CLARA: Confidence of Labels and Raters

Viet-An Nguyen
Facebook

Peibei Shi
Facebook

Jagdish Ramakrishnan
Facebook

Udi Weinsberg
Facebook

Henry C. Lin
Facebook

Steve Metz
Facebook

Neil Chandra
Facebook

Jane Jing
Facebook

Dimitris Kalimeris
Harvard University

ABSTRACT

Large online services employ thousands of people to label content for applications such as video understanding, natural language processing, and content policy enforcement. While labelers typically reach their decisions by following a well-defined "protocol," humans may still make mistakes. A common countermeasure is to have multiple people review the same content; however, this process is often time-intensive and requires accurate aggregation of potentially noisy decisions.

In this paper, we present CLARA (Confidence of Labels and Raters), a system developed and deployed at Facebook for aggregating reviewer decisions and estimating their uncertainty. We perform extensive validations and describe the deployment of CLARA for measuring the base rate of policy violations, quantifying reviewers' performance, and improving their efficiency. In our experiments, we found that CLARA (a) provides an unbiased estimator of violation rates that is robust to changes in reviewer quality, with accurate confidence intervals, (b) provides an accurate assessment of reviewers' performance, and (c) improves efficiency by reducing the number of reviews based on the review certainty, and enables the operational selection of a threshold on the cost/accuracy efficiency frontier.

CCS CONCEPTS

• **Information systems → Crowdsourcing**; • **Computing methodologies → Latent variable models**.

KEYWORDS

Bayesian methods; crowdsourcing; probabilistic models

ACM Reference Format:
Viet-An Nguyen, Peibei Shi, Jagdish Ramakrishnan, Udi Weinsberg, Henry C. Lin, Steve Metz, Neil Chandra, Jane Jing, and Dimitris Kalimeris. 2020. CLARA: Confidence of Labels and Raters. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403304

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08.
https://doi.org/10.1145/3394486.3403304

1 INTRODUCTION

The wide adoption of artificial intelligence for automation across disciplines has yet to mitigate the need for human intervention and support in a range of tasks. One domain where this need manifests is that of online abuse. While online services have made great strides in leveraging machine-learned models to fight abuse (for example, Facebook reported removing 99.8% of fake accounts before they were reported),[1] they also employ thousands of human reviewers to enforce their content policies.[2]

In this paper, we focus on human reviews (we use the terms *reviews* and *labels* interchangeably) in three tasks important for fighting abuse: measuring content violations, quantifying reviewer accuracy, and investing labeling resources efficiently. We study the implementation of these tasks at Facebook, which employs thousands of content reviewers to carry out these and other tasks. These reviewers are trained to follow a *protocol*, i.e., a set of guidelines, in order to enforce Facebook's Community Standards.[3]

Despite the existence of this protocol, an obvious challenge that arises when relying on humans for content labeling is that humans are inherently *noisy* and potentially *biased* decision makers [20]. While bias is a trait of the individual, noise can result from subjectivity or ambiguity of the protocol, or from simple mistakes. These mistakes can have different consequences for the different tasks we study in this paper. For example, Facebook measures *prevalence*, defined as the percentage of policy-violating content seen by Facebook users, by sampling content and sending it to reviewers, who then review it for violations.[4] Failing to consider mistakes in these reviews can lead to incorrect prevalence estimates and/or confidence intervals. When community standards are being enforced, an incorrect decision can result in taking down a benign piece of content from the platform or leaving violating content on the platform. For training models, mistakes can result in noisy "ground truth" labels that might lead to inaccurate models.

A straightforward method to reduce the likelihood of mistakes is to assign multiple reviewers to the same task; another approach is to leverage non-human signals such as scores from machine learning models. A key challenge, which has been studied extensively in the crowd-sourcing literature since the seminal work of Dawid and Skene [8], is how to aggregate multiple decisions for a given task, and provide an estimate for the certainty of the decision.

In this paper, we present CLARA (Confidence of Labels and Raters) a statistical framework used at Facebook to estimate the

[1] https://transparency.facebook.com/community-standards-enforcement
[2] https://newsroom.fb.com/news/2019/02/commitment-to-content-reviewers/
[3] https://www.facebook.com/communitystandards/
[4] https://newsroom.fb.com/news/2019/05/measuring-prevalence/

uncertainty in human-generated decisions. CLARA is built using a generative model, which defines the process of how the observed reviewer decisions are generated based on latent (unobserved) variables, such as the true (latent) decision, the likelihood of reviewer mistakes, and the underlying prevalence. Given this generative model, we perform posterior inference to estimate the values of the latent variables given the observed labels. We show that this method provides an unbiased estimate of the true (latent) label, given one or more observed labels. We emphasize that we do not attempt to quantify *biased* decisions, a task for which Facebook deploys additional systems [4].

The contribution of this paper is in describing the extensive validations and deployment of CLARA in Facebook. While similar underlying models have been described in the past (e.g., [7, 24]), this work provides the details of a large-scale real-world deployment of the complete system, with both offline and online aggregation and uncertainty estimation capabilities. A key enabler of such deployment is the foundational understanding of the unbiased and calibrated outcomes of CLARA, which we detail in this work through extensive simulations.

We describe how CLARA's review aggregation and uncertainty estimates apply to three applications at Facebook:

- *Violation rates.* CLARA provides an unbiased estimator of violation rates that, unlike other methods, is calibrated, robust to changes in reviewer quality, and provides accurate confidence intervals.
- *Reviewers performance.* CLARA provides an unbiased estimate of the performance of reviewers, in two deployed settings – single reviewer per task (measured using auditors) and multiple reviewers per task (measured against majority vote).
- *Efficiency.* Using CLARA's uncertainty estimates, we can provide a cost/accuracy trade-off curve, enabling an operational decision to limit additional reviews based on levels of uncertainty. We show that in some cases this can save 20% of additional reviews with only a 1% loss in accuracy.

2 RELATED WORK

There has been a significant amount of work on label aggregation for inferring *truth* from crowdsourced data [1, 17, 33, 43, 48, 51]. The seminal work of Dawid and Skene (DS) [8] proposed a model to aggregate clinical diagnoses from multiple doctors using expectation maximization (EM) algorithm. Carpenter [7] and Kim & Ghahramani [24] generalized the DS model using Bayesian approaches and used Gibbs sampling for posterior inference, which is similar to the method we use in this paper. Other inference techniques were also proposed including belief propagation [21], variational inference [11, 27, 40], and spectral methods [49].

Capturing the labelers' accuracy and correcting for their bias is an important component of crowdsourcing models [18, 19, 39]. A common way to characterize labelers [24, 32, 40] is using a confusion matrix to model their ability to correctly label items of different categories. We follow this line of work in CLARA to model labelers with confusion matrices. Some papers have made further extensions, e.g., using a hierarchical prior to improve the estimate for individuals with few labels [7], using a community-based approach to group similar labelers [44], and using Dirichlet Process as the prior to allow infinite number of clusters to be discovered from the data [28]. In addition, recent work tried to capture the correlations among labelers [5, 26, 30], while some approaches avoided modeling confusion matrices altogether and instead used Item Response Theory [46] or minimax entropy principle [52].

Crowdsourcing methods have been applied to numerous domains. The DS model has been used in a range of natural language processing tasks including affect recognition [42], word sense disambiguation [32], and document labeling [12]. In computer vision, various crowdsourcing models have been proposed for image labeling [37, 41, 45, 46, 52]. Extensive research on *truth inference* has also been conducted in the database community to discover truth from noisy and potentially conflicting sources for data integration [25, 50, 51].

