Learning to Rank From a Noisy Crowd

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

We study how to best use crowdsourced relevance judgments learning to rank [1, 7]. We integrate two lines of prior work: unreliable crowd-based binary annotation for binary classification [5, 3] and aggregating graded relevance judgments from reliable experts for ranking [7]. To model varying performance of the crowd, we simulate annotation noise with varying magnitude and distributional properties. Evaluation on three LETOR test collections reveals a striking trend contrary to prior studies: single labeling outperforms consensus methods in maximizing learner accuracy relative to annotator effort. We also see surprising consistency of the learning curve across noise distributions, as well as greater challenge with the adversarial case for multi-class labeling.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Crowdsourcing, learning to rank, active learning

1. INTRODUCTION

Crowdsourcing platforms like Amazon Mechanical Turk[1] are changing the dynamics of how we train our learners. While labeled data is no longer as difficult to obtain, individual labels tend to be noisier and require greater quality assurance, e.g. by requesting redundant labels from multiple annotators and resolving disagreements automatically via *consensus* [5, 3]. When annotation is noisy, how do we best utilize labeling effort to maximize learning? Do we label additional examples (improve coverage), or request more labels for already labeled examples to reduce label noise [5]? How should we compute consensus with such multi-labeling? For learning to rank [1, 7], how sensitive is the learner to different quanitities and distributions of label noise?

Prior work compares single labeling (**SL**) each example to multi-labeling for binary classification [5]. Given a fixed seed set of N singly-labeled examples and an infinite pool of unlabeled examples, SL grows this set of labeled examples with each new label (to increase example coverage), whereas multi-labeling requests additional labels for examples in the seed set (to increase label accuracy). With $2N$ labels, SL covers $2N$ examples with a single label while multi-labeling covers the N examples with two labels each in round-robin fashion. Simple majority vote (**MV**) is used for consensus.

Our prior work [3] on binary classification compared SL to multi-labeling with Naive Bayes (**NB**) [6] as well as MV. We studied effects of modeling vs. ignoring worker accuracy and saw across methods that modeling worker accuracy significantly improved classifier accuracy, indicating a limitation of the oft-used simple majority vote with noisy annotation. While a variety of crowd behaviors and noise may arise in practice, both prior studies [5, 3] assumed assumed uniform noise, as well as each label coming from a unique annotator. Other worker behaviors and noise characteristics may be observed in practice and could be usefully modeled.

Prior work by Yang et al. [7] studied learning to rank (with graded judgments) rather than binary classification, evaluating SL, MV, and other consensus algorithms for ranking with LambdaRank. They assumed labels come from reliable experts and provided limited analysis of the relationship between consensus method and the resulting learning curve.

This paper extends our earlier study from binary classification to learning to rank, and we consider learning under different noise quantities and distributions. We compare SL, MV, and NB for consensus, and we measure resulting ListNet [1] ranking accuracy on three LETOR [4] collections: `OHSUMED`, `MQ2007` and `MQ2008`. We observed similar results across all three and so present results on `OHSUMED` only due to space constraints. We respect LETOR's standard 5-fold partition with 3 training folds and the others for validation and testing. While training labels come entirely from the crowd, we make a significant assumption of having expert labels for the entire validation fold ($\approx$ 3500 examples). Note this validation data is used only by ListNet, not by consensus methods. We also use expert labels as ground truth for evaluation. This reflects a scenario in which more costly expert annotation suffices for validation and testing, but larger volumes of more affordable data is desired for training.

We use a seed set size of $N = 800$ (potentially noisy) singly-labeled examples, reflecting a minimal training size to obtain stable results. The learning curve is then measured as a function of adding L additional labels. For each setting

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

This paper was separately reviewed and accepted to HCOMP 2011 [2]. It is published only in the proceedings of SIGIR'11.

SIGIR'11, July 24–28, 2011, Beijing, China.
ACM 978-1-4503-0757-4/11/07.

L	**0**	**800**	**1600**	**3200**	**6400**	**Avg**
No noise (**SL**)	30.9	33.5	36.4	36.9	38.3	36.3

Distribution	**L**	**SL**	**MV**		**NB**	
		Rank	Rank	Label	Rank	Label
$\mathcal{N}(0.7, 0.2)$	0	21.2				
	800	23.5	27.6	61.8	25.2	62.3
	1600	29.0	27.3	63.6	28.6	75.3
	3200	33.0	27.0	70.2	30.6	90.6
	6400	35.3	26.4	77.0	30.4	96.9
	Average	30.2	27.1	68.1	28.7	81.2
$\mathcal{N}(0.5, 0.2)$	0	21.1				
	800	23.7	24.5	52.6	22.2	54.7
	1600	28.8	28.0	54.2	25.6	66.4
	3200	31.7	24.5	55.1	29.1	81.3
	6400	36.1	26.5	58.4	31.0	90.1
	Average	30.1	25.9	55.0	27.0	73.1
$\mathcal{N}(0.4, 0.2)$	0	17.0				
	800	19.5	22.6	46.8	22.9	49.0
	1600	25.8	24.0	44.7	23.9	61.4
	3200	30.4	22.7	44.4	28.4	76.1
	6400	34.0	21.9	39.5	27.0	85.7
	Average	27.4	22.8	43.85	25.6	68.0
$\ln\mathcal{N}(0.4, 0.2)$	0	18.4				
	800	21.1	22.3	40.6	21.7	41.1
	1600	26.9	23.4	40.1	22.2	40.0
	3200	27.9	19.8	35.8	21.7	37.1
	6400	28.6	19.7	30.5	21.8	36.7
	Average	26.1	21.3	36.7	21.8	38.7
$\mathcal{U}(0.2, 0.6)$	0	17.7				
	800	19.0	21.3	37.4	20.5	38.5
	1600	26.6	21.1	33.8	17.1	35.6
	3200	26.7	18.8	29.6	19.8	35.2
	6400	26.4	17.3	23.9	23.2	32.4
	Average	24.7	19.6	31.1	20.1	35.4

