Data-driven Queueing Systems (QS) Capacity Mining

Yawo Kobara 1  Opher Baron 2  Dmitry Krass 2

1Odette School of Management, University of Windsor
2Rotman School of Management, University of Toronto

Introduction

Data-driven capacity estimation is one of the most significant issues in practical queueing theory.

• **Motivation and Background:**
  - Queueing systems (QSs), especially healthcare data such as patient flow data is often incomplete in the record.
  - Healthcare system processes are either unobservable or difficult to record.
  - Determining the right number of servers in a system is impossible with queueing models due to the lack of all the model parameters.

• **Problem Description:** Suppose we have multi-server queueing system that outputs a dataset \((A_k, D_k), k = 1, \ldots, K\), where \(A_k\) and \(D_k\) are the arrival and departure times of the \(k\)th customer.

• **Question:** Determine the effective capacity of a QS using only the arrival and departure timestamps of customers, what is the number of servers (service time distribution) of the queueing system?

![Figure 1. Problem setting visualization](image)

Methodology

1. **Krivulin based estimator** The effective number of servers in the system is estimated as the smallest number that keeps all Krivulin estimated service times positive. Mathematically, it is given as?

   \[
   \hat{c}_1 = \min\{c : S_k = D_k - E_k(c) > 0, \text{for } k = 1, \ldots, K, c = 1, 2, \ldots\} \tag{1}
   \]

   Where \(E_k(c) = \max(A_k, D_k)\) is the Krivulin (1994) estimate of the entering service time of the \(k\)th customer, supposing there are \(c\) servers.

2. **Overtake based estimator** Recursively count people who arrived before a customer but departed after her and select the maximum.

   \[
   \hat{c}_2 = \max\{C^+\} + 1 \tag{2}
   \]

   where \(C^+\) is the number of customers an arriving customer overtakes. \(C^+ \in \{0, 1, \ldots, c - 1\}\).

3. **Krivulin based Zero variance deterministic service algorithm**
   - Variance of the estimated service times of all customers is evaluated for all possible numbers of servers, \(c\).
   - Service time is deterministic, for the true number of servers.
   - Estimate service time for all patients varying the number of servers.
   - The number of servers, \(c\) that return a 0 (minimal) variance is the true number of servers.

4. **Linear Model Based Algorithm**
   - First compute the sojourn time, \(S_k\), of each customer \(k\) as \(S_k = D_k - A_k\).
   - Then compute the number of patients in the system when customer \(k\) arrived as \(N_k = \text{count all customers } j, \text{with arrival time } A_j < A_k\) and their departure time \(D_j > A_k\).
   - Assume there are \(c\) servers in the system.
   - Select only observations with \(N_k \leq c^* + 1\).
   - Run a simple linear regression between \(A_k\) and \(N_k\) \((\hat{a}_0 + \beta N_k)\).
   - If \(p\)-value > 0.05, \(\beta = 0\) then \(c^* < c\).
   - Now select observations with number in the system \(N_k \leq c^* = c^* + 1 \text{ and repeat } 1 - 3.\)
   - Continue to increase until you obtain a significant \(\beta\). Then the true number of servers is the first \(c^* = 1\) for the first \(c^*\) value that gives a significant \(\beta\).

Results for the first two methods

- **Theorem 1** if we observe that people depart from the system in the same order they entered, then, we can’t estimate the capacity.
- **Theorem 2** if we observe that at least one person overtook someone, then, we can estimate the capacity.

![Figure 2. Sample Results](image)

Table 1. \(\chi^2\) values of the \(\chi^2\) tests comparing the observed number of overtake in the NYGH data to the simulated number of overtake using the capacity estimated.

<table>
<thead>
<tr>
<th>Data Size</th>
<th>KW Est</th>
<th>C2 Est</th>
<th>KW Sim</th>
<th>C2 Sim</th>
<th>P* Sim</th>
<th>P* Sim</th>
<th>C-value (0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>3</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
<td>2</td>
<td>21</td>
<td>6</td>
<td>45</td>
<td>45</td>
<td>96</td>
</tr>
<tr>
<td>125</td>
<td>4</td>
<td>4</td>
<td>180</td>
<td>234</td>
<td>171</td>
<td>243</td>
<td>152</td>
</tr>
<tr>
<td>250</td>
<td>4</td>
<td>4</td>
<td>427</td>
<td>393</td>
<td>459</td>
<td>382</td>
<td>288</td>
</tr>
<tr>
<td>550</td>
<td>4</td>
<td>4</td>
<td>1028</td>
<td>1188</td>
<td>909</td>
<td>1135</td>
<td>606</td>
</tr>
<tr>
<td>1000</td>
<td>4</td>
<td>4</td>
<td>1874</td>
<td>2118</td>
<td>2046</td>
<td>1948</td>
<td>1075</td>
</tr>
<tr>
<td>2000</td>
<td>7</td>
<td>7</td>
<td>6546</td>
<td>8190</td>
<td>6687</td>
<td>8292</td>
<td>2105</td>
</tr>
<tr>
<td>5000</td>
<td>8</td>
<td>8</td>
<td>16824</td>
<td>23826</td>
<td>17213</td>
<td>23175</td>
<td>18014</td>
</tr>
</tbody>
</table>

Conclusions

- Determined sufficient conditions to mine the capacity of a QS under incomplete information.
  - Randomness
  - Continuity of service time distribution

- Two algorithms to estimate the effective number of servers were developed and they worked well.
- Only a little bit of data is needed with a small time for the algorithm to perform. The small data size required for convergence shows that the algorithms are cost-effective and fast.
- Other application includes
  - Competitive advantage: you can only observe your competitor from the inside and not from the outside and estimate their capacity to make your decisions in a business game.
  - Compliance auditing: you can audit the efficiency, effectiveness and compliance of your servers to see if they work as expected.

References