Staleness Control for Edge Data Analytics

Atakan Aral
Vienna University of Technology
Vienna, Austria
atakan.aral@tuwien.ac.at

Melike Erol-Kantarci
University of Ottawa
Ottawa, Ontario, Canada
melike.erolkantarci@uottawa.ca

Ivona Brandić
Vienna University of Technology
Vienna, Austria
ivona.brandic@tuwien.ac.at

ABSTRACT

A new generation of cyber-physical systems has emerged with a large number of devices that continuously generate and consume massive amounts of data in a distributed and mobile manner. Accurate and near real-time decisions based on such streaming data are in high demand in many areas of optimization for such systems. Edge data analytics bring processing power in the proximity of data sources, reduce the network delay for data transmission, allow large-scale distributed training, and consequently help meeting real-time requirements. Nevertheless, the multiplicity of data sources leads to multiple distributed machine learning models that may suffer from sub-optimal performance due to the inconsistency in their states. In this work, we tackle the insularity, concept drift, and connectivity issues in edge data analytics to minimize its accuracy handicap without losing its timeliness benefits. Thus, we propose an efficient model synchronization mechanism for distributed and stateful data analytics. Staleness Control for Edge Data Analytics (SCEDA) ensures the high adaptability of synchronization frequency in the face of an unpredictable environment by addressing the trade-off between the generality and timeliness of the model.

CCS CONCEPTS

• **Information systems** → **Online analytical processing engines**; • **Computer systems organization** → **Distributed architectures**; • **Networks** → *Mobile ad hoc networks*; • **Computing methodologies** → *Planning and scheduling*.

KEYWORDS

Edge computing, data stream processing, concept drift, staleness control, non-stationarity, reinforcement learning.

ACM Reference Format:
Atakan Aral, Melike Erol-Kantarci, and Ivona Brandić. 2020. Staleness Control for Edge Data Analytics. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts), June 8–12, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3393691.3394219

1 INTRODUCTION

The past decade has seen the rapid development of the IoT and the introduction of an entirely new generation of Internet services that radically changed many traditional industries. In the IoT paradigm, all things – regardless physical or digital – are connected, and therefore, able to interact with each other remotely. The sectors that are immediately affected and revolutionized by the IoT are healthcare (smart medical devices), manufacturing (smart factories), energy (smart power grids) as well as urban development and transportation (smart cities and vehicles). As a repercussion of this so-called smart revolution, there exists an ongoing paradigm shift from core data analytics to edge data analytics (EDA) [3]. Strict latency requirements and unprecedented velocity of data are the main driving factors for the disruption of core analytics. Many services in the aforementioned areas depend on near real-time decisions based on big streaming data, which render data aggregation and analytics at a central data center infeasible, due to high network delay [8].

Distributed ML models, when deployed at the edge nodes (ENs), can be independently trained and periodically synchronized through a central parameter server. Using EDA techniques, trained model parameters, instead of training data, are transmitted and aggregated. This brings not only lower network latency and less bandwidth usage but also better privacy. Edge computing [6] is a natural fit as it enables processing in close proximity (e.g. at network gateways) or even right at the data source [2]. EDA also enables higher scalability due to the concurrent use of a high number of resources. However, current EDA architectures for distributed learning, such as Federated Learning [5] or Large-Batch Training [4] are not intended for near real-time applications, and therefore, do not consider dynamic synchronization of ML models. This is especially critical when consistently accurate decisions are required despite non-stationary ML models. For instance, concept drift, which is defined as the transformation of the target system over time in unforeseen ways [7], is a great threat to online data analytics. Although there exist effective solutions in a centralized setting, the problem escalates in a distributed and networked system, particularly under intermittent connectivity. Existing quorum or bound based consistency management techniques leave these challenges unanswered due to their rigidness in the face of unpredictability [1]. Thus, novel EDA techniques are needed to address time-sensitivity along with the consistency challenges that intermittent connectivity brings.

2 STALENESS CONTROL PROBLEM

A trivial example is given in Figure 1 to illustrate the staleness control problem. There exist four ENs in this scenario and the chart shows the arrival times of their model updates to the parameter server. At time t_0, two ENs, n_0 and n_1, are connected to the parameter server and deliver their updates with a short network delay. The other two ENs, n_2 and n_3, on the other hand, are not accessible at t_0 and can only deliver their updates at t_1 and t_2, respectively. Building a global model and broadcasting it at each update would disseminate the timeliest information to ENs; however, this would also result in high network overhead due to the abundance and

SIGMETRICS '20 Abstracts, June 8–12, 2020, Boston, MA, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-7985-4/20/06.
https://doi.org/10.1145/3393691.3394219

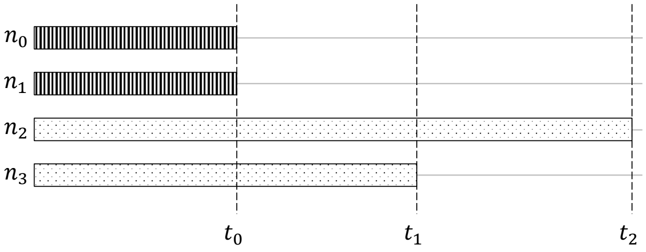

Figure 1: Example Scenario with Two Delayed Updates.

