Improving Consensus Accuracy via Z-Score and Weighted Voting

Hyun Joon Jung
Dept. of Electrical and Computer Engineering
University of Texas at Austin
hyun@mail.utexas.edu

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

Abstract

Using supervised and unsupervised features individually or together, we (a) detect and filter out noisy workers via Z-score, and (b) weight worker votes for consensus labeling. We evaluate on noisy labels from Amazon Mechanical Turk in which workers judge Web search relevance of query/document pairs. In comparison to a majority vote baseline, results show a 6% error reduction (48.83% to 51.91%) for graded accuracy and 5% error reduction (64.88% to 68.33%) for binary accuracy.

Introduction

The Cranfield paradigm for evaluating Information Retrieval (IR) systems (Cleverdon 1997) depends on human judges subjectively assessing documents for topical relevance. While recent advances in stochastic evaluation methods are reducing the number of such assessments needed for reliable evaluation (Carterette, Allan, and Sitaraman 2006; Yilmaz, Kanoulas, and Aslam 2008), human judging remains expensive and slow. Moreover, as we evaluate over ever-larger document collections, keeping pace with today's growing collection sizes, traditional expert judging of large document pools has simply become intractable for practical evaluation of systems. Consequently, crowdsourcing has been much welcomed as a new avenue for alleviating these existing limitations (Alonso, Rose, and Stewart 2008).

Given the subjective nature of relevance judging, even experts may show under 50% agreement (Voorhees 1998). Despite these low agreement levels, Voorhees and others have shown consistency of IR system rankings when evaluated against judgments from different experts. Nonetheless, the possibility that crowdsourcing may yield even lower agreement rates between judges is a cause for significant concern in proving reliability of crowdsourcing for IR evaluation.

A common strategy for improving quality is to request a plurality of judgments for the same example in order to arrive at a single label via a *consensus* method, such as simple majority vote (SM) or a more sophisticated strategy (Sheng, Provost, and Ipeirotis 2008). Intuitively, if we have a single perfect worker, he is all we need, and if all workers are totally unreliable, adding more will not help (Kumar and Lease 2011). The sweet spot for repeated labeling lies between these extremes, where more labels from good but imperfect workers increasingly cancel out noise and bias to improve consensus accuracy. While research on consensus methods has often focused on binary classification tasks with simulated studies, more recent work has considered multi-class judgments with real crowd annotations (Ipeirotis, Provost, and Wang 2010). *Spammers*, who demonstrate very low accuracy, represent a particular threat to consensus accuracy if not detected and handled in some fashion.

In this paper, we report a large-scale consensus study on roughly 20K labels from 766 Mechanical Turk[1] workers who judged ClueWeb09[2] Web pages for topical relevance to different search topics. Z-score[3], a popular measure for outlier detection, was used to filter out noisy workers. We compute the Z-score using various supervised and unsupervised features, individually and in combination. We also evaluate use of features to weight votes for consensus labeling. Results show improvement over both SM and Expectation-Maximization (EM) baselines (Dawid and Skene 1979).

Experiment

Data. Workers were provided a TREC[4] format *title*, *description*, and *narrative* for each search topic using a prebuilt judging interface (Grady and Lease 2010). Relevance was judged on a 0-2 scale indicating non-relevant, relevant, and highly relevant categories. Workers labeled 3,277 query/document examples with a total of 19,232 labels (with ≈ 6 labels per example). All examples had prior TREC expert judgments: 1,501 non-relevant, 863 relevant, and 913 strongly relevant. As an additional safeguard, we also randomly inserted additional 1,183 broken links to be judged, with an explicit judgment option for workers to mark these.

Features. We computed seven features for each worker:

1. graded accuracy vs. gold (GACG)
2. binary accuracy vs. gold (BACG)
3. graded accuracy vs. majority vote (GACM)

[1] https://www.mturk.com
[2] http://lemurproject.org/clueweb09
[3] http://en.wikipedia.org/wiki/Standard_score
[4] http://trec.nist.gov

4. binary accuracy vs. majority vote (BACM)
5. graded distance vs. gold (GDSG)
6. graded distance vs. majority vote (GDSM)
7. binary accuracy vs. broken-links (AHNP).

Whereas *graded accuracy* compares ternary accuracy, *binary accuracy* conflates relevant and highly relevant categories, distinguishing only relevance vs. non-relevance. *Distance* indicates average distance (unsigned magnitude) of a worker's labels to the reference label. For all three feature types, we compare worker label vs. expert label (supervised) and vs. majority vote label between workers (unsupervised). AHNP is the only feature computed on broken-link examples. All features are normalized to $[0, 1]$: accuracy features by definition, and distance features explicitly.

Worker filtering. Let $f^i_{1:n}$ denote feature i's values for all n workers. Let μ^i and σ^i denote the mean and standard deviation for $f^i_{1:n}$. Worker w_j's Z-score z^i for feature i is defined by: $z^i_j = \frac{f^i_j - \mu^i}{\sigma^i}$. For each feature i used, any worker w_j whose Z-score $z^i_j > \gamma$ is filtered out, where γ is a parameter tuned by linear sweep over $\gamma = [0.1, 4.0]$ (by 0.1). When multiple features are used, we filter separately for each and take the intersection of remaining workers across features. We use 5-fold cross-validation for both tuning and testing.

