EP2420 Project 2 - Online learning with sample selection

Rolf Stadler

November 5, 2020

Project Objective

Data-driven network and systems engineering is based upon applying AI/ML (Artificial Intelligence / Machine Learning) methods to measurement data collected from network or compute infrastructure in order to build functionality and management capabilities. This is achieved through learning tasks that use this data for training. Examples of such tasks are KPI prediction using regression models and anomaly detection using clustering techniques.

The ML models are usually trained offline with measurements collected through monitoring. This method achieves good results but has drawbacks. First, the process takes a long time, often hours, since the system must be observed in various states to obtain an accurate model. Second, model training incurs a high computational cost, which increases (at least) linearly with the number of measurements. Third, changes in the system or the infrastructure may require new measurements and model re-computation.

In this project, we follow an online approach for model training, which allows for shorter learning times and lower computational costs. The approach involves a cache of fixed size to store measurement samples and re-computation of ML models based on the current cache. Key to this approach are sample selection algorithms that decide which samples are stored in the cache and which are evicted. For evaluation, we use traces gathered from a testbed at KTH that runs a video-on-demand service and a key-value store under dynamic load.

Project tasks

You will receive data sets with traces collected from a KTH testbed. The traces originate from running a service (either VOD or KV) on the testbed infrastructure while monitoring device and service metrics.

For all tasks in this project

- apply one of the methods described in Project 1, Task 1 to pre-process the trace;
- remove possible outliers;
- reduce the number of features in the data set to 16 through tree-based feature selection;
- use random forest regression for prediction and measure prediction accuracy in NMAE.

Task I - Offline learning on a small-size training set

1. The objective is to train a model for offline prediction on a small training set.
2. Select uniformly at random n samples from the trace to build the training set. Obtain a test set by selecting 1000 samples uniformly at random from the trace which have not been chosen for training. Perform this procedure 10 times, for $n = 32, 128, 512, 2048$. For each value of n, you have now created 10 pairs of test and training sets. Train predictors on theses data sets. Produce a graph that shows the prediction accuracy in function of n.

3. Perform the above procedure using the RR-SS algorithm for sample selection, as follows. First, choose 5000 samples uniformly at random and build a sequence of 5000 elements. This sequence is the input to RR-SS, which processes the samples one by one and fills a cache with n samples. Like above, for each value of n, you create 10 pairs of test and training sets. Train predictors on theses data sets. Produce a graph that shows the prediction accuracy in function of n.

4. Combine both graphs in a plot. One curve in this plot relates to sample selection through random sampling, the other through RR-SS.

5. Add to plot a horizontal line, which shows the accuracy of the predictor when trained offline on the entire data set. (Split the trace into training and test samples and create a training set with 70% of the samples and a test set with 30% of the samples.)

6. Describe the plot and explain the different behavior of the three offline learning methods.

Task II - Online learning on a small-size training set

1. The objective is to train and evaluate models for online prediction using a training set of size n. Use random forest regression for prediction and measure prediction accuracy in NMAE.

2. To build the training set, the observations (ie, samples) $(x_1, y_1), ..., (x_T, y_T)$ are processed one by one by a sample selection algorithm, which fills a sample cache of size n whereby $n \leq T$. Use two sample algorithms, reservoir sampling (RS) and RR-SS (see appendix).

3. Choose a random starting point t_0 on the trace, by choosing a sample time stamp uniformly at random. Use the sequence $(x_{t_0}, y_{t_0}), ..., (x_{t_0+T-1}, y_{t_0+T-1})$ as input for the sample selection algorithm, which fills a cache with n samples. Train a predictor using this cache. Evaluate the predictor on the observations $(x_{t_0+T}, y_{t_0+T}), (x_{t_0+T+1}, y_{t_0+T+1}), (x_{t_0+T+2}, y_{t_0+T+2}), ...$

4. Perform the above procedure 10 times for the cache sizes $n = 32, 128, 512, 2048$ using the sample algorithms reservoir sampling (RS) and RR-SS. Set $T = 1000$.

5. Produce a plot that shows the prediction accuracy in function of n. The graph includes two curves, one related to reservoir sampling (RS), the other to RR-SS.

6. Describe the plot and explain the different behavior of the two online sampling algorithms and their impact on the prediction accuracy.

7. Compare the offline learning methods and results of Task I with the online methods and results of Task II. Discuss your observations.

Task III - Online learning with model re-computation

1. The objective is to train and evaluate models for online prediction using a training set of size n. In contrast to Task II, the model and the cache is periodically adapted. Use random forest regression for prediction and measure prediction accuracy in NMAE.

2. In this investigation, you (a) pick a starting point t_0 on the trace, (b) collect and process T samples using a sample selection algorithm to fill a cache of size n, (c) train the initial model, (d) evaluate the model for $T1$ subsequent samples, (e) collect and process $T2$ samples using the sample selection algorithm to update the cache, (f) re-train the model using the current cache, (g) goto (d).

3. Perform the above procedure 10 times for the cache sizes $n = 32, 128, 512, 2048$ using the sample algorithms reservoir sampling (RS) and RR-SS. Set $T = 1000, T1 = 1000, T2 = 100$.

4. Produce a plot that shows the prediction accuracy in function of n. The graph includes two curves, one related to reservoir sampling (RS), the other to RR-SS.

5. Describe the plot and explain the different behavior of the two online sampling algorithms and their impact on the prediction accuracy.

6. Compare the learning methods and results of this Task (model re-computation) with that of Task II (fixed model). Discuss your observations.

Task IV - Minimizing the overhead of online learning while maintaining prediction accuracy

1. The objective is to find parameters for online learning so that the learning overhead is minimized while the prediction accuracy is maintained. We measure the overhead as the number of y observations we collect for model computation and re-computation. The prediction accuracy we want to achieve is that measured in Task III.

2. This means that you must find the best values for $T, T1, T2$. The parameters should give good results for both sample algorithms reservoir sampling (RS) and RR-SS. For the investigation, select a fixed cache size n.

3. In this task, it is up to you how you conduct the investigation and present the results. Discuss your findings.

4. Optional: Consider that the parameters $T1, T2$ can be dynamically adjusted as time progresses, eg, according to the number of sample encountered or the accuracy achieved at a particular point in time.

Appendix: Reservoir Sampling (RS) and RR-SS

References