Exploiting Ontological Knowledge in Crowdsourcing

Grant Strimel
Computer Science Department
Carnegie Mellon University
gstrimel@andrew.cmu.edu

Ivan Stelmakh
Computer Science Department
Carnegie Mellon University
stiv@cs.cmu.edu

Abstract

We present and analyze techniques for leveraging ontology structure when performing crowdsourced data aggregation in an approval voting setting. We present both theoretic and experimental results as well as giving efficient practical algorithms with strong guarantees. We demonstrate that there are several independent approaches for utilizing the ontology and that all of these approaches can be used in conjunction with one another.

1 Introduction

Platforms such as Amazon Mechanical Turk are widely used for various crowdsourcing applications. Researchers and engineers use data gathered from human workers and use it in an aggregated form to build machine learning models. One of the most popular applications is that of multi-class image classification. This is especially useful in Computer Vision where researchers often require large amounts of precisely labeled data to train classification models (e.g. Neural Networks). For example, if we wish to build a system that identifies which traffic signs are pictured in an image, we would likely require hundreds or thousands of examples of *correctly* labeled image-signs pairs where each image is tagged with a set of road signs appearing in the image.

For applications, data in this form is typically gathered in a multi-class/multiple-choice paradigm where each worker is presented a series of tasks. For each task (e.g. labeling a specific image), a set of possible choices that can be "approved" are presented. There is an extensive corpus of literature that studies the problem of multi-class classification in crowdsourcing settings. For example, [KOS13] study the problem of eliciting the right labels from noisy answers of workers in the setting when the worker is asked to approve only one alternative. Another direction in the literature studies the way the question should be asked. For example, [VVV14] proposes to decompose the problem of multi-class classification into series of simple binary class problems.

In what follows, we consider another way to tackle the problem of multi-class classification where there is a known structure relating the alternatives we can utilize. Let us briefly describe the key observations that motivate of our approach. First of all, as pointed out in [SZP15], it is very restrictive to ask a person to select only one class. Indeed, as studied in [CMW56], people tend to firstly cross out the alternatives that they believe are wrong and then guess from what remains. This implies that if one is forced to select only one answer and is unsure which is correct, it increases the chance he/she will select the wrong alternative. In contrast, if one is asked to approve all the alternatives that he/she believes might be the right answer, then it is more likely that at least one of the approved alternatives will be correct.

Secondly, there is an interesting line of research in psychology that studies categorization and generalization. As summarized in [FGM09], the ability of a person to correctly categorize objects in coarse categories often significantly exceeds the ability to identify the finer classes of the objects. As a result, one might infer that people in their mind first do a coarse categorization before deciding between

Final project report for course CMU 10709 "Fundamentals of Learning from the Crowd", Fall 2017. Instructor: Nihar B. Shah, TA: Ritesh Noothigattu. The formatting style file is borrowed from the NIPS conference.

finer labels that belong to that category. For example, we can consider multi-class classification in crowdsourcing settings. If a worker is given the object and asked to determine to which class the object belongs, he/she may first categorize all the possible alternatives by similarity, eliminate categories which do not apply, then do finer selection. Using this heuristic might be easier than directly selecting the true class out of a large group of alternatives.

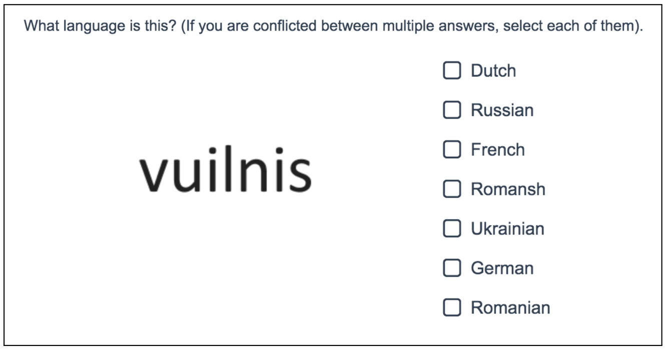
Finally, we note the following observation about human behavior when performing object recognition, "visual processing for object recognition typically proceeds in a coarse-to-fine way, with initial coarse or general processing being followed by fine or detailed processing" [EK15]. Since this way of perception is biologically innate, one may assume that workers follow this way of processing when trying to label the image.

Motivated by these considerations, in this work we investigate using a known categorization of the alternatives, which we call an *ontology*, to help in the in the process of labeling images via crowdsourcing.

2 Problem Formulation

2.1 Setting

In our setting, we have a set of n questions of similar nature which have the same set of d proposed alternatives. We have k crowdsourcing workers. In each question, a worker is given an image object and is instructed to select every alternative he/she believes the object may belong to. Let $x_{ij} \in \{0,1\}^d$ denote worker i's response to question j where x_{ij}^u is 1 if the u-th alternative is selected and 0 otherwise. To make things concrete, our experimental setting will consider the task of language determination. Similar to the example in [SZP15], we ask workers to identify which language appears in an image for a set of seven languages: Dutch, French, German, Romanian, Romansh, Russian, and Ukrainian. An example of this type of question is presented in Figure 1.



What language is this? (If you are conflicted between multiple answers, select each of them).

vuilnis

□ Dutch
□ Russian
□ French
□ Romansh
□ Ukrainian
□ German
□ Romanian

Figure 1: Example question.

