CME307/MSE&311 Project: Resource Allocation

Ananthakrishnan Ganesan (SUID#:06031325), Jayanth Ramesh (SUID#:06041612)
Gregory DePaul (SUID#:06154532), Stephanie Sanchez (SUID#:06108340), Vishal Subbiah (SUID#:06154113)

March 20, 2017

1 Initial Approaches to SCPM and SLPM

We consider the models for an Online Combinatorial Auction (without consideration of resource allocation). Specically the models SCPM and SLPM. But in order to evaluate our models, we need to generate large random data sets. To do so, we perform Algorithm 1.

Algorithm 1 Generate Random Data

---

initialize a = random number between (1,2) of size k x m
initialize p = random number between (0,1) of size m x 1
initialize q = random number between (10,20) of size k x 1
initialize pi = $a^T p$ + random number between (0, 0.2) of size k x 1

Which results in the quantities:

$$\begin{aligned} A &:= \text{combination betting matrix} \\ \bar{p} &:= \text{grand truth vector} \\ \pi &:= \text{biding share prices} \\ q &:= \text{max shares} \end{aligned}$$

These values can then be used to optimize either of our two models. Initially, consider the SCPM Approach in Algorithm 2.

Algorithm 2 SCPM

$u_1(s) = \max\{pi^T x - z + w\Sigma(\log s_i/m)\}$
$u_2(s) = \max\{pi^T x - z + w\Sigma(1 - e^{-s_i}/m)\}$
Perform Offline Assumption(n)
Let $x_{\text{fixed}} := x[0:n]$
Let $F := [x[0:n] - x_{\text{fixed}} == 0, a^T x - z + s == 0, x <= q, x >= 0, s >= 0]$
Solve the linear programs (u_1, F) and (u_2, F)

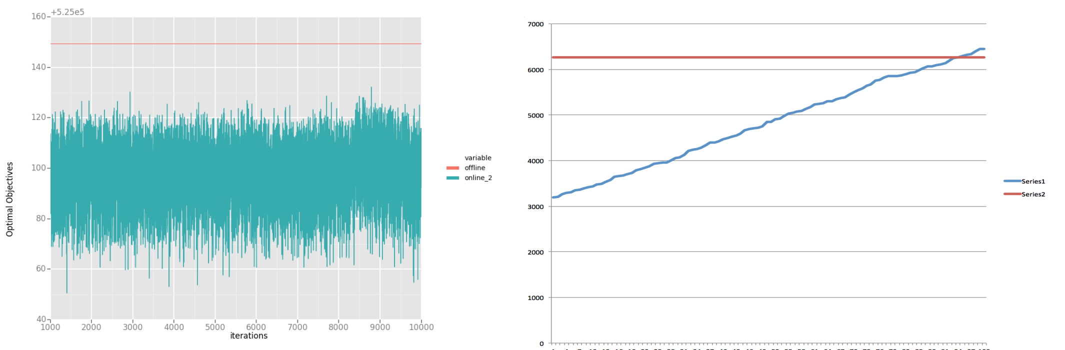
Of course we define our offline assumption for a partition index n to be the linear program:
Offline Assumption:

$$\begin{aligned} \max_{x_1,\ldots,x_n} \quad & \pi^T x - z \\ \text{subject to} \quad & \forall \quad 1 \le i \le m, \sum_{j=1}^{n} a_{ij} x_j - z \le 0 \\ & 0 \le x_j \le q_j, \forall 1 \le j \le n \end{aligned}$$

Clearly, $\forall$ j > n, $x_j = 0$.
Now consider the different approach of using SLPM to solve the online prediction market, summarized in Algorithm 3.
Upon running our algorithms, we plot the current optimal online solution at each iteration past the offline assumption. This is shown in Figure 1.

Algorithm 3 SLPM

Perform Offline Assumption(n)
$y \leftarrow$ *Dual of Offline Assumption*(n)
for $1 \leq j \leq k$ **do**
  if $\pi_j > A_j^T y$ **then**
    $x_j = q_j$
  else
    $x_j = 0$
  end if
end for



(a) SCPM (b) SCLM

Figure 1: Performance Comparison between SCPM and SCLM Models

Notice that the SCLM model produces a final optimal value that is incredibly close to the offline result. Also, when we measure the diff-norm between the shadow prices and our grand truth vector (Figure 2), we do see a general decline. But this metric isn't necessarily sufficient in showing whether or not the two converge.

