Evaluation coder abilities and labeling quality in crowd-sourced judgments of populist elite critique[*]

Hauke Licht *Department of Political Science, University of Zürich*

Whereas anti-elite discourse is recuring phenomenon of contemporary political discourse in many Western democracies, comparative research on anti-elitism in electoral competition is currently stymed by the lack of measurement instruments that are capable to scale to large sets of partisan actors and windows of time. In this paper, I present and evaluate crowd-sourced measurement as an approach that is devised to overcome the scalability limitations of existing measurement instruments. I address three questions. First, I assess crowd coders' abilities to classify elite critique, as well as the variability in abilities across coders. Second, I investigate the quality of measurements obtained by aggregating crowd-sourced judgments, and how this quality changes as the number of coded instances and the number of judgments aggregated per instance are increased. Third, I examine how measurement quality differes between aggregation methods, comparing majority voting and a Bayesian model-based approache. I use both real judgments collected by Hua, Abou-Chadi, and Barberá (2018, 2019) and simulated judgments to answer these questions. My findings indicate that crowd coders exhibit on average rather low true-positive detection abilities compared to their performances in true-negative detection and that true-positive detection abilities vary more strongly across coders. Moreover, I show that aggregating increasing numbers of judgments per instance increases the quality of instance-level measurements as well as the overall accuracy of measurements, whereas the qualty gains achieved by increasing the number of coded instances are negligible. Finally, comparing changes in the accuracy of model-based and majority-voting induced labelings, I show that as numbers of judgments aggregated per instance is increased, (i) measurement quality improvements occur more quickly with model-based labeling and (ii) model-base lableling tends to outperform majority voting—particularly so when the number of coded instances relatively high. My study thus has important implications for reliability considerations pertaining to the crowd-sourced measurement of elite critique in political texts.

Keywords: crowd-sourced measurement, crowd coding, elite critique, labeling quality

Introduction

Anti-elitism is a recuring phenomenon of contemporary political discourse in many Western democracies (Oliver and Rahn 2016; Hobolt and Tilley 2016; Polk et al. 2017; Engler, Pytlas, and Deegan-Krause 2019). While often associated with populism (Taggart 1996; Mudde

[*] Report written for project group *Crowd-sourced measurement of anti-elite rhetoric.* Version: May 19, 2019. Author contact: hauke.licht@uzh.ch

and Kaltwasser 2013), anti-elitism is not genuinely populist (Barr 2009), and an important component of the electoral appeals of many different types of parties (Abedi 2004; Sikk 2012 ; Zulianello 2018; Engler 2018).

Whereas anti-elite discourse is thus widespread, comparative research on anti-elitism in electoral competition is currently stymed by the lack of measurement instruments capable of eliciting valid measurements of partisan actors' uses of anti-elite appeals for large sets actors of partisan actors and windows of time. This is due to the important limitations of Existing measurement instruments that either rely on a classical content-analytical approach (e.g. Jagers and Walgrave 2007), on expert surveys (Polk et al. 2017), or on a computerized dictionary-based approach (e.g. Pauwels 2011). The first two methods are limited in terms of their scalability, as they presupposes the availability of suitable experts with sufficient domain-specific knolwedge and/or require rather high resource investments per measurement unit (cf. Krippendorff 2004, 127–9). Dictionary-based measurement instruments of anti-elitism, in turn, exhibit questionable validity and can be considered ill-adapted to scale across political contexts (cf. Pauwels 2017; Bergman 2018).

In this paper I therefore explore the advantages and limitations of a third approach: one that combines the strengths of the content-analytical approach with the scalability advantage of crowd-sourced measurement (cf. Hua, Abou-Chadi, and Barberá 2018, 2019). I use both real crowd-sourced as well as simulated data to assess the properties of this measurement approach.

The remainder of this paper proceeds as follows. The next section states the goals of the analysis presented in this paper. Thereafter, I explain what data I use to reach these goals, and my empirical stratgy. Then follows a results section. The last section concludes.

Note on terminology

Before proceeding, a brief note on terminology is due: *Content-analytical measurement* involves tasking humans with the judgment of texts according to a coding scheme (**???**). The process of applying a coding scheme to texts is referred to as *coding*. The humans acting as content analyists are refered to as *coders* (in the statistical learning literature, coders are also referred to as *labelers*). A coder's coding of text yields a *judgment*. When the coding scheme is categorical, coders' judgments are also referred to as *classifications*. Texts constitute the level of measurement and are referred to as *instances* (a term borrowed from the statistical learning literature). Since instances are usually judged by multiple coders, *measurements* are obtain by aggregating coders judgments at the level of instances. A measurement obtained for an instance by means of judgment aggregation is referred to as *label*; and the measurements induced by judgment aggregation as *labeling* (again borrowing from the statistical learning literature). *Crowd-sourcing* is the distribution of a task that requires human judgment to a large set of workers Here, crowd-sourced measurement thus refers to the distribution of a content-analytical task to a large crowd of coders, who are thus referred to as *crowd coders*.

Goal statments

My first goal is to quantify coders' abilities to correctly classify texts. It is well documented that crowd workers are noisy labelers, that is, that they tend to make more mistakes than experts or trained coders when judging data [snow_cheap_2008; sheng_get_2008; ipeirotis_repeated_2014]. As the quality of judgments affects the quality of instance-level measurements (Dawid and Skene 1979), a first crucial step to assess the potential of a crowd-sourced measurement of anit-elitism thus is to estimate how noisy coders' judgments of elite critique are.

A second aim is to asses the extent to which aggregating noisy human judgments of elite critique affects measurement quality, and how this changes as the number of judgments aggregated per instance as well as the number of coded instances is increased. This question arises when one considers that depending on human coders' ability to correctly judge instances of text as well as the aggregation method used to obtain measurements at the level of instances, coding error induced by noisy coders may impair measurement quality. In fact, the nosier coders are, the more likely they agree in error (i.e., on assigning an text the wrong label). Coders with poor judgment abilities thus induce high agreement-in-error rates, which, in turn, inflates conventional inter-coder agreement metrics (Passonneau and Carpenter 2014), and tends to impair the quality of measurements obtain by means of majority voting or averaging relative to instances' true labels [Sheng, Provost, and Ipeirotis (2008); guan_who_2018].

A final goal is to compare how labelings induced by majority voting compares to labelings obtained by aggregating judgments using a Bayesian model that accounts for coders varying abilities.

Data and empirical strategy

Crowd-sourced codings of elite critique in textual data

I analyze both real and simulated human judgments of elite critique in political texts. The real data was originally collected by Hua, Abou-Chadi, and Barberá (2018, hereafter HAB) via the crowd-sourcing platform *FigureEight*. HAB have collected social media posts (instances) created by of accounts of parliamentary parties and those of their leaders in six Western European countries, and recruited crowd coders to judge a sample of these posts according to the following question:

> Does this tweet/post criticize or mention in a negative way political leaders or parties, institutions, governments, media, academics, financial elites, etc?

