SQUARE: A Benchmark for Research on Computing Crowd Consensus

Aashish Sheshadri
Department of Computer Science
The University of Texas at Austin
aashishs@cs.utexas.edu

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

Abstract

While many statistical consensus methods now exist, relatively little comparative benchmarking and integration of techniques has made it increasingly difficult to determine the current state-of-the-art, to evaluate the relative benefit of new methods, to understand where specific problems merit greater attention, and to measure field progress over time. To make such comparative evaluation easier for everyone, we present SQUARE, an open source shared task framework including benchmark datasets, defined tasks, standard metrics, and reference implementations with empirical results for several popular methods. In addition to measuring performance on a variety of public, real crowd datasets, the benchmark also varies supervision and noise by manipulating training size and labeling error. We envision SQUARE as dynamic and continually evolving, with new datasets and reference implementations being added according to community needs and interest. We invite community contributions and participation.

1 Introduction

Nascent human computation and crowdsourcing (Quinn and Bederson 2011; Law and von Ahn 2011; Lease 2011) is transforming data collection practices in research and industry. In this paper, we consider the popular statistical aggregation task of *offline consensus*: given multiple noisy labels per example, how do we infer the best consensus label?

While many consensus methods have been proposed, relatively little comparative benchmarking and integration of techniques has occurred. A variety of explanations can be imagined. Some researchers may use consensus methods to improve data quality for another research task with little interest in studying consensus itself. A natural siloing effect of research communities may lead researchers to develop and share new consensus methods only within those communities they participate in. This would lessen awareness of techniques from other communities, especially when research is tightly-coupled with domain-specific tasks. For whatever reason, it has become increasingly difficult to determine the current state-of-the-art in consensus, to evaluate the relative benefit of new methods, and to demonstrate progress.

In addition, relatively few reference implementations or datasets have been shared. While many researchers in other communities simply want to know the best consensus method to use for a given task, lack of a clear answer and reference implementations has led to predominant use of simple majority voting as the most common method in practice. Is this reasonable, or do we expect more sophisticated methods would deliver significantly better performance?

In a recent talk on computational biology, David Tse (2012) suggested a field's progress is often driven not by new algorithms, but by well-defined challenge problems and metrics which drive innovation and enable comparative evaluation. To ease such comparative evaluation of statistical consensus methods, we present SQUARE (**S**tatistical **QU**ality **A**ssurance **R**obustness **E**valuation), a benchmarking framework with defined tasks, shared datasets, common metrics, and reference implementations with empirical results for a number of popular methods. Public shared implementations and/or datasets are used when available, and we provide reference implementations for other methods.

We focus here on evaluating consensus methods which do not require feature representations for examples. This requires consensus to be computed purely on the basis of worker behaviors and latent example properties, excluding hybrid solutions which couple automatic classification with human computation. In addition to measuring performance across datasets of varying scale and properties, SQUARE varies degree of supervision, and we realistically simulate varying noise by preserving empirical traits of each dataset. Beyond empirical analysis, examining multiple techniques in parallel further helps us to organize and compare methods qualitatively, characterizing distinguishing traits, new variants, and potential integration opportunities. We envision SQUARE[1] as a dynamic and evolving community resource, with new datasets and reference implementations added based on community needs and interest.

2 Datasets

We begin by identifying and describing a number of public datasets that are online and provide the foundation for SQUARE 1.0. An early design decision was to include only datasets containing real crowd judgments, thereby increasing validity of experimental findings. While synthetic data can also be useful for sanity checks, carefully controlled experiments, and benchmarking, relatively little synthetic data has been shared. This likely stems from its lesser perceived value and a belief that it can be easily re-generated by others (provided that the generation process is fully and

[1] ir.ischool.utexas.edu/square

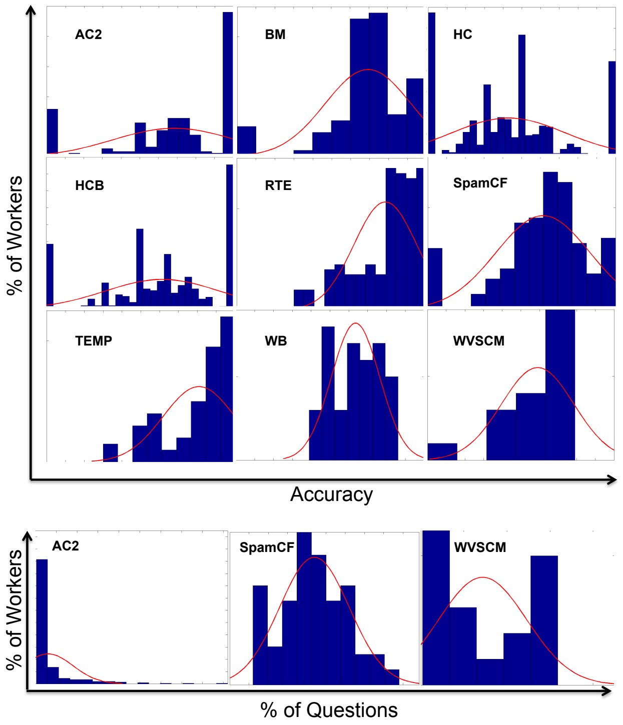

Figure 1.1: **Top**: a histogram shows the distribution of worker accuracies across nine of the datasets considered. **Bottom**: a histogram shows examples labeled per worker.

aptly described, and that reproduction does not introduce errors). As Paritosh notes (2012), reproducibility is both important and challenging in practice, and we posit such reproducibility is essential as a foundation for meaningful benchmarking and analysis. GLAD (Whitehill et al. 2009) and CUBAM (Welinder et al. 2010) valuably not only provide source code for the methods evaluated, but also for generating the synthetic data used in reported experiments. Most recently, Nguyen et al. (2013) present a different benchmarking study and framework based on synthetic data.

