Towards Understanding End-to-end Learning in the Context of Data: Machine Learning Dancing over Semirings & Codd's Table

[Extended Abstract; DEEM 2021 Invited Talk]

Wentao Wu
wentao.wu@microsoft.com
Microsoft Research, USA

Ce Zhang
ce.zhang@inf.ethz.ch
ETH Zurich, Switzerland

ABSTRACT

Recent advances in machine learning (ML) systems have made it incredibly easier to train ML models given a training set. However, our understanding of the behavior of the model training process has not been improving at the same pace. Consequently, a number of key questions remain: *How can we systematically assign importance or value to training data with respect to the utility of the trained models, may it be accuracy, fairness, or robustness? How does noise in the training data, either injected by noisy data acquisition processes or adversarial parties, have an impact on the trained models? How can we find the right data that can be cleaned and labeled to improve the utility of the trained models?* Just when we start to understand these important questions for ML models in isolation recently, we now have to face the reality that most real-world ML applications are way more complex than a single ML model.

In this article—an extended abstract for an invited talk at the DEEM workshop—we will discuss our current efforts in revisiting these questions for an end-to-end ML pipeline, which consists of a noise model for data and a feature extraction pipeline, followed by the training of an ML model. In our opinion, this poses a unique challenge on the *joint analysis of data processing and learning*. Although we will describe some of our recent results towards understanding this interesting problem, this article is more of a "confession" on our technical struggles and a "cry for help" to our data management community.

ACM Reference Format:
Wentao Wu and Ce Zhang. 2021. Towards Understanding End-to-end Learning in the Context of Data: Machine Learning Dancing over Semirings & Codd's Table: [Extended Abstract; DEEM 2021 Invited Talk]. In *International Workshop on Data Management for End-to-End Machine Learning (DEEM'21), June 20–25, 2021, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3462462.3468878

1 INTRODUCTION

Recent years have witnessed the incredible progress of machine learning (ML) systems, driven by the joint efforts from different communities including machine learning, systems, and data management. Because of efforts — just to name a few example from our community, System ML [2] Snorkel [23], ZeroER [28], TFX [3, 19], "Query 2.0" [29], Krypton [20], Cerebro [21], ModelDB [26], MLFlow [32], HoloClean [24], ActiveClean [16], NorthStar [15], mlinspect [7] and some of our own previous work [1, 18, 33]—given a training set, automatically constructing an end-to-end ML pipeline has become something possible for a user without much expertise. Despite various remaining challenges [30, 31], producing *some* (not necessarily the best) ML model given a training set is no longer "rocket science" for many applications.

As a natural next step, it comes MLOps, the DevOps process of machine learning. *How to provide principled guidance for a developer to continuously improve the quality, may it be accuracy, fairness, and robustness, of a given ML model?* As the quality of an ML model heavily relies on the quality of its input data, one core problem in MLOps is to reason about the *influence* of a data example or a subset of data examples, in the training set, to the trained model. Recently, researchers have developed a range of techniques to address this problem [6, 9, 10, 13, 14, 17, 25].

An Emerging Gap. Most, if not all, endeavors on analyzing the influence of data quality confine themselves in studying a *trivial* pipeline that only contains the ML model itself. In reality, however, most real-world applications contain more complex pipelines that typically consist of a sequence of feature transformations then followed by the training of an ML model. *How can we reason about the influence of data quality for such* nontrivial *ML pipelines?*

2 OUR CURRENT APPROACHES

Figure 1 illustrates our current thinking. We consider an end-to-end ML pipeline that consists of the following four components:

Noise Model on Training Set. The input to training is a *distribution* $\mathcal{D}$ over all possible datasets $\mathbb{D}$. Uncertainty can come from various sources, e.g., noise in data acquisition, weak supervision, or manually injected noise for robustness.

Feature Extraction Pipeline. A feature extraction pipeline consists of a sequence of transformations, forming a function $q : \mathbb{D} \mapsto \mathbb{D}$ mapping from one training set to another transformed dataset. In frameworks such as sklearn and ML.Net, these transformations include feature normalization, data augmentation, etc.

ML Training. Followed by a feature extraction pipeline, we then train an ML model $\mathcal{A} : \mathbb{D} \mapsto \mathbb{M}$ that takes as input a training set D and outputs $\mathcal{A}(D)$ as an ML model.

Model Validation. Given an ML model, we assume the existence of a utility function $U : \mathbb{M} \mapsto \mathbb{R}$, which maps each model to some utility, e.g., validation accuracy, fairness, robustness, etc.

