

Training Models on Dirty Data Under Time Constraints

ABSTRACT

{{TODO}}

1. INTRODUCTION

Large and growing data are often susceptible to various forms of corruption, or *dirtyness*, such as missing, incorrect, or inconsistent values. These corruptions can negatively affect subsequent analysis in subtle but significant ways. Increasingly, analytics consists of Machine Learning “data products”, such as recommender systems, spam detectors, and forecasting models, which be very sensitive to data quality. Robust statistical methods have been extensively studied over the last 40 years, and trade off statistical efficiency and/or model bias for reduced sensitivity to the corruptions. While utilizing robust estimates can significantly reduce the generalization error of a statistical model, they essentially learn models that avoid the corruptions. As a result, certain populations of frequently corrupted data may be systematically mispredicted.

A data cleaning approach, complementing existing robust statistical techniques, can mitigate this concern for some types of error. Instead of avoiding the problem, cleaning works by repairing (or approximately repairing) the corruption. Data cleaning is an established line of research and there are numerous recent proposals of systems and techniques [2,4,9,10]. However, cleaning large data can be expensive, both computationally and in human effort, as an analyst has to program repairs for all errors manifest in the data [9]. In some applications, simple data transformations may not be reliable necessitating the use of even costlier crowdsourcing techniques [7,11].

An emerging solution to the growing costs of data cleaning is sampling [1,13,14]. Analysts can sample a large dataset, prototype changes on the sample, and evaluate whether these changes have the desired affect. The case for sampling is analogous to arguments for Approximate Query Processing (AQP) [3], where a timely approximate answer is more desirable than an exact slow answer. Our goal is an *anytime* framework for training Machine Learning models on dirty data. An analyst can clean as much of the data as possible within an allocated time-budget, and then the framework returns a best-effort model. While studied in aggregate query processing (e.g., BlinkDB [3] and Online Aggregation [5,8]), Machine Learning is far more sensitive to sample size (i.e., training data). The key problem is returning a result that is accurate enough for the analyst to judge the significance of their cleaning operations.

There are a few important observations that can allow us to address this problem. First, we might be able to use knowledge about the application (Machine Learning model) to improve efficiency of data cleaning. Increasingly, the consensus in Machine Learning research is that all training data are not created equal and some data are more informative than others [6,12]. This would imply that using the model to guide sampling towards informative data can mitigate the sensitivity to sample size. On the surface, this seems like an Active Learning problem [12], however, most Active Learning approaches only consider label acquisition for unlabeled data. Our second observation is that errors are often happen in batches are often tightly clustered, that is they affect similar data. This motivates an adaptive approach where as an analyst cleans data we learn how data is cleaned to further guide the cleaning.

In this paper, we propose CleanML (CML), an anytime framework for training Machine Learning models with data cleaning. CML supports a class of models called regularized-convex loss problems which includes linear regression, logistic regression, generalized linear models, and support vector machines. Algorithmically, we treat the analysts actions as part of a Stochastic Gradient Descent (SGD). The basic idea of SGD is to draw a data point at random, calculate the gradient at that point, and then update a current best estimate with that gradient. In this work, we start with an initialization (the dirty model) and iteratively update this initialization as we get more clean data. SGD and its variants are well-studied and there are lower-bounds on the convergence rates using these techniques. So we will use SGD as the scaffolding to build and analyze an adaptive algorithm. We also apply several techniques from database systems to optimize the model training. For example, if we intelligently partition data we believe is clean and dirty data, we can avoid expensive scans of this data. Next, when we do have to scan the data, we can amortize these scans using shared cursors to do multiple operations at each datum.

In summary, our contributions are

- We propose CleanML which given dirty data, a convex loss model, and a time budget, returns a model trained with respect to the clean data.
- CleanML is implemented with an non-uniform importance sampling approach that prioritizes points in the clean model that are most informative, and we provide analysis to identify the key tradeoffs.
- **{{Systems Contributions}}**
- We evaluate CleanML on real and synthetic datasets to show that non-uniform sampling achieves improved

performance in comparison to uniform sampling, and dirtiness-agnostic prioritization.

2. BACKGROUND

2.1 Machine Learning and Loss Minimization

In parametric Machine Learning, the goal is learn a set of model *parameters* θ from training examples. The hope is that the parameters θ can predict future phenomena. A common theoretical framework in Machine Learning is empirical risk minimization. We start with a set of training examples $\{(x_i, y_i)\}_{i=1}^N$ on which we minimize a loss function ϕ at each point parametrized that is parametrized by θ .

