ActiveClean: Interactive Data Cleaning For Modern Machine Learning

Sanjay Krishnan, Jiannan Wang, Eugene Wu †, Michael J. Franklin, Ken Goldberg
UC Berkeley, †Columbia University
{sanjaykrishnan, jnwang, franklin, goldberg}@berkeley.edu
ewu@cs.columbia.edu

ABSTRACT

Data cleaning is often an important step to ensure that predictive models, such as regression and classification, are not affected by errors such as inconsistent, out-of-date, or outlier data. Identifying dirty data is often a manual and iterative process, and can be challenging on large datasets. However, many data cleaning workflows can introduce subtle biases into the training processes due to violation of independence assumptions. We propose ActiveClean, a progressive cleaning approach where the model is updated incrementally instead of re-training and can guarantee accuracy on partially cleaned data. ActiveClean supports a popular class of models called convex loss models (e.g., linear regression and SVMs) and record-by-record user-defined data cleaning operations. ActiveClean also leverages the structure of a user's model to prioritize cleaning those records likely to affect the results. Evaluation on four real-world datasets suggests that for a fixed cleaning budget, ActiveClean returns more accurate models than uniform sampling and Active Learning when corruption is systematic and sparse.

1. INTRODUCTION

Building distributed frameworks to facilitate model training on large and growing datasets is a key data management challenge with significant interest in both industry and academia [1, 2, 4, 5]. While these frameworks abstract much of the difficult details of distributed Machine Learning (ML), they seldom offer the analyst any support in terms of constructing the model itself, such as which features to use or how to represent their data. The model construction process is still highly iterative, where through trial-and-error an analyst makes these choices eventually converging onto a model with the desired accuracy. To further complicate matters, data often arrives *dirty*, including missing, incorrect, or inconsistent attributes, due to faulty sensors, software, time delays, or hardware. Thus, part of the iterative model construction process involves identifying potentially dirty data, understanding how they affect the model, and applying techniques to mitigate their effects. While data cleaning is an extensively studied problem, the high dimensionality of many models can amplify even a small amount of erroneous records [7], and the relative complexity (in comparison to SQL analytics) can make it difficult to trace the consequnces of an error.

In prior work, we have noted the choice of data cleaning algorithm can significantly affect results even when using robust ML techniques [3, 6]. In one fraud prediction example, we found that simply applying Entity Resolution before model training improved true positive detection probabilities from

Figure 1: (a) Systematic corruption in one variable can lead to a shifted model. (b) Mixed dirty and clean data results in a less accurate model than no cleaning. (c) Small samples of only clean data can result in similarly inaccurate models.

62% to 91%. Despite this importance, in theory and in practice, the academic community has decoupled the data cleaning problem from featurization and ML. This is problematic because many ML techniques often make assumptions about data homogeneity and the consistency of sampling, which can be easily violated if the analyst applies data cleaning in an arbitrary way.

To understand how this may happen, consider an anlyst training a regression model on dirty data. At first, she may not realize that there are outliers and train an initial model directly on the dirty data. As she starts to inspect the model, she may realize that some records have a large residual value (not predicted accurately). Once she confirms that those records are indeed dirty, she has to design data cleaning rules or scripts to fix or remove the offending records. After cleaning, she re-trains the model–iterating until she no longer finds dirty data. This iterative process is the de facto standard, and in fact encouraged by the design of the increasingly popular interactive "notebook" ML development environments (e.g., IPython), but makes the implicit assumption that model training commutes with incremental data cleaning. This assumption is wholly incorrect; due to the well-known Simpson's paradox, models trained on a mix of dirty and clean data can have very misleading results even in simple scenarios (Figure 1).

