Towards Active Simulation Data Mining*

Mirko Bunse[1], Amal Saadallah[1], and Katharina Morik[1]

TU Dortmund, AI Group, 44221 Dortmund, Germany
{firstname.lastname}@tu-dortmund.de

Abstract. Simulations have recently been considered as data generators for machine learning. However, the high computational cost associated with them requires a smart sampling of what to simulate. We distinguish between two scenarios of simulation data mining, which can be optimized with active learning and active class selection.

Keywords: Simulation · Active learning · Active class selection.

1 Introduction

Simulations are powerful tools for investigating the behavior of complex systems in science and engineering. Recently, there is an increase of attention towards the employment of simulated data in machine learning, an integration that is sometimes termed *simulation data mining* [11,2,4,12]. Its applications range from integrated circuit design [13] over milling processes [9], mechanized tunneling [8], robotized surgery [7], and cancer treatment [5] to astro-particle physics [3].

The goal of simulation data mining is to reason about a real system under study by learning from data which is generated by a simulation of that system. The benefit of this paradigm is that less or even no data is required from the actual system. Acquiring "real" data would often be costly or even be infeasible, e.g. if the actual system is still in the design phase and not yet deployed. Oppositely, simulations have the potential to provide large volumes of data, only at the expense of their computation. However, the need for accurate simulations often leads to complex simulation models (e.g. 3D numerical Finite-Element simulations), which result in high costs associated with data generation. The time and computational resources required by simulations motivate the active sampling of data, more precisely active learning (AL) [10] and active class selection (ACS) [6]. Both of these frameworks seek to select the minimal amount of training data while maximizing the performance of a prediction model trained with that data. In this short paper, we argue that there are two different strategies for the simulation of training data which distinctively correspond to AL and ACS. In fact, a simulation may either generate labels from a set of input features [11,12,13,2,9,8] or it may generate feature vectors from input labels [7,1]. The need for cost efficiency thus makes simulation data mining an imminent application scenario for methods from AL and from ACS.

[*] This work has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876. "Providing Information by Resource-Constrained Data Analysis", projects C3 and B3. http://sfb876.tu-dortmund.de

© 2019 for this paper by its authors. Use permitted under CC BY 4.0.

CEUR-WS.org/Vol-2444/ialatecml_shortpaper2.pdf

2 Active Sampling from Simulated Data

Every simulation is based on some kind of generative model. Such a *simulation model* may comprise analytical, geometric, agent-based, and probabilistic modeling approaches which represent the dynamics of the studied system. Namely, such a model represents how the state $\boldsymbol{s} \in \mathcal{S}$ of the system evolves over time:

$$Sim_{\boldsymbol{\rho}}(\boldsymbol{s}_t, \Delta t) \;=\; \boldsymbol{s}_{t+\Delta t}\,, \quad 0 \leq t \leq T, \tag{1}$$

where $\boldsymbol{\rho} \in \mathcal{P}$ is a vector of simulation parameters, which can be directly related to the parameters of the real system or process. In this view, the simulation is a fixed black box which encodes domain knowledge up to minor details. In the following, we distinguish between two scenarios in which machine learning models are trained on simulated data.

2.1 Forward Learning Scenario

In the first learning scenario, the simulation model has the same direction of inference as the machine learning model $f : \mathcal{X} \to \mathcal{Y}$ that is to be trained. This means that the initial state $\boldsymbol{s}_0 \in \mathcal{S}$ of the simulation is a function of the feature vector $\boldsymbol{x} \in \mathcal{X}$. The simulation then comprises multiple steps $\boldsymbol{s}_1 \in \mathcal{S}, \boldsymbol{s}_2 \in \mathcal{S}, \dots$ until a label $y \in \mathcal{Y}$ is obtained in the the final state $\boldsymbol{s}_T \in \mathcal{S}$. Thus, the simulation and the machine learning model both infer y from $\boldsymbol{x}$, as illustrated in Fig. 1. This learning scenario is probably the most common to date, being approached for example in [11,12,13,2,9,8].

$$\begin{tikzcd}
& \boldsymbol{s}_0 \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \boldsymbol{s}_1 \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \cdots \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \boldsymbol{s}_T \arrow[d, dashed] \\
{} \arrow[r, dashed] & \boldsymbol{x} \arrow[u, dashed] \arrow[rrr, "f"'] & & & y
\end{tikzcd}$$

Fig. 1. In the *forward* scenario, the prediction model $f : \mathcal{X} \to \mathcal{Y}$ has the same direction of inference as the simulation $Sim_{\boldsymbol{\rho}}$ from Eq. 1.

Since the mappings from $\boldsymbol{x}$ to $\boldsymbol{s}_0$ and from $\boldsymbol{s}_T$ to y are given by the problem statement, we could use the simulation to predict y directly—without learning another model f from simulated data. However, simulations often encompass even those details of the analyzed system that are only minor for the prediction task at hand. The computational resources required to compute data from such a precise model limit the resource efficiency of the simulation with respect to the prediction task. It is therefore often not feasible to run a simulation for prediction, particularly for resource-aware or real-time applications. Machine learning can then be used to build surrogate models which solve the prediction task efficiently [9,8]. The simulation can take the role of an oracle $o_{\mathrm{AL}} : \mathcal{X} \to \mathcal{Y}$, so that an AL technique can optimize the data being simulated.