We assume that there exists some ontology between the alternatives, i.e alternatives are related to one another in some fashion. Furthermore, we assume this ontology is known to us. For our setting we will consider simple *graph ontologies*. Namely, we restrict our attention to a binary notion of similarity: alternatives/concepts are vertices and if two alternatives are "similar", an edge exists between their corresponding vertices. We denote the ontology graph as $G = (V, E)$. See that $|V| = d$. We have selected a working example such that the notion of ontology (similarity) is unambiguous: we treat two languages as similar if and only if they come from the same language group.

- **East Slavic**: Russian, Ukrainian
- **Romance**: French, Romanian, Romansh.
- **West Germanic**: Dutch, German

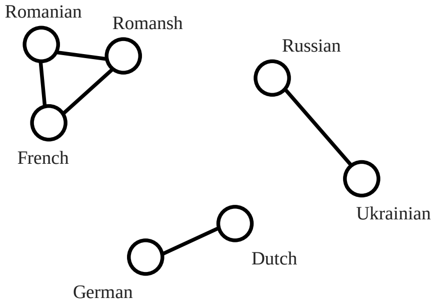


Figure 2: Ontology Graph for Languages.

For our approach, it will also be convenient to consider the complement ontology graph $G' = (V, E')$ where an edge exists in E' if and only if it does not appear in E.

2.2 Proposed Idea

We now give our general approach on how to exploit the ontology in an approval voting setting. We suggest three lines of attack where the prior knowledge of the ontology structure can be exploited.

First, the ontology gives us a natural notion of an *unlabeled error rate*. Since the true labels for each question are not known ahead of time (that would defeat the purpose of crowdsourcing), it might be difficult to evaluate the strength of each worker. However, the ontology gives us a natural proxy for evaluating a workers strength. Those workers that commonly violate the ontology we can designate as weak while those who answer more consistent with the ontology are more likely to be strong. We note a similar observation is used in [BBM13] to relate an unlabeled error rate to a machine learning model's labeled error rate. For us however, we define the unlabeled error rate of a worker i to be

$$\text{err}_{unl}(i) = \frac{1}{n} \sum_{j=1}^{n} \sum_{(u,v) \in E'} x_{ij}^{u} \cdot x_{ij}^{v}$$

In effect, our unlabeled error rate of a worker is a normalized count of the number of edges of G' that are covered by the vertices of the worker's answers. Our goal is to use high unlabeled error rates as a method for identifying spammers. We show how to do this in a simple hammer-spammer setting in Section 3.

Secondly, we can also use our unlabeled error rate to "weight" workers. It is a standard crowdsourcing aggregation technique to take a majority vote over worker answers for each question and declare the true answer of the question to be the vote winner. In paradigms where there strength of the workers are known, it common to use a weighted majority vote where more accurate workers have more "voting power" than the less accurate workers. In these scenarios, the weighted majority vote can far surpass the performance of traditional majority vote. We explore this possibility experimentally in Section 5.2 by considering a voting scheme that employs the unlabeled error rate.

Last, we can use our ontology to direct our data collection by informing the workers in advance of the ontology structure/coarse categorization of alternatives. Along with the general intuition described in Section 1, we note that it is easy to imagine a scenario when a worker has no knowledge about one or more of the proposed alternatives. For example, say there are three alternatives, A, B and C and a worker is familiar with alternatives A and B but completely unfamiliar with alternative C. If the worker is told that C is similar to A, then he/she can use that information to help reduce the number of mistakes on questions for which he/she is not sure of the answer. On a question where the true answer is C and the worker recognizes the object as similar to that of an A object but is not 100% sure, the worker will be more inclined to choose both A and C. On a question where the true answer is B and the worker recognizes the object as a B type object but is not 100% sure, the worker will be more inclined to choose just B and ignore C.

A convenient aspect of these three approaches is that none are mutually exclusive. All of the approaches can be used in conjunction simultaneously. In the remainder of this work, we analyze the proposed ontology exploitation from both theoretical and experimental contexts.

3 Hammers and Spammers

In this section, we consider an analysis of a hammer-spammer setting for general ontology graphs under a restricted idealized class of hammer and spammer workers. We define the class of q-spammers as workers who select each alternative with equal probability q. Namely,

$$x^u_{\text{spammer},j} = \begin{cases} 1 & \text{with prob } q \\ 0 & \text{with prob } 1-q \end{cases}$$

for all alternatives u and questions j. We suggest that this is a reasonable model for a spammer's behavior due to its simplicity of implementation.

We define a p-hammer worker as follows. For any question j with believed correct answer u, the number of alternatives v approved such that $(u, v) \in E'$ is distributed in a geometric fashion with parameter p. Namely, $\ell > 0$ alternatives v are approved where $(u, v) \in E'$ with probability upper-bounded by p^ℓ; the event where no alternatives $v \in E'$ are approved occurs with probability at least $1 - \frac{p-p^{\Delta'}}{1-p}$ where Δ' is the maximum degree of G'. We grant this an idealized model but we contend its adequacy by the behavioral properties of the probability dropping exponentially in the number of "ontology mistakes".

With these classes of spammers and hammers, and a lower bound q_{min} on all parameters q, we present a simple algorithm for doing classifying spammers and hammers. Note Δ is the maximum degree of G.

