Poster abstract: Link Quality Estimation: A Case Study for On-line Supervised Learning in Wireless Sensor Networks

Eduardo Feo-Flushing, Michal Kudelski, Jawad Nagi, Luca M. Gambardella, and Gianni A. Di Caro

Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno - Switzerland

{eduardo,michal,jawad,luca,gianni}@idsia.ch

Abstract. We focus on the implementation issues of on-line, batch supervised learning in computationally limited devices. As a case study, we consider the use of such techniques for link quality estimation. We compare three strategies for the on-line selection of the data samples to be kept in memory and used for learning. Results suggest that strategies that keep balanced the set of training samples in terms of ranges of target values provide better accuracy and faster convergence.

1 Introduction

In real-world scenarios, wireless sensor networks (WSNs) might need to operate in harsh conditions, facing dynamic variations in the environment. In many of these situations, it is useful, or even required, to model the variability and the uncertainty of the environments in order to understand the current situation and/or to make predictions about future events. To this end, *machine learning* techniques have been proven to be a flexible and effective approach to construct models able to capture complex relationships among many different variables. Unfortunately, these techniques often consume considerable amounts of computational resources. This might constitute a barrier to the use of machine learning in WSNs, since these networks are commonly made of devices with limited computational capabilities, both in terms of processing and storage. These limitations raise the need to optimize, or set strict bounds to, the consumption of computational resources during implementation and execution of machine learning algorithms, effectively balancing performance and consumption of resources.

Among the large variety of machine learning techniques, *supervised learning* plays a central role, both historically and in terms of actual applications. In supervised learning a system automatically learns on the basis of a set of labeled training data given as input. In WSNs facing variable situations, on-line supervised learning schemes in which the system adaptively *re-learn* or learn *incrementally* can turn out to be very useful in practice. In a very broad sense, the implementation of on-line supervised learning techniques, can be realized using two main approaches. In the *batch learning* approach, iteratively, a node first gathers batches of data to learn from and use them to re-build (from scratch) the prediction model. For instance, Support Vector Machines can be conveniently used at this aim. In the *incremental approach*, a node incrementally updates the

This research has been partially funded by the Swiss National Science Foundation (SNSF) Sinergia project SWARMIX, project number CRSI22_133059.

model after a new training sample is gathered. Although incremental approaches allow for significant savings in computing resources (mainly processing), their internal models might continually increase on size as new data samples arrive. Moreover, it might difficult to restrict the size of those models, or to control the trade-off between memory requirements and prediction accuracy. An example of incremental approaches which suffers from these problems is the popular *Locally Weighted Projection Regression* (LWPR) [1]. Instead, in batch learning, resource consumption is controllable by deciding/limiting the maximum size of the training set and the frequency of rebuilding the model, making them more suitable to be implemented in computationally limited devices such as WSNs.

In this work, we focus on the implementation issues of on-line, batch learning in computationally limited, embedded networked devices. As a case study, we consider the use of such techniques for *link quality estimates*. This is motivated by our recent work [2] in which we proposed LQL, a protocol for the *on-line supervised learning* of link quality estimates in wireless networks. LQL relies on passive channel monitoring to gather information about the quality of the current wireless links. This information is used to learn a regression mapping between the local network configuration and the expected link quality. Considering packet reception rate (PPR) as quality metric, it was shown that LQL allows to obtain fast and reliable estimates both in simulation and on a real testbed. In [2] we used LWRP for incremental learning and we did not address the issue of the limited computational capabilities, especially for *memory*, which we address here: as data is continually gathered, a node needs to decide which data to include in its training set and which not, since not all training information can be stored because of limited memory. We study different selection strategies and compare their performance using a dataset obtained from a real WSN testbed.

2 On-line training data selection for batch strategies

As each new sample arrives, we face the decision of whether this sample should be included or not in the training set, and, in the positive case, which sample must be then removed (if needed) to keep the set size within the prescribed limits. We evaluated three simple, but representative, strategies. The first is called FIFO, and it treats the current training set as a *first in, first out* queue: a new sample is always accepted and the oldest sample is always removed. In the RANDOM strategy, the decisions are taken according to a probabilistic rule. The probability of a new sample entering the set is 0.5. In the positive case, if the set is full, an existing sample to be removed is chosen uniformly at random. The SLOTTED strategy consists of uniformly partitioning the $[0, 1]$ interval of possible link quality values in n=10 intervals, ($q_1 = [0, 0.1], \dots, q_{10} = [0.9, 1]$). The training set is partitioned accordingly into n slots. Each slot i contains the samples whose link quality value (PRR in our case) falls inside the corresponding interval q_i. The strategy, aims to maintain a set of training data that covers more or less uniformly the range of link quality values (i.e., good and bad links should be equally represented in the set). All slots are managed with FIFO. For all strategies, the first model is built when the training set reaches the maximum allowed size. Then, after replacing half of the samples, a new model is built.

3 Experimental evaluation

To implement batch learning, we used *Gaussian Process Regression* (GPR) [3], a nonparametric, flexible regression method characterized by good prediction accuracy, computational tractability, and its relatively simple implementation [4].

