Machine Learning based session drop prediction in LTE networks and its SON aspects

Bálint Daróczy, András Benczúr
Institute for Computer Science and Control (MTA SZTAKI)
Hungarian Academy of Sciences

Péter Vaderna
Ericsson Research

5th International Workshop on Self-Organizing Networks IWSON, May 11th, 2015, Glasgow
Outline

Introduction

Dataset

Problem statement: analysis of session drops

New method for real-time drop prediction

Evaluation

SON aspects

Conclusions and future work
Introduction

Motivation

• Understand the potential and benefits of using Machine Learning (ML) techniques in SON
• Combine expert knowledge with ML techniques
• Apply the results in SON functions

Objective

• Root-cause analysis of session drops
• Drop prediction for individual sessions

Method

• Evaluation on LTE live network data
Data

- Data is based on eNodeB CELLTRACE logs from a live LTE network
- Event monitoring
- Data streaming is possible

- RRC connection setup / Successful handover into the cell
- UE context release / Successful handover out of the cell

Per UE measurement reports (RSRP, neighbor cell RSRP list)
- Configurable period
- Not available in this analysis

Per radio UE measurement (CQI, SINR)
- Period: 1.28sec

Event monitoring
- Data streaming is possible
## Data

### Session record

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp (s)</td>
<td>0.1 - 30</td>
<td>UNIX timestamp</td>
</tr>
<tr>
<td>Duration (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell Id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release cause</td>
<td>~20 different cause codes</td>
<td></td>
</tr>
<tr>
<td>Drop category</td>
<td>drop / no drop</td>
<td></td>
</tr>
<tr>
<td>Cqi_avg</td>
<td>1 - 15</td>
<td>Channel quality index</td>
</tr>
<tr>
<td>Harq_nack_dl</td>
<td>0 - 1</td>
<td>HARQ NACK ratio in downlink</td>
</tr>
<tr>
<td>Harq_nack_ul</td>
<td>0 - 1</td>
<td>HARQ NACK ratio in uplink</td>
</tr>
<tr>
<td>Rlc_dl</td>
<td>0 - 1</td>
<td>RLC NACK ratio in downlink</td>
</tr>
<tr>
<td>Rlc_ul</td>
<td>0 - 1</td>
<td>RLC NACK ratio in uplink</td>
</tr>
<tr>
<td>Sinr_pusch_avg (dB)</td>
<td>-4 - 18</td>
<td>Signal to Interference plus Noise Ratio on Uplink Shared Channel</td>
</tr>
<tr>
<td>Sinr_pucch_avg (dB)</td>
<td>-13 - 3</td>
<td>Signal to Interference plus Noise Ratio on Uplink Control Channel</td>
</tr>
</tbody>
</table>

*Independent variables are time series (arrays) of varying length*
Drop prediction

› Detailed dataset (report every 1.28 sec)
› Examples for typical individual sessions (last 30 sec)
Drop prediction

Average over all samples (dropped and not dropped)

<table>
<thead>
<tr>
<th>Period I.</th>
<th>Period II.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dropped sessions are usually in bad conditions already well before the drop

Feature selection:
- Time series for the technical parameters
- Minimum, maximum, most frequent item, mean, variance, median
- Same statistics of the gradient

<table>
<thead>
<tr>
<th></th>
<th>full</th>
<th>no drop, sample</th>
<th>drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>All time series</td>
<td>27.4M</td>
<td>210,000</td>
<td>210,000</td>
</tr>
<tr>
<td>At least 15 measurement points</td>
<td>2.8M</td>
<td>23,000</td>
<td>27,440</td>
</tr>
</tbody>
</table>
AdaBoost over statistics

- Weighted sum of weak learners (decision stumps)
- Used as baseline method (see Proactive call drop avoidance in UMTS networks, INFOCOM, page 425-429. IEEE, (2013))
- Feature ranking is possible:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RLC uplink Max</td>
<td>1.55</td>
</tr>
<tr>
<td>2. RLC uplink Mean</td>
<td>0.44</td>
</tr>
<tr>
<td>3. RLC uplink Mean</td>
<td>0.44</td>
</tr>
<tr>
<td>4. HARQ NACK downlink Max</td>
<td>0.29</td>
</tr>
<tr>
<td>5. HARQ NACK downlink Max</td>
<td>0.19 (second threshold)</td>
</tr>
<tr>
<td>6. Difference of RLC uplink Mean</td>
<td>0.24</td>
</tr>
<tr>
<td>7. SINR PUSCH mean</td>
<td>0.33</td>
</tr>
<tr>
<td>8. SINR PUSCH mean</td>
<td>0.26 (second threshold)</td>
</tr>
<tr>
<td>9. HARQ NACK downlink Max</td>
<td>0.21 (third threshold)</td>
</tr>
<tr>
<td>10. HARQ NACK downlink Max</td>
<td>0.15 (fourth threshold)</td>
</tr>
</tbody>
</table>

Fig.: Schapire
Dynamic Time Warping (DTW)

› Utilizing time evolution

Any distance (Euclidean, Manhattan, …) which aligns the $i$-th point on one time series with the $i$-th point on the other will produce a poor similarity score.