Table 1: Label accuracy and ListNet rank accuracy (%) achieved by SL vs. MV and NB consensus methods for varying L and quantity and distribution annotation noise (normal $\mathcal{N}(\mu, \sigma)$, log normal $\ln\mathcal{N}(\mu, \sigma)$, and uniform $\mathcal{U}(min, max)$. Expected label accuracy for SL is defined by noise parameters (mean μ or $\frac{min+max}{2}$; we report empirical accuracy for MV and NB. L additional labels are added to the seed set of $N = 800$ singly-labeled examples. We also report average accuracy of each method across $L = \{800, 1600, 3200, 6400\}$. We repeat experiments 5 times and average for stability.

of L, we compute consenus labels (no-op for single labeling) and then train ListNet using them. We report label accuracy achieved as well as the resultant ranking accuracy achived by ListNet. We measure this across different noise settings.

We simulate noisy annotation via a fixed-size pool of 100 annotators who select between $C = 3$ possible labels (ternary graded relevance classes: non-relevant, relevant, or highly relevant). Each annotator i has a unique parameter p_i denoting the probability he will produce the correct label for a given example. Otherwise he produces one of the other two possible labels (uniformly) at random. New labels are generated by selecting an annotator i from the pool at random and then generating a label according to p_i as just described.

Results without annotation noise and for five possible noise settings are shown in Table 1. Noiseless ranking accuracy with $N = 800$ $L = 0$ provides an approximate upper-bound for MV and NB consensus results across settings of L since perfect consensus would restore us to the noiseless condition. While level of noise clearly impacts the learning curve (Figure 1), we see relatively little impact of different noise distributions on ranking accuracy. Overall, it seems when average accuracy exceeds 50%, sufficient "good" annotators exist to overcome the noise of their less reliable peers.

Between $N = 800$ and $N = 1600$, **SL** begins to consistently outperform **NB** and **MV** across noise distributions, with greater example coverage apparently more important than label accuracy. Effects here may be task-specific or learner-specific, and having expert validation labels may benefit SL more than MV and NB since SL labeling accuracy on training examples is lowest. We aso see **NB** typically outperform **MV** across noise distributions.

We define an adversarial annotator for multi-class annotation with C classes as one whose $p_i < \frac{1}{C}$. In such cases, a simple way fix is to randomly pick one of the other $C - 1$ classes. We saw little benefit from doing so. Suppose an annotator has accuracy 0.2. Assuming a uniform prior over remaining classes, each has probability 0.4, so not much higher than the class originally labeled. We expect more benefit from handling adversarial labeling when accuracy is extremely low (i.e. strongly anti-correlated), or when we have a better prior for selecting between remaining classes.

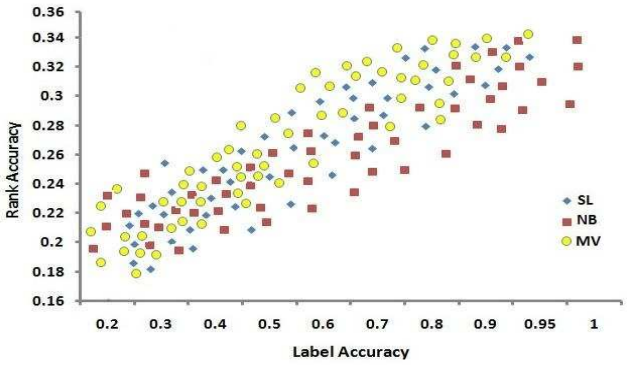

Figure 1: Consensus label accuracy vs. ranking accuracy of the ListNet learner shows a strong linear relationship across consensus methods and noise distributions considered (not shown). This suggests one can simply optimize for label accuracy with confidence of improving rank accuracy as a result.

Future work includes: a similar study with real crowd workers and data, developing more representative models for simulation, studying additional consensus methods and noise settings, and dynamic example selection for labeling.

Acknowledgments. We thank the anonymous reviewers for their valuable feedback. Eunho Yang provided the ListNet implementation. This work was partially supported by a John P. Commons Fellowship for the second author.

2. REFERENCES

[1] Z. Cao, T. Qin, T.-Y. Liu, M. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

[2] A. Kumar and M. Lease. Learning to rank from a noisy crowd. In *3rd Human Computation Workshop (HCOMP) at AAAI*, 2011.

[3] A. Kumar and M. Lease. Modeling annotator accuracies for supervised learning. In *WSDM Workshop on Crowdsourcing for Search and Data Mining*, 2011.

[4] T. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR Learning to Rank Workshop*, 2007.

[5] 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.

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

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