Crowdsourcing is often used for training and validating machine learning models [16, 23, 31, 43, 47, 48]. Work on this topic typically first uses a crowdsourcing model to aggregate labels, and then uses the estimates to train supervised models [41]; other work jointly learns reviewer uncertainties and the model [36, 37]. More recently, a "weakly-supervised" approach was proposed [2], via a system called Snorkel [35]. Snorkel aggregates "labeling functions" into a probabilistic outcome, which is then used directly for training models. We use Snorkel as a state-of-the-art baseline; however, as we show in the paper, for our use cases, Snorkel requires some form of binarization of the ML score, which results in a biased outcome. Specifically, we show that the outputs produced by Snorkel are not calibrated, which works well for their objective of training models but is not suited for prevalence measurement.

3 MODELING UNCERTAINTY

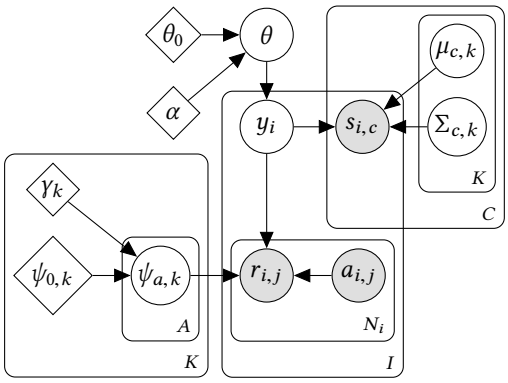
3.1 Data Properties

The input dataset consists of I items, each of which receives N_i labels denoted by $\{r_{i,j}\}$. Here, $r_{i,j}$ is the jth label of item i and is chosen among K possible label categories by reviewer $a_{i,j} \in [1, A]$. In addition, the data also contain C machine learning models (or classifiers), which generate a K-dimensional score vector $s_{i,c}$ for each item. These classifiers can be considered *artificial reviewers* that provide additional useful signals to learn about items and reviewers' latent structures. We assume that $s_{i,c}$ is a probability vector where $\sum_{k=1}^{K} s_{i,c,k} = 1$. For a complete list of notations used in this paper, see Table 1.

Table 2 illustrates an example of a dataset with three unique label categories $\{a, b, c\}$ and a single classifier. Here, the classifier provides a 3-dimensional score vector for each item. Note that the scores can be missing for a subset of the items.

3.2 Modeling Approach

We follow previous work on modeling crowdsourced data [8, 32, 33] and take a Bayesian probabilistic approach to define different latent variables and the generative process of the observed data. Building on the DS model [8], CLARA assumes that reviewers are different from one another in their ability to correctly label items of different true label categories (i.e., *confusion matrix*) and these true label categories occur at different rates in the dataset (i.e., *prevalence*). In addition, since we also observe continuous scores from machine



- Draw a Mult $\theta \sim \text{Dir}(\alpha \cdot \theta_0)$ capturing the *prevalence* of K label categories
- For each reviewer $a \in [1, A]$ and for each label category $k \in [1, K]$
  - Draw a Mult $\psi_{a,k} \sim \text{Dir}(\gamma_k \cdot \psi_{0,k})$, capturing the *confusion* of reviewer a when labeling an item with true label k
- For each classifier $c \in [1, C]$ and for each label category $k \in [1, K]$
  - Draw a mean vector $\mu_{c,k}$ and a covariance matrix $\Sigma_{c,k}$ from a Normal-inverse-Wishart prior
- Given prevalence, confusion matrices and classifier scores, for each item $i \in [1, I]$
  - Draw a true label $y_i \sim \text{Mult}(\theta)$
  - For each observed label $r_{i,j}$ by reviewer $a_{i,j}$ that item i receives
    - Draw $r_{i,j} \sim \text{Mult}(\psi_{a_{i,j}, y_i})$
  - For each score vector $s_{i,c}$ from classifier c
    - Draw $s_{i,c} \sim \text{Normal}(\mu_{c,y_i}, \Sigma_{c,y_i})$

Figure 1: Graphical representation and generative process of the Bayesian model used in CLARA. Circle nodes represent random variables (shaded ones are observed), diamond nodes represent hyperparameters, edges are probabilistic dependencies, and plates represent repetition.

Notation	Description
I	Number of items
N_i	Number of raw labels that item i receives
A	Number of unique reviewers
K	Number of unique label categories
C	Number of unique classifiers
$a_{i,j}$	j^{th} reviewer of item i
$r_{i,j}$	j^{th} label of item i (by reviewer $a_{i,j}$)
$s_{i,c}$	K-dim score vector of item i from classifier c
y_i	True label category of item i
θ	Overall prevalence (K-dim probability vector)
ψ_a	$K \times K$ confusion matrix of reviewer a
$\psi_{a,k}$	K-dim probability vector of reviewer a for item of label category k
$\mu_{c,k}$	$(K-1)$-dim mean vector of the Gaussian corresponding to classifier c and label category k
$\Sigma_{c,k}$	$(K-1) \times (K-1)$ covariance matrix of the Gaussian corresponding to classifier c and label category k

Table 1: Notations used in this paper

learning models, CLARA assumes that the scores of items belonging to each label category come from different Gaussian distributions.

Figure 1 shows the graphical representation of CLARA and its detailed generative process. Given the observed data, we perform posterior inference to estimate the posterior distribution over the latent variables using a collapsed Gibbs sampling approach. Further details about the inference procedure are in the Appendix.[5]

4 SIMULATIONS

Simulations are a crucial part of this work, as they enable us to control the values of labeling accuracy, and measure how well CLARA is able to infer them from observed data, for a wide range of settings. This provides us the confidence that, when deployed in production, CLARA infers the correct values, despite never observing them directly. Each simulation in this section corresponds with a real-world deployment, which we detail in Section 5. The parameters we use explore the space that fits the real deployment.

[5]Implementation of the Gibbs sampler and code to generate the simulated data used can be found at https://github.com/facebookresearch/clara.

Item	Labels	Reviewers	Classifier 1's scores
i	$r_{i,j}$	$a_{i,j}$	$s_{i,1}$
1	$[a, b]$	$[R_1, R_2]$	$\{a: 0.4, b: 0.5, c: 0.1\}$
2	$[a, a, c]$	$[R_2, R_3, R_4]$	$\{a: 0.8, b: 0.05, c: 0.15\}$
3	$[a, b, c]$	$[R_3, R_1, R_5]$	missing
4	$[b]$	$[R_1]$	$\{a: 0.2, b: 0.6, c: 0.2\}$
5	$[a, b, c, a]$	$[R_3, R_1, R_5, R_2]$	missing
...	...	...	...

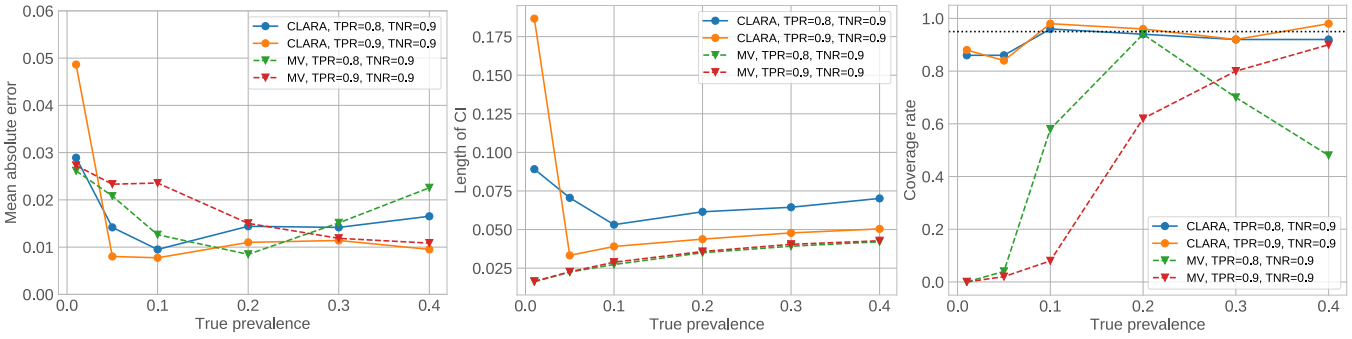
Table 2: Example of input data with $K = 3$ unique label categories and $C = 1$ classifier

4.1 Inferring Prevalence

First, we study CLARA's ability to accurately recover the prevalence, only using human-generated labels. We discuss the results when using classifier scores in the Appendix 7.3.