HAMMER-SPAMMER CLASSIFICATION ALGORITHM

- From the observed data, compute $\text{err}_{unl}\,(i)$ for all workers i.
- If $\text{err}_{unl}\,(i) < 2(\Delta + 1) + \frac{|E'|q^2_{\text{min}}}{2}$, label the worker a hammer, else label the worker a spammer.

Theorem 3.1. *If all p-hammers have a parameter $p \le 0.5$ and all q-spammers have $q \ge q_{min}$ with $q_{min} \ge c\sqrt{\frac{\Delta+1}{|E'|}}$, $c > 2\sqrt{2}$, then the Hammer-Spammer Classification Algorithm will classify each individual worker correctly with probability at least $1 - \exp\left(-\left(\frac{1}{2} - \frac{4}{c^2}\right) nq^4_{min}\right)$.*

From the above theorem, the following corollary comes as a consequence.

Corollary 3.1.1. *If the conditions of Theorem 3.1 are satisfied, then the Hammer-Spammer Classification Algorithm will correctly classify at least a $(1 - \epsilon)$ fraction of workers with probability $(1 - \delta)$ if $n \ge \frac{1}{\left(\frac{1}{2}-\frac{4}{c^2}\right)q^4_{min}} \ln\left(\frac{1}{\epsilon\delta}\right)$.*

Note that the sample complexity given in the above corollary is that it is independent of the number of workers k . The proofs of these statements are given in Appendix A.

4 Experiment

4.1 Design

To test the effects of the ontological approach, we conducted an experiment with real human responses. We designed the experiment to mainly explore the three uses of the ontology: finding spammers, proxy-weighting workers, and directing workers by teaching them the ontology. As mentioned in Section 2.1, our experimental example will be that of language identification. We present the worker with a task to identify which of seven languages is shown in an image and ask the worker to select all the languages he/she believes the word may belong to. We selected the language alternatives in a specific way that they form three distinct groups. Languages inside one group are similar to each other. In this case, by similarity we mean the similarity of alphabets, character patterns and lexical similarity - two languages share words in common. The proposed languages, groups and justification are the following:

- **Russian, Ukrainian** (East Slavic). This group should be easily identified by the vast majority of workers. We believe that when the worker encounters a word from one of these languages he or she will select either one of these languages or both.
- **French, Romanian, Romansh** (Romance). We expect that people generally are not familiar with the Romansh language and the connection between Romanian and French. So being presented the ontology in advance may aid in the classification involving these languages.
- **Dutch, German** (West Germanic). Here we again expect that workers are not as familiar with the Dutch language compared to German and that the connection between the two might be non-obvious.

The data was collected in a survey format. Each worker was presented a series of $n = 45$ questions. Each question contains an image of a word and a list of seven checkboxes, one for each language. See Figure 1.

The experiment was conducted on Amazon Mechanical Turk in conjunction with SurveyGizmo. We use Mechanical Turk as the platform for assigning surveys and compensating workers. Mechanical Turk directs, via a hyperlink, the worker to the list of questions presented by SurveyGizmo. Upon completion of the questions, a survey code is presented to the worker to enter back into Mechanical Turk. The use of SurveyGizmo was chosen because there are many useful built-in features which reduced development effort. These include survey design formatting, recording, reporting, randomization, and A/B testing.

In order to ensure statistical validity, we implemented the following features into the experiment design. The order of the questions is randomized for every worker to avoid biases. Additionally, the order of the options is randomized for every worker to avoid biases. Now we describe below how our experiment addresses each of the ontology uses.

1. To test our ability to catch spammers, after real human data was collected, we augment our data with generated spammer data. We try to limit our true responses from the survey to known humans by taking advantage of Mechanical Turk Features. Namely, we put the following filters on our Turk HIT. Each worker must have at least a 98% HIT approval rating with over 500 HITs completed. Also, we require workers to be *Masters* - a classification which Mechanical Turk tracks and assigns internally using their own statistical modeling. Last, we have a condition applied through SurveyGizmo that every question must have at least one alternative selected. Afterwards we add our generated spammer data according to the q-spammer class. We describe this in more detail in Section 5. We then use the algorithm presented in the previous section to identify spammers.
2. With the results gathered from the experiment, we use our ontology to evaluate each workers unlabeled error. We are then able to use the unlabeled error rate to weight each of our workers to perform a weighted majority vote and compare this against the unweighted version. Note we do not include our generated spammer data.
3. To determine whether or not informing the workers of the ontology before answering the questions alters worker performance, each participant is randomly assigned to one of two groups where the instructions are modified in the second group to inform them of the language classes. We denote the control group as Group A and the test group with the modified instructions as Group B. See the Appendix B for figures displaying different instructions. To measure the effect of this, we define special metric which we refer to as the γ-score. Letting u_j be the correct alternative for option j, then we define the γ-score for worker i as

$$\gamma\text{-score}\,(i) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\left\{x_{ij}^{u_j} = 1\right\} \prod_{v:x_{ij}^{v}=1, v\neq u_j} (1-\gamma)^{\mathbb{1}\{(u_j,v)\in E\}}(\gamma)^{\mathbb{1}\{(u_j,v)\notin E\}}.$$

Intuitively, the γ-score captures a combination of labeled and unlabeled error. For a particular question, if the worker does not select the correct answer they receive a score of zero. On those questions which the correct answer was selected, a small multiplicative penalty is applied for every alternative selected agreeing ontologicaly with the correct answer and a large multiplicative penalty is paid for alternatives selected disagreeing ontologicaly with the correct answer. We will test how the γ-score changes in each of the groups.