2.2 Backward Learning Scenario

In the second scenario, the goal is to learn a prediction model of the "opposite direction" of the simulation. In other words, the prediction task to find the causes of observed effects. This task is modeled by the label y defining the input of the simulation and a corresponding feature vector $\boldsymbol{x}$ being produced, as outlined in Fig. 2. Since the machine learning model now solves another task than the simulation, it is able to achieve analysis goals which can not be achieved with the simulation alone. This second scenario is applied, for example, in robotized surgery, where the force which caused a deformation is predicted [7], or in astro-particle physics, where particle properties are predicted from indirect observations [1,3].

$$\begin{tikzcd}
 & \boldsymbol{s}_0 \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \boldsymbol{s}_1 \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \cdots \arrow[r, dashed, "Sim_{\boldsymbol{\rho}}"] & \boldsymbol{s}_T \arrow[d, dashed] \\
{} \arrow[r, dashed] & y \arrow[u, dashed] & & & \boldsymbol{x} \arrow[lll, "f"']
\end{tikzcd}$$

Fig. 2. In the *backward* scenario, the goal is to predict causes of observed effects. Thus, the direction of infence differs between $Sim_{\boldsymbol{\rho}}$ and f.

Other than in the forward scenario, a "backward" simulation can not predict y from $\boldsymbol{x}$. It can thus not be used as an AL oracle. However, we can use the simulation as the data generator $o_{\text{ACS}} : \mathcal{Y} \rightarrow \mathcal{X}$ that is assumed by ACS. One reason for distinguishing the two scenarios is thus the applicability of active sampling techniques. AL is only amenable in the forward scenario, ACS only in the backward case.

2.3 Active Sampling with Simulation Parameters

The goal of AL and ACS is to reduce the cost of training data generation. Starting from an initial data set, the simulation candidates are scored according to a selection criterion s and the best candidates are being simulated until a stopping criterion is met after some iterations. In this framework, AL scores feature vectors and ACS—in contrast—scores labels.

$$\begin{aligned} s_{\text{AL}} &: \mathcal{X} \rightarrow \mathbb{R} \\ s_{\text{ACS}} &: \mathcal{Y} \rightarrow \mathbb{R} \end{aligned}$$

Having a simulation, we can generalize this concept to a scoring of all simulation inputs, also comprising the auxiliary simulation parameters $\boldsymbol{\rho} \in \mathcal{P}$. Namely, AL can score each $(\boldsymbol{x}, \boldsymbol{\rho})$ and ACS can score each $(y, \boldsymbol{\rho})$ to have a higher chance of identifying the relevant input sub-spaces and to improve efficiency further.

3 Conclusion

We distinguish between two scenarios in which machine learning models are trained from simulated data. Our distinction corresponds to the applicability of AL and ACS, a property not previously detailed in simulation data science. Moreover, we conceive that active sampling techniques can be improved by accounting for the parameters of the simulation.

In upcoming work, we will further elaborate the paradigm of learning from simulations. In this regard, we deem data quality a particular issue because simulated data does not always picture the real system exactly. This problem may be tackled with transfer learning or domain adaptation techniques, which make the differences between multiple data sources—the simulation and the real system—explicit. Therefore, we consider simulation data science a promising use case also for combinations of active sampling and transfer learning.

References

1. Bockermann, C., Brügge, K., Buss, J., Egorov, A., Morik, K., Rhode, W., Ruhe, T.: Online analysis of high-volume data streams in astroparticle physics. In: Proc. of the ECML-PKDD, Part III. LNCS, vol. 9286, pp. 100–115. Springer (2015)
2. Brady, T.F., Yellig, E.: Simulation data mining: a new form of computer simulation output. In: Proc. of the 37th Winter Simulation Conf. pp. 285–289. IEEE (2005)
3. Bunse, M., Piatkowski, N., Morik, K., Ruhe, T., Rhode, W.: Unification of deconvolution algorithms for Cherenkov astronomy. In: Proc. of the 5th Int. Conf. on Data Science and Advanced Analytics (DSAA). pp. 21–30. IEEE (2018)
4. Burrows, S., Stein, B., Frochte, J., Wiesner, D., Müller, K.: Simulation data mining for supporting bridge design. In: Proc. of the 9th Australasian Data Mining Conf. (AusDM). CRPIT, vol. 121, pp. 163–170. Australian Computer Society (2011)
5. Deist, T., Patti, A., Wang, Z., Krane, D., Sorenson, T., Craft, D.: Simulation assisted machine learning (2018), http://arxiv.org/abs/1802.05688, under review
6. Lomasky, R., Brodley, C.E., Aernecke, M., Walt, D., Friedl, M.A.: Active class selection. In: Proc. of the ECML. LNCS, vol. 4701, pp. 640–647. Springer (2007)
7. Mendizabal, A., Fountoukidou, T., Hermann, J., Sznitman, R., Cotin, S.: A combined simulation and machine learning approach for image-based force classification during robotized intravitreal injections. In: Proc. of the 21st Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI). LNCS, vol. 11073, pp. 12–20. Springer (2018)
8. Saadallah, A., Alexey, E., Cao, B.T., Freitag, S., Morik, K., Meschke, G.: Active learning for accurate settlement prediction using numerical simulations in mechanized tunneling. Procedia CIRP (2019)
9. Saadallah, A., Finkeldey, F., Morik, K., Wiederkehr, P.: Stability prediction in milling processes using a simulation-based machine learning approach. In: 51st CIRP Conf. on Manufacturing Systems. Elsevier (2018)
10. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012)
11. Shao, Y., Liu, Y., Ye, X., Zhang, S.: A machine learning based global simulation data mining approach for efficient design changes. Advances in Engineering Software **124**, 22–41 (2018)
12. Trittenbach, H., Gauch, M., Böhm, K., Schulz, K.: Towards simulation-data science – a case study on material failures. In: Proc. of the 5th Int. Conf. on Data Science and Advanced Analytics (DSAA). pp. 450–459. IEEE (2018)
13. Wang, L., Marek-Sadowska, M.: Machine learning in simulation-based analysis. In: Proc. of the Int. Symp. on Physical Design (ISPD). pp. 57–64. ACM (2015)