We generate 50 datasets consisting of $I = 2000$ items, each belonging to one of two categories ($K = 2$). We assign reviewers to items from a set of $A = 3$ distinct reviewers, in the following way: each item i is assigned to two reviewers, and if they do not agree, we assign a third reviewer to "break the tie" (for every item i, $N_i \in \{2, 3\}$). We simulate different values of prevalence $\theta \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$. We set a single confusion matrix for all three reviewers, so that their true negative rate $TNR = 0.9$ (i.e., generally correct when deciding the "negative" category), and true positive rate $TPR \in \{0.8, 0.9\}$ (i.e., have varying success when deciding on the "positive" category). For each configuration of (θ, TPR, TNR), we report:

- **Mean absolute error**: average of the absolute difference between the inferred mean estimate and the true value
- **Length of CI**: the length of the 95% confidence interval
- **Coverage rate of 95% CI**: the fraction of times the true value falls within the 95% confidence interval



(a) Mean absolute error **(b) Length of confidence interval** **(c) Coverage rate of 95% CI**

Figure 2: Prevalence estimation: CLARA versus majority vote

The simulation results are shown in Figure 2, where the dotted lines represent majority vote (MV) results and solid lines represent CLARA results. We also compared CLARA with Snorkel [35], another framework for label aggregation, however, Snorkel provide uncalibrated probabilistic labels, which results in poor prevalence estimation (see Appendix 7.4 for more details). Therefore, we focus on comparing CLARA and MV.

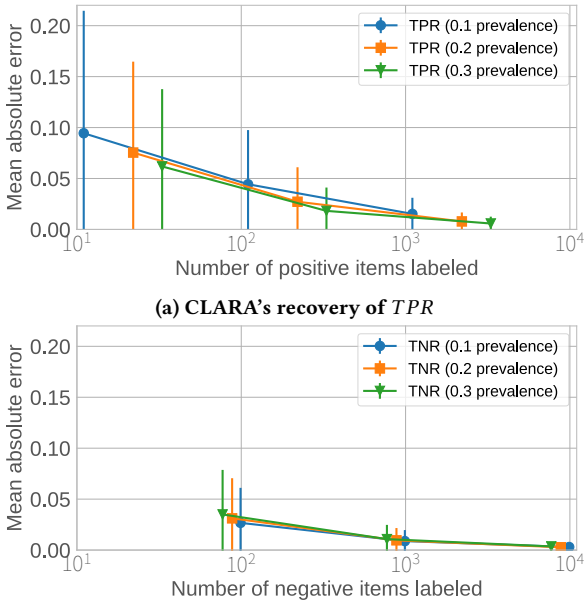
In general, CLARA provides estimated prevalences with smaller error when compared with MV (Figure 2a). The bias of CLARA is consistently small over different prevalence values, except when the true prevalence is close to 0. The magnitude of bias of MV method has a U-shape for $TPR = 0.8$ with larger bias on the two ends of the prevalence plot. Although MV method works well for some prevalence points, its performance is not stable across different settings, which is problematic in a real-world deployment, where the underlying prevalence is unknown.

The length of CI based on MV is quite small, as it fails to consider the uncertainty from labeling error (Figure 2b), while CLARA's CI is wider and more accurate. In addition, CLARA provides wider CI for smaller TPR, which is reasonable since smaller TPR indicates more uncertainty in the labeling output and it leads to wider confidence interval. Due to the bias and underestimation of the width of the confidence interval, inference of prevalence based on MV has higher error in most of the settings and the coverage rate of 95% confidence interval is even close to zero when the true prevalence is small (Figure 2c). Overall, CLARA provides accurate confidence intervals for different prevalences, though it is difficult to make an accurate inference when true prevalence is close to zero and the prevalence is smaller than 0.1.

4.2 Inferring Reviewer Performance

Next, we study CLARA's ability to measure reviewers' performance in two production settings: 1) each task is reviewed by several reviewers without any notion of "ground truth", and 2) a task is reviewed by a single reviewer, and a subset of these reviews are sent to a separate set of auditors, who are highly-skilled, yet still potentially noisy.

4.2.1 ***Performance without audits.*** In this use case, CLARA infers reviewer performance by observing agreements and disagreements among reviewers over multiple tasks, for both positive and negative labels. Therefore, our simulations vary the number of items reviewed by each reviewer, and using different prevalence settings, which determine the number of observations we will have for each reviewer across the different outcomes.

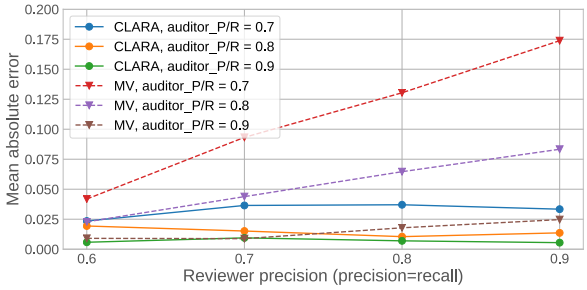


(a) CLARA's recovery of TPR

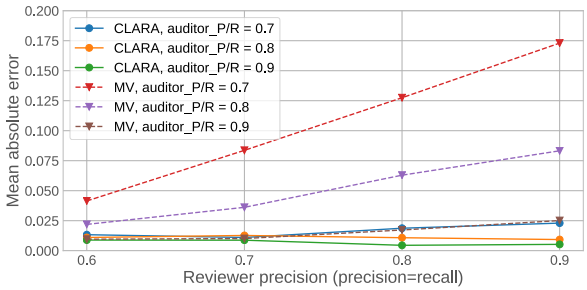
(b) CLARA's recovery of TNR

Figure 3: CLARA's recovery of TPR and TNR as a function of number of items labeled by each reviewer

Our dataset consists of $I = 50,000$ items, each with a different number of reviewers, $A_1 = 10$, $A_2 = 100$, and $A_3 = 1000$ to simulate having reviewers review more or less items on average. As in the previous simulation, each item is assigned with two random reviewers, and we "break ties" with a third random reviewer if needed. In order to simulate varying accuracy across reviewers, each reviewer's TPR and TNR are sampled according to a normal distribution with mean 0.9 and standard deviation 0.05 to simulate reviewers being around 90% accurate.



(a) Errors of precision estimates



(b) Errors of recall estimates

Figure 4: Errors of the estimates of reviewers' precision and recall by majority vote (MV) and CLARA, measured against auditors with varying precision and recall

For each of these three sets of reviewers, we generate $N = 50,000$ items with three different prevalence values, $\theta \in \{0.1, 0.2, 0.3\}$ (corresponding to the positive class). For the 9 resulting datasets, we repeat the simulation 50 times. Figure 3 shows the mean absolute error and standard deviation of CLARA's error in estimating TPR and TNR for each reviewer, averaged over the 50 simulated runs.

For both TPR (Figure 3a) and TNR (Figure 3b), the inference improves as the number of items labeled by each labeler increases. Moreover, after each reviewer reviews around 100 items for each class (positives and negatives), CLARA is able to recover the TPR and TNR rates within less than 4.5% error on average. For around 100 items labeled of each class, the TPR has a slightly higher error rate of around 4.5% for prevalence 0.1 compared to the 2.7% error for inferring TNR. Overall, the error in inferring both TPR and TNR continues to decline with the number of items labeled, and decreases to below 1.5% with over 1000 items labeled for each class.