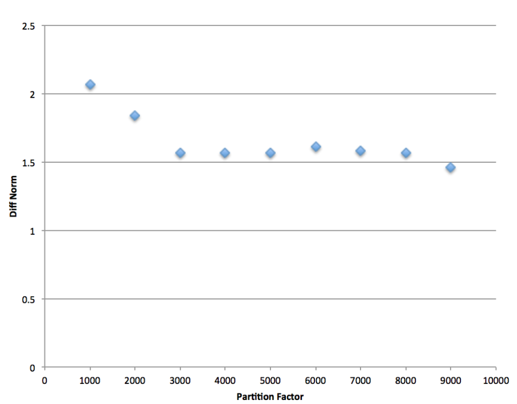


Figure 2: Diff Norm of State and Grand Price Vectors versus K (Initial Knowledge)

2 Theoretical Questions

We now consider a different model, specifically to accomodate resource allocation:

<u>CPCAM</u>

$$\begin{aligned}
\max_{x,s} \quad & \sum_j \pi_j x_j + u(s) \\
\text{subject to} \quad & \sum_j a_{ij} x_j + s_j = b_i, \forall i = 1, 2, ...m, \\
& 0 \leq x_j \leq 1, \forall j = 1, ...n, \\
& s_i \geq 0, \forall i = 1, ...m.
\end{aligned}$$

where b_i is the fixed resources (a limit on shares sold).

1)

Observe, The Lagrangian can be written as

$$L(x, s) = \pi^T x + u(s) - y^T (Ax + s - b) + r^T (x - 1)$$

and the KKT conditions can be written as

$$\begin{aligned}
\pi_j - \sum_{i=1}^m y_i a_{ij} + r_j \leq 0 \text{ for } 1 \leq j \leq n \\
x_j(\pi_j - \sum_{i=1}^m y_i a_{ij} + r_j) = 0 \text{ for } 1 \leq j \leq n \\
\sum_{i=1}^m y_i = 1 \\
\frac{\partial u}{\partial s_i} - y_i \geq 0 \text{ for } 1 \leq i \leq m \\
s_i(\frac{\partial u}{\partial s_i} - y_i) = 0 \text{ for } 1 \leq i \leq m \\
r_j(x_j - 1) = 0 \text{ for } 1 \leq j \leq n \\
r \leq 0
\end{aligned}$$

Claim: The KKT Conditions for Optimality are Sufficient and the dual multiplier values are unique.

Proof. Again following the CPCAM can be re-written as

$$\begin{aligned}
\max_{\bar{x}} f(\bar{x}) = c^T \bar{x} + \bar{u}(\bar{x}) \\
\text{such that } \bar{A}\bar{x} = \bar{b} \\
\text{and } \bar{x} \geq 0
\end{aligned}$$

where

$$\bar{x} = \begin{bmatrix} s \\ x \\ u \end{bmatrix} c = \begin{bmatrix} 0 \\ \pi \\ 0 \end{bmatrix} \bar{A} = \begin{bmatrix} I & A & 0 \\ 0 & I & I \end{bmatrix} \bar{b} = \begin{bmatrix} b \\ e \end{bmatrix}$$

and

$$\bar{u}(\bar{x}) = u(\bar{x}[(m + 1) : (m + n)])$$

The re-written problem has a strictly concave objective and a convex feasible region. Suppose $\exists \bar{x}_1 \neq \bar{x}_2$ such that they are both optimal. Then it means a strictly concave function has the same value in two distinct points in a convex region, which implies $\exists \bar{x}_3$ such that $\bar{A}\bar{x}_3 = \bar{b}$ and $f(\bar{x}_3) > f(\bar{x}_2) = f(\bar{x}_1)$, a contradiction.

Therefore, the maximiser $\bar{x}$ of f is unique, and hence the optimal values of x and s, x^* and s^* are unique. Therefore y^* is also unique.
The uniqueness also implies that the KKT conditions are sufficient.

□

In the case our SCPM adjusted model, $u(s)$ can be thought of as a weighted penalty function (disutility function) for the market organizer.
2)
Now, what if we consider a slight varient of the CPCAM model, where we drop the summation constraint for:

$$a_{ik}x_k + s_i = b_i - q_i^{k-1}, \forall 1, 2, ...m$$

The Lagrangian can be written as

$$L(x_k, s) = \pi_k x_k + u(s) - \sum_{i=1}^{m} y_i(a_{ik}x_k + s_i - b_i + q_i^{k-1}) + r_k(x_k - 1)$$

and the KKT conditions can be written as

$$\begin{aligned}
\pi_k - \sum_{i=1}^{m} y_i a_{ik} + r_k &\leq 0 \\
x_k(\pi_k - \sum_{i=1}^{m} y_i a_{ik} + r_k) &= 0 \\
\sum_{i=1}^{m} y_i &= 1 \\
\frac{\partial u}{\partial s_i} - y_i &\geq 0 \text{ for } 1 \leq i \leq m \\
s_i(\frac{\partial u}{\partial s_i} - y_i) &= 0 \text{ for } 1 \leq i \leq m \\
r_k(x_k - 1) &= 0 \\
r_k &\leq 0
\end{aligned}$$

Claim: This model can be solved within some constant proportion of the Offline Problem

Proof. When $k = 1$, solving the online problem is n times faster than the offline problem since there is only x_1 to be optimized. Since each time a new x_k value comes, s values can be updated efficiently and only the new x values need to be optimized, this is more efficient than the offline version.