Coders were asked to answer Yes or No; all judgments obtained are thus binary.

HAB have crowd-sourced judgments from 352 different coders for a set of 5040 different instances. To control the quality of judgments, they have used 40 'gold-standard' instances (Ipeirotis, Provost, and Wang 2010). Gold-standard instances are texts for which 'true' labels are known. Before accepting a coder as judge in their task, HAB have require each to correctly judge at least seven out of ten gold-standard instances. In addition, such screener tasks were randomly seeded among instances to continously controll judgment quality. This use of gold-standard instances as screeners is considered best practice (Berinsky, Margolis, and Sances 2014, @benoit_crowd–sourced_2016).

Figure 1 shows that the number of judgments per gold-standard instance is high (it lies in the range [123, 165]), which is due to the fact that in HAB's data there are only few gold-standard instances but that each coder is asked to judge a minimum of ten gold-standard instances. Although varying proportions of coders have misclassified gold-standard-instances, *FigureEight*'s weighted majority voting scheme[1] induces correct labels of all but one gold-standard instances.

[1] On *FigureEight*, each coder is assigned a 'trust' score that is reflecting her coding quality. To obtain a labeling, a majority winner is induced using coders' trust scores as weights.

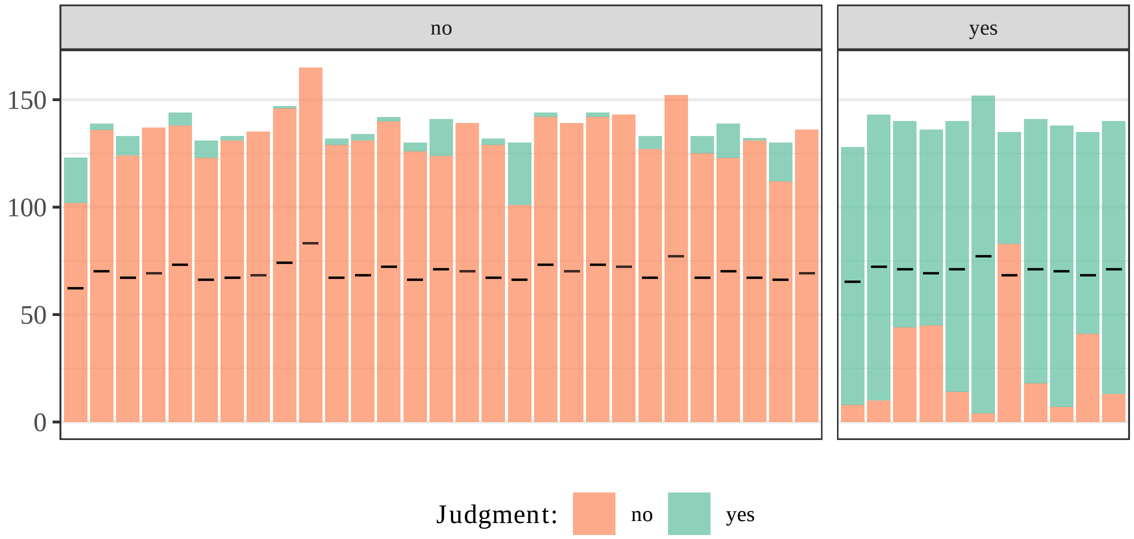

Figure 1: Labelings of gold-standard instances induced by coder-trust weighted majority voting. Plot panel columns separate instances in actual gold standard labels, i.e., in true negatives (left) and true positives (right). Vertical dashes plot on top of bars indicates decision threshold, illustrating that this aggregation method induce one false-negative labeling in gold standard items (seventh instance from left in right panel).

With regard to instances that were not in the gold-standard set, Table 1 shows that each of these free instances was coded three times, resulting in different yes:no patterns.

Table 1: Number of instances with different number of judgments in HAB's crowd-sourced elite critique coding data.

Label	Decision (yes:no)	Number of judgments	Number of instances
no	0:3	3	2454
no	1:2	3	918
yes	2:1	3	865
yes	3:0	3	763

Figure 1 and Table 1 already hint at substantial levels of disagreement among coders. Whereas it is common to aggregate judgments at the instance-level by means of majority voting, I additionally employ a Bayesian modeling approach here. This approach is motivated by the consideration that the crowd coder population is rather heterogenous in its abiltiy to correctly judge instances. In presence of such ability heterogeneity, majority voting can be shown to be biased relative to true labels [Sheng, Provost, and Ipeirotis (2008); guan_who_2018]. The Bayesian model I apply to induce instance-level labelings, in turn, estimates coders abilities and takes them into account when estimating instances' labels. This judgment noise-sensitive approach is expected to mitigate the bias found in majortiy-winner labelings.

The Bayesian Beta-Binomial by Annotator model

I assume that elite critique is a latent binary feature of political texts, and hence crowd coders act as human content-analysts whose judgments I want to aggregate at the instance-level to estimate whether a given instance features elite critique. The setup can be described as a four-tuple $\langle \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{Y} \rangle$, where

- $\mathcal{I}$ is the set of *instances* $i \in 1, \ \ldots, \ n$ distributed for crowd coding,
- $\mathcal{J}$ is the set of *coders* $j \in 1, \ \ldots, \ m$,
- $\mathcal{K}$ is the set of *classes* $k \in \{0, 1\}$ defined by the categorical coding scheme used during crowd-coding, and
- $\mathcal{Y}$ is the set of *judgments* $y_{i,j} \in \{0, 1\}$ recorded for instance i by coder j.

Importantly, while coders' judgments are observed, instances' true class membership (labels) $c_i \in \mathcal{K}$ are unknown *a priori* for all $i = 1, \ \ldots, \ n$. In this setup, an assignment of instances into classes obtained from a set of judgments $\rho(\mathcal{Y}) \Rightarrow \mathcal{C}$ is called a *labeling.* I obtain such labelings by fitting the following model to the judgment data:

Model 1: Bayesian hierarchical Beta-Binomial by Annotator model

$$
\begin{aligned}
c_i &\sim \text{Bernoulli}(\pi) \\
\theta_{0j} &\sim \text{Beta}(\alpha_0, \beta_0) \\
\theta_{1j} &\sim \text{Beta}(\alpha_1, \beta_1) \\
y_{ij} &\sim \text{Bernoulli}(c_i \theta_{1j} + (1 - c_i)(1 - \theta_{0j})) \\
\\
\pi &\sim \text{Beta}(1, 1) \\
\alpha_0/(\alpha_0 + \beta_0) &\sim \text{Beta}(1, 1) \\
\alpha_0 + \beta_0 &\sim \text{Pareto}(1.5) \\
\alpha_1/(\alpha_1 + \beta_1) &\sim \text{Beta}(1, 1) \\
\alpha_1 + \beta_1 &\sim \text{Pareto}(1.5)
\end{aligned}
$$

Note: c_i is the 'true' (unobserved) class of instance i, y_{ij} is coder j's judgment of instance i, θ_{0j} is coder j's specificity (true-negative detection rate), θ_{1j} is her sensitivity (true-positive detection rate), and π is the 'true' prevalence of the positive class. $(\alpha_., \beta_.)$ are hyperparameters governing the distributions of coders ability parameters. The distributions of coders' abilities are parameterized in terms of the means and scales of their respective hyper-distributions and given uninformative priors.

