QURAMI challenge Estimating the waiting time

Aleksandra Stojanova, Biljana Zlatanovska, Dusan Bikov,
Mirjana Kocaleva, Zlatko Varbanov

September 2, 2016

Part III: Multi-service framework

Multi-service framework

- QURAMI app provide virtual ticket to users to improve waiting time in office queue.
- Until now app provides the user with number of people in the line before his turn, and sends real time updates on the line progress.

- Our goal was to provide estimation of the user's expected time in the waiting line in order to improve on time service.

Multi-service framework, approach

- The proposed problem was finding a solution in a case when app is offering only one service in a queue.
- Our approach is more like extended version of the problem and we offer more generalized solution.
- We estimated waiting time in a case when is possible to have more than one service in the queue, or when the app offers opportunity to chose a type of service.

Multi-service framework, notation

- m is the number of services;
- T (t_1, t_2, …, t_m) is an information vector for mean service time (w.l.o.g. $t_1 \le t_2 \le \ldots\ldots \le t_m$);
- $V = (v_1, v_2, \ldots, v_m)$ is an information vector for variance service time;
- $R(V) = (r_1(v_1), r_2(v_2), \ldots, r_m(v_m))$ is a vector of (already calculated) range values for all services;
- n is the current length of line and $P = (p_1, p_2, \ldots., p_n)$ are the mean service times for the people waiting now in the line (practically, any p_i is an element of T).

Multi-service framework, inputs and outputs

- *INPUT:*
  - Times $t_1, t_2, \ldots, t_m$;
  - Values r_1 (v_1), r_2 (v_2), …, r_m (v_m) ;
  - Service number *j*.
- *OUTPUT:*
  - A ticket (with a denoted range)

Algorithm in multi-service framework

Step 1: Start with n – the number of people who are waiting to be served;

Step 2: If $n = 0$ go to the last step (with a range $[0, r_j]$);

Step 3: If $n>0$ then we give $p_1, p_2, \ldots, p_n$ as the mean service times;

Algorithm in multi-service framework

Step 4: Calculate the vector $A = (a_1, a_2, \ldots, a_m)$ where $a_k = t_k$ if service k was not used till now (even if requested) and

$$a_k = \frac{rt(k)}{ns(k)}$$

where $rt(k)$ is the real time spent today for service k and $ns(k)$ is the number of people that used the service k today till now (we need to keep this daily information).

Algorithm in multi-service framework

Step 5: Calculate $s = \sum_{i=1}^{n}(w_1 a_{p_i} + w_2 p_i)$ where $a_{p_i} \in A$ is the coefficient corresponding to the i^{th} service in the line (that takes time p_i) and $w_1 + w_2 = 1$;

Step 6: Calculate the range $(s - r_j, s + r_j)$;

Step 7: Take a ticket.

Conclusion

- QURAMI could expand and give useful data to users
- smart ranges of the time
- deep and robust statistical approach to be undertaken
- need of real data for testing purposes

Thank you very much for your attention.