A non-linear (elastic) alignment produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis.
**Sample set (S):**

- **Session 1:**
  - sinr_pusch: [s11, s12, ...]
  - rlc_dl: [r11, r12, ...]
  - ...  
  - drop: yes/no

- **Session 2:**
  - sinr_pusch: [s21, s22, ...]
  - rlc_dl: [r21, r22, ...]
  - ...  
  - drop: yes/no

- **Session R**
  - ...

**Training set:**

- **Session T:**
  - sinr_pusch: [sT1, sT2, ...]
  - rlc_dl: [rT1, rT2, ...]
  - ...  
  - drop: yes/no

**Session to predict:**

- **Session X:**
  - sinr_pusch: [sX1, sX2, ...]
  - rlc_dl: [rX1, rX2, ...]
  - ...  

- **Session to predict:**
  - ...

**4 algorithms:**

- Adaboost over statistics (baseline)
- Adaboost over DTW
- Support Vector Machine (SVM) over statistics
- SVM over DTW (using Similarity kernel)
Similarity kernel

- Represent session X as Random Field $P(X|\theta)$ generated by distances from other sessions $s_1, s_2, …$
- Sample set S can be all training set, a sample, or cluster representatives

The simplest Random Field

- Fisher score of X is vector $G_X = \nabla \theta \log P(X|\theta)$
- Define a mapping $x \rightarrow G_x F^{-1/2}$
- Yields the Kernel $K(X, Y) = G_x^T F^{-1} G_y$

Intuitive interpretation: $G_X$ is the direction where the parameter vector $\theta$ should be changed to fit best to data point X.

- For a Gibbs Random Field (with an approx. Fisher Information matrix) we get a natural normalization of the distances from the sample $S$:

$$G^i_X = F_{ii}^{-\frac{1}{2}} G^i_X = \frac{E_\theta[\text{dist}(x, s_i)] - \text{dist}(x, s_i)}{E_\theta[\text{dist}(x, s_i)] - \text{dist}(x, s_i))^2}$$
Evaluation metrics

› Class confusion matrix

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>(drop)</td>
<td>+ (drop)</td>
</tr>
<tr>
<td></td>
<td>TP</td>
</tr>
<tr>
<td>(nodrop)</td>
<td>- (nodrop)</td>
</tr>
<tr>
<td></td>
<td>FN</td>
</tr>
<tr>
<td></td>
<td>FP</td>
</tr>
<tr>
<td></td>
<td>TN</td>
</tr>
</tbody>
</table>

› Receiver Operator Characteristic (ROC) curve

AUC = Area under the Curve

AUC_{red} = 0.6
AUC_{blue} = 0.5

› Based on the ranked order of drop „likelihood”
› No need for decision threshold
› Stable comparison of prediction methods
Evaluation

DTW similarity kernel calculations, one session
## SON aspects

### SON Operation

**Phase 1 (training):**

- **Labelled training set**
- **PREDICTOR**
- **OFFLINE**

**Phase 2 (application):**

- **Instant attribute set**
- **PREDICTOR**
- **Classification result (drop/no drop)**
- **REALTIME**

### Configuration of the predictor

<table>
<thead>
<tr>
<th>Gain matrix</th>
<th>Predicted class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual class</td>
<td>drop</td>
</tr>
<tr>
<td>drop</td>
<td>+2</td>
</tr>
<tr>
<td>no-drop</td>
<td>-2</td>
</tr>
</tbody>
</table>

### Examples

- **Protect applications sensitive to drops** (see *Proactive call drop avoidance in UMTS networks, INFOCOM, page 425-429. IEEE, (2013))*
  - Example: during parallel active voice call and data usage, block data usage when the call is predicted to be dropped

- **Handover optimization**
  - Improving HO decision by querying a Drop Predictor

1. Trigger HO decision
2. Set HO parameters e.g. hysteresis, time-to-trigger
3. Integrate Drop Prediction into the HO decision e.g. trigger HO preparation directly
Conclusions and Future work

› Conclusions
  – High-granularity measurements enable for using data analytics and learning techniques in RAN
  – AdaBoost highlighted that uplink channel caused most of the drops in the examined network
  – New ML approach utilizing time evolution has been introduced
  – DTW + SVM (with similarity kernel) method outperforms the baseline methods
  – Assessing the potential of predictions: many seconds ahead, using only short session history
  – Method is shown how to configure the predictor for SON functions by using the ROC curve

› Future work
  – Further drill-down (e.g. using reason code, RSRP measurement reports, etc.)
  – Assess the potential of learning techniques in real use-cases
Thank you for your attention