Carpenter (2008) refers to this model as the Beta-Binomial by Annotator (BBA) model.[2] Without knowing instances' true classes and no prior knowledge of coders' abilities, the BBA model is perfect to scrutinize coder abilities, as it allows to estimate the sensitivities and specificities of coders who have contributed their judgments of instances' class membership.

Evidence of crowd coders varying abilities in classifying elite critique

Estimation

In order to asses coders true-positive and true-negative detection rates in elite critique codings, I fit a BBA model to HAB's crowd-sourced codings data using an implementation in JAGS (Plummer 2003). Specifically, I first updated the model with 5000 burn-in iterations, and then obtain MCMC estimates for three chains of 10,000 iterations each, and only every 10th iteration was retained to mitigate within-chain auto-correlation in estimates.[3]

All priors were chosen to be uninformative, reflecting a situation where we have no domain-specific prior knowledge about instances classes, coders' abilities, and positive class prevalence. Because of the abundance of parameters of the BBA model ($2 \times J$ ability parameters, I instance class estimates, and prevalence and hyperparameter estimates), I used the Deviance Information Criterion to judge model convergence, and find the model to be well-converged and that chains are well-mixed.

Assessing coder ability estimates

Figure 2 plots posterior mean estimates of coders' sensitivity and specificity parameter estimates (Figure X in the Appendix reports mean estimates and 90% credibility intervals). Sensitivity refers to true-positive detection rate, that is, a coder's ability to correctly judge an instance that truely features elite critique. Conversely, specificity refers to true-negative detection rate, that is, a coder's ability to correctly judge an instance that truely features *no* elite critique.

[2] This name is due to its property that, given a conjugate beta prior, the posterior densities of instances' class membership follow a beta-binomial distribution.

[3] These choices are based on inspecting convergence and auto-correlation in initial models with fewer iterations and less (or no) thinning. Using these fitting parameters yields a shrinkage factor that is sharply declining within the first post-burn-in iterations and then is very close to 1. Moreover, with the thinning parameter set to 10, auto-correlation in posterior estimates is negligible.

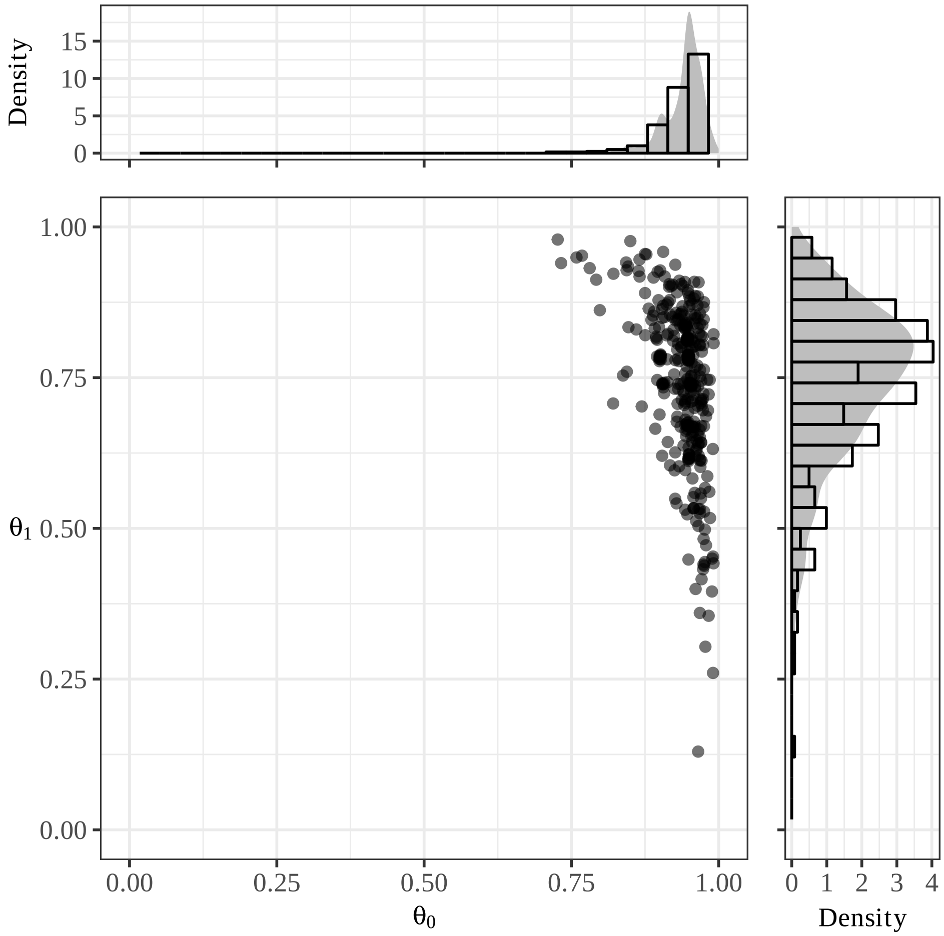

Figure 2: Mean posterior estimates of coders' abilities for elite critique classification. θ_0 and θ_1 refer to specificity and sensitivity, respectively.

Coders are found to perform generally very well in detection true negative instance (i.e., texts featuring no elite critique), while the recruited coders are more heterogeneous with regard to their ability to correctly judge positive instance. Importantly, there are not only coders that perform only moderately (those with mean sensitivity estimates in the range [.5, .7]), but also some adversarial coders with substantial posterior density mass below .5.

Inspecting the distribution of coders' mean posterior ability estimates (marginal distributions 2), we can conclude that coders are overall better in correctly judging texts featuring no elite critique than they are in judging those that feature elite critique.