4.2 Data Collection

Data was collected on Amazon Mechanical Turk over a 48 hour period from November 20 - November 22, 2017. The response count breakdown is given Table 4.2. The average response times were within the 10 - 14 minute range over the 45 questions. We present our full data analysis on the gathered data in Section 5.

	Group A	Group B	Total
Response Count	89	73	162

Table 1: Response Counts.

5 Results and Analysis

We advance the results collected from the experiment and analyze the data according to each perspective of the ontological uses.

5.1 Finding Spammers

For the task of diagnosing spammer behavior, we augment our data sets by adding a supplemental synthetic data set of 100 q-workers. Each spammer is added with q parameter generated uniformly at random in the range $[1/2, 1]$. For each group, Group A, Group B, and Spammers, we now plot the the sample size n against the fraction of correctly classified workers. To simulate smaller sample sizes $n < 45$ for Group's A and B we randomized the permutation of questions and consider just the first n questions.

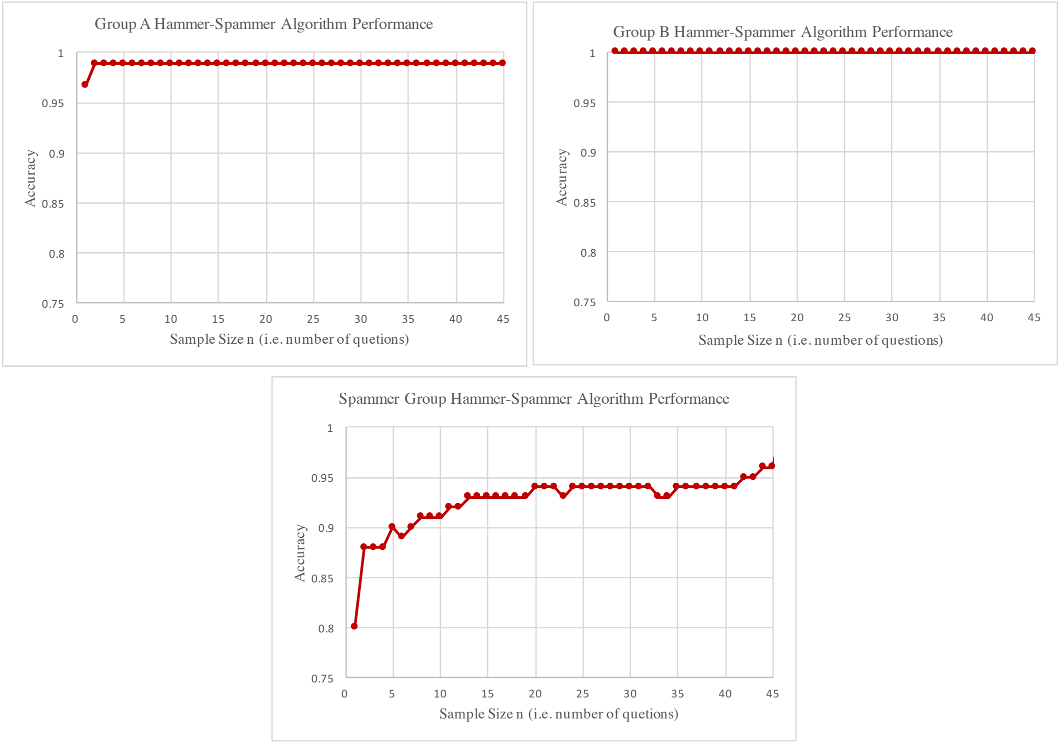


Figure 3: HAMMER-SPAMMER CLASSIFICATION Alg Performance.

It is apparent from Figure 3 that the HAMMER-SPAMMER CLASSIFICATION Algorithm does extremely well in identifying the hammers with small sample size but in our simulation, the algorithm requires a higher sample size to catch the spammers.

5.2 Weighting Workers

As alluded to in Section 2.2, we can use the unlabeled error as a proxy for determining the strength of each worker. If we knew the accuracy of each worker, we could perform a weighted majority vote in an attempt to outperform unweighted majority vote. A classic weighting scheme given worker accuracies w_i for majority vote is the log-odds voting scheme [BK14]:

$$\text{answer reported for question } j = \underset{u\in[d]}{\operatorname{argmax}} \sum_{i=1}^{k} \ln \frac{w_i}{1-w_i} x_{ij}^u$$

Since we do not know the true w_i of each worker i, we adapt our weighted voting scheme by setting

$$w_i = 0.95 - 0.45 \frac{\text{err}_{unl}(i) - \min_{i'} \text{err}_{unl}(i')}{\max_{i'} \text{err}_{unl}(i') - \min_{i'} \text{err}_{unl}(i')}$$

as a heuristic. Basically this sets the weight for those workers who are the most consistent with the ontology as $w_i = 0.95$ and those who are the least consistent with $w_i = 0.5$, and all else have a linear interpolation between the extremes. For both Group A and Group B, we took the majority vote over the 45 questions and compared it to the accuracy of the heuristic weighted majority vote. The results are given in Table 5.2 below.

