Queue Mining: Queueing Theory meets Process Mining

AIS Group Meeting

Arik Senderovich

10/11/2016
Short Bio

- Postdoc @ Technion
- BSc in IE&M (Knowledge Systems)
- MSc in STAT (Service Engineering)
- PhD in Data Science (Queue Mining)
Favorites: Service Processes

Processes where (efficient and effective) service is the desired business outcome:

- Call centers
- Hospitals
- Transportation
SEELab and SEEData
Hand-made Performance Modeling

Yom-Tov and Mandelbaum [2014]
Automated process modeling based on data = Process mining

Logging
Focus: Performance Analysis

Illustration by Wil van der Aalst
Queue mining = Process mining with Queueing Theory

State-of-the-art: Approach I

From Rozinat et al. [2009]; Rogge-Solti et al. [2013]
Queue Mining: Approach I

Queueing models:
- Analytically simple models (efficiency) – no need for simulation
- (Often) accurate performance analysis w.r.t. data (robust/generalize well)
State-of-the-art: Approach II

From van der Aalst et al. [2011]
Queue Mining: Approach II

Queueing features added to state:
- Examples: queue-lengths, delays, classes
- Input for machine learning techniques
Outline

- Background

- Single-station queueing models
  - Single-class
  - Multi-class

- Queueing networks
  - Pre-defined routing
  - Random routing

- Conformance checking with queueing networks

Feel free to ask questions
Single-Station Single-Class Queues

Are these useful models?
Single-Station Queues

Are these useful models?

- Building block of networks
Single-Station Queues

Are these useful models?

- Building block of networks
- Single-resource type processes
  - Total time is delay (queueing) and process time
Queueing Model: Building Blocks

Kendall’s notation – A/B/C/Y/Z+X:
- A – arrivals, B – service times
- C – static server capacity (n servers); Y – queue size
- Z – service policy (FCFS, LCFS, Processor Sharing…)
- X – (Im)patience

Abandonments
Example: M/M/n

Assumptions (A/B/C/Y/Z+X):

- Dropped notation Y,Z,X (defaults are taken): infinite queue size, FCFS policy, no abandonments
- M - Poisson arrivals (completely random, one at a time, constant rate)
- M - Exponentially distributed service times
- Easy to analyze when parameters are known (data)
Problem: Delay Prediction

How long will the target customer wait?

- Online prediction problem
- Approach I – fit q-model (&parameters) from the log
- Approach II – transition system + learning

CAiSE2014 paper with Weidlich, Gal, Mandelbaum
Notation and Accuracy Measure

- The actual waiting time of a customer: $W_i$
- Delay predictor from a certain method: $\theta^i_{\text{method}}$
- Accuracy via the root of average squared-error (RASE):
  $$\sqrt{\frac{1}{n} \sum_{i=1}^{n} (W_i - \theta^i_{\text{method}})^2}$$
- Systemic errors in assumptions- avg. absolute bias:
  $$\left| \frac{1}{n} \sum_{i=1}^{n} (W_i - \theta^i_{\text{method}}) \right|$$
Approach I: Queueing Model is Fitted

G/M/n+M model:
- Exponential service times and (im)patience
- General arrival rates, FCFS policy, unlimited queue
Two families of delay predictors:
1. Queue-length (state based)
2. Snapshot principle (history based)
Queue-Length Predictors

\[ \theta_{QLM-NA} = \frac{QL + 1}{n\mu} \]

\[ \theta_{QLM} = \sum_{i=0}^{QL} \frac{1}{n\mu + i\alpha} \]

Whitt [1999]
Snapshot Prediction: Last-to-Enter-Service

(Armony et al., 2009; Ibrahim and Whitt, 2009)

Prediction:
The last customer to enter service waited \( w \) in queue

\[
\theta_{LES} = w
\]
Approach II: Transition System Based

Transition system with queueing features:

- Queue lengths are clustered (heavy, moderate, typ.)
- Prediction is based on QL cluster + progress
Results I: Bank’s Call Center Data

![Bar chart showing RASE for all loads, heavy load, moderate load, and typical load across different categories: No Information, K-Means, QLM, Last-to-Enter-Service.](chart.png)
Results II: Bank’s Call Center Data

[Bar chart showing absolute bias for different loads and methods: All Loads, Heavy Load, Moderate Load, Typical Load. The chart compares No Information, K-Means, QLNA, QLM, and Last-to-Enter-Service methods.]
Single-Station Multi-Class Queues

(b) Queueing perspective

Useful?
Single-Station Multi-Class Queues

Useful?

- Different types of customers (VIP vs. Regular; Urgent vs. Ambulatory)
Useful?