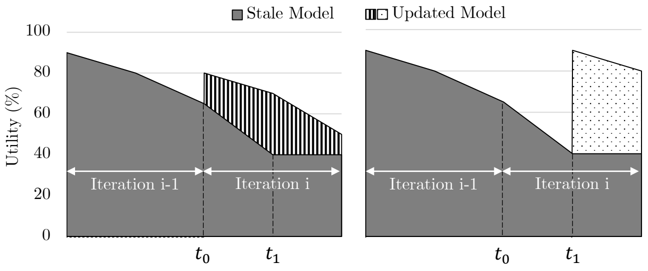

Figure 2: An Illustration of Alternative Solutions (i) and (ii).

wide-area distribution of ENs. Therefore, we consider the case that local models are updated online, whereas, the global model is periodically synchronized (i.e. once in a predefined iteration length). Ultimately, the solution options of a staleness controller in our example are (i) to build and broadcast the model immediately; (ii) to wait until one more EN responds before broadcasting; and (iii) to wait until both ENs respond before broadcasting.

Figure 2 illustrates the performance of solutions (i) and (ii) with a utility metric (e.g. mean accuracy). At iteration $i - 1$, each EN hosts a model, however, the utility of this model (solid grey) decreases over time due to insularity and non-stationarity. Thus, ENs should be updated with a model that incorporates global information at iteration i. Solution (i) on the left has the advantage of timeliness (update at t_0) so the ENs avoid the stale model from iteration $i - 1$. They would receive the updated model as soon as they are accessible by the parameter server. However, this would mean that the updates from lagging ENs (n_2 and n_3) are ignored for the iteration i. Solution (ii) on the right, on the other hand, updates the global model with additional information from n_3 at the cost of delaying the update. The updated model could eventually have higher utility, but the stale one has to be tolerated between t_0 and t_1.

3 CONTRIBUTIONS

Our main hypothesis in this work is that there exists an inconstant point to broadcast during each iteration, where the trade-off between generality and staleness yields the optimum accuracy. Waiting longer after this point does not increase generality but decreases accuracy due to staleness. We model the staleness control problem as a Markov decision process, a discrete-time stochastic control process that allows partly controlled and partly random outcomes. In staleness control, although it is possible to control whether to wait for the next update, the arrival time and the source of the update (i.e. the next state) are unknown. Based on this model, we propose an efficient reinforcement learning based algorithm SCEDA, which makes dynamic scheduling decisions by learning individual network connectivity trends of ENs as well as the significance of their updates. In designing SCEDA, particular attention is paid that it satisfies the following properties to facilitate its use in practice: (i) it does not require human intervention and automatically learns from experience, (ii) after limited initial bootstrapping, it continues improving and adapting to the changes, (iii) it does not require any monitoring at the ENs and operate with the data already available at the parameter server, and (iv) it computational overhead is low enough to allow real-time decision making.

We evaluate the runtime performance of the SCEDA algorithm through extensive simulation and emulation enhanced with real-world data from two EDA use cases: electric vehicle integration into the smart grid (with real commute traces) and virtual reality headset movement prediction (with real head movement traces).

4 DISCUSSION

Experimental results show that SCEDA minimizes the age of information while maximizing its value. Moreover, edge data analytics with the proposed dynamic model synchronization mechanism can achieve a comparable level of accuracy as core data analytics, yet with near real-time decisions. This confirms our hypothesis by showing that not every source is necessary to reach maximum accuracy and partial information is sufficient provided that it is effectively disseminated. SCEDA also outperforms the baseline algorithms, including the state-of-the-art quorum mechanism. The impact of this work goes far beyond our initial use case scenarios and it is possibly applicable to many stateful analytics tasks on distributed and streaming big data, in general.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and our shepherd Ganesh Ananthanarayanan, for many insightful comments and constructive suggestions that helped us improve the quality of the paper greatly. We also thank Dogan Altan for the valuable discussions regarding reinforcement learning techniques. This work has been partially funded through the Rucon project (Runtime Control in Multi Clouds), FWF Y 904 START-Programm 2015.

REFERENCES

[1] Atakan Aral and Ivona Brandic. 2018. Consistency of the Fittest: Towards Dynamic Staleness Control for Edge Data Analytics. In *Int'l European Conference on Parallel and Distributed Computing Workshops*. Springer, Turin, Italy, 40–52.

[2] Atakan Aral and Tolga Ovatman. 2018. A Decentralized Replica Placement Algorithm for Edge Computing. *IEEE Transactions on Network and Service Management* 15, 2 (2018), 516–529.

[3] Aakanksha Chowdhery, Marco Levorato, et al. 2018. Urban IoT Edge Analytics. In *Fog computing in the internet of things*. Springer, Cham, Switzerland, 101–120.

[4] Priya Goyal, Piotr Dollár, et al. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *CoRR* abs/1706.02677 (2017), 12.

[5] Brendan McMahan, Eider Moore, et al. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, Lauderdale, FL, USA, 1273–1282.

[6] Mahadev Satyanarayanan, Paramvir Bahl, et al. 2009. The Case for VM-based Cloudlets in Mobile Computing. *IEEE Pervasive Computing* 8, 4 (2009), 14–23.

[7] Geoffrey I Webb, Roy Hyde, et al. 2016. Characterizing concept drift. *Data Mining and Knowledge Discovery* 30, 4 (2016), 964–994.

[8] Eric P Xing, Qirong Ho, et al. 2015. Petuum: A New Platform for Distributed Machine Learning on Big Data. *IEEE Transactions on Big Data* 1, 2 (2015), 49–67.