Finding Patterns in Noisy Crowds: Regression-based Annotation Aggregation for Crowdsourced Data

Natalie Parde and **Rodney D. Nielsen**
Department of Computer Science and Engineering
University of North Texas
{natalie.parde,rodney.nielsen}@unt.edu

Abstract

Crowdsourcing offers a convenient means of obtaining labeled data quickly and inexpensively. However, crowdsourced labels are often noisier than expert-annotated data, making it difficult to aggregate them meaningfully. We present an aggregation approach that learns a regression model from crowdsourced annotations to predict aggregated labels for instances that have no expert adjudications. The predicted labels achieve a correlation of 0.594 with expert labels on our data, outperforming the best alternative aggregation method by 11.9%. Our approach also outperforms the alternatives on third-party datasets.

1 Introduction

Publicly-available labeled datasets are scarce for many NLP tasks, and crowdsourcing services such as Amazon Mechanical Turk[1] (AMT) offer researchers a quick, inexpensive means of labeling their data. However, workers employed by these services are typically unfamiliar with the annotation tasks, and they may have little motivation to perform high-quality work due to factors such as low pay and anonymity. To further complicate matters, some workers may produce spam or malicious responses. Thus, it is not uncommon for workers to correlate poorly with one another.

Researchers using crowdsourcing services commonly aggregate the labels they receive via simple strategies such as using the majority or average label. These methods are best suited for simple, straightforward tasks; with noisier data such as that which may be obtained for more difficult or subjective tasks, these strategies may produce skewed labels that misrepresent the instance. Thus, it is desirable to devise more effective aggregation strategies that consider factors such as label distribution and worker quality, while still avoiding manual adjudication of all instances.

In this work, our contributions are as follows: (1) we develop a regression-based method for automatically aggregating crowdsourced annotations of varying quality, with poor agreement and minimal expert-adjudicated data, that addresses multiple potential flaws or biases in non-expert human annotation. To do so, we (2) crowdsource annotations for a difficult NLP task, metaphor novelty scoring, and (3) describe a process by which we automatically detect untrustworthy workers. We then (4) introduce a feature set that captures label distribution and trustworthiness, and extract the features from our crowdsourced annotations. Finally, (5) we train a regression model that predicts aggregated labels for unseen instances and compare the predictions to expert annotations, finding that our method outperforms the best alternative approach. We evaluate our approach both on our data and on existing crowdsourcing datasets. All datasets and source code are available for the research community to improve on our results.[2]

2 Related Work

Several methods have been proposed to identify low-quality workers in crowdsourced data. Jagabathula et al. (2016) filtered adversarial workers in binary labeling tasks by identifying those with outlier labeling patterns, and Lin et al. (2014) identified when additional labels for binary tasks should be crowdsourced to optimize classifier accuracy. Unlike these approaches, our filtering algorithm is suitable for multi-class annotation tasks.

[1] www.mturk.com

[2] Our data can be downloaded at http://hilt.cse.unt.edu/resources.html, and our source code is available at https://github.com/natalieparde/label-aggregation.

Various methods have also been explored as intelligent modes of label aggregation. Most (Snow et al., 2008; Raykar et al., 2010; Karger et al., 2011; Liu et al., 2012; Hovy et al., 2013; Felt et al., 2014; Huang et al., 2015) have built upon the probabilistic item-response model first proposed by Dawid and Skene (1979), which simultaneously estimates annotator quality and aggregated labels using an expectation-maximization algorithm. MACE (Hovy et al., 2013) is a popular implementation inspired by this that aggregates labels as a function of the annotation and a learned binary variable indicating whether the annotator is a spammer. We posit that although annotator quality is an important factor in predicting accurate aggregations, the interplay between it and other factors is more nuanced. Thus, rather than adapting the item-response method, our learning approach incorporates features that address multiple potential flaws or biases in crowdsourced annotations.

Some researchers have also used data-aware approaches to predict aggregations (Raykar et al., 2010; Felt et al., 2014, 2015, 2016). We do not use the data itself in this work, to avoid skewing labels in a way that makes it trivial to learn classifiers based on the same data. To the best of our knowledge, our work is the first to frame label aggregation as a regression task, with features based solely on workers and their labels, that learns entirely from a small amount of expert-adjudicated crowdsourced annotations.