We also include only datasets with ground-truth *gold* labels for evaluation. We are agnostic here about the provenance of these gold labels and refer the reader to the source descriptions for more details. Nevertheless, the possibility of varying gold *purity* (Klebanov and Beigman 2010) should be considered in interpreting benchmark results. Not all studies creating gold labels report inter-annotator agreement statistics, and errors in gold could impact the comparative evaluation of methods considered (Cormack and Kolcz 2009).

Table 2.1 provides summary statistics for each dataset. Figure 1.1 plots a histogram of worker accuracies for nine of the datasets, above a histogram of the number of examples labeled per worker. While AC2 shows the oft-discussed exponential distribution of a few workers doing most of the work (Grady and Lease 2010), SpamCF and WVSCM show strikingly different work distributions. The empirical worker accuracy distributions shown here provide an important characterization of real crowd data, and our experiments which artificially vary label noise (Section 4) carefully preserve Figure 1.1's empirical worker accuracy distributions.

Dataset	Categories	Examples	Workers	Labels	MV *Acc.*
AC2	4	333	269	3317	88.1
BM	2	1000	83	5000	69.6
HC	3	3275	722	18479	64.9
HCB	2	3275	722	18479	64.8
RTE	2	800	164	8000	91.9
SpamCF	2	100	150	2297	66.0
TEMP	2	462	76	4620	93.9
WB	2	108	39	4212	75.9
WSD	3	177	34	1770	99.6
WVSCM	2	159	17	1221	72.3

Table 2.1: Public datasets used in the SQUARE benchmark.

NLP Datasets. The five Natural Language Processing datasets described below span three tasks: binary classification (BM, RTE, and TEMP), ordinal regression (AC2), and multiple choice selection (WSD).

AC2 (Ipeirotis, Provost, and Wang 2010) includes AMT judgments for website (ordinal) ratings $\{G, PG, R, X, B\}$.

BM (Mozafari et al. 2012) contains negative/positive sentiment labels $\{0, 1\}$ assigned by AMT workers to tweets.

RTE, **TEMP**, and **WSD** (Snow et al. 2008) provide AMT labels. RTE includes binary judgments for textual entailment (i.e., whether one statement implies another). TEMP includes binary judgments for temporal ordering (i.e., whether one event follows another). **WSD** includes ternary multiple choice judgments (not multi-class classification) for selecting the right sense of word given an example usage.

Other Datasets

WVSCM (Whitehill et al. 2009) includes AMT binary judgments distinguishing whether or not face images smile.

WB (Welinder et al. 2010) has AMT binary judgments indicating whether or not a waterbird image shows a duck.

SpamCF (Ipeirotis 2010) includes binary AMT judgments about whether or not an AMT HIT should be considered a "spam" task, according to their criteria.

HC (Buckley, Lease, and Smucker 2010; Tang and Lease 2011) has AMT ordinal graded relevance judgments for pairs of search queries and Web pages: *not relevant*, *relevant*, and *highly-relevant*. **HCB** conflates relevant classes to produce only binary labels (Jung and Lease 2011; 2012).

3 Models & Algorithms

Many models and estimation/inference algorithms have been proposed for *offline* consensus. Algorithms predominantly vary by modeling assumptions and complexity (Liu and Wang 2012), as well as degree of supervision. Since many workers label only a few items, more complex models are particularly susceptible to the usual risks of poor estimation and over-fitting when learning from sparse data. To limit scope, we currently exclude *online* methods involving data collection, as well as methods performing *spammer* detection and removal. We also exclude consideration of *ordinal regression* methods (Lakshminarayanan and Teh 2013), though multi-class classification methods are applicable (if not ideal). Finally, we do not consider open-ended tasks beyond multiple choice (Lin, Mausam, and Weld 2012).

While the space of proposed algorithms is vast (far beyond what space constraints permit us to cite, describe formally, or evaluate), we consider a variety of well-known methods which provide a representative baseline of current practice. In particular, we include models which vary from ignoring worker behavior entirely, modeling worker behavior irrespective of the example, and modeling varying worker behavior as a function of example properties. We briefly summarize and discuss each method below. Complementing empirical analysis presented in Section 4, our conceptual review of methods below emphasizes relationships between them, distinguishing traits, and possible variants.

3.1 Majority Voting (MV)

MV represents the simplest, oft-applied consensus method which often performs remarkably well in practice. MV assumes high quality workers are in the majority and operate independently, and it does not model either worker behavior or the annotation process. It is completely task-independent with no estimation required, provides lightening-fast inference, and trivially generalizes from binary classification to multi-class classification and multiple-choice. However, this simplicity *may* come at the cost of lower label quality.

While many alternative tie-breaking strategies might be used (e.g., using an informative class prior), our formulation follows the usual practice of unbiased, random tie-breaking. Similarly, while MV assumes high quality workers dominate, a lightly-supervised variant (not reported) could detect helpful vs. adversarial workers, filtering the latter out, or with binary labeling, exploit anti-correlated labels by simply "flipping" them (Kumar and Lease 2011).

3.2 ZenCrowd (ZC)

A natural extension to MV is to weight worker responses intelligently, e.g., by the worker's corresponding reliability/accuracy. Demartini, Difallah, and Cudré-Mauroux (2012) do so, using Expectation Maximization (EM) to simultaneously estimate labels and worker reliability. Their approach appears to be derived from first principles rather than earlier EM consensus methods (Dawid and Skene 1979; Smyth et al. 1995), or Snow et al. (2008)'s passing mention of such a simplified model. Like MV, ZC makes simplifying assumptions of workers acting independently and without modeling varying worker behavior as a function of each example's true class assignment. The modeling of one parameter per worker is more complex than MV but simpler than estimating a full confusion matrix per worker. This single parameter per worker also enables detection and handling of adversarial workers, which MV cannot do without additional light supervision. An advantage of having worker reliability as the only free parameter, besides reduced model complexity for sparse data, is that the model trivially generalizes to multi-class or multiple choice tasks with no increase in complexity (though by the same token may be less effective with increasing classes or choices).

