University of California, Santa Cruz

CMPS203 Final Project

Spring, 2018

Data Programming: A New Paradigm For Unlabelled Data

Author:
Dhawal Joharapurkar
dhawal@ucsc.edu

Supervisor:
Dr. Cormac Flanagan
cormac@ucsc.edu

June 15, 2018

Contents

Abstract

This is a survey on the recent progress in the improvement of data labelling processes. I study the newly introduced "Data Programming", a paradigm which deals with programmatic creation of datasets. Then, I look at "Snorkel", a system that allows users to train models without hand labelling any data. The users can write labelling functions representing heuristics to label data, and can vary in coverages and accuracies. "Snorkel" denoises their outputs and combines them to provide probabilistic data labels. Then, I study "Babble Labble", which is an extension built using Snorkel, that processes heuristics described in natural language to labelling functions and thereby used to label data. These works constitute an important foray in this space of data labelling, and is right on the horizon of the massive data wave currently under way.

1 Introduction

This report captures the recent progress in the space of improving labelling processes to make unlabelled data more readily available for machine learning methods to use. Since most end-use machine learning models are discriminative, they suffer from the dearth of labelled data. This lack of availability of labelled data is due to several domain specific reasons, but also because it's difficult to collect, expensive, and might constantly change based on new discoveries or improvement in understanding of the space. The body of work that I study tries to sidestep this issue, by processing the unlabelled data and providing labels and this data generation process also lends itself into the model making phase, thus enabling the production of state-of-art models, quickly and cheaply.

In short, I survey three works:

- Data Programming: A new paradigm for the programmatic creation of training sets called Data Programming [Ratner et al., 2016].
- Snorkel: A system that enables users to train machine learning models without manually labelling data, by writing labelling functions representing heuristics [Ratner et al., 2017].
- Babble Labble: A framework for training classifiers using natural language explanations provided by annotators for each labelling decision [Hancock et al., 2018].

These works originate from Stanford's DAWN project, whose mission is to democratize AI by simplifying the building of AI applications.

2 Related Work

2.1 Data Programming Related Work

This work is a continuation of previous work in machine learning, termed as distant supervision. For example, in relation extraction from text, the input corpus is heuristically mapped to a knowledge base containing relations between entities [Mintz et al., 2009] [Craven et al., 1999]. There are various extensions to these methods, such as multiple instance learning one, discriminative feature based models, generative models, etc. This approach is similar to the generative models, but differs in that the generative models are not built on the user's inputs, which is the case with this approach. There are other approaches [Shin et al., 2015] [Mallory et al., 2015] that use user's heuristics to directly label unlabelled data, but they don't deal with the noise generated by this labelling functions.

Other similar approaches include crowd sourcing [Krishna et al., 2017] [Gao et al., 2011], in which the classical question of modeling the accuracies of various labellers without using gold data arises. This work is different in that, not only does it satisfy the conventional crowd sourcing patterns, it also allows for users to describe dependencies between themselves. This work also focuses on being able to label large data using a few functions, an opposite of crowd sourcing settings wherein large number of labellers label small bits of the dataset.

Co-training and Boosting are other well studied procedures similar to Data Programming, but differ in that they don't allow for explicit modelling of dependencies between views of data and that labelled data is explicitly necessary, respectively.

2.2 Snorkel Related Work

Combining various sources of weak supervision [Dalvi et al., 2013] [Joglekar et al., 2015] [Zhang et al., 2014] is an important challenge in being able to leverage these sources together and effectively. Researchers have looked at estimating the accuracy of label sources without access to the gold data,

mostly set in the crowd sourcing setting, where every user is looked as a source of labels with unknown accuracy. These methods use generative probabilistic models to estimate a latent variable, the true class label, based on noisy observations. Some other methods use user-specified dependency structures to estimate labels. Snorkel is different in that it supports various sources of weak supervision, by learning the correlational structure amongst these sources without ground data.

Snorkel is also related to other forms of supervision such as semi-supervised learning, transfer learning, and active learning, but Snorkel differs in that it's focused on managing weak sources of supervision and doesn't focus on combining itself with other types of supervision.

2.3 Babble Labble Related Work

Babble Labble is closely related to natural language explanations/instructions modeling and weak supervision. Most work [Ling and Fidler, 2017] [Liang et al., 2013] [Srivastava et al., 2017] on the natural language processing side, converts these explanations/instructions into features for a discriminative classifier straight away. Babble Labble on the other hand, converts them into labelling functions, sources of weak supervision, which it then uses to label data, and the uses the learnings from the correlation structures of these functions as information whilst building the discriminative classifier.