DEEM'21, June 20–25, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8486-5/21/06…$15.00
https://doi.org/10.1145/3462462.3468878

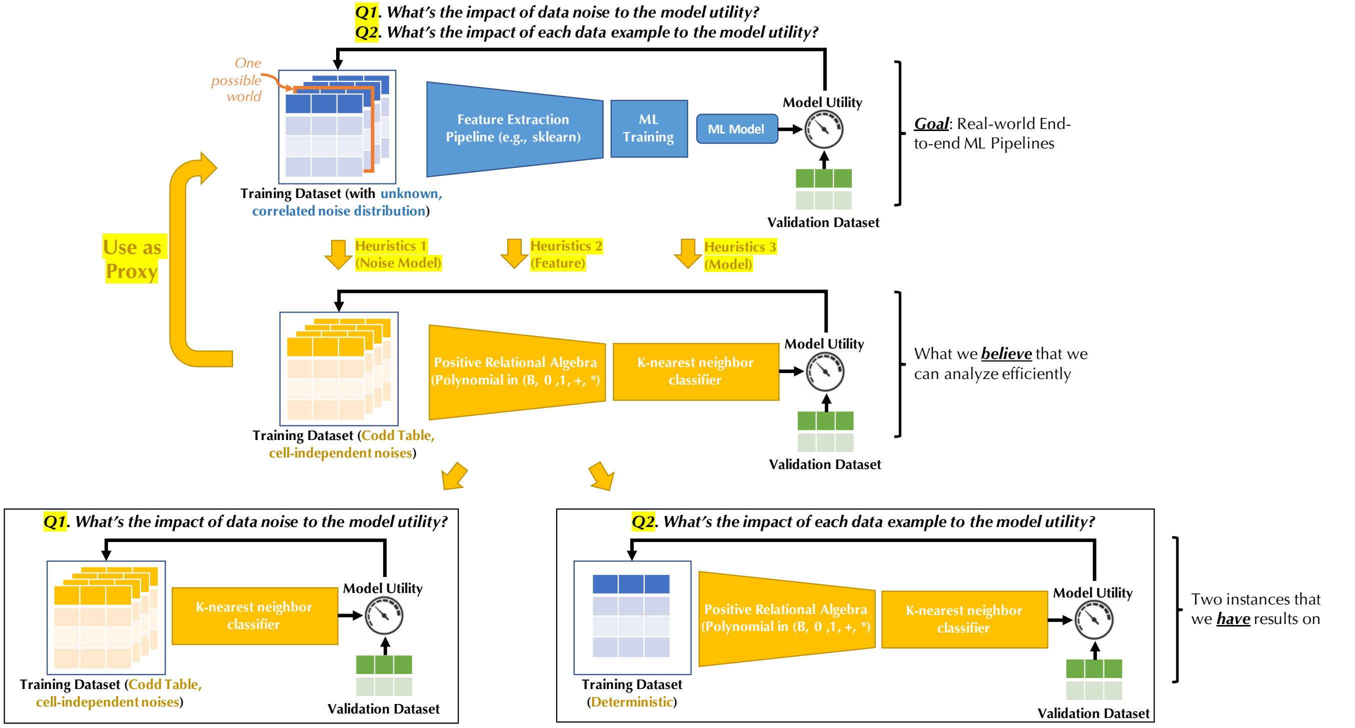
Q1. What's the impact of data noise to the model utility?
Q2. What's the impact of each data example to the model utility?
One possible world
Feature Extraction Pipeline (e.g., sklearn)
ML Training
ML Model
Model Utility
Goal: Real-world End-to-end ML Pipelines
Training Dataset (with unknown, correlated noise distribution)
Validation Dataset
Use as Proxy
Heuristics 1 (Noise Model)
Heuristics 2 (Feature)
Heuristics 3 (Model)
Positive Relational Algebra (Polynomial in (B, 0 ,1, +, *)
K-nearest neighbor classifier
Model Utility
What we believe that we can analyze efficiently
Training Dataset (Codd Table, cell-independent noises)
Validation Dataset
Q1. What's the impact of data noise to the model utility?
K-nearest neighbor classifier
Model Utility
Training Dataset (Codd Table, cell-independent noises)
Validation Dataset
Q2. What's the impact of each data example to the model utility?
Positive Relational Algebra (Polynomial in (B, 0 ,1, +, *)
K-nearest neighbor classifier
Model Utility
Training Dataset (Deterministic)
Validation Dataset
Two instances that we have results on