4.2.2 ***Performance with audits.*** Some of the decisions used for enforcing Facebook's community standards are done by a single reviewer. In order to measure the accuracy of these decision, Facebook employs a smaller set of *auditors*, who go through more extensive training sessions, and audit a subset of all decisions. However, despite their extensive training, they might also make mistakes. Therefore Facebook sends each audited task to two auditors, and if they do not agree, the task is sent to a third one who "breaks the tie". Given this setup, we want to be able to measure the performance of the reviewers, with respect to the auditors' labels, while taking into account that both the original reviewer and the auditors can make mistakes.

The performance of both set of reviewers are measured by precision and recall, where both can be computed based on TPR, TNR and prevalence from CLARA's output. When generating simulated datasets, we let precision equal recall for both reviewers and auditors. We simulate with precision (recall) equals 0.6, 0.7, 0.8, 0.9 for reviewers, and precision (recall) equals 0.7, 0.8 and 0.9 for auditors. For each combination of precision/recall from the above two reviewer groups, we generate datasets consisting of $N = 5000$. We use a prevalence (ratio of positive labels) of 0.5, as this resembles the setup used at Facebook for auditing tasks, where the positive class is upsampled. Each item has one reviewer rating and two or three random auditors' ratings (based on a "tie-breaking" rule).

We estimate precision and recall of reviewers using two methods. First, the baseline is using the majority vote of the auditors as the "ground truth label", and measure precision and recall of reviewers against it. Second, using CLARA, we treat reviewers as one group (all sharing a single confusion matrix) and the auditors as a separate group (again, all sharing a single confusion matrix). We then use CLARA's inferred confusion matrix of reviewers to compute precision and recall.

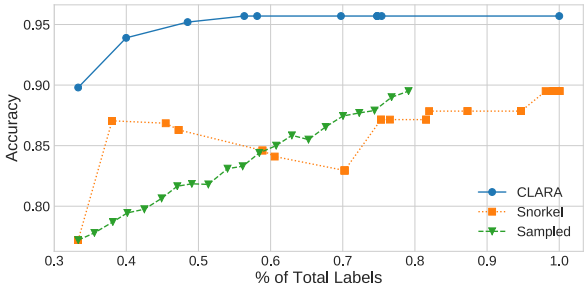
Figure 4a and 4b depict the mean absolute error of estimated precision and recall for regular reviewers respectively, averaging over 50 simulated runs for each configuration. CLARA consistently outperforms majority vote, specifically when the auditors have a low precision (recall). Intuitively, when auditors make more mistakes, using majority vote is more likely to produce an incorrect label, which leads to inaccurate inference of the reviewer's performance. CLARA, on the other hand, is capable of observing and correcting for such cases, thus produces a significantly better estimate.

4.3 Labeling Efficiency

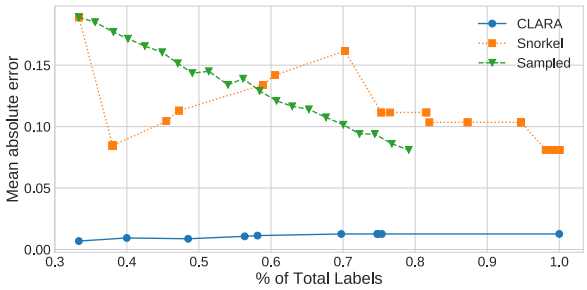
Finally, we explore the usage of CLARA for an efficient review process – instead of using tie-breaking, we can leverage the confidence of the label as provided by CLARA, and obtain more labels only when the confidence is not sufficiently high. Here, confidence is defined as the maximum of the per class probabilities over all classes. Unlike prevalence measurement, in this case, the confidence does not need to be calibrated, enabling us to compare with uncalibrated methods. We consider the following escalation methods:

Random stratified sampling. As a baseline, we extend the tie-breaking method and use a stratified sampling approach where we get additional reviews for a given percentage of items stratified by the class of the first review. If the first two reviews disagree, a final review is requested to break the tie.

Snorkel. We use the confidence from a trained Snorkel model to determine when to stop requesting for additional reviews. That is, for a particular threshold and a given number of reviews, we stop requesting additional reviews if the confidence from the Snorkel model is above a threshold. Once stopped, we use majority vote to determine the final label. We also tried binarzing Snorkel's output using 0.5 as a threshold; however that performed even worse than majority vote aggregation. It's important to note that Snorkel's output is intended to be used as for training an ML model, rather than binarizing the output. Its uncalibrated output is most likely the cause for the low performance we further observe in this analysis.



(a) Accuracy



(b) Mean absolute error

Figure 5: Accuracy of ground truth and prevalence estimation when using dynamic labeling policies for reducing labeling volume

CLARA. Similar to the Snorkel approach, we use the confidence from CLARA to determine when to stop requesting for additional labels. Once stopped, we use a threshold of 0.5 on the CLARA probability output to determine the final label.

We use $A = 3$ reviewers for every item with a train dataset and validation dataset, each consisting of $I = 2000$ items. The train set is used for training Snorkel and CLARA, and the validation set is used to compare all the methods. Here, we assume a prevalence of 0.1. The main difference compared to previous simulations is that we assume three different types of raters or confusion matrices, specifically each with different TPR and TNR. This is because varying worker quality is a crucial factor in reducing the number of labels, e.g., with a single review and single confusion matrix, there are only two confidence values from either Snorkel or CLARA for the first review. We use $TPR = TNR$, with 0.6, 0.8, and 0.9, corresponding to the different labeler types.

Figure 5 shows the trade-off curves between total number of labels and the accuracy, measured as aggregated labels against the simulated true label (Figure 5a), and estimated prevalence against simulated prevalence (Figure 5b). The maximum amount of labels is reached when all three reviewers label all tasks, which not always reached when using the tie-breaking method; indeed, using our parameters, about 20% of the labels are not used for the tie-breaking method because many jobs do not need a third review. Also, note that since we use majority vote to aggregate Snorkel's outputs, the right-most value of both plots is the same for the Snorkel and "Sampled" methods.

Figure 5b shows that CLARA significantly outperforms the baseline sampling and Snorkel, and is able to recover the true label with 95% accuracy even when 50% of the labels are not used. Similarly, Figure 5b shows that CLARA estimates the true prevalence well regardless of the percent of total labels used while both the sampling and Snorkel methods have biased prevalence estimates that get worse with fewer labels. These results emphasize the importance of validating the label aggregation method to the specific task.

5 DEPLOYMENT

In this section, we describe the deployment of CLARA in production at Facebook, and how we leverage it to accommodate three closely related tasks: (1) measuring the prevalence of different types of bad content, (2) obtaining unbiased estimates of reviewers' performance, and (3) improving the efficiency of our labeling workforce.

5.1 System Overview

Figure 6 illustrates an overview of how CLARA is deployed at scale in production at Facebook. Content is sampled and sent to review using a centralized reviewing tool. The tool strips private details, and renders the content so that it can be reviewed. The decisions are passed to CLARA in both offline and online fashions.