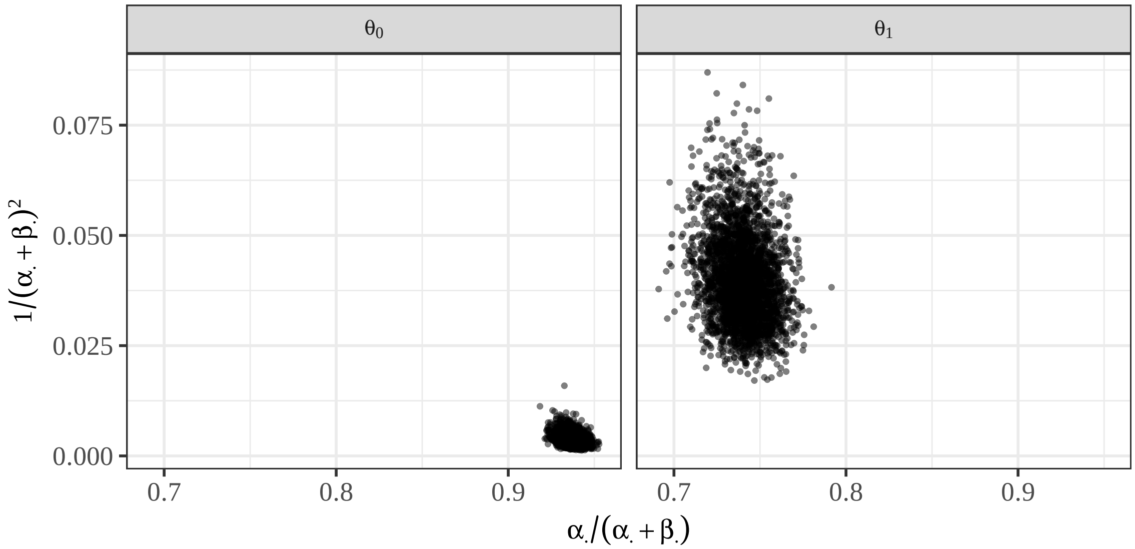

Figure 3: Posterior estimates of hyperparamters of ability distribtions in elite critique classification. Hyperdistribution parameters reported in terms of their means, $\alpha/(\alpha+\beta)$, and their invers squared scales, $1/(\alpha+\beta)^2$.

When looking at the distributions of mean posterior hyper-parameters estimates of sensitivity and specificity distributions in Figure 4, we see that this larger variation in mean sensitivity estimates in recruited coders abilities' shapes our posterior belief about the coder population at large: we are much more uncertain about the mean of the sensitivity distribution than about that of the specificity distribution (indicated by larger horizontal spread). Similarly, the precision of our belief in the coder population's sensitivity is lower than it is for specificity (indicated by the larger vertical spread).

Classification performance of model-based labeling

In order to assess how model-based labeling performs relative to instances' true labels, but true labels in HAB's data are knonw only for 40 out of total 5040, I use apply a simulation approach. Spercifically, I simulated judgments according to the data generating process captured by Model 1.

Simulation parameters

The true prevalence π has been simulated at 0.394, the mean posterior estimate in the elite critique codings data analyzed in the previous section. Coders' ability parameters were drawn from the Beta-distributions displayed in Figure 5.

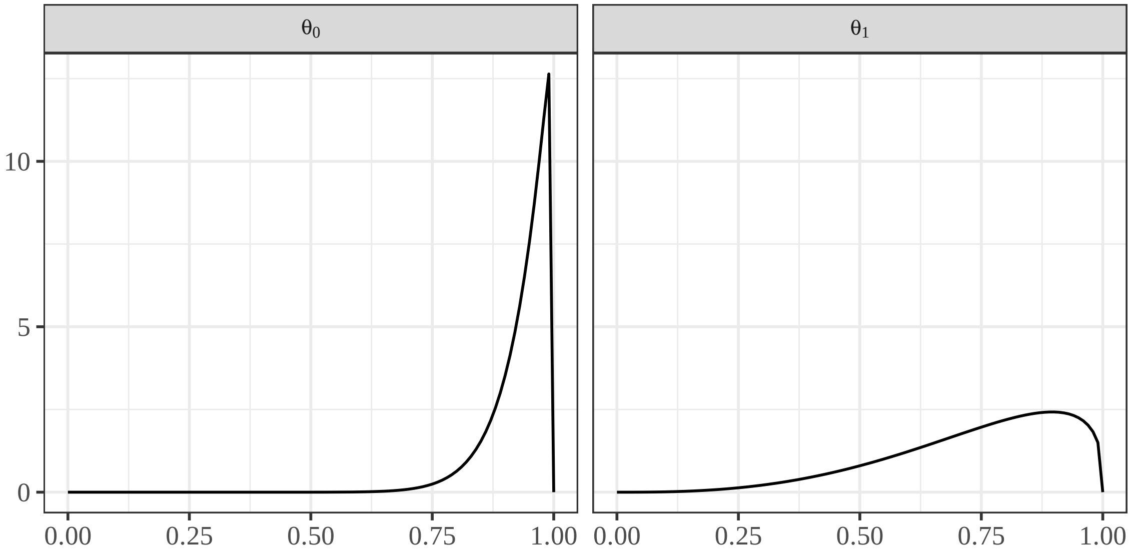

Figure 4: Densities of coder ability hyperdistribtions. θ_0 and θ_1 refer to specificity and sensitivity, respectively

From these specificity and sensitivity distributions, tuples of ability parameters were randomly drawn for 40 coders. The empirical distribution of coders' simulated abilities is shown in Figure 6.

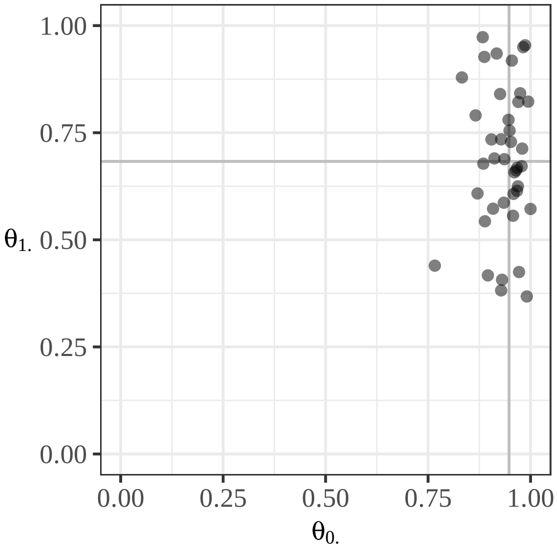

Figure 5: Distribution of coders' simulated ability parameters. $\theta_{0.}$ and $\theta_{1.}$ refer to specificity and sensitivity, respectively

In total, I have randomly sampled 500 instances from a Bernoulli distribution with the simulated π value of 0.394. Given a missingness rate of .75 (i.e., each instance was judged by only 10 out of total 40 coders), I have then generated total 10 judgments for each of these instances. In order to examine how model-based labelings of instances perform as a function of the total number of instances coded, n, I have then split from the entire 500 instances blocks of $n \in \{200, 250, \dots, 500\}$ instances, such that instances in smaller sized blocks are nested in respective larger sized blocks (i.e., all instances in block 200 are also in the 250 block, etc.). This is thought to imitate the situation where we collect increasing amounts of judgments for new instances.