3 Methods

3.1 Data Collection

We evaluated our approach on our new metaphor novelty dataset, as well as on third-party datasets. To build our dataset, we crowdsourced annotations for 3112 potentially metaphoric word pairs, and randomly divided the instances into training (1036), validation (1038), and test (1038) subsets. We developed features and selected our regression algorithm using the training and validation sets only; the test set was withheld until the evaluation.

3.1.1 Annotation Task

Instances were comprised of pairs of words from 1840 sentences in the VU Amsterdam Metaphor Corpus (VUAMC) (Steen et al., 2010). The VUAMC consists of documents for which individual words are labeled as metaphors. The novelty of those metaphors varies widely, from highly conventional to quite novel. Each sentence for which we collected annotations contained a content word (noun, verb, adjective, or adverb) labeled as being metaphoric, and one or more other content words or personal pronouns that were syntactically related to the metaphoric word. Word pairs containing a metaphoric word and a syntactically-related content word or personal pronoun were considered instances. AMT workers ("Turkers") were asked to score each instance on a discrete scale from non-metaphoric (0) to highly novel metaphor (3). Some examples are shown in Table 1.[3]

Example	Score
Alice looked up, and there stood the Queen in front of them, with her arms folded, *frowning* like a **thunderstorm**.	Novel Metaphor (3)
'Once,' said the Mock Turtle at last, with a **deep** *sigh*, 'I was a real Turtle.'	Conventional Metaphor (1)
A large rose-tree **stood** near the *entrance* of the garden: the roses growing on it were white, but there were three gardeners at it, busily painting them red.	Non-Metaphor (0)

Table 1: Sample word pairs provided to Turkers.

Instances were grouped into Human Intelligence Tasks (HITs) containing all instances associated with 10 sentences each. Five worker assignments were requested per HIT, and Turkers were paid $0.20 per HIT. Overall, 237 Turkers annotated 942 assignments, with an average correlation of 0.269 per HIT (the poor agreement suggests this is a very difficult annotation task). An expert adjudicated all 3112 instances; those labels were considered the gold standard.

3.1.2 Data Filtering

Spam and malicious workers were identified during data collection using a filtering algorithm that compared annotations with those completed by "potentially good annotators" (PGA). Alg. 1 describes this process. Letting H_i be a set of HITs collected, A_i be the set of annotators who annotated H_i, and A=$\cup(A_1, \ldots, A_j)$ be the set of all annotators, the algorithm computes three sets of annotators: good annotators (GA), spammers or malicious annotators (Bad Robots, or BR), and annotators of currently unknown quality UQA.

$\text{R}(a_j, a_k)$ computes the correlation coefficient between two annotators a_j and a_k, where a_k is a potentially good annotator whose annotations overlap with a_j's, and $\text{AVG\_R}(a_j)$ computes the average correlation between a_j and all a_k. HITs

[3] Sentences are from Lewis Carroll's *Alice in Wonderland.*

Algorithm 1 Worker Filtering for Annotation Set i

$PGA \leftarrow A \setminus BR$
repeat
  for a_j in A **do**
    $A^j \leftarrow \{a \in PGA\}$ who annotated ≥ 1 unfiltered HIT in common with a_j
    for a_k in A^j **do**
      $r_{j,k} \leftarrow \text{R}(a_j, a_k)$
    $r_j \leftarrow \text{AVG\_R}(a_j)$
  $B^- \leftarrow \{a_j \in A | r_j < 0.0\}$
  $B^0 \leftarrow \{a_j \in A | r_j == 0.0 \text{ or } r_j == \infty\}$[4]
  $B^+ \leftarrow \{a_j \in A\}$, of size $|B^-|$, with the lowest $r_j > 0.0$
  $B^{<.1} \leftarrow \{a_j \in A | r_j < 0.1\}$
  $PGA = A - (B^- + B^0 + B^+ + B^{<.1})$
until convergence or iterations = max
$GA \leftarrow \{a_j \in A | r_j > 0.35\}$
$BR \leftarrow B^- + B^0 + \text{BOTTOM}(\text{ROUND}(\frac{2}{3}|B^-|), B^+)$

completed under a minimum time threshold were also filtered. Following algorithm completion, filtered HITs and unpaid HITs from members of BR were rejected, and annotators in BR were disqualified from accepting future HITs. 116 total assignments were rejected by the filtering algorithm. Annotators in UQA (UQA=$A-GA-BR$) who had completed ≥ 2 HITs and had an $r_j < 0.1$ were also disqualified. All other HITs were accepted.

