Modeling Annotator Accuracies for Supervised Learning

Abhimanu Kumar
Department of Computer Science
University of Texas at Austin
abhimanu@cs.utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@ischool.utexas.edu

ABSTRACT

Crowdsourcing [5] methods are quickly changing the landscape for the quantity, quality, and type of labeled data available to supervised learning. While such data can now be obtained more quickly and cheaply than ever before, the generated labels also tend to be far noisier due to limitations of current quality control mechanisms and processes. Given such noisy labels and a supervised learner, an important question to consider, therefore, is how labeling effort can be optimally utilized in order to maximize learner accuracy? For example, should we (a) label additional unlabeled examples, or (b) generate additional labels for labeled examples in order to reduce potential label noise [12]? In comparison to prior work, we show faster learning can be achieved for case (b) by incorporating knowledge of worker accuracies into consensus labeling [13]. Evaluation on four binary classification tasks with simulated annotators shows the empirical importance of modeling annotator accuracies.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

crowdsourcing, human computation, active learning

1. INTRODUCTION

Historically, supervised learning methods often outperformed their unsupervised counter-parts since providing a learner with more information can enable it to more quickly and effectively learn a desired pattern. Recent years saw this trend reverse, however, due to the massive growth of the Web having provided unsupervised methods with free and seemingly limitless training data [9]. Now the advent of crowdsourcing (e.g. via Amazon's Mechanical Turk[1]), has introduced another potentially disruptive shift: labeled data can suddenly also be obtained far cheaper, easier, and faster than ever before. A significant obstacle remains, though: crowdsourcing methodologies tend to suffer from poor quality control. Consequently, crowdsourced labels are typically quite noisy and exhibit high variance. An important research question, then, is how to most efficiently utilize a crowdsourced-based method for obtaining new labels in order to maximize learning rate with respect to annotation time and cost?

In this paper, we expand on Sheng et al.'s investigation [12] of how labeling effort may be best utilized in order to maximize learner accuracy. Should we (a) label additional unlabeled examples, or (b) generate additional labels for labeled examples in order to reduce potential label noise? In comparison to Sheng et al., the key difference of our work is incorporating knowledge of annotator accuracies into the model, which has large impact on resultant accuracies achieved by the learner. In another line of work by Snow et al. [13], a (slightly less) simple Naive Bayes approach is used to construct a weighted ensemble for consensus labeling in which labels are weighted proportionally to the accuracy of the annotator they come from. Snow et al. assume a fixed number of labels are obtained per example and do not investigate learning rates from consensus labeling. We integrate these two lines of prior work by using knowledge of annotator accuracy to more effectively aggregate labels and thereby improve the learning rate of our supervised model. Results on four binary classification tasks using C4.5 [10] show the empirical effectiveness of our approach, as well as suggesting potential benefit for other tasks and learning models.

2. RELATED WORK

Recent years have seen significant growth in label aggregation research. For example, Raykar et al. model label expertise via the EM algorithm to predict underlying labels [11], building on earlier work by Dawid and Skine [3]. Ipeirotis et al. differentiate error and bias in labeling mistakes with the idea that the bias can still be helpful for learning [6]. Dekel and Shamir give a unique approach to solve noisy label problem by pruning out experts who produce the most noise [4]. Whitehill et al. follow a different approach in that the labeler accuracies are not known a priori to them [15]. Yan et al. provide a predictive algorithm that reduces number of overlapping labels required for label prediction by determining which labels obtained need verification [16].

Alonso et al. use crowd workers to assess relevance [1]. Yang et al. predict the number of overlapping expert labels needed when there is substantial disagreement among the experts [17]. Both Alonso et al. and Yang et al. perform aggregation by simple majority vote. Mason et al. investigate the effect of

[1] https://www.mturk.com

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM'11). Hong Kong, China, Feb. 9, 2011.

compensation on worker accuracies [8]. Little et al. provide a comparative evaluation of collaborative and independent labeling approaches for labeling accuracy and cost [7].