Figure 1: The goal of our endeavor is to understand two questions for an end-to-end ML pipeline: *Q1. What is the impact of data noise to the utility of a trained model? Q2. What is the impact of each data example to the utility of a trained model?* We hope to characterize these two questions in a principled way: to compute the entropy for Q1 and Shapley value for Q2. Both quantities are #P-hard to compute in general. To this end, we conduct three heuristics to approximate such a pipeline into something that we *believe* should give us PTIME algorithm for both quantities, by approximating the noise model, the feature extraction pipeline, and the ML model, into simpler counterparts. Our current results cover two simpler instances.

2.1 Entropy and Shapley Value

We are curious about two questions:

> *Q1. What is the impact of data* noise *on the utility of a trained model?*
>
> *Q2. What is the impact of each* individual *data example on the utility of a trained model?*

Our current thinking is to understand these two questions by analyzing the following two quantities.

Q1 and Entropy. To understand the impact of data noise, we are curious about the *entropy* of the utility over the distribution of input training set $\mathcal{D}$:

$$h = H\left(U \circ \mathcal{A} \circ q(\mathcal{D})\right).$$

Intuitively, when the noise in the input training set *does not matter*, we expect this entropy to be zero. Thus, a principled framework for "Data Cleaning for ML" is to clean data examples to decrease the expected entropy as much as possible. This process can benefit from decades of study on *sequential information maximization* [4]. This term also has an interesting connection to the *robustness* of the ML pipeline, via techniques such as randomized smoothing [5].

Q2 and Shapley Value. To understand the impact of each individual data example d_i, we are curious about the *Shapely value* of d_i. Let D be a training set:

$$s = \sum_{D' \subseteq D \backslash d_i} \binom{|D|}{|D'|}^{-1} \left(U \circ \mathcal{A} \circ q(D \cup \{d_i\}) - U \circ \mathcal{A} \circ q(D)\right)$$

Intuitively, this computes the expected marginal contribution of d_i over all possible permutations of other data examples. Previous research has shown the effectiveness of Shapley value in helping many downstream tasks [11].

2.2 Hardness, Heuristics, and Proxies

The technical challenge is that computing both the entropy and the Shapley value is #P-hard for general distributions, pipelines, and ML models. Although one can resort to Markov chain Monte Carlo (MCMC) to approximate these values, our current approach aims at understanding scenarios where they can be computed *exactly*.

To achieve this, we approximate an end-to-end ML pipeline $(\mathcal{D}, q, \mathcal{A}, U)$ with a *proxy* pipeline $(\tilde{\mathcal{D}}, \tilde{q}, \tilde{\mathcal{A}}, U)$:

(1) $\mathcal{D} \mapsto \tilde{\mathcal{D}}$: We approximate the noise model over training set as *Codd's table*, assuming cell-independent noises.
(2) $q \mapsto \tilde{q}$: We approximate the feature extraction pipeline as an expression in positive relational algebra equipped with a *provenance semiring* $(\mathbb{B}, \vee, \wedge, \texttt{false}, \texttt{true})$ [8].
(3) $\mathcal{A} \mapsto \tilde{\mathcal{A}}$: We approximate the ML model as a *KNN classifier*.

2.3 Entropy: KNN over Codd's Table

In our previous work [12] we show that it is possible to compute, in PTIME, the entropy of the proxy pipeline $(\tilde{\mathcal{D}}, \tilde{q}, \tilde{\mathcal{A}}, U)$ when

(1) The feature extraction pipeline $\tilde{q}$ consists of only `map` operations;
(2) The utility function U is *validation accuracy*.

2.4 Shapley Value: KNN over Positive Relational Algebra

In our previous work [9] we show that it is possible to compute, in PTIME, the Shapley value of the proxy pipeline $(\tilde{\mathcal{D}}, \tilde{q}, \tilde{\mathcal{A}}, U)$ when

(1) The feature extraction pipeline $\tilde{q}$ consists of only `map` operations;
(2) The utility function U is *validation accuracy*.

Also, in our ongoing work, we believe that we can compute, in PTIME, the Shapley value for the proxy pipeline $(\tilde{\mathcal{D}}, \tilde{q}, \tilde{\mathcal{A}}, U)$ when

(1) The feature extraction pipeline $\tilde{q}$ contains `map`, `fork`, and `1-to-many join` operations.
(2) The utility function U is *validation accuracy*.