We see that there is an improvement with the proxy weighted majority vote over the traditional majority vote but the improvement is only marginal.

	Majority Vote	Weighted Majority Vote
Group A	25	26
Group B	22	24

Table 2: Comparison of Number of Correct Answers Given by Different Voting Schemes

5.3 Informing Workers

Now we present our analysis on the effect of informing the workers of the ontology/categorization before answering the questions. Our main method for this analysis is to evaluate the differences in behavior of the γ-score across our two groups with $\gamma = 0.05$. Figure 4 shows the discretized density of the γ-scores in each treatment group.

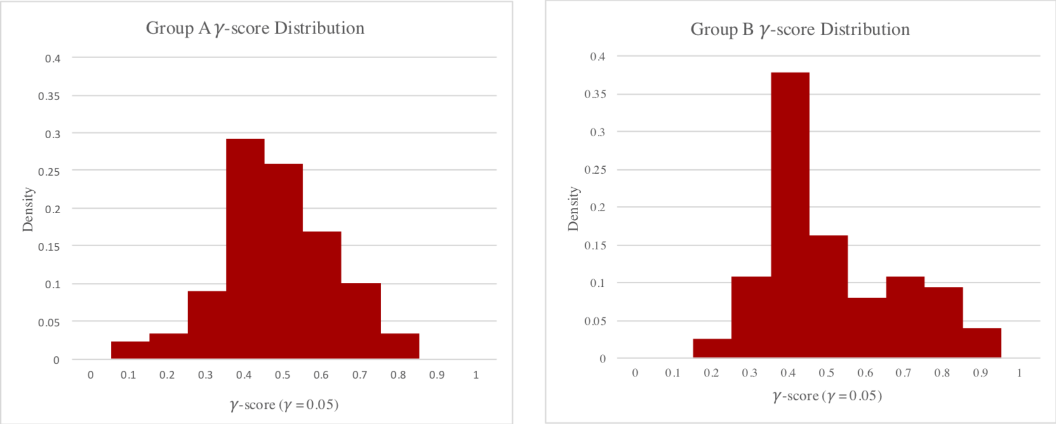


Figure 4: γ-score Distributions Across Treatment Groups.

From observation, it appears there is difference in the nature of the γ-scores between Group A and Group B. To confirm this observation we perform a statistical chi-squared test. We discretize the score by into 5 bins. This breakdown is presented in Table 5.3. Our test is set up using the following hypothesis statements: *null hypothesis* - the distribution of γ-scores is independent from treatment group, *alternative hypothesis* - the distribution of γ-scores is dependent on the treatment group.

	[0.0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8,1.0]	Totals
Group A	5	34	38	12	0	89
Group B	2	36	18	15	3	74
Totals	7	70	56	27	3	163

Table 3: γ-score Contingency Table.

From the contingency table, we compute a chi-square value of $\chi^2 = 10.53$. The critical value of χ^2 with 4 degrees of freedom is 9.49 at the 5% level of significance. Since $10.53 > 9.49$, therefore we reject the null hypothesis and conclude that the γ-score (with $\gamma = 0.05$) is affected by presenting the workers with the ontology structure. Note that our analysis does not state whether or not the γ-score has improved in Group B as compared to Group A. Though the mean is marginally higher in Group B compared to Group A (0.454 vs 0.4316), we do not draw inference on these types of measures of increase. We intentionally avoid such an analysis as it is unclear whether a strict improvement in say mean γ-score is a desired property. This is because we do not pretend to know how the data across workers will be aggregated (majority vote, EM, etc.). For example, it may be more desirable to have variance and separation amongst scores for the aggregation process. We conclude that there is a significant different in γ-score when teaching the ontology to workers and there is likely some form of improvement in performance of the group as a whole.

Our last form of data exploration centers on understanding the affect of the γ-scores when adjusting the γ parameter. Namely, we look at the difference in mean γ-score between Groups A and B for different levels of γ over the range $\gamma = 1$ (no punishment for agreement with the ontology, full punishment for a disagreement) to $\gamma = 0.5$ (equal punishment for agreement and disagreement with the ontology). Our findings are displayed in Figure 5.

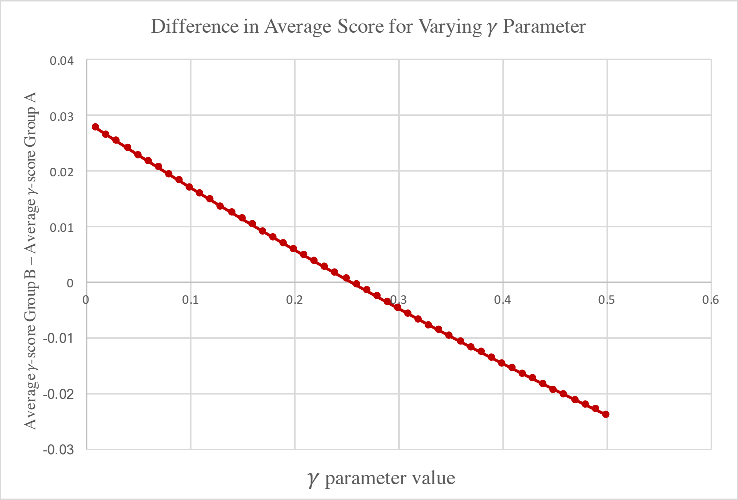


Figure 5: Difference in γ-score for Varying γ Parameter.