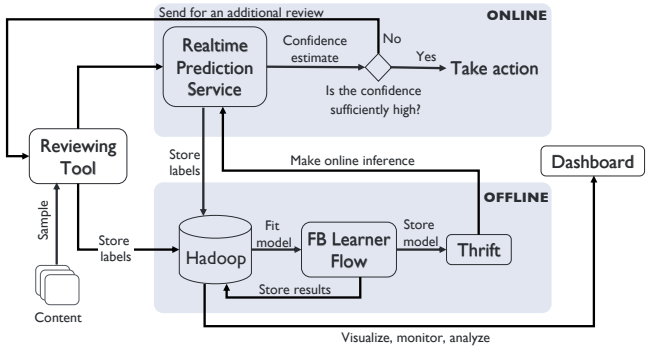


Figure 6: System overview of CLARA

Offline. Labels are stored in Hadoop – a distributed file system and periodically to train CLARA models using Facebook's FBLearner Flow [10]. The trained model (its binary outcome) is stored together with all its infered outcomes – prevalence, reviewer precision and recall, which are used for monitoring and analytical purposes, including external reporting. The trained model is stored in a Thrift structure (https://thrift.apache.org/), which enables cross-language services deployment and inference calls. This is used in the online deployment, providing uncertainty estimates in real-time.

Online. Using Facebook's real-time prediction service (FBLearner Predictor), CLARA uses the pre-trained models to estimate the confidence of the decision. If the estimated confidence is less than a threshold (which is pre-selected using CLARA's cost/accuracy trade-off analysis), the item is re-enqueued in the reviewing tool for additional reviews. This is done until the obtained certainty (using all provided reviews) is sufficiently high, in which case the appropriate action is taken (e.g., remove content that was decided to be violating).

5.2 Prevalence Measurement

Facebook publicly reports the *prevalence* of different violation categories on the platform. Prevalence is defined as the percentage of views that were of violating content, and thus is used to measure how "bad content" is seen on Facebook.

Facebook uses the following process to measure prevalence for a set of violations: 1) sampling uniformly at random a subset of items from the population of interest, 2) sending the sampled data to reviewers, and 3) using statistical methods to infer the mean and confidence interval of prevalence in the population from the labeled data. It is worth emphasizing that here we focus on estimating the prevalence using *only* human labels and assume that we do not have access to the whole unlabeled population. This is in contrast to the body of research on prevalence estimation [6, 22], also known as *quantification* [3, 13–15, 29] or *class prior estimation* [34, 38], which use supervised learning to train a classifier and make predictions on unlabeled data to infer the prevalence in the population.

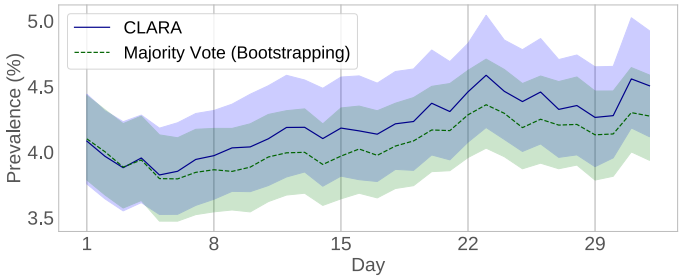
5.2.1 ***Data.*** We consider two prevalence metrics, A and B, measuring two violation types on Facebook. To measure the two metrics, everyday we sample uniformly at random a subset of items (about 2000 for metric A and 400 for metric B) from the population and send for human reviewers to label as either "violating" or not. To reduce the impact of labeling mistakes, we use multi-review for both metrics in which each item is first labeled by two reviewers and is sent for an additional reviewer to break tie if the first two disagree.

5.2.2 ***Prevalence estimation.*** We compare two approaches to estimate the prevalence mean and 95% confidence interval (CI):

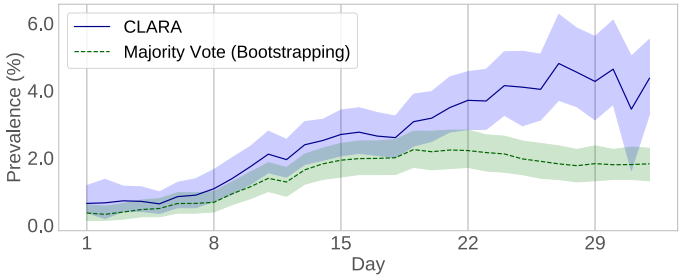
- **Majority vote with bootstrapping (MV).** This is a baseline approach which (1) aggregates each item's multiple labels using majority vote, (2) compute the fraction of items whose aggregated label is "violation" to estimate the prevalence mean, and (3) use bootstrapping [9] to generate 1000 point estimates to compute the 95% CI.
- **CLARA.** CLARA outputs the estimated prevalence of the labeled dataset θ, which can be used to estimate the prevalence in the population since the labeled dataset is sampled uniformly at random. More concretely, during Gibbs sampling, we collect 1000 point estimates of θ and use them to compute the prevalence mean and 95% CI.

We collect a month of labeled data and use the two methods to estimate the prevalence for both metrics. Figure 7 shows the prevalence estimates (i.e., the mean and 95% CI) from MV and CLARA.

5.2.3 ***Incorporating labeling error in prevalence estimates.*** One main limitation of using majority vote for prevalence estimation, as shown in Section 4, is that it does not account for labeling error which can lead to biased estimates. CLARA, on the other hand, jointly infers the prevalence and the confusion matrix of reviewers, providing estimates which are corrected for labeling error. The estimates from MV and CLARA align relatively well for metric A (Figure 7a), most likely due to very low error rates. On the other hand, the two estimates diverge for metric B especially after day 15 (Figure 7b), which is likely due to a change in labeling quality, e.g., as a result of updated guidelines.



(a) Metric A



(b) Metric B

Figure 7: Prevalence estimates (mean and 95% confidence intervals) for metrics A and B, comapring CLARA and majority vote

Furthermore, using majority vote to aggregate multiple labels into a single final label ignores important information about the uncertainty of the labeling process, especially when reviewers disagree with each other, and thus underestimates the prevalence confidence intervals. As shown in Figures 7a and 7b, CLARA's CIs are generally larger than MV for both metrics, capturing additional uncertainty from labeling (in addition to sampling uncertainty).

5.2.4 ***Identifying the cause of prevalence changes.*** Another key value of estimating and monitoring prevalence using CLARA over time is the ability to identify if the change in prevalence is due to a change in labeling quality. In our example, the estimates from MV and CLARA for metric A track closely with each other and are stable over time due to the stability of TPR/TNR estimates and disagreement rate over time. However, as discussed above for metric B, CLARA can identify a spurious movement in prevalence starting around day 15 where the estimated prevalence increases and the CI becomes much wider. These changes can be explained by a change in the TPR/TNR estimates in which TNR remains high and stable over time but there is a drop in TPR starting from day 15. Diagnosing the metric further reveals that there was a re-training for reviewers around this time; this led to a higher disagreement rate among reviewers, resulting in a change in the labeling quality and behavior for metric B.

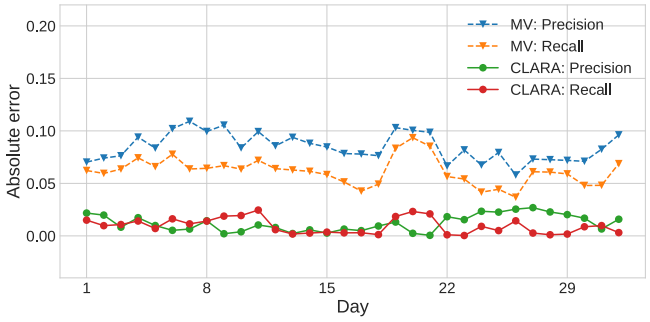


Figure 8: Absolute error of precision and recall estimates using Majority Vote and CLARA, compared with ground truth precision and recall

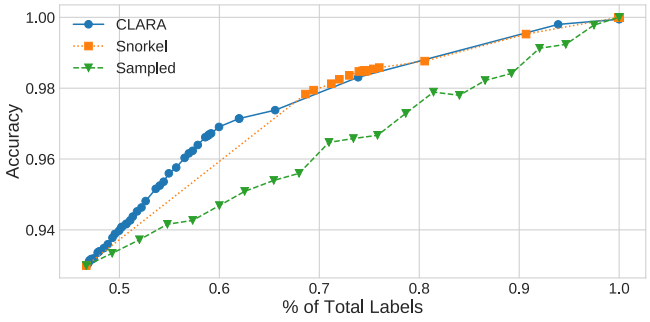


Figure 9: Labeling efficiency in production, enabling the selection of an operating point on the "efficient frontier" of the cost/accuracy tradeoff

5.3 Reviewer Performance Measurement

As described in Section 4.2.2, we use CLARA to measure reviewer performance using labels obtained from auditor reviewers. In Facebook, these performance measures are used to identify reviewers that need additional training, and cases where the protocols that they use are not sufficiently clear.