2.5 Empirical Results

We also have empirical results that for many pipelines, especially those that we studied in real-world representative scenarios [22], the proxy pipeline $(\tilde{\mathcal{D}}, \tilde{q}, \tilde{\mathcal{A}}, U)$ is a good proxy for many end-to-end ML pipelines. This helps us to enable a range of applications efficiently, including data cleaning for ML [12], certifiable defense against backdoor attacks [27], and data debugging [11].

On the other hand, we also identify two cases where we fail, which are related to the decision on using a KNN classifier as a proxy for more complex ML models:

(1) **(Shapely value)** When the utility function reflects some global, populational utility (e.g., *fairness*), the KNN proxy tends to fail since it captures more of the local structures.
(2) **(Entropy)** When the downstream ML model reflects a different noise tolerance profile as KNN, the KNN proxy tends to fail.

3 TECHNICAL STRUGGLES AND THE "CRY FOR HELP"

Our current efforts reflect more about what we do not know than what we actually know. In the following, we list several key struggles that we have, hoping for feedback and help.

3.1 What is a Feature Extraction Pipeline? What is a Good Proxy?

One key struggle is that *we do not know how to characterize a feature extraction pipeline precisely*. On one hand, we have a generic abstraction that treats it as a data flow graph, which, despite its tremendous success in practice, does not give us the necessary specifics to reason about its theoretical property. On the other hand, treating it as a positive relational algebra query gives us a way to reason about data provenance, which, however, completely ignores all *computations* happening inside the pipeline. We are in dire need on some *formalism* of the pipelines that is more powerful than a polynomial in the provenance semiring while at the same time can give us more logical properties than what a data flow graph offers.

3.2 What is a Good Proxy for ML Models?

So far, we have been using kNN as proxy of the downstream ML model. This is a compromise we made to trade off between accuracy and computational efficiency. While having kNN as our proxy makes our problems computationally tractable, we could lose on accuracy when the downstream ML model itself is complex. We are in dire need of more advanced proxies that, similar to kNN, are amenable to mathematical analysis (so that we can design efficient algorithmic solutions), but can be more accurate than kNN when approximating downstream ML models.

3.3 What is a Good Metric for Measuring Data Quality Impact?

Is entropy the best way to reason about data cleaning for ML? Is Shapley value the best way to reason about data influence? How about data subgroups? How about influence functions [14]?

Although we believe that both entropy and Shapley value are reasonable metrics to measure the impact of data quality in our problem context, they are by no means the only ones. Since our current solutions are tied to these two metrics, a natural question to ask is then, if we replace the metrics by something else (e.g., influence functions), would that change our main observations and conclusions? For example, maybe our current algorithmic frameworks are no longer efficient and therefore we need brand-new solutions for such new metrics. It would be interesting and perhaps challenging to investigate the generalizability of our techniques when facing a new data quality metric, which can even be application-driven. On the other hand, it would also be interesting to study the commonality of the popular metrics used in practice and see its impact on the computational problem structure, which may bring in new insights to our algorithm design and generalization as well.

REFERENCES

[1] Leonel Aguilar, David Dao, Shaoduo Gan, Nezihe Merve Gurel, Nora Hollenstein, Jiawei Jiang, Bojan Karlas, Thomas Lemmin, Tian Li, Yang Li, Susie Rao, Johannes Rausch, Cedric Renggli, Luka Rimanic, Maurice Weber, Shuai Zhang, Zhikuan Zhao, Kevin Schawinski, Wentao Wu, and Ce Zhang. 2021. Ease.ML: A Lifecycle Management System for MLDev and MLOps. http://cidrdb.org/cidr2021/papers/cidr2021_paper26.pdf. Accessed: 2021-6-4.

[2] Matthias Boehm, Michael W Dusenberry, Deron Eriksson, Alexandre V Evfimievski, Faraz Makari Manshadi, Niketan Pansare, Berthold Reinwald, Frederick R Reiss, Prithviraj Sen, Arvind C Surve, and Shirish Tatikonda. 2016. SystemML: Declarative Machine Learning on Spark. *Proceedings VLDB Endowment* 9, 13 (Sept. 2016), 1425–1436.

[3] Eric Breck et al. 2019. Data Validation for Machine Learning. In *SysML*.

[4] Yuxin Chen, S Hamed Hassani, Amin Karbasi, and Andreas Krause. 2015. Sequential Information Maximization: When is Greedy Near-optimal? 40 (2015), 338–363.