In order to examine how repeated judgment of instances affects model-based labeling quality (i.e., increasing the number of judgments aggregated per coding, n_i), I have sampled different numbers of judgments for each instance, such that $n_i \in \{3, 4, \dots, 10\}$. Again, I have applied a nesting logic when splitting the entire judgments dataset. That is, for a given instance, all judgements that are in the $n_i = 3$ subset are also in the $n_i = 4, ..., 10$ subsets, etc. Mirroring the logic of fitting BBA models to differently n-sized but nested codings datasets, this simulation strategy mimics a situation where one collects increasing numbers of repeated judgments from *different* coders for a given instance.

Estimation

I fitted a BBA model to each $n \times n_i$ dataset.[4] I kept uninformative priors for coders ability parameters and ability distributions' hyper-parameters, but specified a more informative prior for the positive class prevalence, $\pi \sim \text{Beta}(2, 8)$. The latter choice was thought to reflect, where results from an early analysis suggest that the prevalence lies in the range [0.071, 0.429] with 90% confidence. Moreover, I constrained coder ability estimates to lie in the range [0.0001, 0.9999], since otherwise the slicer of the JAGS' MCMC algorithm would get stuck in infinite density regions.

Labeling quality of posterior classifications

For each model, I induced posterior labelings by assigning instances to the positive and negative classes according to whether their mean posterior class estimate c_i is $> .5$ (equivalent to assigning the mode posterior class estimate, i.e., majority voting, in binary classification).

[4] Because depending on n and n_i, convergence was achieved after varying numbers of iterations, and models also exhibited different levels of auto-correlation in chains, I ran the models with different burn-in and thinning configurations that are reported in Tables 4 and 5 the Appendix. However, the number of retained iterations was always adapted to the thinning parameter so that for each model 1000 estimates were obtained for each of three chains.

Posterior classification uncertainty

A point of concern that pertains to labeling quality is how certain we are about these posterior-estimate based assignments of instances into classes. To assess this question, I define *posterior classification uncertainty* (PCU) as the standard deviation in an instance i's label posterior estimates $\mathbf{c}_i = (c_{i1}, \ldots, c_{iT})$ across chains and iterations:

$$\mathrm{SD}(\mathbf{c}_i) = \sqrt{\frac{\sum_{t=1}^{T}(c_{it} - \bar{c}_i)^2}{T-1}},$$

where t indexes the t^{th} estimate, and here $T = 1000 \times 3$ (iterations times chains). The theoretical maximum of PCU is reached if an instance is estimated to be a member of the positive class exactly $T/2$ times, and this maximum approaches .5 as $T \to \infty$.

The distributions of PCUs for different values of n and n_i (Figure 7) illustrates that posterior classifications are comparatively uncertain if we aggregate only few judgments per instance. As n_i is increased, however, the PCU of most instances is reduced substantially, and in the extreme case of $n_i = 10$ is reduced to negligible levels in virtually all instances We can also see that increasing n contributes only little to change this pattern.

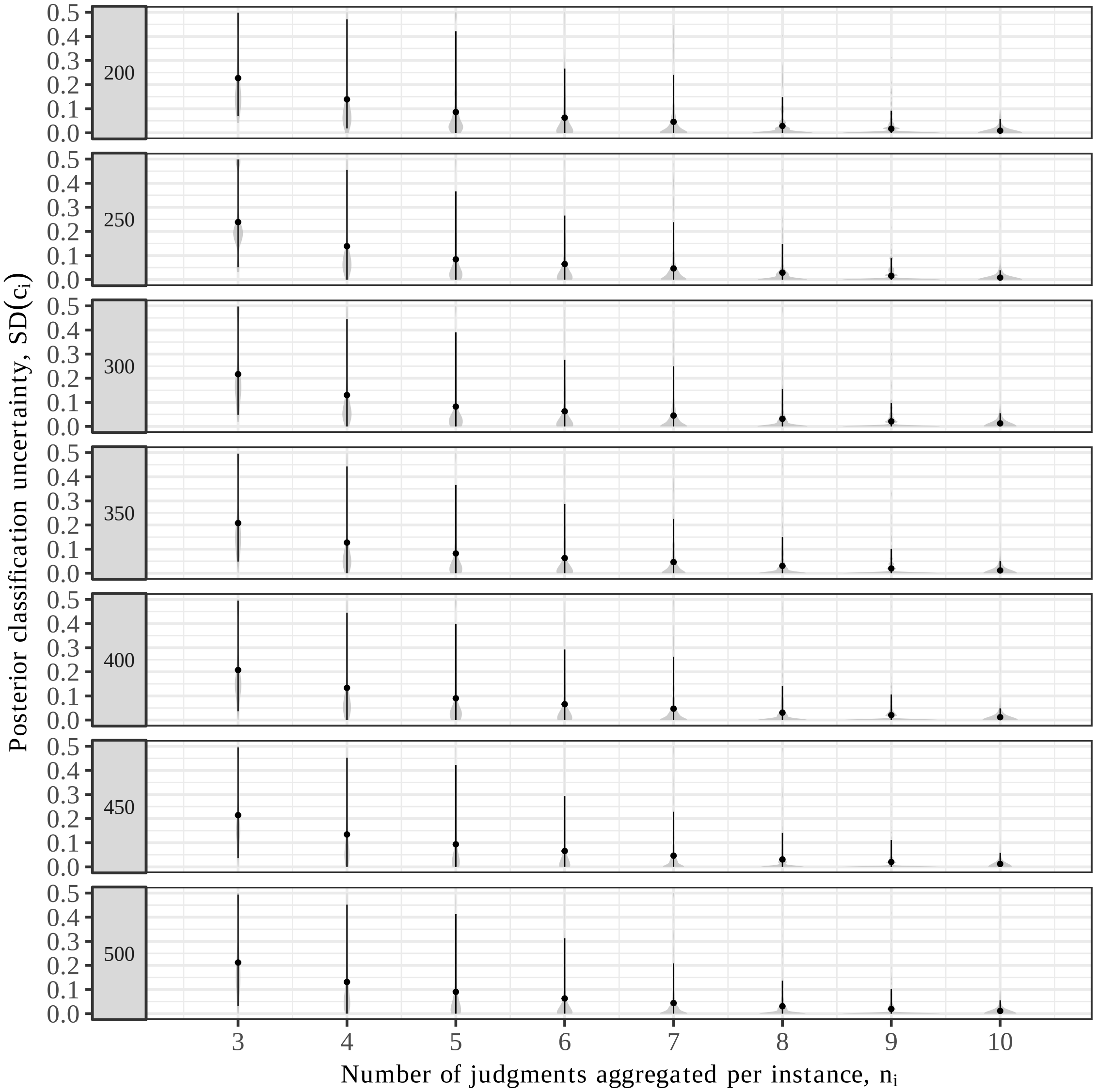

Figure 6: Distributions of posterior classification uncertainty by n and n_i. Results obtained by fitting BBA model to simulated elite critique judgments.