5.3.1 ***Data.*** We collected decisions that were sampled for auditor reviewers for 30 days, each of which contains about 20K items. Each item in the dataset is labeled as either *violating* or *non-violating* by (1) a single regular reviewer, and (2) two or three (for breaking ties) auditors. Similar to Section 4.2.2, we estimate the precision and recall of regular reviewers using two methods: majority vote of auditors and CLARA. For validation, we use a set of ground truth labels to compute the ground truth precision and recall.

5.3.2 ***Result.*** Figure 8 shows the absolute error of estimated precision and recall for regular reviewers using both methods over a 30-day period, compared with the ground truth precision and recall. We can see that CLARA's precision and recall estimates are almost unbiased with the absolute errors are consistently close to zero for all 30 days, while the errors using MV are significantly higher.

5.4 Labeling Efficiency

We deployed CLARA to improve the efficiency of human labeling at Facebook for a number of settings, including but not limited to enforcement (i.e., deciding whether an object is violating Facebook's policies) and labeling data for machine learning models. As described in Section 4.3, the main idea is to estimate the confidence of human labels and only send for additional reviews when the confidence is not sufficiently high.

Here, we are interested in reducing the number of labels while achieving minimal loss in accuracy compared to the majority voted label. A cruicial outcome of using the uncertainty is the ability to plot the efficient frontier of the cost (number of needed labels) vs. accuracy (as measured against the expensive majority vote). This enables team to select an operating point, balancing the savings with the accuracy.

5.4.1 ***Data.*** We used human labeled data over 60 days for a certain violation type on Facebook, originally reported from either users or various automated models. The dataset consists of two classes with roughly 280K items in the train set obtained from the first 53 days and 5.5K items in the validation set from the last 7 days. The data collection is done through a tie-breaking approach with up to three reviewers for every item; when the first two reviewers disagree, a final review is used to break the tie. About 10% of the reported data has a positive violation label in train data when using majority vote, and there are 148 distinct reviewers over the 60 day time period. We use labeling functions and individual confusion matrices for every reviewer, respectively, for training Snorkel and CLARA.

5.4.2 ***Result.*** In Figure 9, we show the trade-off curve for the percent of total labels and accuracy of the final decision compared to majority vote. We estimate the confidence of the first review (maximum of the per class probabilities over all classes) for Snorkel and CLARA, and request additional reviews only if it is above a threshold. The tradeoff curve is the result of varying the threshold between 0.5 and 1.0 with steps of 0.01. As shown, we can save 20% of the labels with about 1% loss in accuracy compared to majority vote if we use CLARA or Snorkel as the escalation method. Note that in the simulations, CLARA was able to do a better job of recovering the true label compared to Snorkel and the baseline random stratified sampling approach. However, in production, the true label is unknown, and we compared against the majority voted label, which could be the reason for the discrepancy.

6 CONCLUSION

In this paper, we presented CLARA, a system developed at Facebook to estimate human labeling uncertainty. We showed three use cases of CLARA at Facebook – improved prevalence estimation, more accurate assessment of reviewer performance, and more efficient labeling. In its current implementation, CLARA leverages machine learning scores by considering them to be similar to a non-binary "reviewer". However, we observe that human mistakes are correlated with the difficulty of the task, which can be reflected in the machine learning score. Therefore, in future work, we plan to extend the discrete confusion matrix that CLARA infers to instead learn a continuous "confusion" function and continuous prevalence function, which may vary based on the difficulty of the task as modeled by a machine learning score.

REFERENCES

[1] Lora Aroyo, Anca Dumitrache, Oana Inel, Zoltán Szlávik, Benjamin Timmermans, and Chris Welty. 2019. Crowdsourcing Inclusivity: Dealing with Diversity of Opinions, Perspectives and Ambiguity in Annotated Data. In *WWW*. 1294–1295.

[2] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the structure of generative models without labeled data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

[3] Antonio Bella, Cesar Ferri, Jose Hernandez-Orallo, and Maria Jose Ramirez-Quintana. 2010. Quantification via Probability Estimators. In *ICDM*.

[4] Joaquin Quiñonero Candela. 2019. Facebook and the Technical University of Munich Announce New Independent TUM Institute for Ethics in Artificial Intelligence. https://newsroom.fb.com/news/2019/01/tum-institute-for-ethics-in-ai/

[5] Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. 2019. Max-MIG: an Information Theoretic Approach for Joint Learning from Crowds. In *ICLR*.

[6] Dallas Card and Noah A Smith. 2018. The Importance of Calibration for Estimating Proportions from Annotations. In *NAACL-HLT*, Vol. 1.

[7] Bob Carpenter. 2008. Multilevel Bayesian models of categorical data annotation. *Unpublished manuscript* 17, 122 (2008), 45–50.

[8] A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* 1 (1979).

[9] B. Efron. 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics* 7, 1 (1979), 1–26.

[10] Facebook. 2016. Introducing FBLearner Flow: Facebook's AI backbone. https://code.fb.com/core-data/introducing-fblearner-flow-facebook-s-ai-backbone/

[11] Paul Felt, Kevin Black, Eric Ringger, Kevin Seppi, and Robbie Haertel. 2015. Early Gains Matter: A Case for Preferring Generative over Discriminative Crowdsourcing Models. In *NAACL-HLT*. 882–891.

[12] Paul Felt, Eric Ringger, and Kevin Seppi. 2016. Semantic annotation aggregation with conditional crowdsourcing models and word embeddings. In *COLING*.

[13] George Forman. 2005. Counting Positives Accurately Despite Inaccurate Classification. In *ECML*.

[14] George Forman. 2008. Quantifying Counts and Costs via Classification. *Data Min. Knowl. Discov.* 17, 2 (Oct. 2008), 164–206.

[15] Pablo González, Alberto Castaño, Nitesh V. Chawla, and Juan José Del Coz. 2017. A Review on Quantification Learning. *ACM Comput. Surv.* 50, 5 (Sept. 2017).

[16] Melody Y. Guan, Varun Gulshan, Andrew M. Dai, and Geoffrey E. Hinton. 2017. Who Said What: Modeling Individual Labelers Improves Classification. In *AAAI*.

[17] Nguyen Quoc Viet Hung, Nguyen Thanh Tam, Lam Ngoc Tran, and Karl Aberer. 2013. An evaluation of aggregation techniques in crowdsourcing. In *WISE*.

[18] Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang. 2014. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery* 28, 2 (2014), 402–441.

[19] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on Amazon Mechanical Turk. In *SIGKDD Workshop on Human Computation*.

[20] Daniel Kahneman, Andrew M. Rosenfield, Linnea Gandhi, and Tom Blaser. 2016. Noise: How to Overcome the High, Hidden Cost of Inconsistent Decision Making. https://hbr.org/2016/10/noise

[21] David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative Learning for Reliable Crowdsourcing Systems. In *NeurIPS*. 1953–1961.

[22] Katherine A. Keith and Brendan O'Connor. 2018. Uncertainty-aware generative models for inferring document class prevalence. In *EMNLP*.