Voting. We compute consensus labels via simple majority vote (**SM**), or feature-weighted voting with a single feature (**SWM**) or multi-featured (**MWM**). Let $f' \subseteq f^{1:7}$ denote the features used, and let $\lambda_j = \prod_{f \in f'} f_j$ be worker w_j's weight. Let $\{y^k_j == c^i\}_{0,1}$ indicate if worker w_j's label y^k_j for example x^k is category c^i. We then predict x_k's category $\hat{c_k}$ by:

$$\hat{c_k} = \underset{i}{\operatorname{argmax}} \sum_{j=1}^{n} \lambda_j \{y^k_j == c^i\}_{0,1} \quad (1)$$

Results. Figure 1 shows six columns: graded and binary accuracy for each voting method. Rows evaluate the impact of different feature combinations. The "Baseline" evaluates each voting method without any Z-score worker filtering. Entries show both accuracy achieved and the associated threshold parameter γ (with typical range $\gamma \in [0.5 - 2.0]$).

Not shown in Figure 1, we also evaluated EM as an additional baseline for binary accuracy (only). On a nearly-equivalent example set (using all but 200 of the 19K labels), EM achieved 0.666 vs. SM achieving 0.639.

To understand the impact of different feature combinations, we evaluate individual features, feature pairs, ablation configurations (all but one feature), and the entire feature set. The most effective feature sets are shown. Individually, the most effective features were GACG (Feature 1), GDSG (Feature 5), and AHNP (Feature 7).

Regarding voting methods, multi-feature weighted majority voting performed slightly better than single feature weighted majority voting, achieving 6% relative error reduction (48.83% to 51.91%) vs. baseline for graded accuracy and 5.32% (64.88% to 68.33%) for binary accuracy.

	GA +SM	BA +SM	GA+SWM	BA+SWM	GA+MWM	BA+MWM
Baseline	0.4751	0.6549	0.4822	0.6402	0.4883	0.6488
f1(GACG)	0.4867(0.5)	0.6597(0.4)	0.5041(0.7)	0.6747(1.5)	0.5072(1.5)	0.6811(1.5)
f2(BACG)	0.4944(0.5)	0.6692(0.5)	0.4965(0.5)	0.6769(1.0)	0.5038(1.0)	0.6781(1.0)
f3(GACM)	0.4751(2.0)	0.6546(2.0)	0.4940(0.6)	0.6674(0.7)	0.5066(1.0)	0.6704(1.0)
f4(BACM)	0.4751(3.0)	0.6561(1.8)	0.4901(0.8)	0.6689(0.8)	0.5050(0.8)	0.6750(0.8)
f5(GDSG)	0.4883(0.6)	0.6622(0.6)	0.4995(1.3)	0.6744(1.3)	0.5072(1.3)	0.6805(1.3)
f6(GDSM)	0.4764(1.6)	0.6549(2.2)	0.4986(0.9)	0.6704(1.0)	0.5063(1.1)	0.6708(1.1)
f7(AHNP)	0.5035(1.6)	0.6823(1.7)	0.4974(1.7)	0.6457(1.7)	0.4999(1.6)	0.6515(1.7)
f1+f7	**0.5176(0.6)**	0.6860(0.7)	**0.5194(0.6)**	0.6790(1.3)	**0.5191(1.5)**	**0.6833(1.5)**
f1+f5	0.5075(0.3)	0.6619(0.6)	0.5011(0.4)	0.6744(1.3)	0.5069(1.5)	0.6805(1.5)
f5+f7	0.5151(0.6)	**0.6875(1.1)**	0.5172(1.1)	**0.6793(1.2)**	**0.5191(1.2)**	0.6827(1.3)
ALL	0.5044(1.8)	0.6830(2.0)	0.5154(1.1)	0.6781(1.5)	0.5179(1.3)	0.6811(1.9)

Figure 1: Consensus label accuracy using Z-score worker filtering (with different features) vs. different voting schemes. Bold result indicates best accuracy in the given column. Entries also show the threshold parameter value (γ) used. SM, SWM, and MWM refer to the different voting schemes.

Conclusion

We described a set of supervised and unsupervised features, used individually and in combination, for worker filtering and weighted voting. On a large collection of Web relevance judgments, results showed that filtering with multi-feature weighted voting improved consensus accuracy by 3.08% absolute for graded accuracy and 3.45% for binary accuracy.

Acknowledgments

. We thank the anonymous reviewers for their valuable feedback. Wei Tang performed the EM evaluation. This work was partially supported by an Amazon Web Services grant, and a John P. Commons Fellowship for the second author.

References

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

Carterette, B.; Allan, J.; and Sitaraman, R. 2006. Minimal test collections for retrieval evaluation. In *SIGIR*, 268–275.

Cleverdon, C. 1997. The cranfield tests on index language devices. *Readings in Information Retrieval* 47–59.

Dawid, A., and Skene, A. 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.

Grady, C., and Lease, M. 2010. Crowdsourcing document relevance assessment with mechanical turk. In *NAACL-HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 172–179.

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *KDD Workshop on Human Computation (HCOMP)*.

Kumar, A., and Lease, M. 2011. Modeling annotator accuracies for supervised learning. In *WSDM Workshop on Crowdsourcing for Search and Data Mining*, 19–22.

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

Voorhees, E. 1998. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Information Processing and Management*, 315–323.

Yilmaz, E.; Kanoulas, E.; and Aslam, J. A. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR*, 603–610.