3. TASK

Our task formulation largely mirrors that of Sheng et al. [12]: training a supervised learner for binary classification. We are given a large pool of unlabeled examples from which to draw examples for labeling. Our goal is to maximize classifier accuracy relative to labeling effort (unlike Sheng et al., we assume unlabeled examples are freely obtained). We assume each label requires a fixed cost to produce, regardless of the specific example or the annotator involved (we ignore issues of varying example difficulty or annotator expertise with regard to labeling cost). In addition to the pool of unlabeled examples, we assume a small set of seed examples already assigned a single label. Seed data provides a minimal training set for the classifier to which additional labeled examples may be added. All labels are potentially noisy.

We expect classifier accuracy to improve with more accurate and/or plentiful training data, suggesting a tradeoff for using labeling effort. At each labeling opportunity, should we (a) label an additional, previously unlabeled example or (b) generate a new label for a previously labeled example ("multi-labeling"). While individual labels may be noisy, effective aggregation of multiple labels can potentially yield more accurate consensus labels for training. As in Sheng et al., examples to be labeled are chosen as follows: for (a), uniformly at random from the pool of unlabeled examples, and for (b), using a fixed round-robin schedule which visits each (previously labeled) example once before repeating. We do not consider selection of examples to maximally benefit the learner, to maximally reduce uncertainty of existing labels, or based on example difficulty or the annotator expertise. We also assume the system's choice of (a) or (b) is fixed *a priori*; a more difficult task would require the system to repeatedly choose between (a) and (b) at run-time.

4. METHODS

We compare performance of several methods which differ in two dimensions: how labeling effort is utilized ((a) or (b) above), and for (b), how label aggregation is achieved.

Single Labeling (SL) [12]. Always label a previously unlabeled example; examples are never multi-labeled.

Multi-Labeling with Majority Voting (MV) [12]. Always generate an additional label for a previously labeled example. Labels are aggregated via simple majority vote.

Multi-Labeling with Naive Bayes (NB) [13]. As with MV, always re-label a previously labeled example. Labels are aggregated via Naive Bayes. Given labels $Y^j_{1:w}$ generated by w workers for example j, NB predicts label $X^j = \hat{x}$ via:

$$\begin{aligned}\widehat{x} &= \operatorname*{argmax}_x P(X^j = x|Y^j_{1:w}) \\ &\propto P(Y^j_{1:w}|X^j)P(X^j) \\ &= \prod_{i=1}^{w} P(Y^j_i|X^j)P(X^j)\end{aligned}$$

where we assume each annotator's labels are conditionally independent. Rather than model the full conditional distribution $P(Y|X)$,we instead model annotator i's accuracy by a single accuracy parameter $p_i = P(Y = X)$.

5. EVALUATION

Simulation. As in Sheng et al. [12], we assume each label is generated by a unique annotator with accuracy independent of the particular example. Given example j with true label $X^j = x$, annotator i generates a label $Y^j_i = x$ with probability p_i. Unlike Sheng et al., we assume these accuracies are known to the system, e.g. established from past work (this assumption will be further discussed later in the paper). Annotator accuracies are drawn from a uniform distribution whose interval is varied to simulate different annotator behaviors. As in Sheng et al., we assume each annotator generates exactly one label. Whenever a new label is needed, the simulator first samples a new annotator accuracy from this uniform distribution, then samples a correct or incorrect binary label based on this accuracy and the example's true label (known to the simulator but not to the system).

Data. We report on four benchmarks also used by Sheng et al.: `Mushroom`, `Spambase`, `Tic-Tac-Toe` and `Chess:King-Rook vs. King-Pawn`[2]. Since inspection of the datasets revealed minimal class imbalance (empirical proportion of examples with $X = 1$ is 48.2%, 39.4%, 65.3%, and 52%, respectively), for the NB method we assume a simplifying uniform prior for $P(X)$ which can be ignored. Results on the last two datasets exhibited similar trends as on the first two, so we omit these latter results due to space constraints.

Learning. We adopt the same C4.5 decision tree classifier [10] implemented by J48 in WEKA[3] as used by Sheng et al. We also follow the same experimental setup of a 70/30 train/test partition, and report results of averaging across 10 trials with different random partitions. We fix the number the of seed examples at 64, and as in Sheng et al., we generate labels in pairs to avoid tie breaking for MV. As an example, assume 64 labels are generated beyond the seed set. This would yield a total of 128 single-labeled training examples for SL, while for MV and NB, we would have 32 examples with 3 labels and 32 examples with one label.