While ZC is unsupervised as proposed, it can be fully-supervised by maximum-likelihood (ML), as in Snow et al. (2008), lightly-supervised by only providing an informative class prior, or semi-supervised by using gold labels where available and standard EM estimation otherwise (Wang, Ipeirotis, and Provost 2011).

3.3 Dawid and Skene (DS) & Naive Bayes (NB)

Dawid and Skene (1979)'s classic approach models a confusion matrix for each worker and a class prior, using EM to simultaneously estimate labels, confusion matrices, and the prior. Snow et al. (2008) adopt the same model but consider the fully-supervised case of ML estimation with Laplacian (add-one) smoothing. Like MV and ZC, workers are assumed to operate independently (Wang, Ipeirotis, and Provost 2011). Unlike MV and ZC, confusion matrices let DS/NB capture differential worker error behavior as a function of each example's true class. For example, unlike MV and ZC, DS/NB can detect and model a worker who produced perfect labels for examples of one class and opposite (adversarial) labels for the other class. While this greater modeling power can exploit more specialized statistics, sparsity can be more problematic. Also, while confusion matrices easily generalize to the multi-class labeling task, they do not generalize to the multiple choice selection task, where available choices are independent across examples.

Like ZC, DS/NB can be generalized to semi-supervised and lightly-supervised cases. A variant estimation procedure can distinguish correctable bias vs. unrecoverable noise (Wang, Ipeirotis, and Provost 2011). Whereas MV is agnostic of worker behavior, and ZC models worker behavior as irrespective of the input, DS/NB model varying worker behavior given an example's true underlying class. Moreover, whereas ZC models a single parameter per worker, DS/NB model one free parameter per class per worker.

3.4 GLAD

Like ZC and unlike DS/NB, GLAD (Whitehill et al. 2009) models only a single parameter per worker (the *expertise* α), with similar tradeoffs in modeling complexity. Like ZC/DS, GLAD uses unsupervised model estimation via EM, but estimation is more complex, requiring gradient ascent in each *M-step*. Like DS/NB, GLAD models varying worker behavior as a function of the input example. However, rather than considering the underlying class, GLAD models example difficulty β. An extension to multi-class is described (but not found in their public implementation). Like MV and ZC, GLAD easily generalizes to multi-choice selection classification. Like ZC and DS/NB, gold data may be used for supervision when available (e.g., fixing known labels in EM).

3.5 Raykar 2010 (RY)

DS and NB both estimate a confusion matrix, while DS imposes a class prior and NB uses Laplacian (add-one) smoothing. Raykar et al. (2010) propose a Bayesian approach to add worker specific priors for each class. In the case of binary labels, each worker is modeled to have bias toward the positive class α_i (sensitivity) and toward the negative class β_i (specificity). A Beta prior is assumed for each parameter. As with ZC, DS, and GLAD, an unsupervised EM method is derived to simultaneously estimate labels and model parameters (like GLAD, involving gradient descent).

RY's novelty lies in using an automatic classifier to predict labels, but this classifier also requires a feature representation of examples. However, when such a representation does not exist, as here, the method falls back to maximum-a-posteriori (MAP) estimation on DS, including priors on worker bias to each class. The multi-class extension is made possible by imposing Dirichlet priors, on each worker's class bias, and the class prior itself. However, the presence of class specific parameters inhibits extension to multi-choice, where the available choices are independent for each example.

3.6 CUBAM

Methods above model annotator noise and expertise (GLAD, ZC), annotator bias (DS,NB,ZC), and example difficulty (GLAD). Welinder et al. (2010) incorporate all of these along with a normalized weight vector for each worker, where each weight indicates relevance to the worker. Like prior assignments in RY, a Bayesian approach adds priors to each parameter. Worker labels are determined by an annotator-specific threshold τ_j on the projection of the noisy/corrupted input x_i and worker specific weight vector w_j. Probability of label assignments is maximized by unsupervised MAP estimation on the parameters, performing alternating optimization on x_i and worker-specific parameters $< w_i, \tau_j >$ using gradient ascent. Apart from label estimates, the surface defined by projection $w^T \tau_j$ enables viewing worker groupings of bias and expertise. CUBAM can generalize to multi-class classification but not multi-choice selection. No direct supervised extension is apparent.

4 Experimental Setup

This section describes our benchmarking setup for comparative evaluation of consensus methods (Section 3). We vary: 1) the dataset used and its associated task; 2) the degree of supervision; and 3) the level of annotation noise.

1. Data and Task. All experiments are based upon real-world crowdsourcing datasets. Whereas our first set of experiments measure performance on each dataset as-is, our second set of experiments simulate carefully-controlled increase or decrease in annotation noise, as discussed below.

2. Degree of supervision. We evaluate unsupervised performance and 5 degrees of supervision: 10%, 20%, 50%, 80%, and 90%. In each case, we use cross-fold validation, i.e. for the 10% supervision setting, estimation uses 10% train data and is evaluated on the remaining 90%, this procedure is repeated across the other nine folds, finally, average performance across the folds is reported. We report unsupervised performance on the 10-fold cross-validation setup, using 90% of examples in each fold for estimation (*without* supervision) and report average performance.

In the unsupervised setting, uninformed, task-independent hyper-parameters and class priors are unlikely to be optimal. While one might optimize these parameters by maximizing likelihood over random restarts or grid search, we do *not* attempt to do so. Instead, with *light-supervision*, we assume no examples are labeled, but informative priors are provided (matching the training set empirical distribution). Finally, *full-supervision* assumes gold-labeled examples are provided. To evaluate ZC, RY, DS and GLAD methods under full-supervision, labels are predicted for all examples (without supervision) but replaced by gold labels on training examples.

3. Label noise. We present two sets of experiments. Whereas our first set of experiments measure performance on each real-world dataset as-is (Section 5.1), our second set of experiments (Section 5.2) simulate more or less label noise for each dataset while rigorously maximizing data-realism (and thereby the validity and generalization of empirical findings). Data-realism is maximized by: 1) preserving which workers labeled which examples in each dataset, altering only the labels themselves; and 2) measuring the variance of each dataset's worker accuracy empirical distribution and using this variance to parameterize a normal distribution from which worker accuracies are sampled.