These results are summarized in Figure 8 that plots the change in mean PCU and the standard deviation in PCUs for different combinations of n and n_i. While we see now more clearly that differences in n make no significant difference—neither for changes in the mean PCU (left-hand panel), nor in the standard deviation of PCUs (right-hand panel)—, for all values of n there occurs some reduction in mean PCU values as n_i is increased. Similarly, The standard deviation in PCUs is lower for higher values of n_i. Taken together, we thus see that both average PCU values and their variability decreases as n_i is increased.

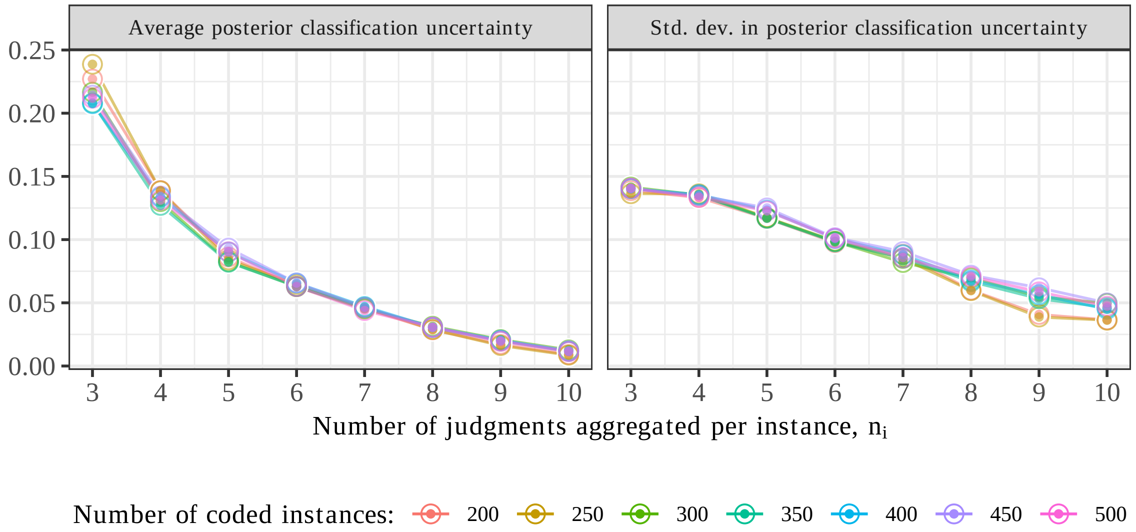

Figure 7: Change in posterior classification uncertainty. Results obtained by fitting BBA model to simulated elite critique judgments.

Figure 9 says little about the significance of the reductions in mean PCUs, however. Therefore, Table 6 in the Appendix reports the proportion of instances whose PCU is *increased* if aggregating $n_i + l$ instead of n_i judgments for $l \in (1, \ldots, 7)$ (displayed in columns 3–9). The proportion of instances with positive PCU change from n_i to $n_i + l$ can be interpreted as a significance test: if less than 5% of instances see increase, we are 95% confident that the change in PCU induced by collecting and aggregating an additional l judgments leads to an average decrease in PCUs. Indeed, adding just one or two more judgments per instance leads to an average decrease in instance-level PCUs in more than 95% of instances for most values of n_i. This changes little as n is increased. In order to reduce average PCU levels, it thus seems advisable in most cases to collect multiple judgments per instance (i.e., increase n_i) rather than judgments for new instances (i.e., increase n).

Posterior classification performance

In order to assess models' classification performance, I have first obtained instances' true class labels from the simulated judgments dataset. Next, I have compared model-based labelings of instances for each value of n, $n[i]$, each chain, and each iteration to instances' true class labels. Using this data, I have then computed statistics (mean, and 5% and 95% percentiles) of performance measures across chains and iterations for each combination of n and n_i. Figure 9 visualizes these statistics for different classification performance metrics. (Note the scaling of the y-axis.)