Results. Figures 1-5 compare results of SL, MV, and NB methods across five experimental conditions which vary the range of annotator accuracies simulated. Results are summarized in each Figure's caption. Note the x-axis in these figures denotes the number of *additional* labels beyond the seed data (when the methods begin to be applied). While for multi-labeling methods it would have been interesting to directly measure consensus label accuracy achieved on training data, we focus our analysis instead on the effect of these consensus labels on classifier accuracy. Since datasets used exhibit minimal class imbalance, we report simple accuracy rather than measuring precision and recall.

Overall, NB tends to perform as well or better than the other two methods. When single label accuracies are already high (Figure 1), multi-labeling has little benefit and we should

[2] `http://archive.ics.uci.edu/ml/datasets.html`
[3] `http://www.cs.waikato.ac.nz/ml/weka`

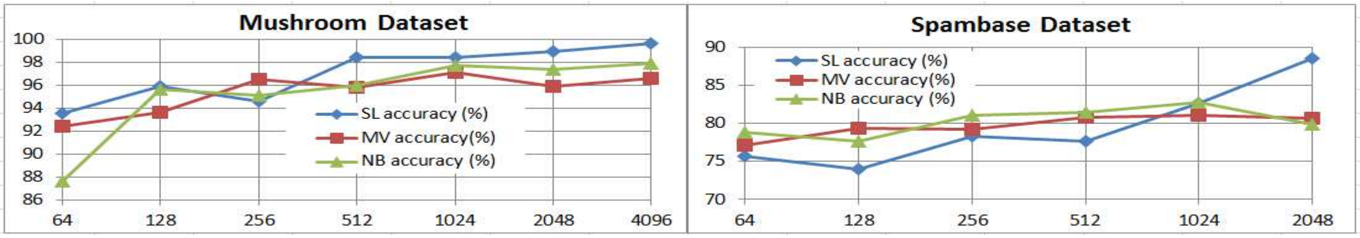

Figure 1: $p_{1:w} \sim \mathbf{U}(0.6, 1.0)$. With very accurate annotators, generating multiple labels (to improve consensus label accuracy) provides little benefit. Instead, labeling effort is better spent single labeling more examples.

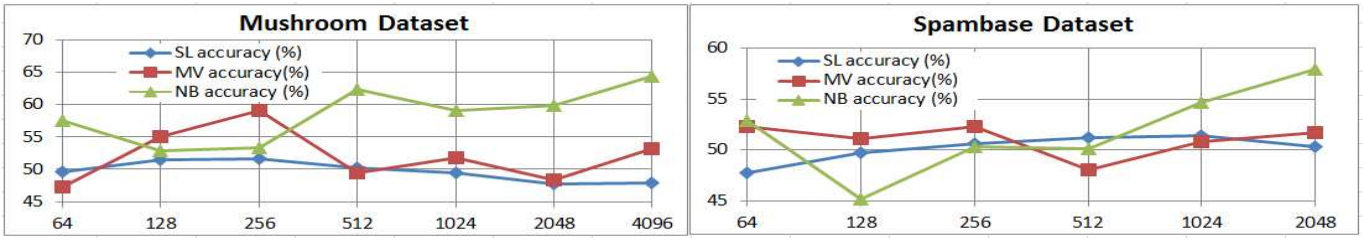

Figure 2: $p_{1:w} \sim \mathbf{U}(0.4, 0.6)$. With very noisy annotators, single labeling yields such poor training data that there is no benefit from labeling more examples (i.e. a flat learning rate). MV just aggregates this noise to produce more noise. In contrast, by modeling worker accuracies and weighting their labels appropriately, NB can improve consensus labeling accuracy (and thereby classifier accuracy).