Notice that the difference is most magnified at the extreme values with Group B performing best for the largest ontology violation penalty and Group A performing the best when there is equal punishment for selecting wrong alternatives regardless of their ontological position. We find it note worthy that the graph is nearly a perfect straight line and that it crosses the x-axis at 0.25 almost exactly, the center of the two extremes where it so happens both Groups A and B have equal γ-scores. It is not currently well understood why the line crosses exactly at 0.25 but we do not find this a coincidence and a deeper theoretic analysis is needed to explain this phenomenon. We would like to pursue this line of study in subsequent research.

6 Discussion

This work focused on the use of a known ontology structure to aid in crowdsourcing approval voting tasks. We demonstrated the ability for the ontology to find spammers from both a theoretic and experimental perspective. We used the ontology to compute a proxy score for worker skill level and exploited this score to perform weighted majority voting with some success. Last, we found that presenting the workers the ontology before answering the questions impacts worker performance and yields a different distribution of the γ-score. It would be interesting to see if the model assumptions could be relaxed. In particular, extending the results to more general ontologies and perhaps different spammer behaviors. We would like to see in future work a stronger theoretic foundation for weighted majority voting using the ontology. Also, it would be interesting to know if there is natural theoretic description of the behavior of the γ-score for Group A and Group B for varying γ values.

References

[BBM13] Nina Balcan, Avrim Blum, and Yishay Mansour. Exploiting ontology structures and unlabeled data for learning. pages 1112–1120, 2013.

[BK14] Daniel Berend and Aryeh Kontorovich. Consistency of weighted majority votes. pages 3446–3454, 2014.

[CMW56] Clyde H Coombs, John Edgar Milholland, and Frank Burton Womer. The assessment of partial knowledge. *Educational and Psychological Measurement*, 16(1):13–37, 1956.

[EK15] Michael W. Eysenck and Mark T. Keane. *Cognitive Psychology: A Student's Handbook*. Psychology Press, 7 edition, 2015.

[FGM09] N. H. Feldman, T. L. Griffiths, and J. L. Morgan. The influence of categories on perception: Explaining the perceptual magnet effect as optimal statistical inference. *Psychological Review*, 116(4):752–782, 209.

[KOS13] David R. Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. *Proceedings of the ACM SIGMETRICS*, 41(1):81–92, 2013.

[SZP15] Nihar B. Shah, Dengyong Zhou, and Yuval Peres. Approval voting and incentives in crowdsourcing. *arXiv:1502.05696*, 2015.

[VVV14] Aditya Vempaty, Lav R. Varshney, and Pramod K. Varshney. Reliable crowdsourcing for multi-class labeling using coding theory. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):81–92, 2014.

A Proofs

Proof of Theorem 3.1. We begin by analyzing the expected value of unlabeled error of hammers with $p \leq 0.5$. Recall by assumption the p-hammer operates so that for answer u believed to be correct by the worker, $\mathbf{Pr}\left[\sum_{v\neq u} x_{ij}^{v} = \ell\right] \leq p^{\ell}$ for $\ell \geq 1$ and $\mathbf{Pr}\left[\sum_{v\neq u} x_{ij}^{v} = 0\right] \geq 1 - \frac{p-p^{\Delta'}}{1-p}$.

$$\begin{aligned}
\mathbf{E}\left[\mathrm{err}_{unl}\left(\mathrm{hammer}\right)\right] &= \mathbf{E}\left[\frac{1}{n}\sum_{j=1}^{n}\sum_{(u,v)\in E'} x_{ij}^{u}\cdot x_{ij}^{v}\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}\left[\sum_{(u,v)\in E'} x_{ij}^{u}\cdot x_{ij}^{v}\right] \\
&= \mathbf{E}\left[\sum_{(u,v)\in E'} x_{ij}^{u}\cdot x_{ij}^{v}\right] \\
&\leq 0\cdot\left(1-\frac{p-p^{\Delta'}}{1-p}\right)+\sum_{\ell=1}^{\Delta'-1}\left(\binom{\ell+1}{2}+\ell\Delta\right)p^{\ell}
\end{aligned}$$

Let

$$\begin{aligned}
S &= \left(\binom{2}{2}+\Delta\right)p+\left(\binom{3}{2}+2\Delta\right)p^{2}+\left(\binom{4}{2}+3\Delta\right)p^{3}+\cdots \\
\implies \quad pS &= \left(\binom{2}{2}+\Delta\right)p^{2}+\left(\binom{3}{2}+2\Delta\right)p^{3}+\left(\binom{4}{2}+3\Delta\right)p^{4}+\cdots
\end{aligned}$$

Subtracting we get

$$\begin{aligned}
& & (1-p)S &= (1+\Delta)p+(2+\Delta)p^{2}+(3+\Delta)p^{3}+\cdots \\
\implies & & (1-p)S &= \frac{p}{(1-p)^{2}}+\frac{\Delta}{1-p}-1 \qquad \text{by known sum formulas.} \\
\implies & & \mathbf{E}\left[\mathrm{err}_{unl}\left(\mathrm{hammer}\right)\right] &\leq \frac{p}{(1-p)^{3}}+\frac{p\Delta}{(1-p)^{3}} \\
\implies & & \mathbf{E}\left[\mathrm{err}_{unl}\left(\mathrm{hammer}\right)\right] &\leq 4(\Delta+1) \qquad \text{by } p \leq 0.5
\end{aligned}$$