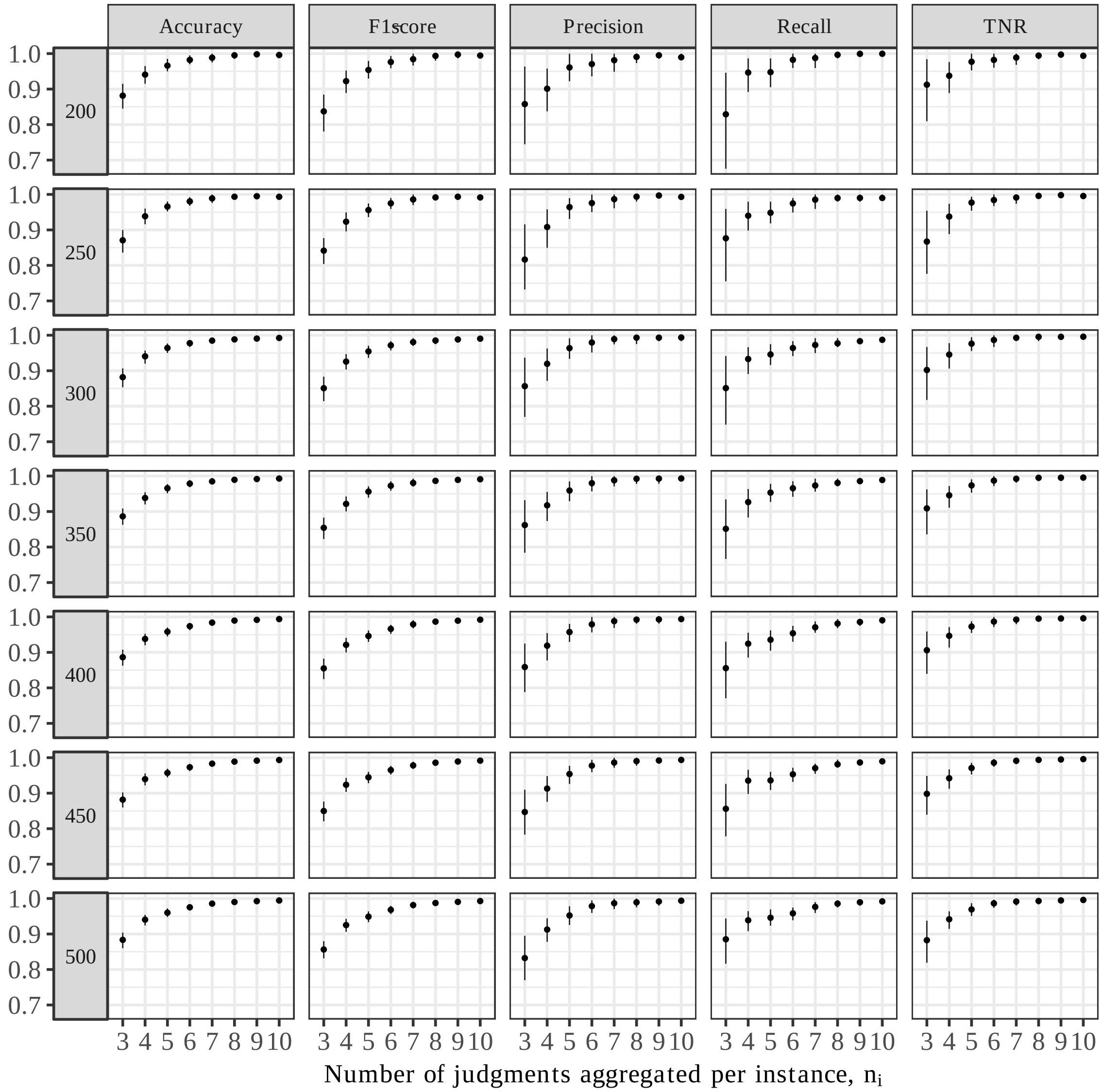

Figure 8: Mean and 90%-CIs of performance metrics by n and n_i. Results obtained by fitting BBA model to simulated elite critique judgments. Classifications induced by assigning instances' their posterior mode class, and comparing model-based classifications to simulated true values.

Figure 9 allows the following conclusions. *Accuracy*, defined as the proportion of correctly classified instances, increases as n_i is increased. There exist no substantial accuracy differentials across values of n, however. (Only 90%-CIs get tighter as n is increased.) *Recall* (also: true-positive rate, *TPR*), defined as the number of true-positive classifications over the sum of true-positive and false-negative classifications (i.e., over the number of all positive instances), exhibits much more variability across values of n_i. Similarly, *precision*, defined as the number of true-positive classifications over the sum of all positive classification (incl. false-positives), is relatively low for low n_i, but increases quiet rapidly as n_i is increased. The CIs for precision estimates are comparatively wide, however.

In contrast, *TNR*, the true-negative rate defined as the number of true-negative classifications over the number of all negative instances, is very high already for low values of n_i, and thus we observe no substantial improvements in the negative detection rates of models as the number of judgments aggregated per instance is increased. This is due to the fact that with a low positive instance prevalence, our simulated coders judge in expectation true negative instances about four to five times more than positive instance. There is thus more data and hence higher precision in negative detection.

Finally, the *F1-score* that combines recall and precision into one metric[5] increases significantly as n_i is increased. This is because in the denominator low precision depresses the F1-score. Note that the choice to report F1-scores is motivated by the presence of strong class imbalance. In presence of class imbalance, the accuracy is not a good performance criterion, since accuracy of about $1 - \pi$ can be achieved by simply assigning each instance the majority class label. As the F1-score takes both precision and recall into account, we can achieve high F1-scores only if our aggregation method performs also reasonable well in correctly classifying true-positive instances.

Table 7 in the Appendix reports F1-score changes as n_i is increased by l. It reports the proportion of iterations across chains for which an increase from n_i to $n_i + l$ judgments per instance induces a decrease in the F1-score. Again, this can be interpreted as a significance test with a a simple logic: as we want to see F1-score increases as l is increased, the smaller the proportion with a F1-score reduction, the better. Analyzing F1-score changes shows that while it is not possible to assert significant F1-score improvements for higher-values of n_i in models fitted to small-n subsets (due to smaller sample sizes and hence less power), we have reason to be confident that increasing the number of judgments if we have collect only few judgments so far usually leads to improvements in the models classification performance.

Comparison to majority-voting based classifications

A final goal of this analysis is to compare the classification performances of BBA models to those induced by majority voting. Table 2 shows that majority-voting induced and model-based labelings correlated rather strongly.

When plotting performance metrics for majority-voting induced classifications (Figure 10), we see that we get no false-positives with majority voting: the precision and TNR of all models are perfect (false-positives factor in the denominator of these metrics). What is more, because of the strong class imbalance, the accuracies are also very close to perfect. The

[5] The formula is F1-score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Table 2: Correlation between model-based classifications and majority voting

	Number of judgments per instance							
n	3	4	5	6	7	8	9	10
200	0.900	0.856	0.946	0.915	0.926	0.905	0.989	0.968
250	0.772	0.885	0.933	0.893	0.926	0.918	0.992	0.975
300	0.875	0.865	0.931	0.904	0.931	0.904	0.972	0.952
350	0.869	0.855	0.918	0.911	0.935	0.912	0.970	0.947
400	0.870	0.858	0.902	0.896	0.927	0.923	0.953	0.943
450	0.833	0.829	0.899	0.930	0.935	0.931	0.949	0.927
500	0.800	0.853	0.889	0.917	0.922	0.909	0.946	0.938

recall/TPR are much lower, however, and we see that in contrast to model-based labelings, F1-scores improve slower as n_i is increased.