In a parallel trend, the dimensionality of the features used in ML models is also rapidly increasing. It is now common to use 100,000s of features in image processing processing problems with techniques such as Deep Learning. Empirically, such feature spaces have facilitated breakthroughs in previously hard classification tasks such as image classification, robot actuation, and speech recognition. However, the pitfall is that the standard approaches for debugging and rea-

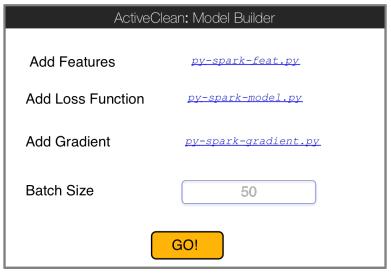

Figure 2: Model Builder

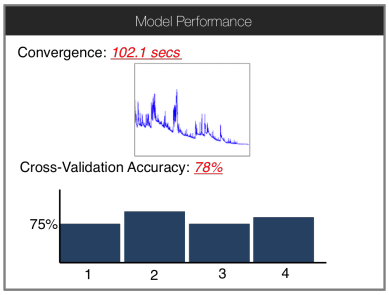

Figure 3: Performance Eval

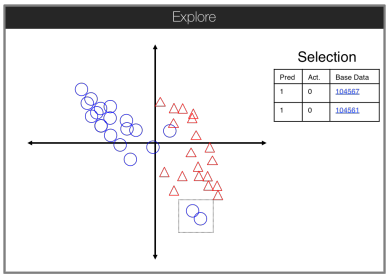

Figure 4: Diagnose Eval

soning about data error may lose their intuition for higher dimensions. In other words, it is often not obvious how an analyst should select which records to clean.

As it stands, there are two key problems in interactive model construction, (1) correctness, and (2) dirty data identification. We address these to problems in a system called ActiveClean which facilitates interactive training-cleaning iteration in a safe way (with expected monotone convergence guarantees) and automatically selects the most valuable data for the analyst to inspect even in the complex models popular in modern ML pipelines. The selection technique applied in ActiveClean uses pointwise gradients to generalize the outlier filtering heuristics to select potentially dirty data even in complex models. The analyst initializes an ActiveClean with an ML model, a featurization function, and the base data, and the ActiveClean initially returns the model trained on the dataset. ActiveClean also returns an array of data sampled from the model that are possibly dirty. The analyst can apply any value transformations to the data and then prompt the system to iterate.

We demonstrate ActiveClean with a visual interface allows analysts to debug complex ML models and understand the effects of dirty data. This interface will allow the analyst to specify the desired model and featurization. It will then visualize a sample of potentially dirty records and allow the analyst to clean the data appropriately. In our demonstration, we will present three experimental scenarios where the models are affected by dirty data:

EXAMPLE 1 (EVENT DETECTION WITH SVMS).

EXAMPLE 2 (VIDEO SEGMENTATION WITH CNNS).

EXAMPLE 3 (TOPIC MODELING WITH LDA).

2. THE ActiveClean INTERFACE

We will first describe the components of the ActiveClean interface.

2.1 Problem Specification

First, the analyst must specify the Machine Learning problem that they are addressing.

2.2 Performance Evaluation

2.3 Diagnose Interface

2.4 Clean Interface

3. REFERENCES

[1] Berkeley data analytics stack. `https://amplab.cs.berkeley.edu/software/`.

[2] A. Alexandrov, R. Bergmann, S. Ewen, J. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke. The stratosphere platform for big data analytics. *VLDB J.*, 23(6), 2014.

[3] A. Arxiv. Activeclean: Arxiv. `http://arxiv.org`.

[4] A. Crotty, A. Galakatos, and T. Kraska. Tupleware: Distributed machine learning on small clusters. *IEEE Data Eng. Bull.*, 37(3), 2014.

[5] G. Inc. Tensorflow. `https://www.tensorflow.org/`.

[6] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, and K. Y. Goldberg. Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression. In *CASE*, 2014.

[7] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli. Is feature selection secure against training data poisoning? In *ICML*, 2015.

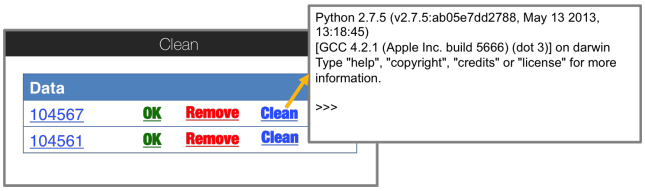

Figure 5: Clean Eval