ldots, p_k$ of k prices in k consecutive time steps. A buyer decides to buy the item during a time step as soon as her valuation is more than or equal to the price offered in that time step. The buyers decision in a time step immediately affects the valuations of other buyers in the same time step. More precisely, a time step is assumed to end when no more buyers are willing to buy the item at the price at this time step. At this time, we move to the next step and offer a new price. Our goal is to find a sequence $p_1, \ldots, p_k$ of k prices that maximizes our expected revenue given the probability distributions $f_{i,S}$ for each buyer i and each subset $S \subset V$, where S is the set of buyers that already own the good.[AGH+09]

4.2 Trivial Solution

$\mathsf{Stable}(k)$ problem has a trivial solution assuming we have n time steps ($k = n$). Consider following algorithm:

4.2.1 Algorithm

1. `Set price` p_1 `to` ∞
2. `Decrease the price untill a person buys the item.`
3. `Wait until network becomes stable and go to step 3.`

4.3 Deterministic $\mathsf{Stable}(k)$

As defined earlier, in $\mathsf{Stable}(k)$, any buyers decision to buy a good affects other buyers valuations of good in the same time step. Below we show that order of the buyers during a time step has no effect on the state after the time step has ended. Let $B^1(S,p) := \{i|v_i(S) \geq p\} \cup S$ i.e.$B^1(S,p)$ specifies the set of buyers who immediately want to buy or already own the good, given a global price

p, and a set S of buyers who already own the good at the beginning. We can recursively define

$$B^k(S,p) = B^1(B^{k-1}(S,p),p) \tag{1}$$

and use the induction to reason that $B^k(S,p)$ will own the items at the end of this time step. Let

$B(S,p) = B^l(S,p)$

where

$l = \max k | B^k(S,p) - B^{k-1}(S,p) \neq \emptyset$

Since valuation function v_i of each buyer i is monotone nondecreasing, it can be aruged that the set $B(S,p)$ identifies the buyers wo own the good at the end of time step, and it does not depend upon the order of buyers in that time step.

4.3.1 Soving Deterministic $Stable(1)$

In $Stable(1)$, our goal is to find p_1 such that revenue $p_1.|B(\emptyset,p_1)|$ is maximized. Let $\beta_i = \sup\{p|i \in B(\emptyset,p)\}$ and $\beta = \{\beta_i | 1 \leq i \leq n\}$. We know that

Lemma 1 *for any a and b such that a ¡ b,* $B(\emptyset,b) \subseteq B(\emptyset,a)$

and

Lemma 2 *The optimal price* p_1 *is in set* β. **Proof.** If p is smaller than any value in β, by increasing p to the smallest price in β which is larger than p, we can achieve better revenue without losing any customers. Similarly If p is greater than β_1, we can decrease p to β_1 to achieve better revenue.

Here is an outline of the algorithm to find p_1. First find all the elements β_i of $beta$. Now consider the profit $\beta_i.B(\emptyset,\beta_i)$ of each of them to find the best result. There are $|\beta|$ steps in total and at step i, we set the price to the maximum valuations of remaining players, considering the set S as buyers who have already bought the good. At the end of step i, $B(\emptyset,\beta_i)$ is the set of buyers have own the good and β_{i+1} is the maximum valuation of the remaining buyers.

Algorithm for ith step: By induction we know that $S = B(\emptyset,\beta_{i-1})$ and $\beta_i < \beta_{i-1}$ is the maximum valuation of the remaining buyers. And from Lemma 1, $S \subseteq B(\emptyset,\beta)$, so if we set global price to β_i, then by the end of time step, set of owners will be $S = B(\emptyset,\beta)$ and maximum valuation of remaining players be β_{i+1}

4.3.2 Soving Deterministic $Stable(k)$

We try to solve $Stable(k)$ by running $Stable(1)$ consisting of m steps and by using dynamic algorithm. We are looking for strictly decreasing sequence of prices $p_1, p_2 \ldots, p_k$. Considering that and Lemma 1, we want to maximize $\sum_{i=1}^{k} |B(\emptyset,p_i) - B(\emptyset,p_{i-1})|.p_i$. Thus the problem $Stable(k)$ can be solved by considering the subproblem $A[j,m]$ where we are to chose an increasing sequence π of j prices from set $\{\beta_1, \beta_2 \ldots, \beta_m\}$ to maximize the profit and setting the price at the last time step to β_m. This subproblem can be solved using following dynamic program:

$$A[j,m] = \max_{1 \leq t < m} A[j-1,t] + |B(\emptyset,\beta_m) - B(\emptyset,\beta_t)|.\beta_m \tag{2}$$

5 $\mathsf{Daily}(k)$ Problem

5.1 Definition

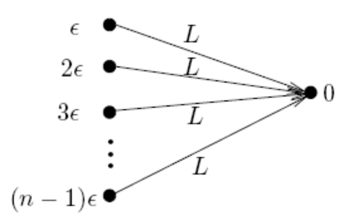
Consider a seller who wants to sell a digital good, and a set V of n buyers each with a valuation $vi(S)$ for each subset $S \subset V \backslash \{i\}$. We assume that $vi(S)$ is a random variable that comes from a probability distribution with an accumulative distribution function $F_{i,S}$. Given a number k, the $\mathsf{Daily}(k)$ problem is to design a pricing policy for k consecutive days time steps. In this problem, a pricing policy is to set a public price p_i at the start of time step i for each $1 \leq i \leq k$. At the start of each time step, after the public price p_i is announced, each buyer decides whether to buy the item or not, based on the price offered on that time step and her valuation. The decision of a buyer during a time step is not affected by the action of other buyers in the same time step. Our goal to find a pricing policy consisting of k prices that maximizes the expected revenue to the seller

5.2 Trivial Solution?

Lets assume that instead of probability distribution, we know the exact valuations $vi(S)$ of each buyer i, then it seems plausible to aruge that if we apply same $\mathsf{Stable}(k)$ algorithm, we do not need more than n time steps because we can set p_i such that at each step i at least one buyer buys the good, and one may think that $\mathsf{Daily}(k)$ problem can be easily solved. But unfortunately this algorithm does not lead to the maximum expected revenue. Consider the following example:

5.2.1 Example

There are n buyers number 1 to n. For buyer $1 \leq i \leq n$, initial valuation of good is $\epsilon.(i-1)$. A purchase by any buyer $i \neq 1$, increases buyer 1's valuation by L. The valuations of the rest of the buyers do not change. L is arbitrarily large while ϵ is arbitrarily small.



Using our algorithm on this problem, seller would sell the good to buyer 1 in second step for price L. But it is obvious to see that if she decides to wait till everyone else has bought the good, she will be able to sell the good to 1 for $(n-1).L$ hence improving overall revenue. Daily problem is not only hard, but also hard to approximate by some factor.

As compared to $\mathsf{Stable}(k)$, in $\mathsf{Daily}(k)$ buyers reacts slowly, and while news spreads through the network before seller can change the price in $\mathsf{Stable}(k)$, in $\mathsf{Daily}(k)$ seller can change the price before the news spreads through the network. But in both cases we have imaptient buyers i.e. if price of a good is less than or equal to buyers valuation then buyer will not wait for the price to change and buy the good at once.

References

[AGH+09] Hessameddin Akhlaghpour, Mohammad Ghodsi, Nima Haghpanah, Hamid Mahini, Vahab S. Mirrokni, and Afshin Nikzad. Optimal Iterative Pricing with Positive Network Externalities. 2009.