3.2 Features

We designed features to capture the distribution and trustworthiness of crowdsourced labels for each instance. The features are described in Table 2. ANNOTATIONS are designed to provide the regression algorithm with label distributions based on label value and worker trustworthiness. AVG. R features are intended to further clarify worker quality, and AVG. R (GOOD) is meant to provide a more selective view of the same characteristic. AVG., WEIGHTED AVG., and WEIGHTED AVG. (GOOD) allow the regressor to consider three different versions of a popular aggregation strategy, and finally, HIT R supplies the algorithm with an estimate of agreement on the current instance to consider when making its prediction.

3.3 Regression Algorithm

The approach utilizes a random subspace regressor, which was selected based on its performance on the training and validation data relative to a large variety of other regression algorithms. Random subspace is similar in nature to bagging and random forests, using multiple decision trees constructed from subsets of features selected randomly without replacement to make its predictions (Ho, 1998). We used the implementation from the Weka library (Frank et al., 2016), with Weka's REPTree classifier as the base decision tree model.

[4]Turkers who assigned the same label to every instance, or whose assignments had already been filtered for some other reason (e.g., violating the minimum time threshold).

[5]We also include a second copy of these features ordered by the annotators' average r values.

Feature	Description
ANNOTATIONS	From highest to lowest label, the five annotations for the instance.[5]
AVG. R	For each annotator, in order of label value, his/her avg. correlation with other workers across all instances he/she annotated.[5]
AVG. R (GOOD)	AVG. R in which each annotator is compared only to annotators with r_j>0.35. If the annotator has no overlapping annotations with those, AVG. R is repeated.
AVG.	Average of the five ANNOTATIONS.
WEIGHTED AVG.	Let l_i be the i^{th} ANNOTATION, and r_i be its annotator's AVG. R. Then, WEIGHTED AVG. $= \frac{\sum_{i=1}^{5}(l_i \times r_i)}{\sum_{i=1}^{5} r_i}$.
WEIGHTED AVG. (GOOD)	Similar to WEIGHTED AVG., with weights (r_i) taken from AVG. R. (GOOD) instead of AVG. R.
HIT R	The average weighted correlation among annotators for the HIT containing the instance. Letting $w_{i,j}$ be the weight for a pair of annotators equal to $\frac{r_i+r_j}{2}$, where r_i and r_j are the AVG. R associated with annotators a_i and a_j, $r_{i,j}$ be the correlation between annotators a_i and a_j for the HIT, and P contain all annotator pairs (a_i, a_j) for the HIT, HIT R $= \frac{\sum_{p \in P} r_{i,j} \times w_{i,j}}{\sum_{p \in P} w_{i,j}}$

Table 2: Features used.

	Affect (Emo.)	Affect (Val.)	WebRel	Ours
Instances	600	100	2439	3112
Annotators	38	38	722	237
Annotators / Instance	10	10	5	5
Label Range	0-100	-100-100	0-2	0-3

Table 3: Dataset Details

4 Evaluation

4.1 Other Datasets

In addition to evaluating our approach on our data, we evaluate it on three existing crowdsourcing datasets that differ in terms of their size, noise level, and number of annotators. Details about each dataset are shown in Table 3, with additional information below. Each third-party dataset was randomly divided into 66% training and 34% test.

Affect (Emotion and Valence). Affect (Emotion) and Affect (Valence) were created for Snow et al.'s (2008) work, and contain emotion (*anger*, *fear*, *disgust*, *joy*, *sadness*, and *surprise*) and valence ratings for 100 headlines from the SemEval affective text annotation task (Strapparava and Mihalcea, 2007) test set. Annotations indicate the degree of emotion in an emotion-headline pair (Affect (Emotion)) and the overall positive or negative valence of a headline (Affect (Valence)). Snow et al. report an average correlation among annotators of 0.669 (emotion) and 0.844 (valence).

WebRel. WebRel was originally created for the TREC 2010 Relevance Feedback Track (Buckley et al., 2010), and its annotations indicate the relevance of web documents retrieved for queries. The full dataset contains crowdsourced annotations for 20,232 topic-document pairs; 3277 of those pairs additionally have gold-standard labels. The number of annotations collected per instance varied. We used the subset of instances with gold standard labels and at least five annotations, and reconstructed their HIT groupings based on the workers that annotated each instance (we assumed all instances annotated by the exact same set of workers were originally from the same HIT). Average correlation per HIT was 0.102 (quite noisy).

