MS&E 310 Course Project I: Online Linear Programming

Yinyu Ye

February 8, 2015

In general resource allocation, we consider a linear program in the form

$$
\begin{array}{ll}
\text{maximize}_{\mathbf{x}} & \sum_{j=1}^{n} \pi_j x_j \\
\text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \ \forall\ i = 1, 2, ..., m \\
& 0 \le x_j \le q_j, \ \forall\ j = 1, ..., n,
\end{array} \tag{1}
$$

where b_i is the known positive quantity of resource i; see [3] and references therein.

The classical LP algorithm would compute the optimal solution $\mathbf{x}$ altogether, while the online algorithm would compute solution sequentially x_1, then x_2,.... Specifically, when compute the decision variables x_1 to x_k, there is no information associated with x_{k+1} and so on. In this course project, you are asked to study and explore some theories of online linear programming.

As an example, consider the prediction market (also called information market) discussed in the class. Assume that there are m states of futures of which exactly one will happen. Each bidder is allowed to bid on any subset of the states, with a price limit and a quantity limit. In what follows, we use $\mathbf{a}_j$ to denote the bid profile of the jth bidder, π_j to denote his price limit and q_j to denote his share quantity limit. The call auction model we studied in class is as follows:

$$
\begin{array}{ll}
\text{maximize}_{\mathbf{x},z} & \left(\sum_j \pi_j x_j\right) - z \\
\text{s.t.} & \sum_j a_{ij} x_j \le z, \ \forall\ i = 1, 2, ..., m \\
& 0 \le x_j \le q_j, \ \forall\ j = 1, ..., n;
\end{array} \tag{2}
$$

where z can be viewed as a variable resource. Few questions can be raised on the model:

Question 1: The dual prices may not be unique when solving the model. In order to get a unique price, people sometimes add another term to the objective function when solving the model; one model which is called CPCAM is defined as follows:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x},z,\mathbf{s}} & \sum_j \pi_j x_j - z + u(\mathbf{s}) & \\ \text{s.t.} & \sum_j a_{ij} x_j + s_i = y,\ \forall\ i = 1, 2, ..., m, & \\ & 0 \le x_j \le q_j,\ \forall\ j = 1, ..., n, & \\ & s_i \ge 0, \forall\ i = 1, ..., m. & \end{array} \tag{3}$$

where $u(\mathbf{s}) = u(s_1, \ldots, s_m)$ is increasing and strictly concave. Write down the first-order KKT conditions for optimality. Are they sufficient? Argue why this problem may likely have unique price (that is, the Lagrange multipliers on the m equality constraints of the problem). How would you interpret $u(\mathbf{s})$ and z? (Hint: Read [5] and [2].)

Question 2: The disadvantage of the call auction model is that it can't tell the bidders whether their bids are accepted or not until the market closes. This is undesirable since sometimes the bidders want to know the results to their bids immediately so that they can modify their bids and submit again. Therefore, in practice, the market is usually implemented in an online version which is defined as follows. Instead of solving the optimization after the market closes, whenever kth bidder submits a bid, the market maker solves the following optimization problem:

$$\begin{array}{lll} \text{maximize}_{x_k,y,\mathbf{s}} & \pi_k x_k - y + u(\mathbf{s}) & \\ \text{s.t.} & \sum_{j=1}^k a_{ik} x_k + s_i = y,\ \forall\ i = 1, 2, ..., m & \\ & 0 \le x_k \le q_k & \\ & s_i \ge 0,\ \forall\ i = 1, ..., m. & \end{array} \tag{4}$$

Note that, in (4), only scalar x_k, z and vector $\mathbf{s}$ are variables, and each quantity $x_j, j < k$ is already "locked down". Here again we assume that $u(.)$ is increasing and strictly concave. Write down the first-order KKT conditions of (4), and argue why this online problem may be solved efficiently comparing to the offline problem.

Question 3: We continue our discussion in Question 2. Assume that the kth bidder has a true valuation $\hat{\pi}_k$ per share for his state selection $\mathbf{a}_k$. We solve (4) for the optimal x_k^*. Then we charge the bidder by

$$\chi(0) - \chi(x_k^*)$$

where

$$\chi(x) = \begin{array}{ll} \text{maximize}_{y,\mathbf{s}} & -z + u(\mathbf{s}) \\ \text{s.t.} & s_i = z - q_i - a_{ik} x,\ \forall\ i = 1, 2, ..., m. \end{array} \tag{5}$$

where $q_i = \sum_{j=1}^{k-1} a_{ij} x_j$ is the outstanding shares of state i that has been already locked before the kth bidder arrives.

Using the incentive compatibility theorem, under this charging method, the optimal strategy for the kth bidder is to bid his or her true valuation $\hat{\pi}_k$ per share (assume that $\mathbf{a}_k$ is already determined and fixed)

Remark: A mechanism is truthful (or incentive compatible) if and only if for every agent, reporting his or her truthful belief is always the optimal strategy, disregarding what others do. In fact, such an optimal strategy is called dominant strategy in game theory. In fact, the above mechanism is a special case of the VCG mechanism; read references [2] and [7].

Question 4: Run the market with online model (4) with utility functions:

$$u_1(\mathbf{s}) = \frac{w}{m} \sum_i \log s_i$$

and

$$u_2(\mathbf{s}) = \frac{w}{m} \sum_i (1 - e^{-s_i})$$

separately for a positive weight parameter, say $w = 1$ and $w = 10$ respectively. Does the choice w make a difference? Is there significant difference between using $u_1(.)$ and $u_2(.)$?

You may run the online and offiline auctions using simulated bidding data. Fix a grand truth price vector $\bar{\mathbf{p}} > 0$ with $\sum_i \bar{p}_i = 1$. One way to generate a sequence of random bids, $k = 1, 2, ...$, is as follows: generate a vector $\mathbf{a}_k$ whose each entry is either zero or one at random, generate random scalar $q_k \in [2, 10]$, then let $\pi_k = \bar{\mathbf{p}}^T \mathbf{a}_k + randn(0, 0.2)$ where $randn(0, 0.2)$ represents the Gauss random variable with zero mean and variance 0.2 in Matlab. Does the state price vector $\mathbf{p}$ generated from the online auction model approaches the grand truth vector $\bar{\mathbf{p}} > 0$? Explain your observations and findings.

Question 5: How these ideas could be applied to the fix-resource model (1). Consider the case $q_j = 1$ for all j. Then consider the case (i) a_{ij} is between 0 and 1; (ii) a_{ij} is between -1 and 1.

References

[1] Shipra Agrawal and Nikhil R. Devanur. Fast Algorithms for Online Stochastic Convex Programming http://arxiv.org/abs/1410.7596, SODA2015

[2] S. Agrawal, E. Delage, M. Peters, Z. Wang and Y. Ye. A Unified Framework for Dynamic Prediction Market Design. Operations Research, 59:3 (2011) 550-568.

[3] S. Agrawal, Z. Wang and Y. Ye. A Dynamic Near-Optimal Algorithm for Online Linear Programming. http://arxiv.org/abs/0911.2974; to appear in Operations Research.

[4] N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani. Algorithmic Game Theory Cambridge University Press, New York, NY, USA.

[5] M. Peters. CONVEX MECHANISMS FOR PARI-MUTUEL MARKETS. Ph.D. Thesis, Stanford, November 2008.

[6] M. Peters, A. M-C. So and Y. Ye. Pari-mutuel Markets: Mechanisms and Performance. *The 3rd International Workshop On Internet And Network Economics*, 2007.

[7] $http://en.wikipedia.org/wiki/Vickrey\_auction.$