The related works pertaining to the combination of weak supervision sources is discussed in the above two sub sections.

3 A New Paradigm For Data Labelling

3.1 Motivation

Machine Learning is bit of a dichotomy in today's world – it's both far more and far less accessible than ever before. On one side, a deep learning model can run state of the art results without any manual feature engineering or algorithm development, thereby making it very accessible. But, there are many ways in which it is opaque, and thought of as black boxes. There aren't ways in which one can specify domain knowledge or heuristics into a model to augment it, and models are often clunky and can't handle a small change in objectives.

However, with the burst of data available, machine learning systems don't need to be hand-programmed at all. These systems simply leverage the data and end up learning the idiosyncrasies and dependencies of the domain from the data. However, this in only true in case of usable data – when it is assembled, clean and debugged, a really expensive and slow task, especially when domain expertise is required. Moreover, in the real world, things are volatile, and the processes might change over time, rendering the old data useless and we'd have to re-iterate the data processing with the updated guidelines. As you can see, this is a horror for scaling.

Hence, for all these reasons, more and more researchers are turning to weaker forms of supervision, such as heuristically generating training data using external knowledge bases, patterns or rules, or other classifiers. These are more or less categorically ways of programmatically generating training data – or, in a catchphrase "Data Programming".

3.2 Data Programming

As in many machine learning application settings, the common problems faced are as follows:

- Hand-labelled data is expensive – and procuring it is slow and tedious.
- Lack of external knowledge bases of the domain – rendering traditional distant supervision unusable.
- Constant change in requirements or processes – forcing us to update the models, which isn't an easy task.

Hence, they propose data programming, a paradigm for programmatic creation of training data sets, allowing systems to benefit from the data generated via these systems. In this paradigm, the data is lablled via heuristic rules called labelling functions, rather than labelling each data point by hand.

To write it more formally, a labelling function is $\lambda_i : \chi \mapsto \{-1, 0, 1\}$, a user defined function that encodes domain heuristic and provides a label to some subset of the data. Considering a binary classification task, the objective is to minimize the logistic loss under a linear model given some features,

$$l(w) = \mathbf{E}_{(x,y)\sim\pi}[log(1 + exp(-w^T f(x)y))]$$

where we have some distribution π over object and class pairs $(x, y) \in \chi \times \{-1, 1\}$.

Labelling functions have varying accuracies and coverages, as it represents a pattern that a user wishes to tell their model. It's a simple way of encoding this information than hand-labelling data points, as it quickens up the process. But, in doing so, these functions will tend to overlap, conflict, and have dependencies which users can provide as part of the specification.

Once the user has specified labelling functions, we can first construct a model in which each function behaves independently, given the true class label. If each function λ_i has some probability β_i for labelling an object, and some probability α_i for labelling the object correctly, and assuming the classes have a probability of 0.5 (equal classes), then the model has a distribution of:

$$\mu_{\alpha,\beta}(\Lambda, Y) = \frac{1}{2}\prod_{i=1}^{m}(\beta_i\alpha_i\mathbf{1}_{\{\Lambda_i=Y\}} + \beta_i(1-\alpha_i)\mathbf{1}_{\{\Lambda_i=-Y\}} + (1-\beta_i)\mathbf{1}_{\{\Lambda_i=0\}}) \quad (1)$$

where $\Lambda \in \{-1, 0, 1\}^m$ contains the output labels by the labelling functions and $Y \in \{-1, 1\}$ is the predicted class. From here, to find the parameters (α, β) which are most consistent with observations, we simple use maximum likelihood estimation (MLE), which in simple words is maximizing the probability that the observed labels produced on training examples occur under the generative model in (1).

From here, once the parameter is learnt, we have to minimize the expected risk over a linear model, so we define the noise aware empirical risk with a regularization term to compute the noise-aware empirical risk minimizer. Since this is a logistic regression problem, it can be solved using stochastic gradient descent as well.

Using empirical evidence from the experiments they conducted, they found that the labelling functions start being dependent on each other and by modelling this structure, they can improve accuracy in some cases. Hence, they extend their model in a way that a user can specify a dependency graph to show how the system can leverage it to better estimate parameters.