Mean worker accuracy is varied from 0.55 to 0.95 by 0.10. We define a normal distribution with this mean and with variance following the dataset's empirical distribution of worker accuracies. For each real worker in the dataset, we discard his actual labels and instead sample a new accuracy w_i for him from this distribution. The worker's labels are then re-generated, matching gold with probability w_i. In the case of disagreement and multiple categories to choose from, a category assignment is made at random with no bias. Experimental setup otherwise follows our earlier setup.

Evaluation metrics. Presently the benchmark includes only accuracy and F_1 metrics. While a wide variety of different metrics might be assessed to valuably measure performance under alternative use cases, a competing and important goal of any new benchmark is to simplify understanding and ease adoption. This led us to intentionally restrict consideration here to two simple and well-known metrics. That said, we do plan to expand the set of metrics in future work, such as to varying loss functions and the benefit vs. cost tradeoff of improving performance by collecting more gold examples (e.g., from an expert) for supervision. Significance testing is performed using a two-tailed, non-parametric permutation test (Smucker, Allan, and Carterette 2007).

Implementations. We used existing public implementations of DS, GLAD and CUBAM algorithms. We provide open source reference implementations in SQUARE for the other methods considered: MV, NB, ZC, and RY.

4.1 Experimental Details of Methods

A variety of important implementation details impact our evaluation of methods. We discuss these details here.

ZC in its proposed form does not impose priors on parameters (Demartini, Difallah, and Cudré-Mauroux 2012). Our implementation does impose priors on both the label category distribution and worker reliabilities. A Beta prior was assumed for worker reliability, and a Dirichlet prior was imposed on label categories. In each experimental setup, the workers were assigned the same prior distribution. In the unsupervised setup, the prior distribution on worker reliability had a mean of 0.7 and a variance of 0.3 (as with RY below) and the label categories were assumed to be uniformly distributed. In the lightly-supervised and fully-supervised setups, both the worker reliability and label category prior parameters were estimated from the train split. The worker reliability prior parameters were set by computing average ML estimates for each worker's reliability in the train split.

NB was implemented to learn each worker's full confusion matrix, with Laplacian (add-one) smoothing (Snow et al. 2008). The algorithm was extended for multi-class using a one-vs-all approach. Since NB strictly depends upon training data, it was used only in the fully-supervised setting.

RY was implemented for binary labeling (Raykar et al. 2010). Beta priors were imposed on worker specificity, sensitivity and positive category prevalence. When unsupervised, the worker sensitivity prior was set to have mean 0.7 and variance of 0.3 (as with ZC above), the same distribution was assumed for specificity, and the label categories were assumed to be uniformly distributed. The lightly-supervised and fully-supervised settings had the prior parameters set to compute average ML estimates for each worker from the train split. Since RY was implemented for binary labeling, results are limited to datasets with two categories.

CUBAM, DS, and GLAD. Lacking supervision, CUBAM hyper-parameters were assigned default priors from the the implementation. Only the unsupervised case was evaluated since the hyper-parameters associated with distributions modeling question transformation, worker competence cannot be inferred from the train splits used.

DS predicts labels without any priors. Under the lightly-supervised and fully-supervised settings, category priors were assigned ML estimates inferred from the training fold.

GLAD is assigned uniform class label likelihood, with priors of 1 for task difficulty and 0.7 for worker expertise. Under the lightly-supervised and fully-supervised settings, class priors were set by ML estimates inferred from the training fold. Worker expertise was set as the average worker accuracy inferred from the training set, and as in the other implementations, the same prior was assigned to all workers. Finally the prior on task difficulty were set to 1.

Both CUBAM and GLAD implementations support only binary class estimation, hence results from the algorithms are reported only on datasets with binary labels.

5 Results

This section presents benchmarking results of methods across datasets and tasks, following the experimental setup described in Section 4. Whereas our first set of experiments measure performance on each real-world dataset as-is (Section 5.1), our second set of experiments (Section 5.2) simulates varying label noise for each dataset.

5.1 Results on Unmodified Datasets

We first report performance on unmodified datasets. Statistical significance testing is limited to results in Table 5.3.

Unsupervised. Figure 4.1 plots performance of each method across each dataset, showing *relative accuracy* in comparison to the baseline accuracy of majority vote (MV). Average performance across datasets is reported both for relative accuracy to MV (Figure 4.1 far right), and for actual accuracy and F_1 in Table 5.1. Classic DS achieves top average performance for both metrics. Each method except RY and ZC also outperforms the others on at least one dataset. More striking, on SpamCF and TEMP datasets, methods show no improvement over baseline MV. Evaluation of the methods under the unsupervised setting, when averaged across all binary labeled datasets, showed DS to outperform the rest of the methods, both on *avg. accuracy* and F_1 score; Table 5.1 tabulates results on all the methods.

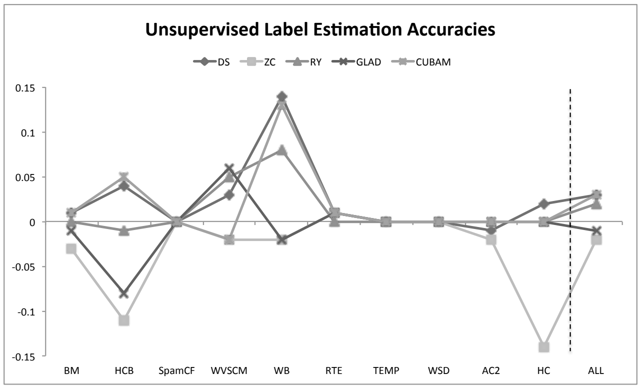

Figure 4.1: Unsupervised performance of consensus methods, as measured across seven binary labeled real datasets. Accuracy is plotted relative to a Majority Vote (MV) baseline. Average performance of methods across all datasets is shown at the right. On multiple choice WSD and multi-class AC2 and HC, results are reported only for DS and ZC.