We consider a dataset obtained from the INDRIYA Testbed, a 3D WSN deployed across three floors of the School of Computing at the National University of Singapore [5]. The network is composed of 139 TelosB sensor motes. Nodes run the TinyOS operating system and are programmed in the NesC programming language. An implementation of LQL was used to collect data, which consisted in vectors of network features playing the role of measurable parameters determining the quality (PPR) of a link. Since INDRIYA nodes are static, in order to maximize the number of observed topologies and simulate changes related to mobility and/or radio switch-off in energy saving modes, the experiments were designed considering sub-networks of 40 logical nodes each, sampled out of the 139 nodes of the INDRIYA network. Each sub-network consists of randomly selected nodes, with the condition that each node has at least one neighbor. Each sub-network operates for 3 minutes. After that, a new sub-network is selected. Individual nodes also change their data generation rates (at the application layer). Rate changes occurred with intervals uniformly distributed between 5 and 45 seconds. Data rates are also uniformly distributed between 5 and 35 Pkts/sec. Data packet size is set to 100 bytes, for a total payload of 114 bytes.

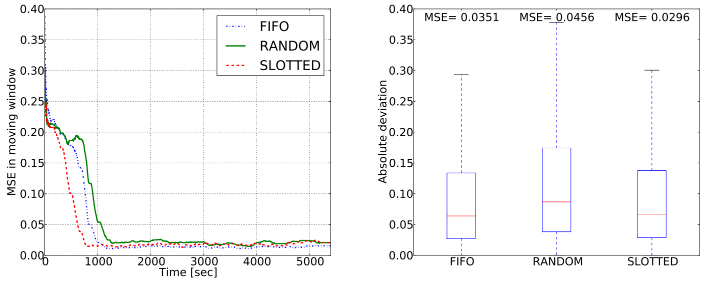
Since all data samples are timestamped, we could recreate the batch learning process for each single logical node (represented by different real nodes in each of the sub-networks) and run GPR considering a maximum training set of size 50 and 100 samples comparing the different selection strategies.

As performance metric, we measure *prediction errors*: the difference between measured and predicted values of PRR. These are calculated for each sample immediately after it is processed. We report the mean squared error (MSE) averaged in a moving window. We also report the final MSE value and the boxplot of the distribution of absolute prediction errors. These results are always averaged over all the nodes.

Figures (1) and (2) show the results for training set size of 50 and 100, respectively. From the analysis of the plots we can observe that the SLOTTED and FIFO strategies provide the best prediction accuracy. Over time, the SLOTTED strategy converges faster (i.e., generates robust models). Although the RANDOM strategy overall does not perform significantly worse, its initial slow convergence suggests that its accuracy could be dramatically reduced in environments that are quite dynamically changing over time.

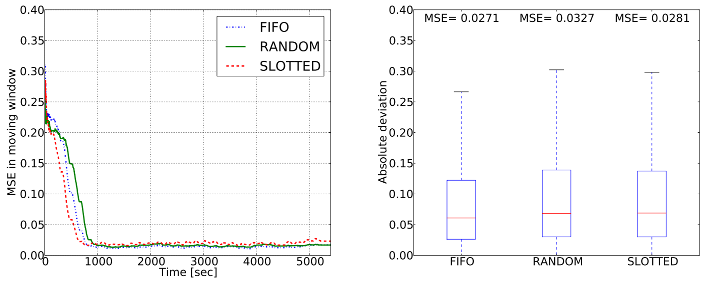
4 Conclusions and future work

We tackled the implementation issues of on-line supervised learning in devices with limited computational resources. We focused on batch learning approaches for processing training data and building a predictor in the context of link quality estimation. Accounting for memory limitations, for buffering training data, we compare three strategies for the on-line selection of the data samples to be



(a) Moving MSE (b) Distribution of the errors

Fig. 1: Performance of selection strategies for a training set of size 50 samples.



(a) Moving MSE (b) Distribution of the errors

Fig. 2: Performance of selection strategies for a training set of size 100 samples.

kept in memory and used for learning. Results suggest that strategies that keep balanced the set of training samples in terms of ranges of target values provide better accuracy and faster convergence. Future work includes the design of more complex strategies that measure the correlation between the samples composing the training set. The very good prediction accuracy achieved even with small number of samples also motivates the effort to implement the on-line supervised learning algorithms in small, embedded devices such as sensor nodes.

References

1. Vijayakumar, S., D'souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation **17** (2005) 2602–2634
2. Di Caro, G.A., Kudelski, M., Feo, E., Nagi, J., Ahmed, I., Gambardella, L.: On-line supervised learning of link quality estimates in wireless networks. In: Proc. of the 12th IEEE/IFIP Med-Hoc-Net. (2013) 69–76
3. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)
4. Gibbs, M., MacKay, D.J.: Efficient implementation of Gaussian processes. Technical report, Cavendish Lab, Cambridge, U.K. (1997)
5. Doddavenkatappa, M., Chan, M.C., Ananda, A.: Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed. In: Proc. of TRIDENTCOM. (2011) 302–316