[23] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. 2018. Learning From Noisy Singly-labeled Data. In *ICLR*.

[24] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics*. 619–627.

[25] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. *SIGKDD* 17, 2 (2016).

[26] Yuan Li, Benjamin Rubinstein, and Trevor Cohn. 2019. Exploiting Worker Correlation for Label Aggregation in Crowdsourcing. In *ICML*. 3886–3895.

[27] Qiang Liu, Jian Peng, and Alexander Ihler. 2012. Variational Inference for Crowdsourcing. In *NeurIPS*. 692–700.

[28] Pablo G. Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Perez-Cruz. 2015. Bayesian Nonparametric Crowdsourcing. *JMLR* 16, 1 (2015).

[29] Alejandro Moreo and Fabrizio Sebastiani. 2019. Learning to Quantify: Estimating Class Prevalence via Supervised Learning. In *SIGIR*.

[30] An T. Nguyen, Byron C. Wallace, and Matthew Lease. 2016. A Correlated Worker Model for Grouped, Imbalanced and Multitask Data. In *UAI*. 537–546.

[31] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2019. Confident Learning: Estimating Uncertainty in Dataset Labels. arXiv:stat.ML/1911.00068

[32] Rebecca J Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *TACL* 2 (2014).

[33] Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian Models of Annotation. *TACL* 6 (2018), 571–585.

[34] Marthinus C. Plessis, Gang Niu, and Masashi Sugiyama. 2017. Class-prior Estimation for Learning from Positive and Unlabeled Data. *Machine Learning* (2017).

[35] Alexander J. Ratner, Stephen H. Bach, Henry E. Ehrenberg, and Christopher Ré. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *VLDB Endowment* 11, 3 (2017), 269–282.

[36] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Anna Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*.

[37] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning From Crowds. *JMLR* 11 (Aug. 2010), 1297–1322.

[38] Marco Saerens, Patrice Latinne, and Christine Decaestecker. 2002. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation* (2002).

[39] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *SIGKDD*. 614–622.

[40] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. 2013. *Dynamic Bayesian Combination of Multiple Imperfect Classifiers*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–35.

[41] Padhraic Smyth, Usama M Fayyad, Michael C Burl, Pietro Perona, and Pierre Baldi. 1995. Inferring ground truth from subjective labelling of Venus images. In *NeurIPS*. 1085–1092.

[42] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *EMNLP*. 254–263.

[43] Jennifer Wortman Vaughan. 2018. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *JMLR* 18 (2018), 193–1.

[44] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based Bayesian Aggregation Models for Crowdsourcing. In *WWW*.

[45] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *NeurIPS*. 2424–2432.

[46] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NeurIPS*. 2035–2043.

[47] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. 2014. Learning from Multiple Annotators with Varying Expertise. *Mach. Learn.* 95, 3 (June 2014), 291–327.

[48] Jing Zhang, Xindong Wu, and Victor S. Sheng. 2016. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review* 46 (2016), 543–576.

[49] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. 2016. Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing. *JMLR* 17, 1 (Jan. 2016), 3537–3580.

[50] Bo Zhao, Benjamin I. P. Rubinstein, Jim Gemmell, and Jiawei Han. 2012. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration. *VLDB Endowment* 5, 6 (Feb. 2012), 550–561.

[51] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *VLDB* (2017).

[52] Dengyong Zhou, Sumit Basu, Yi Mao, and John C. Platt. 2012. Learning from the Wisdom of Crowds by Minimax Entropy. In *NeurIPS*. 2195–2203.

7 APPENDIX

7.1 Implementation and Experiment Details

Code for CLARA's Gibbs sampler and simulated synthetic datasets can be found at **https://github.com/facebookresearch/clara**. For Snorkel comparisons, we used their Python package.[6]. For all simulations, we run the Gibbs sampler for 12000 iterations, collecting 2000 samples after a 2000-burn-in period with a sample lag of 5.

7.2 Posterior Inference

Given the observed data which includes the observed labels $\boldsymbol{r} = \{r_{i,j}\}$ with their corresponding reviewers $\boldsymbol{a} = \{a_{i,j}\}$ and the classifier scores $\boldsymbol{s} = \{s_{i,c}\}$, the inference task is to find the posterior distribution over the latent variables including (1) the prevalence θ, (2) the confusion matrices $\boldsymbol{\psi} = \{\psi_{a,k}\}$ of all reviewers, and (3) the means $\{\mu_{c,k}\}$ and covariance matrix $\{\Sigma_{c,k}\}$ of all Gaussian distributions. Since exact inference is intractable, we approximate the posterior distribution by performing collapsed Gibbs sampling, alternating between (1) sampling the true label y_i of each item after integrating out θ and $\boldsymbol{\psi}$ and (2) estimating the mean $\mu_{c,k}$ and covariance matrix $\Sigma_{c,k}$ of each Gaussian distribution. Algorithm 1 provides an overview of the inference method used.

Algorithm 1: Collapsed Gibbs sampling procedure

```
Initialize y_i with the majority voted label;
for iteration t from 1 to T do
    foreach classifier c and label category k do
        Update μ_{c,k} and Σ_{c,k} using Equations 2 and 3;
    end
    foreach item i do
        Remove the current assignment of y_i;
        Sample y_i according to Equation 1;
        Add the newly sampled assignment of y_i;
    end
end
```

Sampling y_i. The probability of assigning a label k to item i conditioned on all other variables is denoted by $p(y_i = k \mid \star)$ and computed as follow:

$$p(y_i = k \mid \star) \propto p(y_i \mid \boldsymbol{y}^{-i}, \theta; \alpha, \theta_0) \times \prod_{j=1}^{N_i} p(r_{i,j}, a_{i,j} \mid y_i, \boldsymbol{r}^{-i}, \boldsymbol{a}^{-i}, \boldsymbol{\psi}; \boldsymbol{\gamma}, \boldsymbol{\psi}_0) \times \prod_{c=1}^{C} p(s_{i,c} \mid y_i, \boldsymbol{\mu}, \Sigma; \mu_0, \lambda_0, \Sigma_0, \nu_0) \quad (1)$$

where the superscript $^{-i}$ denotes the exclusion of item i from a specific set of variables.

Due to the conjugacy between Dirichlet and Multinomial distributions, we can integrate out both θ and $\boldsymbol{\psi}$ and compute the first two components in Equation 1 as follow:

$$p(y_i = k \mid \boldsymbol{y}^{-i}; \alpha, \theta_0) = \frac{\alpha \cdot \theta_{0,k} + n_k - 1}{\alpha + \sum_{k'=1}^{K} n_{k'} - 1}$$

$$p(r_{i,j}, a_{i,j} \mid y_i = k, \boldsymbol{r}^{-i}, \boldsymbol{a}^{-i}; \boldsymbol{\gamma}, \boldsymbol{\psi}_0) = \frac{\gamma_k \cdot \psi_{0,k,r_{i,j}} + m^{-i}_{a_{i,j},k,r_{i,j}}}{\gamma_k + \sum_{r'=1}^{K} m^{-i}_{a_{i,j},k,r'}}$$

where n_k denotes the number of items being assigned with true label k and $m^{-i}_{a,k,r}$ denotes the number of times that reviewer a assigns an observed label r to an item with true label k excluding item i.