Light-supervision. Figure 5.1 plots MV relative performance for each dataset. The effect of varying supervision is shown in a separate plot for each dataset. Table 5.1 presents average results across all datasets under varying supervision. DS is seen to outperform other methods with 10%-50% supervision on *avg. accuracy* and F_1 score, but RY performs best at 90% supervision. 80% supervision has RY and DS marginally outperforming each other on *avg. accuracy* and F_1 score respectively. Performance on each individual dataset, as observed in the unsupervised setting, did not highlight any individual method consistently performing best. Observations made earlier in the unsupervised case with regard to SpamCF and TEMP also carry-over here, with no improvement over MV for the first two.

Full-Supervision. As with previous light-supervision results, Figure 5.2 plots MV relative performance for each dataset. The effect of varying supervision is shown in a separate plot for each dataset. Table 5.1 presents average results across all datasets under varying supervision.

RY outperforms other methods with 50% or more supervision, contrasting earlier results where DS was consistently best. Note that DS outperformed the other methods for 10% and 20% supervision, but bettered RY only slightly. While NB was expected to outperform other methods with increasing supervision, DS and RY were seen to perform better.

Performance on individual datasets follows the same trend as in the averaged results, with the exception of WVSCM, where GLAD was superior. As with no supervision and light-supervision, TEMP shows similar trends, though MV outperformed DS and NB on SpamCF.

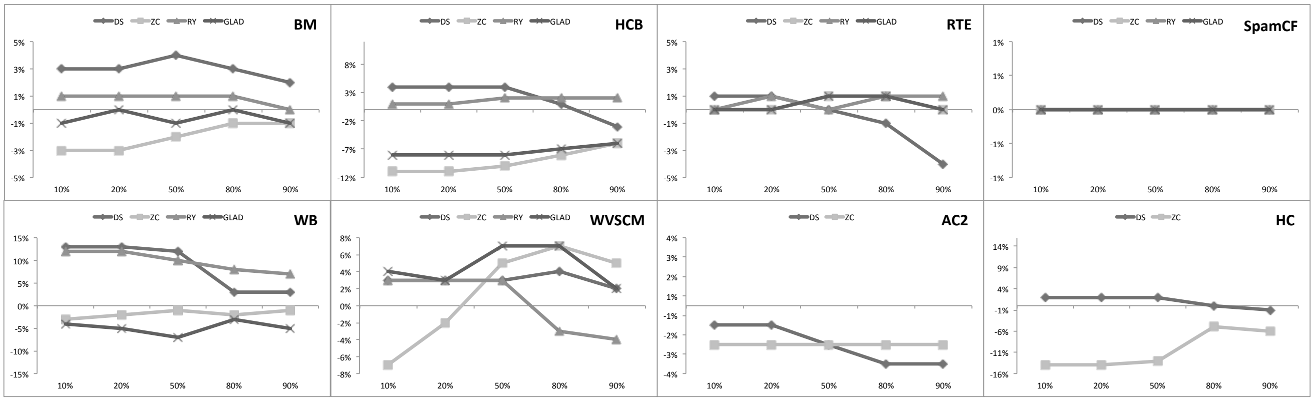

Figure 5.1: **Light-supervision.** Relative accuracy vs. baseline MV of 4 methods (DS, ZY, RY, and GLAD) across 8 (unmodified) crowd datasets (BM, HCB, RTE, SpamCF, WB, WVSCM, AC2, and HC) for 5 training conditions: 10%, 20%, 50%, 80%, and 90%. For multi-class AC2 and HC datasets, only multi-class methods DS and ZY are shown. Note the y-axis scale varies across plots to show dataset-dependent relative differences. Section 4's *Degree of supervision* provides details regarding supervision.

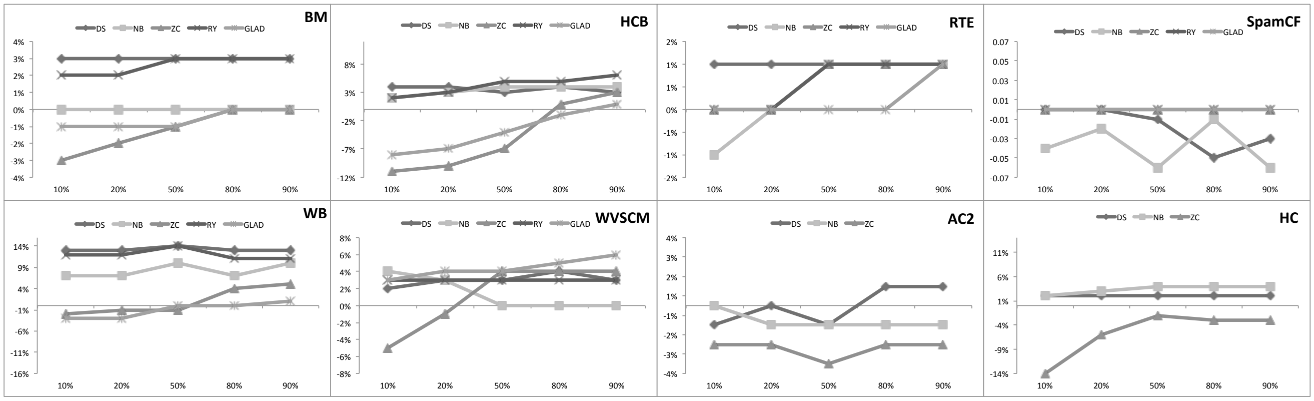

Figure 5.2: **Full-supervision.** Relative accuracy vs. baseline MV with full-supervision. See Figure 5.1 caption for further detail.