[5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1310–1320.

[6] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. (April 2019). arXiv:1904.02868 [stat.ML]

[7] Stefan Grafberger, Julia Stoyanovich, and Sebastian Schelter. 2021. Lightweight Inspection of Data Preprocessing in Native Machine Learning Pipelines. In *CIDR*.

[8] Todd J Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '07* (Beijing, China). ACM Press, New York, New York, USA.

[9] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. 2019. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. In *VLDB*.

[10] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards Efficient Data Valuation Based on the Shapley Value. In *Proceedings of Machine Learning Research (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1167–1176.

[11] Ruoxi Jia, Xuehui Sun, Jiacen Xu, Ce Zhang, Bo Li, and Dawn Song. 2021. Scalability vs. Utility: Do We Have to Sacrifice One for the Other in Data Importance Quantification? *CVPR* (2021).

[12] Bojan Karlaš, Peng Li, Renzhi Wu, Nezihe Merve Gürel, Xu Chu, Wentao Wu, and Ce Zhang. 2021. Nearest Neighbor Classifiers over Incomplete Information: From Certain Answers to Certain Predictions. In *VLDB*.

[13] Pang Wei Koh, Kai-Siang Ang, Hubert H K Teo, and Percy Liang. 2019. On the Accuracy of Influence Functions for Measuring Group Effects. (May 2019). arXiv:1905.13289 [cs.LG]

[14] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 1885–1894.

[15] Tim Kraska. 2018. Northstar: An Interactive Data Science System. *PVLDB* (2018).

[16] Sanjay Krishnan et al. 2016. ActiveClean: Interactive Data Cleaning for Statistical Modeling. *PVLDB* (2016).

[17] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. [n.d.]. ActiveClean: Interactive data cleaning for statistical modeling. http://www.vldb.org/pvldb/vol9/p948-krishnan.pdf. Accessed: 2021-4-7.

[18] Tian Li, Jie Zhong, Ji Liu, Wentao Wu, and Ce Zhang. 2018. Ease.ml: Towards Multi-tenant Resource Sharing for Machine Learning Workloads. *Proceedings VLDB Endowment* (2018).

[19] Akshay Naresh Modi et al. 2017. TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. In *KDD 2017*.

[20] Supun Nakandala et al. 2019. Incremental and Approximate Inference for Faster Occlusion-Based Deep CNN Explanations. In *SIGMOD*.

[21] Supun Nakandala et al. 2020. Cerebro: A Data System for Optimized Deep Learning Model Selection. *PVLDB* (2020).

[22] Fotis Psallidas, Yiwen Zhu, Bojan Karlas, Matteo Interlandi, Avrilia Floratou, Konstantinos Karanasos, Wentao Wu, Ce Zhang, Subru Krishnan, Carlo Curino, and Others. 2019. Data Science through the looking glass and what we found there. *arXiv preprint arXiv:1912. 09536* (2019).

[23] Alexander Ratner et al. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *PVLDB* (2017).

[24] Theodoros Rekatsinas et al. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. *PVLDB* (2017).

[25] Boris Sharchilev, Yury Ustinovskiy, Pavel Serdyukov, and Maarten de Rijke. 2018. Finding Influential Training Samples for Gradient Boosted Decision Trees. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 4577–4585.

[26] Manasi Vartak et al. 2016. ModelDB: A System for Machine Learning Model Management. In *HILDA*.

[27] Maurice Weber, Xiaojun Xu, Bojan Karlas, Ce Zhang, and Bo Li. 2020. RAB: Provable Robustness Against Backdoor Attacks. *arXiv preprint arXiv:2003. 08904* (2020).

[28] Renzhi Wu et al. 2020. ZeroER: Entity Resolution Using Zero Labeled Examples. In *SIGMOD*.

[29] Weiyuan Wu et al. 2020. Complaint-Driven Training Data Debugging for Query 2.0. In *SIGMOD*.

[30] Doris Xin, Hui Miao, Aditya Parameswaran, and Neoklis Polyzotis. 2021. Production Machine Learning Pipelines: Empirical Analysis and Optimization Opportunities. (March 2021). arXiv:2103.16007 [cs.DB]

[31] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. 2021. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. (Jan. 2021). arXiv:2101.04834 [cs.HC]

[32] M. Zaharia et al. 2018. Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.* (2018).

[33] Ce Zhang et al. 2017. DeepDive: Declarative Knowledge Base Construction. *Commun. ACM* (2017).