- Different types of customers (VIP vs. Regular; Urgent vs. Ambulatory)
- Classes = activities (A vs. F – A gets priority)
Single-Station Multi-Class Queues

Useful?

- Different classes/types of customers (VIP vs. Regular; Urgent vs. Ambulatory)
- Classes = activities (A vs. F – A gets priority)
Approach I for Multi-Class Queues

Assuming priority queues model:
- Queue length predictors – derived upper and lower bounds
- Snapshot principle (based on Reiman and Simon [1990])
Approach II for Multi-Class Queues
Results: Telecom Call Center Data

NLR, Tree – similar to De Leoni et al. [2014] (BPM14’ best paper)
What about networks of queues?

Snapshot principle holds in q-networks with **pre-defined routing**: public transport, outpatient clinics,...
Bus Traveling Time Prediction

Information Systems [2015] with Weidlich, Schnitzler, Gal, Mandelbaum
Bus Routes as Q-Networks
Prediction Problem

Bus stops

Time

Bus position

travel time?
Approach I: Snapshot Principle
Approach II: Load-related + Snapshot Features

Feature 1: day of the week
Feature 2: hour-minute-second
Feature 3: snapshot prediction
Feature 4
Ensemble of Regression Trees

**RF**  random forests (bagging)

**ET**  extremely randomized trees

**Intuition:**
build each tree non-optimally and independently from the others

**AB**  adaboost

**GB**  gradient tree boosting

**GBLAD**  robust gradient tree boosting

build trees sequentially, trying to add a tree that correct the flaws of the current ensemble
Boosting over the Snapshot Predictor
Results: Dublin Buses (GPS data)

Accuracy of the prediction over all trips. **Worse, Best, Best of S+xx and xx**

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>MARE (%)</th>
<th>MdARE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>539</td>
<td>23.37</td>
<td>16.15</td>
</tr>
<tr>
<td>RF</td>
<td>539</td>
<td>24.11</td>
<td>16.37</td>
</tr>
<tr>
<td>ET</td>
<td>519</td>
<td>22.05</td>
<td>15.23</td>
</tr>
<tr>
<td>AB</td>
<td>512</td>
<td>27.08</td>
<td>18.05</td>
</tr>
<tr>
<td>S+AB</td>
<td>504</td>
<td>26.32</td>
<td>16.84</td>
</tr>
<tr>
<td>GB</td>
<td>508</td>
<td>20.46</td>
<td>13.84</td>
</tr>
<tr>
<td>S+GB</td>
<td>494</td>
<td>19.95</td>
<td>13.53</td>
</tr>
<tr>
<td>GBLAD</td>
<td>520</td>
<td>19.38</td>
<td>13.86</td>
</tr>
<tr>
<td>S+GBLAD</td>
<td>514</td>
<td>19.06</td>
<td>13.65</td>
</tr>
</tbody>
</table>
What if routing is not pre-defined?

Approximation techniques, e.g. Queueing Network Analyzer (Whitt [1983]):

- Allows concurrency and non-exponential times
- Steady-state approx. (model per hour...)
Idea: PN->GSPN->QN Transformation

Four step approach:
1. Control-flow discovery (e.g., IM)
2. Enrichment (firing times, arrivals, resources,...)
3. Simplification (helps to avoid over-fitting)
4. Translation to QN for analysis (QNA)

BPM [2016] with Shleyfman, Weidlich, Gal, Mandelbaum
Conformance checking: A Queueing Network Perspective

Illustration by Wil van der Aalst
Conformance checking: A Queueing Network Perspective

Information systems [2015] with Yedidsion, Weidlich, Gal, Mandelbaum, Kadish, Bunnel
Conformance checking: A Queueing Network Perspective

The two queueing networks are compared:

1. Detect deviations between planned and actual performance measures
2. Root-cause analysis:
   - Compare structures (unscheduled activities)
   - Building blocks (arrivals, service times,...)

Root-cause of deviations can lead to performance improvement (example is coming up)
Example: Fork-Join Construct

Clinical Assistant (Infusion Vitals)

Pharmacy (Prepare Medication)

Infusion Nurse (Infusion)

Queue
Resource Server
Fork
Join
Step I: Unexpected Queueing

Drug is not ready!

Figure 5: Waiting time for Infusion (after vitals); Sample size = 996, Mean = 25min, Stdev = 29min
Step II: Production time is not the cause!
Step II: Production policy is...!
Process Improvement: Idea

- New policy for sequencing “vitals” patients to reduce waiting and increase throughput
- Dominates the EDD policy – proofs and experiments in the paper
Conclusion

- Queueing models are useful for process mining
- Especially: in service processes with scarce resources
- Happy to collaborate on further integration of queueing theory into process mining

Thank you!
sariks@tx.technion.ac.il