			Light-Supervision					Full-Supevision					
Method	Metric	No Supervision	10%	20%	50%	80%	90%	10%	20%	50%	80%	90%	Count
MV	Acc	79.2	79.2	79.2	79.2	79.3	79.3	79.2	79.2	79.2	79.3	79.3	0
	F_1	77.5	77.5	77.5	77.2	78.0	78.1	77.5	77.5	77.2	78.0	78.1	0
ZC	Acc	77.2	76.3	77.1	78.4	78.9	78.9	76.8	77.6	78.7	80.4	80.8	0
	F_1	76.4	74.2	75.7	76.8	77.7	77.7	75.4	76.1	77.0	79.2	79.6	0
$GLAD$	Acc	78.7	78.1	78.0	78.2	78.9	78.0	78.3	78.5	79.2	79.8	80.3	0
	F_1	77.3	76.8	76.7	77.0	78.6	77.6	76.9	77.1	77.6	79.0	79.5	0
NB	Acc	-	-	-	-	-	-	80.3	80.7	80.5	80.7	80.5	0
	F_1	-	-	-	-	-	-	79.1	79.0	78.5	78.5	78.9	0
DS	Acc	**82.2**	**82.3**	**82.2**	**82.0**	80.4	79.5	**82.2**	**82.2**	82.1	81.8	81.9	6
	F_1	80.2	80.2	80.0	79.4	78.9	77.9	80.1	80.0	79.6	79.2	79.9	7
RY	Acc	80.9	81.6	81.6	81.5	**80.5**	**80.1**	81.9	82.0	**82.5**	**82.3**	**82.3**	5
	F_1	79.1	79.6	79.5	79.2	78.8	78.8	79.8	79.9	79.9	80.4	80.4	4
$CUBAM$	Acc	81.5	-	-	-	-	-	-	-	-	-	-	0
	F_1	79.8	-	-	-	-	-	-	-	-	-	-	0

Table 5.1: **Results on unmodified crowd datasets.** Accuracy and F_1 results when averaged over all seven binary datasets (BM, HCB, RTE, SpamCF, TEMP, WB, and WVSCM) for varying supervision *type* (none, light, and full) and *amount* (10%, 20%, 50%, 80%, and 90%). Maximum values for each metric across methods in each column are bolded (Accuracy) and underlined (F_1). As a simple summary measure, the final column counts the number of result columns (out of 11) in which a given method achieves the maximum value for each metric. Results of statistical significance testing (50% condition only) appear in Table 5.3.

Discussion. CUBAM, with relatively weaker assumptions, was expected to perform best. This was seen on HCB, one of the noisier datasets considered (see Figure 1.1 for its worker accuracy histogram). However, on SpamCF, a dataset with a similar noise profile to HCB, all methods comparable performance to MV. A possible explanation is that SpamCF is far smaller than HCB, challenging estimation. On the flip side, on TEMP and RTE datasets, where workers are mostly accurate, MV appears sufficient, with more complex models providing little or no improvement.

Across experimental setups, GLAD consistently performed best on WVSCM but was outperformed on other datasets. ZC performed similarly, and both model accuracy while bias is ignored. This highlights the usual value of using available domain knowledge and tuning hyperparameters intelligently. Of course, increasingly complex models make estimation more difficult, and beyond the estimation challenge, performance is also ultimately limited by modeling capability. For datasets in which its sufficient to model worker accuracies (i.e., there exists a close to optimal positive worker weight configuration), GLAD and ZC perform well with informed priors or supervision. But they appear to be less robust on datasets with biased or adversarial workers, where methods with weak assumptions like CUBAM appear to thrive. The consistent performance of RY, across datasets, when priors were well informed or when further consolidated with minimal gold standard, suggests sufficiency in model complexity to generalize over most of the real datasets considered. Consistent performance of DS, which is similar to RY (except for the inclusion of worker priors) further corroborates this analysis.

5.2 Results With Varying Noise

Whereas experiments in Section 5.1 measured performance on each real-world dataset as-is, we now present results of carefully varying the label noise for each dataset. We consider two levels of supervision, 20% and 80%. Table 5.2 reports results across noise conditions and methods. We do not report statistical significance of noise-based experiments.

With no supervision, RY performs best, contrasting results in the original setting where DS was superior. ZC is seen to perform considerably better compared to the performance with original labels. Another contrasting observation is the steep degradation in CUBAM performance with noise. With light-supervision, DS is seen to be the best performing method under noise; RY performs best under the noisiest condition. With full-supervision, we see that under the noisier conditions, RY performs best with 20% supervision while DS outperforms the rest with 80% supervision. The performance improved vastly across methods, even for low levels of supervision. The relative performance of NB to lightly-supervised and fully-supervised methods shows the same trend observed on unmodified datasets.

Discussion. MV shows little resilience to high to modest noise levels (55% to 75%), where it is consistently outperformed by other methods. This suggests common use of MV for its simplicity may sacrifice quality in such cases. However, differences are far more pronounced here than with original data, suggesting risk of evaluation artifact. With low noise (e.g., 95%), differences are far smaller, and moreover, the classic DS method performs near-flawlessly.

Dataset	Metric	Best Method-Types	Best Methods
BM	Acc	**5f**, 5l, 6lf, 7u	5-7
	F_1	**5f**, 5l, 6lf, 7u	5-7
HCB	Acc	**6f**, 5u, 7u	5-7
	F_1	**4f**, 5ulf, 6lf, 7u	4-7
RTE	Acc	**4f**, 2ulf, 3ul, 5uf, 6ulf	2-6
	F_1	**4f**, 2ulf, 3ul, 5uf, 6ulf	2-6
$SpamCF$	Acc	**7u**	7
	F_1	**7u**	7
$TEMP$	Acc	**6l**, 1u, 2ulf, 3ulf, 6u, 7u	1-3,6,7
	F_1	**6l**, 1u, 2ulf, 3ulf, 6u, 7u	1-3,6,7
WB	Acc	**4f**, 5ulf, 6lf, 7u	4-7
	F_1	**4f**, 5ulf, 6lf, 7u	4-7
$WVSCM$	Acc	**3l**, 3uf, 2ulf, 5ulf, 6ulf	2,3,5,6
	F_1	**3l**, 3uf, 2ulf, 5ulf, 6ulf	2,3,5,6

Table 5.3: **Statistical significance.** For each (unmodified) binary dataset (BM, HCB, RTE, SpamCF, TEMP, WB, and WVSCM) and quality metric (Accuracy and F_1), we report all (tied) methods achieving maximum quality according to statistical significance tests (Section 4). Methods are indicated by number (1=MV, 2=ZC, 3=GLAD, 4=NB, 5=DS, 6=RY, and 7=CUBAM) and supervision *type* by letter (u=none, l=light, and f=full). For each dataset-metric condition, the top scoring method-type pair is shown first in bold, followed by all tied method-type pairs according to significance tests. Given space constraints, statistical significance is reported only for the 50% supervision *amount* condition. The final column ignores supervision *type* distinctions.