So we can express the probability we misclassify a hammer as

$$
\begin{aligned}
&\mathbf{Pr}\left[\text{classified spammer}|\text{hammer}\right]\\
&\qquad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{hammer}\right) > 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2}\right]\\
&\qquad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{hammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{hammer}\right)\right] > 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2} - \mathbf{E}\left[\text{err}_{unl}\left(\text{hammer}\right)\right]\right]\\
&\qquad\leq \mathbf{Pr}\left[\text{err}_{unl}\left(\text{hammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{hammer}\right)\right] > 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2} - 4(\Delta+1)\right]\\
&\qquad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{hammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{hammer}\right)\right] > \frac{|E'|q_{\text{min}}^2}{2} - 2(\Delta+1)\right]\\
&\qquad\leq \exp\left(\frac{-2\left(\frac{|E'|q_{\text{min}}^2}{2} - 2(\Delta+1)\right)^2}{\sum_{i=1}^n\left(\frac{|E'|}{n}\right)^2}\right)\\
&\qquad\leq \exp\left(\frac{-\frac{|E'|^2q_{\text{min}}^4}{2} + 4(\Delta+1)|E'|q_{\text{min}}^2 - 8\Delta^2}{\frac{|E'|^2}{n}}\right)\\
&\qquad\leq \exp\left(-\frac{nq_{\text{min}}^4}{2} + 4\frac{n(\Delta+1)q_{\text{min}}^2}{|E'|}\right)\\
&\qquad\leq \exp\left(-\left(\frac{1}{2} - \frac{4}{c^2}\right)nq_{\text{min}}^4\right) \qquad\qquad \text{since } q_{\text{min}} > c\sqrt{\frac{\Delta+1}{|E'|}}
\end{aligned}
$$

Now we analyze the expected value of unlabeled error of the spammers. Recall by assumption that a q-hammer operates by selecting each alternative independently with probability q. So we have

$$
\begin{aligned}
\mathbf{E}\left[\text{err}_{unl}\left(\text{spammer}\right)\right] &= \mathbf{E}\left[\frac{1}{n}\sum_{j=1}^n\sum_{(u,v)\in E'} x_{ij}^u \cdot x_{ij}^v\right]\\
&= \mathbf{E}\left[\sum_{(u,v)\in E'} x_{ij}^u \cdot x_{ij}^v\right]\\
&= \sum_{(u,v)\in E'} \mathbf{E}\left[x_{ij}^u \cdot x_{ij}^v\right]\\
&= |E'|q^2.\\
&\geq |E'|q_{\text{min}}^2
\end{aligned}
$$

So we can express the probability we misclassify a spammer as

$$
\begin{aligned}
&\mathbf{Pr}\left[\text{classified hammer}|\text{spammer}\right]\\
&\quad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{spammer}\right) < 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2}\right]\\
&\quad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{spammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{spammer}\right)\right] < 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2} - \mathbf{E}\left[\text{err}_{unl}\left(\text{spammer}\right)\right]\right]\\
&\quad\le \mathbf{Pr}\left[\text{err}_{unl}\left(\text{spammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{spammer}\right)\right] < 2(\Delta+1) + \frac{|E'|q_{\text{min}}^2}{2} - |E'|q_{\text{min}}^2\right]\\
&\quad= \mathbf{Pr}\left[\text{err}_{unl}\left(\text{spammer}\right) - \mathbf{E}\left[\text{err}_{unl}\left(\text{spammer}\right)\right] < 2(\Delta+1) - \frac{|E'|q_{\text{min}}^2}{2}\right]\\
&\quad\le \exp\left(\frac{-2\left(\frac{|E'|q_{\text{min}}^2}{2} - 2(\Delta+1)\right)^2}{\sum_{i=1}^{n}\left(\frac{|E'|}{n}\right)^2}\right)\\
&\quad\le \exp\left(\frac{-\frac{|E'|^2q_{\text{min}}^4}{2} + 4(\Delta+1)|E'|q_{\text{min}}^2 - 8(\Delta+1)^2}{\frac{|E'|^2}{n}}\right)\\
&\quad\le \exp\left(-\frac{nq_{\text{min}}^4}{2} + 4\frac{n(\Delta+1)q_{\text{min}}^2}{|E'|}\right)\\
&\quad\le \exp\left(-\left(\frac{1}{2} - \frac{4}{c^2}\right)nq_{\text{min}}^4\right) \qquad\qquad \text{since } q_{\text{min}} > c\sqrt{\frac{\Delta+1}{|E'|}}
\end{aligned}
$$

Since, the probablity of error for both cases is at most $\exp\left(-\left(\frac{1}{2} - \frac{4}{c^2}\right)nq_{\text{min}}^4\right)$ this completes the proof.

