MS&E 310 Course Project I: Online Linear Programming

Yinyu Ye

December 8, 2017

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

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

where π_j is the gain to allocate a combination of goods/resources to bidder j, a_{ij} is the required quantity of good/resource i for bidder j, and b_i is the total known available quantity of good/resource i; see [3] and references therein. For simplicity, in this project we assume that a_{ij} is either 0 or 1.

The classical offline 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 and perform computational observations on some simulated data.

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:

$$\begin{array}{ll} \text{maximize}_{\mathbf{x},\mathbf{s}} & \sum_j \pi_j x_j + u(\mathbf{s}) \\ \text{s.t.} & \sum_j a_{ij} x_j + s_i = b_i, \ \forall\, i = 1, 2, ..., m, \\ & 0 \leq x_j \leq 1, \ \forall\, j = 1, ..., n, \\ & s_i \geq 0, \forall\, i = 1, ..., m. \end{array} \tag{2}$$

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 will have unique price (that is, the Lagrange multipliers on the m equality constraints of the problem). You may assume that $\frac{\partial u(.)}{\partial s_i}\Big|_{s_i=0} = \infty$ for all i. How would you interpret $u(\mathbf{s})$ and $\mathbf{s}$? (Hint: Read [6] 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:

$$\begin{array}{lll} \text{maximize}_{x_k,\mathbf{s}} & \pi_k x_k + u(\mathbf{s}) & \\ \text{s.t.} & a_{ik}x_k + s_i = b_i - q_i^{k-1}, \ \forall\ i = 1, 2, ..., m & \\ & 0 \leq x_k \leq 1 & \\ & s_i \geq 0, \ \forall\ i = 1, ..., m. & \end{array} \tag{3}$$

where $q_i^{k-1} = \sum_{j=1}^{k-1} a_{ij}\bar{x}_j$ is the good/resources i that is already allocated before the kth bidder arrives.

Note that, in (3), only scalar x_k and vector $\mathbf{s}$ are variables, and each quantity $\bar{x}_j, j < k$ is already "locked down". Here again we assume that $u(.)$ is increasing and strictly concave. Write down the first-order KKT conditions of (3), and argue why this online problem may be solved efficiently comparing to the offline problem.

Question 3: Run the market with online model (3) 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 with $m = 10$ and $b_i = 1,000$ for all i. Fix a grand truth price vector $\bar{\mathbf{p}} > 0$, 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, 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 4: There is another online algorithm described in [3]. Suppose there is a good estimate of total n bidders in the market. Then we wait for the first k bidders arrived and solve the revealed partial linear program

$$\begin{array}{lll}\text{(SLPM):} & \text{maximize}_{x_1,...,x_k} & \sum_{j=1}^{k} \pi_i x_i \\ & \text{s.t.} & \sum_{j=1}^{k} a_{ik} x_k \le \frac{k}{n} b_i, \ \forall\ i = 1, 2, ..., m, \\ & & 0 \le x_j \le 1, \ \forall\ j = 1, ..., k.\end{array}$$

and then use the *dual prices*, say $\bar{\mathbf{y}}^k$ of the partial LP for future online decisions:

$$x_j = 1, \text{ if } \pi_j > \mathbf{a}_j^T \bar{\mathbf{y}}^k; \quad \text{and } x_j = 0, \text{ otherwise.}$$

Let $n = 10,000$. Then run the online (SLPM) algorithm using the same simulated bidding data based on three different sizes of $k = 50, 100, 200$ to see how sensitive the k is.

Question 5: Now let us dynamically update the dual prices at time point $k = 50, 100, 200, 400, 800...$ and use the prices to make decision at the subsequent period. How does the dynamic learning perform? How do SCPM and SLPM compare with? You may use the same data that you generated in answering Question 3. Any idea to combine SCPM and SLPM? (You may read more references [1, 4, 5] in this research line.)

Question 6: One may consider more general resource allocation problems with production costs:

$$\begin{array}{ll}\text{maximize}_{\mathbf{x}} & \sum_{j=1}^{n} (\pi_j x_j - \sum_{k,i} c_{ijk} y_{ijk}) \\ \text{s.t.} & \sum_k y_{ijk} = a_{ij} x_j; \ \forall i, j, \\ & \sum_{i,j} y_{ijk} \le c_k; \ \forall k, \\ & 0 \le x_j \le 1, \ \forall\ j = 1, ..., n; \ y_{ijk} \ge 0\end{array} \tag{4}$$

where c_{ijk} is the cost allocate good/resource i, which is produced by producer $k = 1, ..., K$, to bidder j; and c_k is the production capacity of producer k.

Write out the dual of the resource allocation problems with production costs, and develop an online algorithm to allocate producers to bidders with needed good/resource combinations. Also run the algorithm on simulated bidding data and production costs, and possible theoretical analyses assuming bidders come randomly.

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] Anupam Gupta and Marco Molinaro How the Experts Algorithm Can Help Solve LPs Online. https://arxiv.org/abs/1407.5298, 2014.

[5] Thomas Kesselheim, Klaus Radke, Andreas Tnnis, Berthold Vcking. Primal Beats Dual on Online Packing LPs in the Random-Order Model. https://arxiv.org/abs/1311.2578, STOC 2014.

[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.