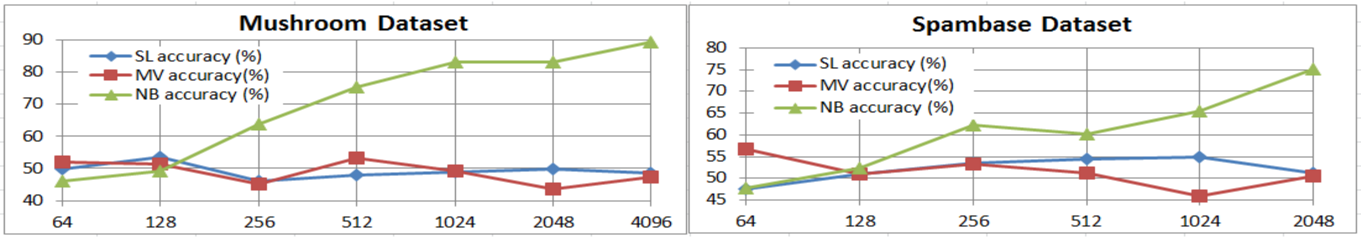

Figure 3: $p_{1:w} \sim \mathbf{U}(0.3, 0.7)$. With greater variance in accuracies vs. Figure 2, NB further improves.

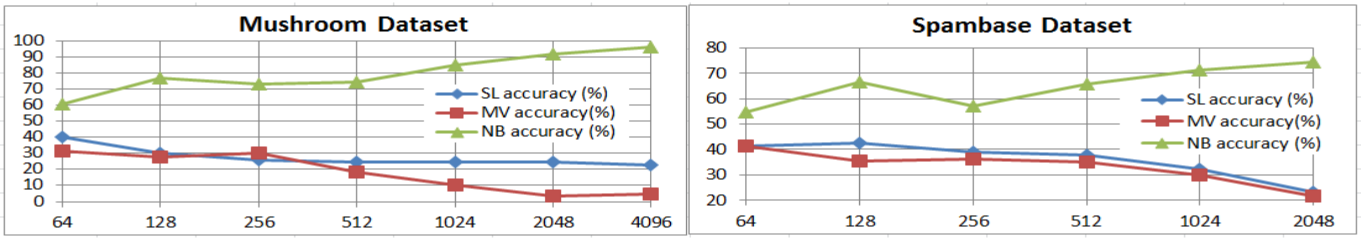

Figure 4: ($p_{1:w} \sim \mathbf{U}(0.1, 0.7)$). When average annotator accuracy is below 50%, SL and MV perform exceedingly poorly. However, variance in worker accuracies known to NB allows it to concentrate weight on workers with accuracy over 50% in order to achieve accurate consensus labeling (and thereby classifier accuracy).

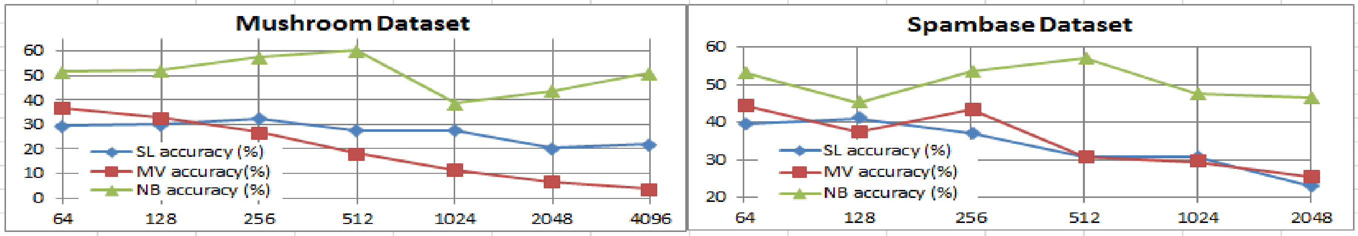

Figure 5: $p_{1:w} \sim \mathbf{U}(0.2, 0.6)$. When nearly all annotators typically produce bad labels, failing to “flip” labels from poor annotators dooms all methods to low accuracy.

simply label more examples. As average annotation becomes noisier, however, we see both SL and MV having diminishing accuracy while NB continues to be able to effectively exploit the most accurate annotators to improve classifier accuracy. Of particular note is the *adversarial* case of annotators achieving below 50% accuracy (average accuracy is below this in both Figures 4 and 5). In the former case, NB achieves accuracies above 90% despite this adversarial average accuracy, whereas the SL and MV schemes analyzed by Sheng et al. [12] perform very poorly. Even in the latter case, NB well-outperforms the other methods.