□

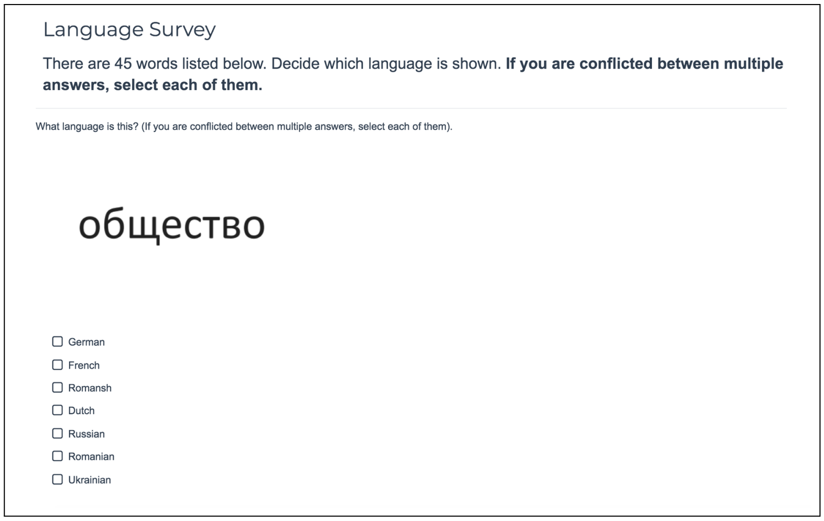
Proof of Corollary 3.1.1. Let Z_i be a random variable such that $Z_i = 1$ if the ith worker is misclassified and $Z_i = 0$ otherwise. By assumption we have:

$$
\begin{aligned}
n &\geq \frac{1}{\left(\frac{1}{2}-\frac{4}{c^2}\right) q_{\text{min}}^4} \ln\left(\frac{1}{\epsilon\delta}\right) \\
n\left(\frac{1}{2}-\frac{4}{c^2}\right) q_{\text{min}}^4 &\geq \ln\left(\frac{1}{\epsilon\delta}\right) \\
-n\left(\frac{1}{2}-\frac{4}{c^2}\right) q_{\text{min}}^4 &\leq \ln\left(\epsilon\delta\right) \\
\exp\left(-n\left(\frac{1}{2}-\frac{4}{c^2}\right) q_{\text{min}}^4\right) &\leq \epsilon\delta \\
\frac{1}{\epsilon}\exp\left(-n\left(\frac{1}{2}-\frac{4}{c^2}\right) q_{\text{min}}^4\right) &\leq \delta \\
\frac{\mathbf{E}\left[Z_i\right]}{\epsilon} &\leq \delta && \text{by Theorem 3.1.} \\
\frac{k\mathbf{E}\left[Z_i\right]}{\epsilon k} &\leq \delta \\
\frac{\mathbf{E}\left[\sum_{i=1}^{k} Z_i\right]}{\epsilon k} &\leq \delta \\
\mathbf{Pr}\left[\sum_{i=1}^{k} Z_i > \epsilon k\right] &\leq \delta && \text{by Markov's Inequality.}
\end{aligned}
$$

Thus the with at least probability of $1-\delta$ less that an ϵ fraction of workers will be misclassified.

□

B Survey Figures



Language Survey

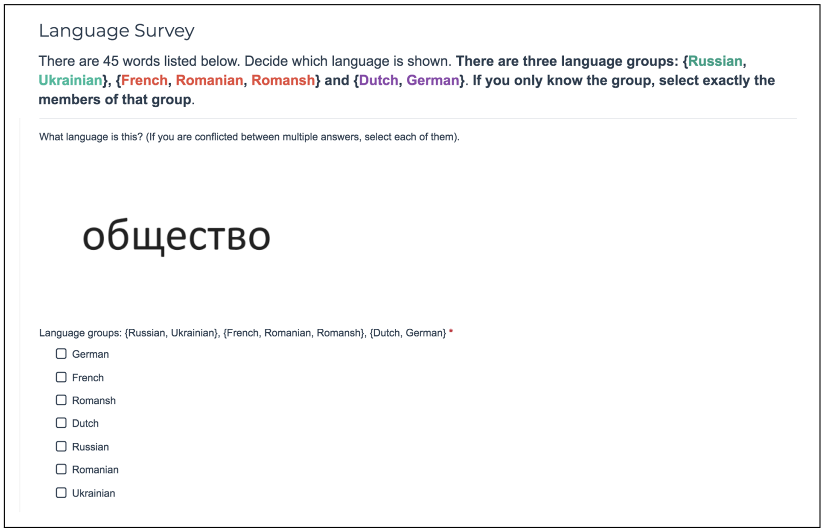
There are 45 words listed below. Decide which language is shown. **If you are conflicted between multiple answers, select each of them.**

What language is this? (If you are conflicted between multiple answers, select each of them).

общество

□ German
□ French
□ Romansh
□ Dutch
□ Russian
□ Romanian
□ Ukrainian

Figure 6: Group A instructions and example question.



Language Survey

There are 45 words listed below. Decide which language is shown. **There are three language groups: {Russian, Ukrainian}, {French, Romanian, Romansh} and {Dutch, German}. If you only know the group, select exactly the members of that group**.

What language is this? (If you are conflicted between multiple answers, select each of them).

общество

Language groups: {Russian, Ukrainian}, {French, Romanian, Romansh}, {Dutch, German} *

- □ German
- □ French
- □ Romansh
- □ Dutch
- □ Russian
- □ Romanian
- □ Ukrainian

Figure 7: Group B instructions and example question.