While GLAD and ZC perform similarly on unmodified datasets, the relative superiority between methods is switched. Taken with the lackluster performance of CUBAM, this further evidences method sensitivity to dataset characteristics: greater model complexity does not necessarily produce better predictions. As highlighted in Section 5.1, setting intelligent priors on parameters when possible from domain knowledge matters. This is seen with unsupervised RY outperforming DS, as consequence of preset priors. In the lightly supervised setting, however, RY clearly shows its dependence on priors, while DS remains consistent, with no imposed prior distribution of worker bias.

6 Discussion & Conclusion

We began this paper by noting continued, frequent use of simple majority voting (MV) to produce consensus labels from crowd data, despite the many more sophisticated methods that have been proposed. To be provocative, does this practice reflect mere ignorance, naivety, or laziness, or do more sophisticated methods offer only modest benefit which is simply not worth the bother? Should "experts" of new methods make stronger claims of superior quality and advocate wider adoption as "best practices" for crowdsourcing?

We observed in our benchmark tests that MV was often outperformed by some other method. Moreover, for anyone willing to tolerate a modicum of additional complexity in modeling worker bias, the classic DS and its exten-

		No Supervision					Light-Supervision					Full-Supervision					
		Avg. Worker Accuracy					Avg. Worker Accuracy					Avg. Worker Accuracy					
Method	Fold Size	55%	65%	75%	85%	95%	55%	65%	75%	85%	95%	55%	65%	75%	85%	95%	Count
MV	20%	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
	80%	-	-	-	-	-	45.1	67.2	82.4	93.6	97.8	45.1	67.2	82.4	93.6	97.8	0
ZC	20%	56.6	77.8	85.2	**99.2**	99.8	55.9	70.9	85.1	**99.1**	**99.8**	66.2	85.9	**95.9**	**99.1**	99.7	5
	80%	-	-	-	-	-	57.8	67.7	86.6	**98.3**	**99.6**	85.9	90.0	95.4	**99.0**	**99.9**	5
$GLAD$	20%	52.6	70.3	84.8	98.4	98.6	53.7	70.8	85.0	98.4	98.7	63.8	71.0	87.3	98.6	99.7	0
	80%	-	-	-	-	-	47.1	69.3	85.9	97.8	98.9	69.6	81.1	95.7	98.4	99.7	0
NB	20%	-	-	-	-	-	-	-	-	-	-	82.2	86.1	93.8	97.2	98.8	0
	80%	-	-	-	-	-	-	-	-	-	-	84.0	88.0	95.0	98.6	99.6	0
DS	20%	45.0	85.3	91.2	99.1	**99.9**	48.3	**85.2**	**91.3**	99.0	99.7	75.3	89.8	95.6	**99.1**	**99.9**	5
	80%	-	-	-	-	-	47.1	**82.0**	**92.1**	98.0	99.4	**86.2**	**90.7**	**96.0**	**99.0**	99.8	7
RY	20%	**56.9**	**86.1**	**95.3**	99.1	99.7	**59.1**	69.9	86.9	98.9	**99.8**	**83.1**	**90.0**	95.8	98.9	99.8	7
	80%	-	-	-	-	-	**71.0**	78.9	89.1	98.1	99.4	85.7	90.3	95.3	98.9	99.7	4
$CUBAM$	20%	52.4	67.6	83.2	97.7	97.9	-	-	-	-	-	-	-	-	-	-	0
	80%	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

Table 5.2: **Results with injected noise.** Accuracy achieved by each method is averaged across all seven binary datasets (BM, HCB, RTE, SpamCF, TEMP, WB, and WVSCM) for three supervision *type* conditions (none, light, and full) and five simulated worker accuracy conditions (55%, 65%, 75%, 85%, and 95%). Injected noise respects empirical dataset properties (Section 4). For light and full-supervision, we report results from training on 20% vs. 80% of each dataset (5-fold cross-validation). In case of no supervision, folds are not used, with results shown arbitrarily in the 20% fold row. While most methods can be evaluated across supervision *type* conditions, NB must be fully-supervised and CUBAM unsupervised. For each result column, the maximum accuracy achieved for each method-fold pair is shown in bold. The final column simply counts the number of result columns (out of 15) in which the maximum value is achieved for each fold. Table 5.3 reports statistical significance.

sion RY (which effectively just adds priors on parameters) performed remarkably well across our tests. Is it a failing of SQUARE's current benchmark tests that we do not observe even more impressive improvements from other, more sophisticated methods? For example, we did not consider hybrid approaches requiring feature representations, nor did we consider worker filtering approaches prior to consensus.

We invite contributions of: 1) better benchmark tests which would more strikingly reveal such improvements; 2) better tuning of included methods in order to maximize their full potential; or 3) additional methods we did not consider. The value of demonstrating clear progress to potential adopters and sponsors would seem to be important for our field to tackle, as well as to better understand where our clear success stories are to date and identify the particular challenge cases motivating greater attention.

On the other hand, the fact that each method was seen to outperform every other method in some condition seems to validate the need both for producing a diversity of approaches, and for multi-dataset testing in making stronger claims of improvement and generalizable performance. The degree of empirical diversity observed was relatively surprising since we did not explicitly construct diverse tests, but merely relied upon "found" methods and data.