4.2 Experimental Setup

We compare our approach to a number of alternative methods, detailed with justifications in Table 4. The alternatives are popular aggregation techniques that address different potential flaws in non-expert annotation. We train our approach on the training (and validation, for our dataset) data, and test on the test set. Since MACE (used for *Item-Response*) learns from and outputs predictions for the same data, we provide it with the entire dataset (training, validation if available, and test), but report its results for the test instances only. We provide input to MACE in an n-dimensional sparse matrix (1 row per instance and 1 column per each of n distinct annotators in the dataset, with filled values only for the annotators who provided annotations for that instance), since the approach requires knowledge of which annotator provided each annotation to function properly.[6]

[6]Note: Item-response approaches are better-suited to scenarios in which fewer workers annotate more instances each, but our results would also improve under such circumstances where a worker's trustworthiness, as measured by average r value, is more reliable.

Approach	Description
Majority Vote	The most frequent label given by annotators for the instance. Ties were broken by taking the highest of the tied labels—assumes the most popular opinion should be trusted.
Highest	The highest label for the instance—assumes those who see a metaphor should be trusted.
Item-Response	The prediction expected from an item-response model. We use MACE (Hovy et al., 2013) to generate predictions since it is a well-documented item-response approach that is publicly available online.
Mode Average	The real-valued average of the mode(s) of the instance's labels (if only one mode, this feature is that mode)—assumes popular opinions should be trusted, and equally popular opinions are equally trustworthy.
Average	The average of all five labels—assumes each annotator's opinion is equally valid.
Rule-Based	Assigns a value of 0 if 4+ annotators labeled the instance as such; otherwise, takes the avg. non-zero label—assumes annotators frequently miss tricky or subtle instances.

Table 4: Alternative Approaches.

We also evaluate the performance of different feature subsets on our data. *All−Averages* contains all features except for AVG., WEIGHTED AVG., and WEIGHTED AVG. (GOOD). Each other subset contains all features except for the respective feature type noted from Table 2. The correlation coefficient (r) and root mean squared error (RMSE) were recorded for each test condition since our estimator produced continuous-valued scores. Since *Mode Average*, *Average*, and *Rule-Based* result in continuous values and *Majority Vote*, *Highest*, and *Item-Response* result in discrete values, we present two versions of our results; in one, predictions were rounded to the nearest integer (forcing a 0, 1, 2, or 3) and in the other, they were left as-is. For the discrete approaches on our data, we also report accuracy.

4.3 Results

The results are presented in Tables 5, 6, and 7. Table 5 compares our method with each alternative approach on our data, and Table 6 compares our method with the alternatives on each third-party dataset. Table 7 shows the results of the feature ablation. On our dataset, our approach outperformed all other approaches, with r = 0.594 with the gold standard and RMSE (0-3) = 0.605. This represented correlation improvements of 18.6%, 11.9%, and 69.2% relative to the continuous alternative approaches (*Mode Average*, *Average*, and *Rule-Based*, respectively). The

Method	*r*	RMSE	Acc.
Majority Vote	0.443	1.011	0.536
Highest	0.295	1.701	0.183
Item-Response	0.362	1.083	0.483
Ours (Rounded)	**0.490**	**0.690**	**0.600**
Mode Average	0.501	0.836	—
Average	0.531	0.743	—
Rule-Based	0.351	1.126	—
Ours (Continuous)	**0.594**	**0.605**	**—**

Table 5: Comparison with alternative methods.

	Method	*r*	RMSE
Affect (Emotion)	Majority Vote	0.510	23.2
	Highest	0.416	52.4
	Item-Response	0.526	21.8
	Ours (R)	**0.578**	**16.6**
	Mode Average	0.506	21.9
	Average	**0.613**	16.7
	Rule-Based	0.462	26.5
	Ours (C)	0.578	**16.6**
Affect (Valence)	Majority Vote	0.423	50.1
	Highest	0.573	75.3
	Item-Response	0.483	46.0
	Ours (R)	**0.938**	**18.4**
	Mode Average	0.644	37.4
	Average	0.926	22.4
	Rule-Based	0.913	19.7
	Ours (C)	**0.938**	**18.4**
WebRel	Majority Vote	0.325	1.0
	Highest	0.219	1.2
	Item-Response	0.385	0.9
	Ours (R)	**0.412**	**0.8**
	Mode Average	0.350	0.9
	Average	0.372	0.8
	Rule-Based	0.282	0.9
	Ours (C)	**0.523**	**0.7**

Table 6: Comparison on third-party datasets.

rounded predictions also outperformed all discrete alternatives (*Majority Vote*, *Highest* and *Item-Response*) with relative correlation improvements of 10.6%, 66.1%, and 35.4%, respectively. All approaches had strong positive statistically significant ($p<<0.0001$) correlations and the improvement of our results over the alternatives was statistically significant ($p<<0.0001$).