Since we are updating the mean μ and covariance Σ of all the Gaussian distributions, the last component in Equation 1 is simply

$$p(s_{i,c} \mid y_i = k, \boldsymbol{\mu}, \Sigma; \mu_0, \lambda_0, \Sigma_0, \nu_0) = \mathcal{N}(s_{i,c}; \mu_{c,k}, \Sigma_{c,k}) = \frac{1}{(2\pi)^{(K-1)/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(s_{i,c} - \mu_{c,k})^T \Sigma^{-1} (s_{i,c} - \mu_{c,k})\right)$$

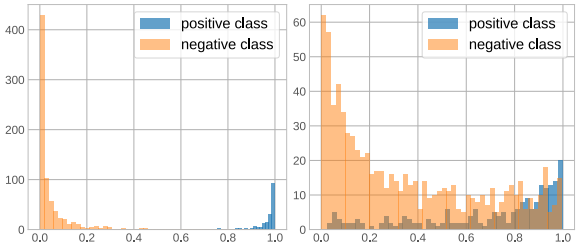
Updating $\mu_{c,k}$ ***and*** $\Sigma_{c,k}$. After assigning a true label to each item, we update the mean and covariance of each Gaussian distribution using maximum likelihood estimation (MLE). Let $\boldsymbol{I}_k$ denote the set of items being assigned to label category k, the mean vector $\mu_{c,k}$ and the covariance matrix are updated as follow:

$$\mu_{c,k} = \frac{1}{|\boldsymbol{I}_k|} \sum_{i \in \boldsymbol{I}_k} s_{i,c} \quad (2)$$

$$\Sigma_{c,k} = \frac{1}{|\boldsymbol{I}_k|} \sum_{i \in \boldsymbol{I}_k} (s_{i,c} - \mu_{c,k})(s_{i,c} - \mu_{c,k})^T \quad (3)$$

It is worth noting that for simplicity and efficiency, we use Maximum Likelihood Estimation (MLE) to update μ and Σ here. When the dataset is highly imbalanced in which MLE might fail if no item is assigned to some low prevalence label categories during Gibbs sampling, we update μ and Σ by sampling from the posterior distributions.

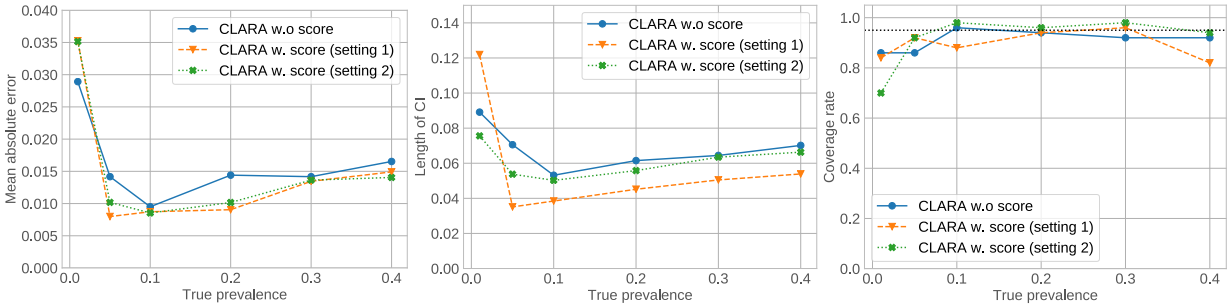
For the experimental results in this paper as well as in production, we run CLARA for a fixed number of iterations, discard all samples during the *burn-in* period (to remove samples from the non-stationary posterior distribution at the beginning of the sampling process), and collect the point estimates of all latent variables after every few iterations (i.e., the *sample-lag* to avoid auto-correlation between consecutive samples). To set the prior means θ_0 and $\psi_{0,k}$, we use an empirical Bayes approach and treat the majority votes as the true labels to estimate the prevalence and confusion matrix from the input data.



(a) Setting 1: $\mu_0 = -4$, $\mu_1 = 4$ **(b) Setting 2:** $\mu_0 = -1$, $\mu_1 = 1$

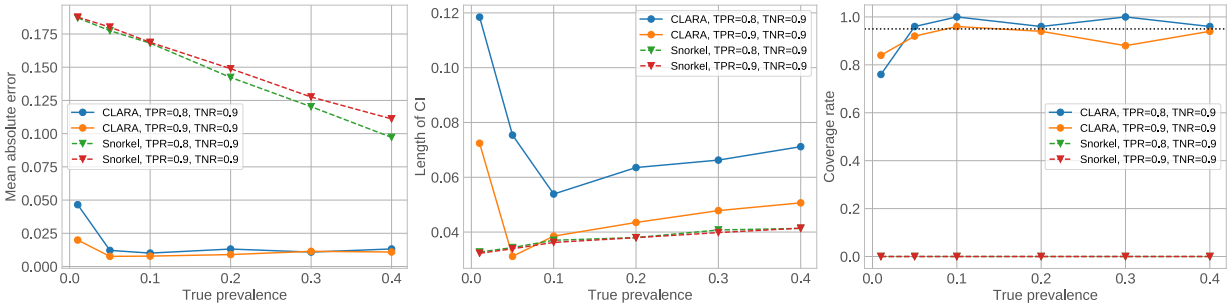
Figure 10: Distribution of classifier scores for two classes, $\sigma_0 = \sigma_1 = 2$

[6]https://github.com/snorkel-team/snorkel



(a) Mean absolute error (b) Length of confidence interval (c) Coverage rate of 95% CI

Figure 11: Prevalence estimation using CLARA with classifier score



(a) Mean absolute error (b) Length of confidence interval (c) Coverage rate of 95% CI

Figure 12: Prevalence estimation using CLARA and Snorkel

7.3 Human Labels with Classifier Scores

We also evaluate CLARA's ability to recover the true prevalence when we have available classifier score for each item (as described in Section 3.1). We first generate dataset in the same process as in Section 4.1, and then generate classifier score as follows: (1) draw a raw score x_i from $\mathcal{N}(\mu_0, \sigma_0)$ if true label is 0 and from $\mathcal{N}(\mu_1, \sigma_1)$ if true label is 1, here $\mu_0 \neq \mu_1$, and (2) use the sigmoid transformation of x_i as the classifier score of item i.

Here classifier scores is generated as an indicator of each item's true rating (with noise). Larger difference between μ_0 and μ_1 indicates more separable distribution between positive and negative class in terms of classifier scores. In practice, this matches the case of a well-performing ML model that provides good quality of classifier scores. In contrast, we might also have ML model that does not perform quite well, where the distribution of scores between the two classes are less separable.

In order to see CLARA's performance under different quality of ML scores, we generate two settings of scores as follows. The first setting uses $\mu_0 = -4, \mu_1 = 4$, and the second setting uses $\mu_0 = -1, \mu_1 = 1$, where in both cases $\sigma_0 = \sigma_1 = 2$. The resulting score distribution is shown in Figure 10. Setting 1 results in clear separation of the scores of the positive and negative classes, while scores from the setting 2 are less separable. For simplicity, we only experiment with the setting where $TPR = 0.8$ and $TNR = 0.9$. The rest of the parameters in the dataset and CLARA training are all identical with those in Section 4.1.

The results of applying CLARA with the two classifier scores are shown in Figure 11. Overall, CLARA's estimates with classifier score perform better compared with its "no score" counterpart. With classifier score, CLARA's prevalence has a smaller bias (Figure 11a), more efficient confidence interval (Figure 11b), without losing the accuracy in terms of coverage rate of the resulting confidence intervals (Figure 11c).

7.4 Inferring Prevalence with Snorkel

Following the same simulation and evaluation procedure in Section 4.1, we evaluate Snorkel's ability to recover the true prevalence in the input dataset. In Figure 12, we report the mean absolute error, length of the CI, and coverage rate of the 95% CI for both CLARA and Snorkel, using their probabilistic estimates on an item level to infer the prevalence. We see that Snorkel does not estimate the prevalence well compared to CLARA; after looking at the individual item level probabilities estimated from Snorkel, we concluded this is because they are not calibrated. This is not surprising as Snorkel was designed for ML models, not for prevalence estimation, where calibration is not necessary, e.g., when weighting the loss function.