While using synthetic data usefully enables carefully controlled experimentation, it is important that we continue to strive and assess its realism when using it for comparative evaluation of methods. While our own noise simulation utilized the actual empirical variance of each dataset in parameterizing the normal distribution from which worker accuracies were sampled, it would have been better to sample from the empirical distribution directly without assuming normality, which worker accuracy histograms clearly show to simplify reality. It would be informative to survey, implement, and assess synthetic data conditions from prior studies, and richer descriptive statistics and better models of worker behavior (Klebanov and Beigman 2010) could provide new insights with intrinsic value, enable more realistic simulation, and let us benefit from faster simulation-based evaluations while preserving confidence of experimental validity and findings carrying-over to operational settings.

Qualitative comparison of techniques helped us to characterize distinguishing traits, new variants, and integration opportunities. Like other open source benchmarks, we envision SQUARE as dynamic and continually evolving, with new tasks, datasets, and reference implementations being added based on community needs and interest. In an independent and parallel effort, Nguyen et al. (2013) recently released another open source benchmark, based on synthetic data, which implements or integrates a subset of methods found in SQUARE plus ITER (Karger, Oh, and Shah 2011) and ELICE (Khattak and Salleb-Aouissi 2011). Another consensus method with reference implementation, MACE (Hovy et al. 2013), was also recently published. We will continue to update SQUARE's website as a central community resource as new developments become known to us, and we invite others to join us in advancing this community benchmark.

Acknowledgments. We thank Yinon Bentor and the anonymous reviewers for their time and valuable feedback. This material is based upon work supported by the National Science Foundation, under Grant No. 1253413, and by a Temple Fellowship.

References

Buckley, C.; Lease, M.; and Smucker, M. D. 2010. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *The Nineteenth Text Retrieval Conference (TREC) Notebook*.

Cormack, G. V., and Kolcz, A. 2009. Spam filter evaluation with imprecise ground truth. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 604–611. ACM.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* 20–28.

Demartini, G.; Difallah, D. E.; and Cudré-Mauroux, P. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proc. WWW*, 469–478.

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

Hovy, D.; Berg-Kirkpatrick, T.; Vaswani, A.; and Hovy, E. 2013. Learning whom to trust with mace. In *Proceedings of NAACL-HLT*, 1120–1130.

Ipeirotis, P.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 64–67. ACM.

Ipeirotis, P. G. 2010. Mechanical Turk: Now with 40.92% spam. December 16. http://www.behind-the-enemy-lines.com/2010/12/mechanical-turk-now-with-4092-spam.html.

Jung, H. J., and Lease, M. 2011. Improving Consensus Accuracy via Z-score and Weighted Voting. In *Proceedings of the 3rd Human Computation Workshop (HCOMP) at AAAI*, 88–90.

Jung, H. J., and Lease, M. 2012. Improving Quality of Crowdsourced Labels via Probabilistic Matrix Factorization. In *Proceedings of the 4th Human Computation Workshop (HCOMP) at AAAI*.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, 1953–1961.

Khattak, F. K., and Salleb-Aouissi, A. 2011. Quality control of crowd labeling through expert evaluation. In *Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds*.

Klebanov, B. B., and Beigman, E. 2010. Some empirical evidence for annotation noise in a benchmarked dataset. In *Proc. NAACL-HLT*, 438–446. Association for Computational Linguistics.

Kumar, A., and Lease, M. 2011. Modeling annotator accuracies for supervised learning. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 19–22.

Lakshminarayanan, B., and Teh, Y. W. 2013. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. *arXiv preprint arXiv:1305.0015*.

Law, E., and von Ahn, L. 2011. Human computation. *Synthesis Lectures on AI and Machine Learning* 5(3):1–121.

Lease, M. 2011. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the 3rd Human Computation Workshop (HCOMP) at AAAI*, 97–102.

Lin, C. H.; Mausam, M.; and Weld, D. S. 2012. Crowdsourcing control: Moving beyond multiple choice. In *AAAI HCOMP*.

Liu, C., and Wang, Y. 2012. Truelabel + confusions: A spectrum of probabilistic models in analyzing multiple ratings. In *Proc. ICML*.

Mozafari, B.; Sarkar, P.; Franklin, M. J.; Jordan, M. I.; and Madden, S. 2012. Active learning for crowd-sourced databases. *CoRR* abs/1209.3686.

Nguyen, Q. V. H.; Nguyen, T. T.; Lam, N. T.; and Aberer, K. 2013. An evaluation of aggregation techniques in crowdsourcing. In *Proceedings of the The 14th International Conference on Web Information System Engineering (WISE 2013)*.

Paritosh, P. 2012. Human computation must be reproducible. In *CrowdSearch: WWW Workshop on Crowdsourcing Web Search*, 20–25.

Quinn, A. J., and Bederson, B. B. 2011. Human computation: a survey and taxonomy of a growing field. In *Proc. CHI*, 1403–1412.

Raykar, V. C.; Yu, S.; Zhao, L. H.; and Valadez, G. H. 2010. Learning from crowds. *JMLR* 11:1297–1322.

Smucker, M. D.; Allan, J.; and Carterette, B. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 623–632. ACM.

Smyth, P.; Fayyad, U.; Burl, M.; Perona, P.; and Baldi, P. 1995. Inferring ground truth from subjective labelling of venus images. *Advances in neural information processing systems* 1085–1092.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. EMNLP*, 254–263.

Tang, W., and Lease, M. 2011. Semi-Supervised Consensus Labeling for Crowdsourcing. In *Proceedings of the ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*.

Tse, D. 2012. The Science of Information: From Communication to DNA Sequencing. In *Talk at the Frontiers of Information Science and Technology (FIST) Meeting*. December 14. Slides at: www.eecs.berkeley.edu/~dtse/cuhk_12_v1.pptx.

Wang, J.; Ipeirotis, P.; and Provost, F. 2011. Managing crowdsourcing workers. In *Winter Conference on Business Intelligence*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*, 2424–2432.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. R. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2035–2043.