3.3 Snorkel

Snorkel is the first end-to-end system that implements data programming. It demonstrates that this paradigm enables users to produce high-quality

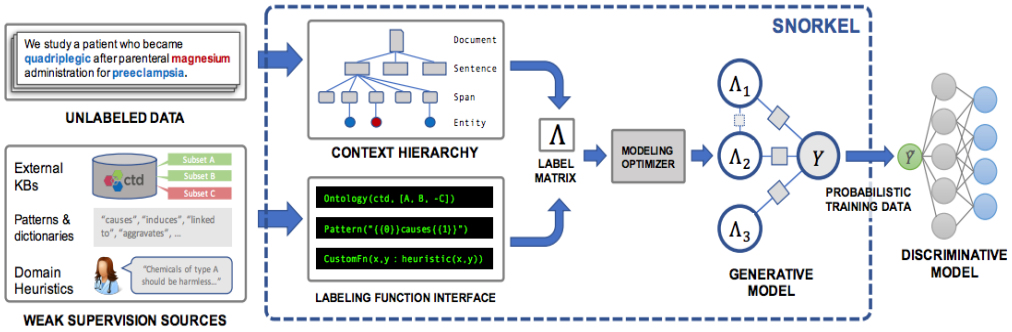

Figure 1: An overview of the Snorkel system

models for a wide range of tasks.

Snorkel's work-flow is designed around data programming and proceeds in three main stages (Figure 1):

1. **Writing Labelling Functions:** Users of Snorkel can specify labelling functions, allowing them to express various weak supervision sources such as external knowledge bases, patterns, heuristics, and more.

2. **Modelling Accuracies and Correlations:** Then, Snorkel automatically learns a generative model over the functions, to estimate their accuracies and correlations, without using any ground-truth data. It instead learns from the agreements and disagreements of the functions. The generative model is specified as follows:

$$p_w(\Lambda, Y) = Z_w^{-1} exp\left(\sum_{i=1}^{m} w^T \phi_i(\Lambda, y_i) \right)$$

where Z_w is a normalizing constant, Y are true labels, and ϕ_i are various factors. To learn this model without access to the true labels Y, we can minimize the negative log marginal likelihood given the observed label matrix Λ:

$$\hat{w} = \underset{m}{\text{argmin}} - log \sum_{Y} p_w(\Lambda, Y)$$

We can optimize this objective by alternating stochastic gradient with Gibbs's sampling, such as in contrastive divergence.

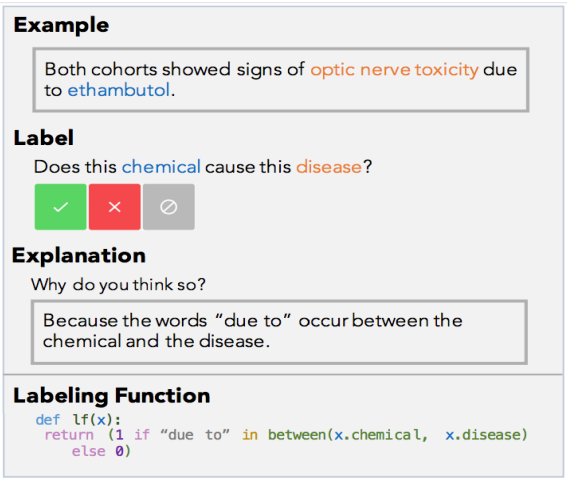

Figure 2: An overview of the Babble Labble system

3. **Training a Discriminative Model:** The output of Snorkel are probabilistic labels, which can be fed into numerous machine learning models (yes, even the discriminative ones now that we have labels). The labels produced by Snorkel are precise but can be low-coverage depending on the functions specified by the user(s), the discriminative model retains this precision but increases coverage and robustness on unseen data.

3.4 Babble Labble

Babble Labble is an extension to Snorkel in which users can provide labels as natural language explanations, and hence don't have to specify programmatic functions. This is a big extension, as now this system becomes accessible to people without a programming background, since they can simply write natural language and the system converts these explanations into code (Figure 2).

There are three key components to the system: a semantic parser which transforms natural language explanations into a set of logical forms representing labelling functions, a filter bank which removes as many incorrect labelling functions as possible, without needing gold truth labels, and a label aggregator which then combines these labelling functions taking care of overlapping and conflicting functions.