There is an important caveat to these results to mention. While we have assumed the system has knowledge of annotator accuracies, in the experiments above, only NB is exploiting this information, putting SL and MV methods at an unfair disadvantage. To remedy this, we tried making a trivial change to SL and MV methods to always "flip" labels produced by adversarial annotators, and we repeated all of our experiments. While not shown here, results are strikingly different: all methods generally perform comparably across conditions (excepting only the case of the most accurate annotators, in which case SL continues to dominate). As such, the key lesson appears to be the importance of modeling worker accuracies at all, rather than the specific method for how these accuracies are used. In practice, annotator accuracies must be estimated from prior observations, and so system knowledge of them will be noisy. This and other assumptions of our setup here will be important to test with actual crowd annotated data in future work.

6. CONCLUSION

This paper expanded upon Sheng et al.'s investigation [12] of how labeling effort can be optimally utilized in order to maximize learner accuracy, assuming a crowdsourced environment in which labels obtain may be very noisy and exhibit high variance. Results with simulated annotators showed that incorporating knowledge of worker accuracies into the model can have a very large impact on classifier accuracy, particularly in adversarial settings. Future work will investigate these issues and findings on real crowd-annotated data.

Other follow-on work includes analysis of crowd data in order to characterize general properties of crowd labor for modeling, e.g. expected number of workers and distribution of worker accuracies as a function of task nature and difficulty, etc. While we assumed all labeling effort was used either for labeling new examples or for re-labeling, a clear generalization will be to decide at each labeling opportunity during run-time which strategy is likely to be most effective. Similarly, we would like to couple this work with traditional active learning methods in which we must decide which example to label next in order to maximally benefit the learner or reduce variance of existing labels, etc. Annotators can be better modeled via: (a) estimating their accuracies from trap-questions or inter-annotator agreeement, (b) tracking and updating dynamic worker accuracies which change over time, and (c) modeling directional errors or biases of annotators rather than modeling accuracy via a single parameter.

"Wisdom of crowds" generally suggests a group of laymen can outperform a smaller number of experts assuming certain conditions are met (e.g. independence of judgment between crowd members) [14]. Similar effects have been observed with automated systems in which combining uncertain predictions from multiple independent learners via ensemble techniques tends to outperform the best individual systems [2]. This suggests an an interesting synergy to investigate between effective ensemble methods for leveraging the crowd and automated systems in tandem. A related trend will see hybrid systems increasingly integrate human effort with automation to "close the loop" and achieve greater functionalities than either can achieve on its own [16].

7. ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers for their valuable suggestions and feedback.

8. REFERENCES

[1] O. Alonso, D. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *ACM SIGIR Forum*, 42(2):9–15, 2008.

[2] R. Bell and Y. Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[3] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. In *Applied Statistics, Vol. 28, No. 1.*, 1979.

[4] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *COLT*, 2009.

[5] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[6] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *KDD-HCOMP*, 2010.

[7] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *KDD-HCOMP*, 2010.

[8] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". In *SIGKDD*, 2009.

[9] P. Norvig. Statistical learning as the ultimate agile development tool. In *CIKM*, 2008.

[10] J. Quinlan. *C4. 5: programs for machine learning.* Morgan Kaufmann, 1993.

[11] V. C. Raykar, S. Yu, L. H. Zhao, and G. H. Valadez. Learning from crowds. In *Journal of Machine Learning Research 11 (2010) 1297-1322*, MIT Press, 2010.

[12] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good? In *EMNLP*, 2008.

[14] J. Surowiecki. The Wisdom of Crowds, 2004.

[15] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, Vancouver, 2009.

[16] T. Yan, V. Kumar, and D. Ganesan. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys*, pages 77–90, 2010.

[17] H. Yang, A. Mityagin, K. M. Svore, and S. Markov. Collecting high quality overlapping labels at low cost. In *SIGIR 2010*, Geneva, 2010.