Software 2.0 and Snorkel: Beyond Hand-Labeled Data

Christopher Ré
Stanford University
Palo Alto, CA
chrismre@cs.stanford.edu

ABSTRACT

In the last few years, deep learning models have simultaneously achieved high quality on conventionally challenging tasks and become easy-to-use commodity tools. These factors, combined with the ease of deployment compared to traditional software, have led to deep learning models replacing production software stacks in not only traditional machine learning-driven products including translation and search, but also in many previously heuristic-based applications. This new mode of software construction and deployment has been called Software 2.0 [2].

A key bottleneck in the construction of Software 2.0 applications is the need for large, high-quality training sets for each task. This talk describes Snorkel, a system that enables users to help shape, create, and manage training data for Software 2.0 stacks. In Snorkel applications, instead of tediously hand-labeling individual data items, a user implicitly defines large training sets by writing programs, called labeling functions, that assign labels to subsets of data points, albeit noisily. This idea of using multiple, imperfect sources of labels builds on work in distant supervision. However, if ignored, the uneven (and unknown) accuracies and coverages of the user-provided labeling functions can easily lead to suboptimal results:

Example. Suppose we have two training sets, T_1 and T_2, which are produced by two processes (or labeling functions). T_1 has high accuracy say 90% but low yield, labeling 10k points while T_2 has lower accuracy, 60%, but higher yield, 1M points. If we put the training sets together, we have a set T of 1.01M points with overall accuracy 60.3%. This could be distressing for a user: a model trained on T_1 seems to lose quality when trained on all of T. Naively combining the training sets fails to account for the different origins of T_1 and T_2.

Snorkel addresses this challenge of uneven training source quality by automatically learning a statistical model of the labeling functions' accuracies and correlation structure. The lack of hand-labeled data when learning this model raises several statistical challenges including estimating accuracies, learning correlations, and selecting features that refine labeling function quality [1,3,4]. Snorkel then uses this model to combine and reweight the labeling functions' labels, producing a set of *probabilistic* training labels, thus effectively passing along key
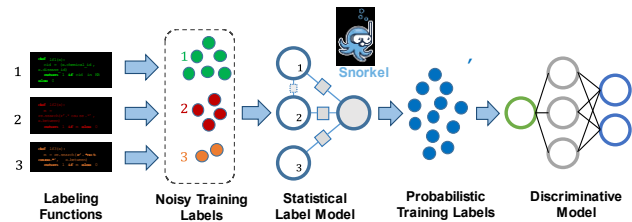
provenance information about the training. Our experimental results and theory show that estimating and accounting for the quality of the labeling functions in this way can lead to improved training set labels and boost downstream application quality— potentially by large margins, e.g., more than ten points of F1 score in NLP applications.

Exploiting the varied quality of supervision is a key building block to help manage the software 2.0 stack—but it's far from the only technique. Indeed, recent extensions of these core themes have led to projects automatically generating data augmentations, synthesizing labeling functions, and programmatically defining multi-task supervision. This does not even touch the many new opportunities for deployment and systems in Software 2.0. Hence, we contend there is a broad research motivated by Software 2.0.

Although only two years old, the Snorkel project powers applications in major tech companies and scientific efforts. It is used in applications in traditional machine learning applications like natural language processing, medical imaging, and prediction. Perhaps more excitingly for the Software 2.0 vision, it's also used in traditional enterprise applications like data cleaning, data integration, and semi-structured extraction— areas that have traditionally been difficult to deploy machine learning for.

For more information about the formal underpinnings and applications of Snorkel, we refer to Snorkel.stanford.edu for open source code, tutorial, and links to technical papers.

Bibliography
[1] S. H. Bach et al. "Learning the structure of generative models without labeled data." *ICML*, 2017: 273-282
[2] A. Karpathy. Software 2.0. https://medium.com/@karpathy.
[3] A. J. Ratner et al. "Data Programming: Creating large training sets quickly," *NIPS 2016:* 3567-3575
[4] A.J. Ratner et al. "Snorkel: Rapid training data creation with weak supervision," P*VLDB* 11(3):269-282, 2