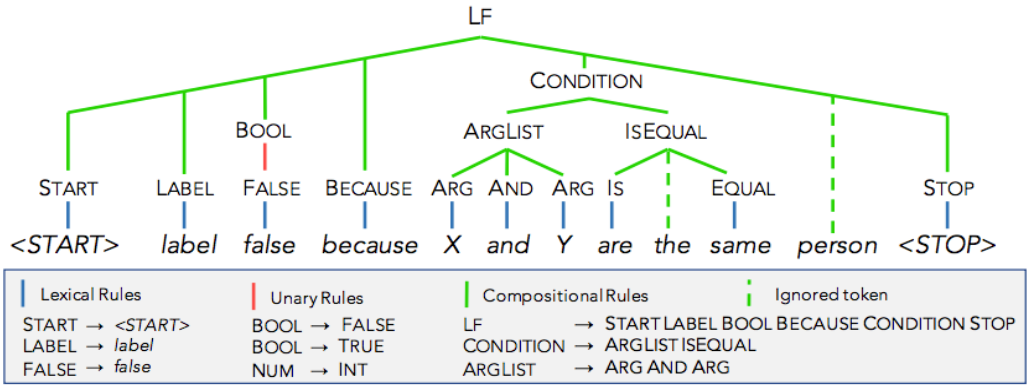

Figure 3: Rule based semantic parser of the Babble Labble system

3.4.1 Explanations

An user looks at subset S of the unlabelled dataset D, where $|S| \ll |D|$ and for each input $x_i \in S$ provides a label y_i and a natural language explanation e_i for why the example should receive that label. The explanation e_i, usually contains part of the example.

3.4.2 Semantic Parser

The role of the parser is the transform the natural language explanations into labelling functions $\{f_1, \ldots, f_k\}$, of the form $f_i : \chi \mapsto \{-1, 0, 1\}$ in a binary classification setting. The parser doesn't have to be an accurate one that results in the single correct parse of the explanation. However, the main focus if for the parser to have a high coverage, because the hypothesis is that many similar parses can be potentially useful. For this reason, they employ a simple rule based parser, which is usable without any training. They predefine their set of tokens and predicates for the domain they are working on, based on empirical inputs from subject matter experts. To identify candidate labelling functions, they recursively construct a set of valid parses for each span of the explanation, based on the substitutions in the grammar rules of the rule based parser. They allow any number of tokens in a span to be ignored to match to a rule, as it allows the parser to handle unexpected input and still result in a valid parse. The parser iterates over increasingly large subspans of the input, thereby generating candidates for each entity of the superset of the inputs tokens (Figure 3).

3.4.3 Label Aggregator

The label aggregator's role is to combine multiple labels which might be overlapping and conflicting, into one probabilistic label per example. This is already discussed in the previous section describing the functioning of Snorkel.

Later, a discriminative model is trained on top, which maintains the accuracy of the label aggregator, but increases the coverage and makes the system more robust.

4 Conclusion

In this report, I've surveyed three papers that are recent advancements in labelled dataset creation. They allow for the usage of weak supervision sources, in the form of labelling functions which are far more scalable than hand-labelling data points. These are important advancements as we move forward, as data is going to become more and more omnipresent, and being able to use it is literally the fuel of the machine learning industry. Without data, there are no models, and without any models there is no (artificial) intelligence. Thanks to this recent bodies of work, we are starting to understand how we can turn unusable data into something we can leverage to build our models upon.

References

[Craven et al., 1999] Craven, M., Kumlien, J., et al. (1999). Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86.

[Dalvi et al., 2013] Dalvi, N., Dasgupta, A., Kumar, R., and Rastogi, V. (2013). Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*, pages 285–294. ACM.

[Gao et al., 2011] Gao, H., Barbier, G., and Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14.

[Hancock et al., 2018] Hancock, B., Varma, P., Wang, S., Bringmann, M., Liang, P., and Ré, C. (2018). Training classifiers with natural language explanations. *arXiv preprint arXiv:1805.03818*.

[Joglekar et al., 2015] Joglekar, M., Garcia-Molina, H., and Parameswaran, A. (2015). Comprehensive and reliable crowd assessment algorithms. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 195–206. IEEE.

[Krishna et al., 2017] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

[Liang et al., 2013] Liang, P., Jordan, M. I., and Klein, D. (2013). Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

[Ling and Fidler, 2017] Ling, H. and Fidler, S. (2017). Teaching machines to describe images via natural language feedback. *arXiv preprint arXiv:1706.00130*, 2.

[Mallory et al., 2015] Mallory, E. K., Zhang, C., Ré, C., and Altman, R. B. (2015). Large-scale extraction of gene interactions from full-text literature using deepdive. *Bioinformatics*, 32(1):106–113.

[Mintz et al., 2009] Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

[Ratner et al., 2017] Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. *arXiv preprint arXiv:1711.10160*.

[Ratner et al., 2016] Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, pages 3567–3575.

[Shin et al., 2015] Shin, J., Wu, S., Wang, F., De Sa, C., Zhang, C., and Ré, C. (2015). Incremental knowledge base construction using deepdive. *Proceedings of the VLDB Endowment*, 8(11):1310–1321.

[Srivastava et al., 2017] Srivastava, S., Labutov, I., and Mitchell, T. (2017). Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1527–1536.

[Zhang et al., 2014] Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. (2014). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in neural information processing systems*, pages 1260–1268.