On WebRel and Affect (Valence), our approach outperformed all other approaches for both the discrete and continuous conditions. On Affect (Emotion), our approach outperformed all alternatives for the discrete condition and had a lower RMSE than all other approaches for the continuous condition (relative reductions in error to RULE-BASED, AVERAGE, and MODE AVERAGE were 37.4%, 0.6%, and 24.2%, respectively), but the predictions from AVERAGE correlated better with the gold standard than did those of our approach.

	Rounded		Continuous	
Feature Set	***r***	**RMSE**	***r***	**RMSE**
All	0.490	**0.690**	**0.594**	**0.605**
All−Annotations	0.440	0.716	0.557	0.627
All−Avg. R	0.480	0.701	0.581	0.611
All−Avg. R (G.)	**0.494**	0.692	0.582	0.611
All−Averages	0.465	0.703	**0.594**	0.607
All−HIT R	0.486	0.693	0.587	0.608

Table 7: Feature subset performance comparison.

Interestingly, Table 7 shows that the discrete version of our approach performed slightly better when the features indicating annotators' correlations with good annotators were removed; this was not the case for the continuous-labeled version. The raw annotations themselves were the most valuable features for both cases. Their removal led to a correlation reduction of 10.2% (rounded) and 6.2% (continuous) relative to using all features.

The results suggest that our approach is a suitable means of automatically aggregating noisy crowdsourced labels, and that reasonable results can be obtained even when training on only a small amount of expert-adjudicated instances. Further, the performance of the alternative approaches suggests that typical aggregation techniques may be less suitable for tasks with many workers who completed relatively few annotations.

5 Conclusion

In this work, we present a regression-based aggregation method that addresses multiple potential flaws or biases in non-expert human annotation. We show that the predictions from our approach correlate at r=0.594 with expert adjudications for a noisy, difficult task, outperforming the best alternative approach by 11.9% on our data and by up to 63.7% on third-party crowdsourcing datasets. This improvement shows that a learning approach can overcome some of the challenges faced by simple label aggregation techniques for these types of tasks. Our data and source code is publicly available for further research by others.

Acknowledgments

This material is based upon work supported by the NSF Graduate Research Fellowship Program under Grant 1144248, and the NSF under Grant 1262860. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Chris Buckley, Matthew Lease, Mark D Smucker, Hyun Joon Jung, Catherine Grady, et al. 2010. Overview of the trec 2010 relevance feedback track (notebook). In *The Nineteenth Text Retrieval Conference (TREC) Notebook*, pages 88–90. Association for Computational Linguistics.

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28.

Paul Felt, Robbie Haertel, Eric Ringger, and Kevin Seppi. 2014. Momresp: A bayesian model for multi-annotator document labeling. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Paul Felt, Eric Ringger, Jordan Boyd-Graber, and Kevin Seppi. 2015. Making the most of crowdsourced document annotations: Confused supervised lda. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 194–203, Beijing, China. Association for Computational Linguistics.

Paul Felt, Eric Ringger, and Kevin Seppi. 2016. Semantic annotation aggregation with conditional crowdsourcing models and word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1787–1796, Osaka, Japan. The COLING 2016 Organizing Committee.

Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. The weka workbench. In *Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, fourth edition. edition. Morgan Kaufmann.

Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with mace. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Ziheng Huang, Jialu Zhong, and Rebecca J. Passonneau. 2015. Estimation of discourse segmentation labels from crowd data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2190–2200, Lisbon, Portugal. Association for Computational Linguistics.

Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. 2016. Identifying unreliable and adversarial workers in crowdsourced labeling tasks. *Journal of Machine Learning Research*, 17(1).

David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1953–1961. Curran Associates, Inc.

Christopher H Lin, Daniel S Weld, et al. 2014. To re (label), or not to re (label). In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2014)*.

Qiang Liu, Jian Peng, and Alexander T Ihler. 2012. Variational inference for crowdsourcing. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 692–700. Curran Associates, Inc.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*, volume 14. John Benjamins Publishing.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.