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \phi(x_i, y_i, \theta)$$

For example, in a linear regression ϕ is:

$$\phi(x_i, y_i, \theta) = \|\theta^T x_i - y_i\|_2^2$$

ϕ is often designed to be *convex*, essential meaning bowl-shaped, to make the training this model tractable. Typically, a *regularization* term $r(\theta)$ is added to this problem. The regularization term acts as penalty to avoid overfitting to the training data.

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \phi(x_i, y_i, \theta) + r(\theta)$$

For example, a popular variant of linear regression is called LASSO which is:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \|\theta^T x_i - y_i\|_2^2 + \lambda \cdot \|\theta\|_1$$

This class of problems includes all generalized linear models and support vector machines.

2.2 Training on Clean Data

The Theoretical Statistics community and Machine Learning community have extensively studied the problem of robustness to statistical outliers and random noise. Various *robust* methods have been proposed to reduce a loss function’s sensitivity to outliers. For example, in the case of linear regression, we can change the L_2 norm to an L_1 norm to mitigate the effect of outliers:

$$\phi(x_i, y_i, \theta) = \|\theta^T x_i - y_i\|_1$$

How exactly does a data cleaning approach differ from a robust method? We often in the problem setting where we are training on dirty data, but need to test or deploy the model on clean data. Consider an image classification task with incorrect labeling. If we train a model with respect to the incorrect labels, while we might have achieve a good out-of-sample accuracy on the incorrect labels, the classifications are incorrect in a semantic sense. Likewise, consider the case where we are predicting future product demand based on corrupted historical data. Training a model with respect to the corrupted data might have a low R^2 cross-validation error, but is incorrect at predicting the future trends.

In such scenarios, we see data cleaning as complementary to robust statistics. Data cleaning gives us information about the “true” data distribution, which is highly beneficial when errors have systematic biases. Without cleaning, certain subpopulations of data might be frequently misrepresented.

2.3 Stochastic Gradient Descent

Our goal is to design an anytime framework for training models on dirty data. Our main idea is to introduce data cleaning into the iterative algorithms that train the models. The problems described in the previous subsections are often trained using a technique called Stochastic Gradient Descent (SGD) or one of its variants. The basic idea of SGD is to draw a data point at random, calculate the gradient at that point, and then update a current best estimate with that gradient.

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma \nabla \phi(x_i, y_i, \theta^{(t)})$$

SGD can also be applied in a “mini-batch” mode, where we draw a subset of data at random and update with the average gradient.

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma \sum_{i \in S^{(t)}} \nabla \phi(x_i, y_i, \theta^{(t)})$$

At a high-level, this problem mirrors the problem explored in this work. We start with an initialization (the dirty model) $\theta^{(0)}$ and iteratively update the existing best model as we get more clean data. SGD and its variants are well-studied and there are lower-bounds on the convergence rates using these techniques.

3. REFERENCES

- [1] Trifacta. <http://www.trifacta.com>.
- [2] Sampleclean. <http://sampleclean.org/>, 2015.
- [3] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. BlinkDB: queries with bounded errors and bounded response times on very large data. In *EuroSys*, pages 29–42, 2013.
- [4] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. Katara: A data cleaning system powered by knowledge bases and crowdsourcing. 2015.
- [5] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, J. Gerth, J. Talbot, K. Elmeleegy, and R. Sears. Online aggregation and continuous query support in mapreduce. In *SIGMOD Conference*, pages 1115–1118, 2010.
- [6] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.
- [7] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *SIGMOD*, 2014.
- [8] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *SIGMOD Conference*, pages 171–182, 1997.
- [9] S. Kandel, A. Paepcke, J. Hellerstein, and H. Jeffrey. Enterprise data analysis and visualization: An interview study. *VAST*, 2012.
- [10] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J.-A. Quiané-Ruiz, N. Tang, and S. Yin. Bigdancing: A system for big data cleansing. 2015.
- [11] H. Park and J. Widom. Crowdfill: Collecting structured data from the crowd. In *SIGMOD*, 2014.
- [12] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.
- [13] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The data tamer system. In *CIDR*, 2013.
- [14] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *SIGMOD Conference*, pages 469–480, 2014.