□

3 Different Programming Model

We now want to programmatically solve the resource allocation problem.
3)
We consider using the variant CPCAM model with SCPM. This time, we set the bidding data to be a (10 x k) vector such that all entries $b_i = 1000$. Our results are then summarized in Figure 3. Notices that the values are above the offline version, which indicates this algorithm is slightly riskier than what we may necessarily want.

4)

We return to SLPM with the intention of applying it to resource allocation. In order to accommodate the new model, we must impose a new constraint, specifically:

$$\forall 1 \leq i \leq m, \sum_{j=1}^{k} a_{ik}x_k \leq \frac{n}{k}b_i.$$

The results still appear to be almost identical to that of Figure 1b. But we also take the liberty to notice the effect a partitioning index has on this data set. Specifically, as we vary the partitioning factor, it has relatively no change on the quotient of the optimal online value by the optimal offline value. This is indicated in Figure 4, where we see the slope of the regression is for all intents and purposes 0.

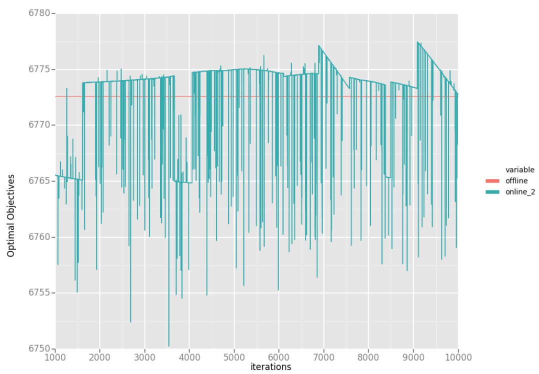


Figure 3: Adjusted SCPM for CPCAM

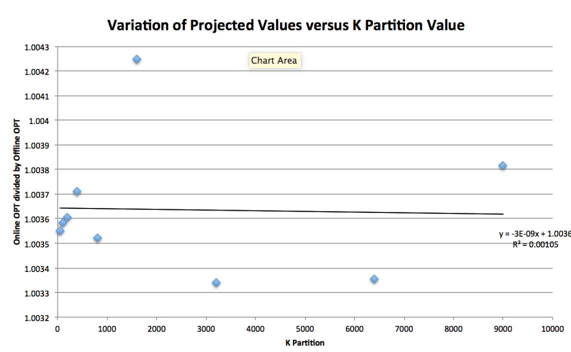


Figure 4: Variability of Partitioning Indices

4 Conclusions and Further Thoughts

5)

Lastly, we consider one final approach to solving the Resource Allocation problem. Specifically, we no return the the SCLM algorithm, but with the intention of logarithmically recalculating the dual prices in order to be receptive to new pricing information over time. Consider Algorithm 4.

Algorithm 4 Dynamically Dual Updating SLPM

Perform Offline Assumption(n)
for $n < K$ **do**
  $y \leftarrow$ *Dual of Offline Assumption*(n)
  for $n \leq j < 2n$ **do**
    if $\pi_j > A_j^T y$ **then**
      $x_j = q_j$
    else
      $x_j = 0$
    end if
  end for
  $n \leftarrow 2n$
end for

Performing this algorithm, we have reduced the risk of our SCLM method, as evident in Figure 5.

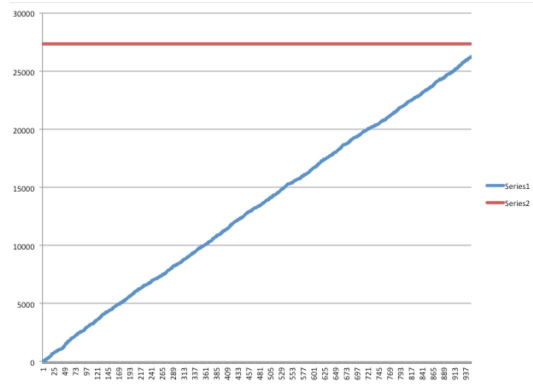


Figure 5: Dynamic Dual Updating SCLM

So now we are left wondering about the performance of SPCM an SPLM. They certainly achieve relatively accurate results, particularly the dynamic SCLM algorithm. The downside to this dynamic model is that it requires many iterations before it yields any useful information. On the other hand, the SCPM lacks the stability of the SCLM algorithm.
Thought: What if you combined the two methods? That is, we if you perform Heuristic Switching over time.

Algorithm 5 Heuristic Switching

for $n \leq i < K$ **do**
$OPTVAL = \max\{SCLM, SCPM\}$
end for

Of course this would be very computationally intense, and you can further refine the heuristic to favor the SCPM upon earlier iterations, while selecting the SCLM over later iterations.

References

[1] Peters, M., So, A.M-C., and Ye, Y., 2006. "A convex parimutuel formulation for contingent claim markets." ACM Conference on Electronic Commerce. ACM, New York, NY, USA