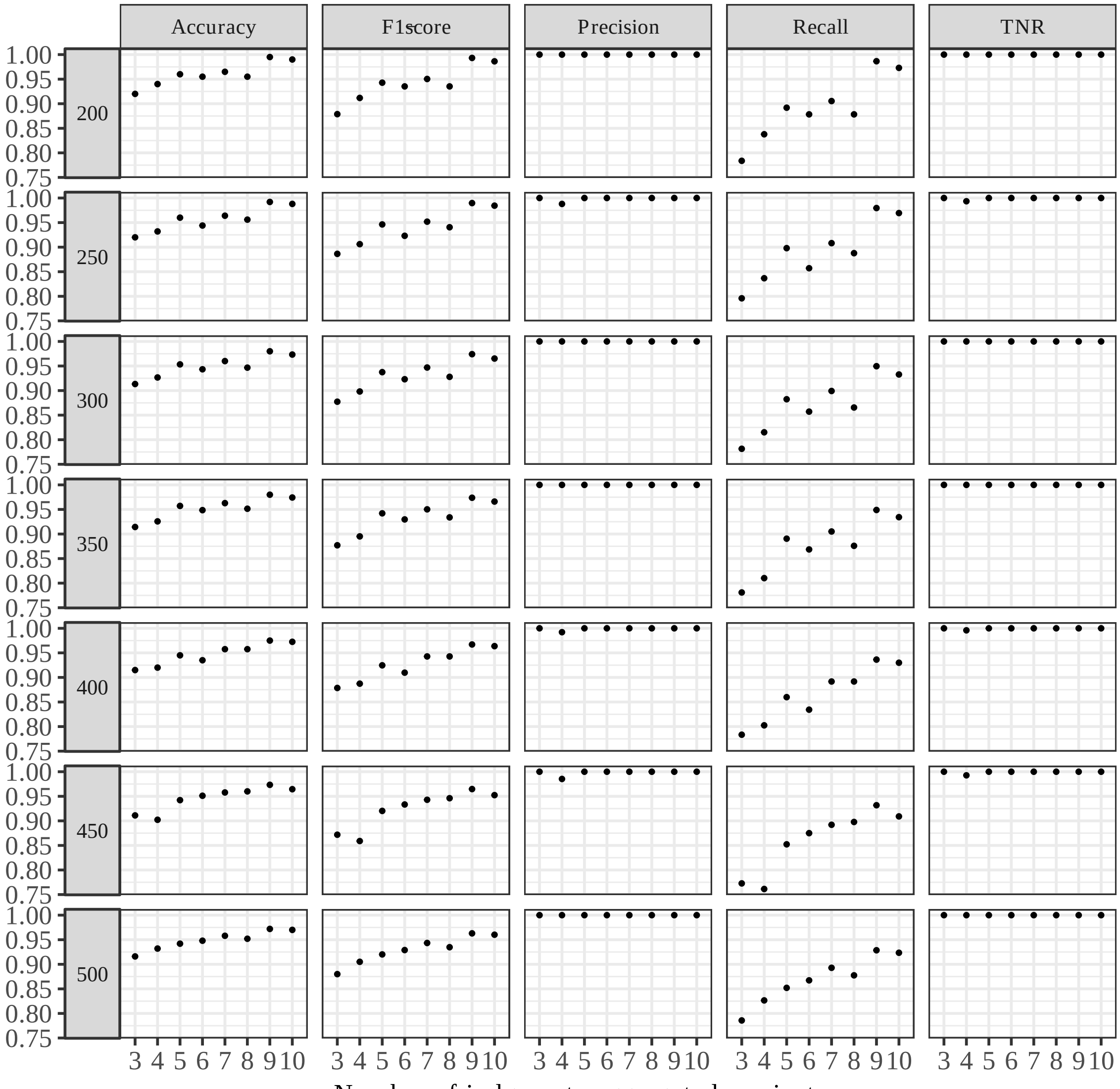

Figure 9: Performance of majority voring by n and n_i. Classifications obtained by inducing majority winner (with random tie-breaking for even n_i) in simulated anti-elitism codings.

Given that we obtain posterior classifications for each iteration and each chain of each model, we can also compute how confident we can be that BBA models and majority voting induce different classification performances. These differences are illustrated for F1-scores in Figure 11.[6]

[6] To obtain 90% confidence bounds, the majority-voting based F1-score point estimates have been subtracted from scores induced by model-based classifications of each iteration. This yielded 3000 differences that were aggregated into means and 90%-CIs.

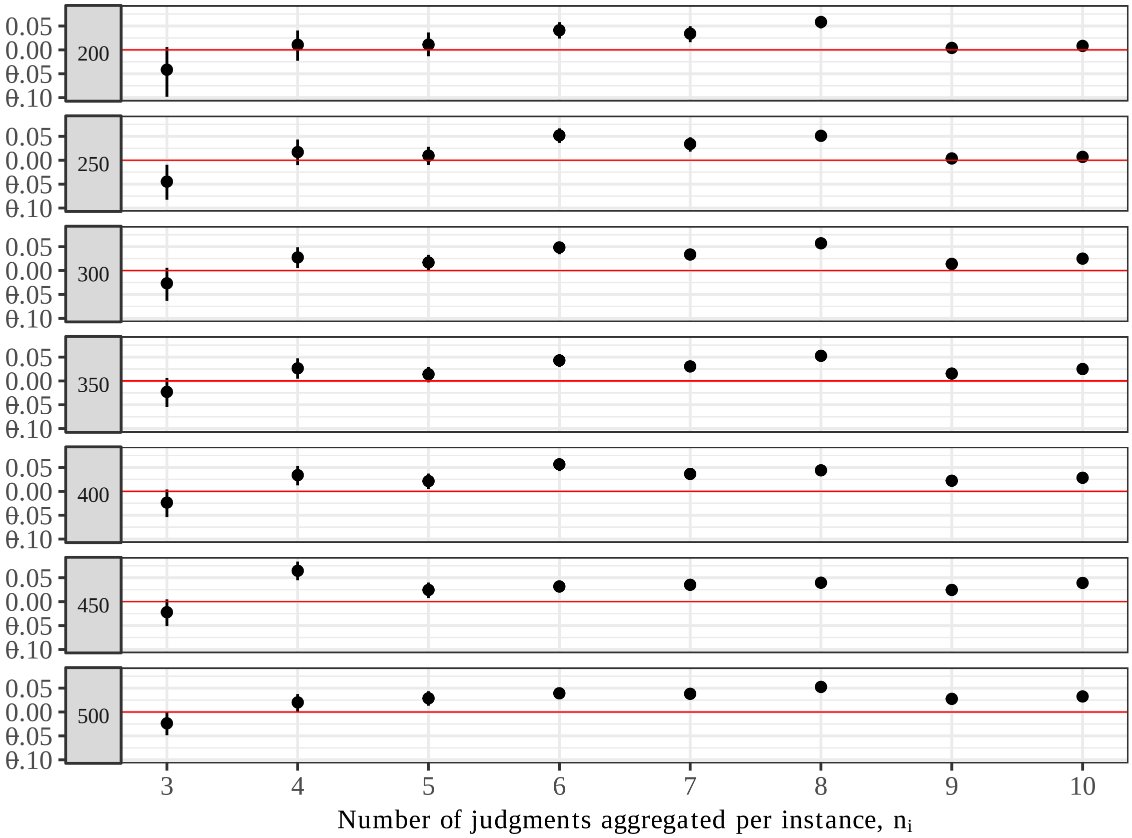

Figure 10: Mean differences and 90%-CIs between F1-scores induced by majority voting and model-based classifications. Positive (negative) differences indicate superiority of model (majority voting).

Figure 11 shows that while majority voting tends outperform model-based classifications in terms of F1-scores for $n_i = 3$ across values of n (though not significantly for all n), we see that for intermediate n_i model-based classifications tends to perform (significantly) better. Specifically, for $5 < n_i < 9$, we are 95% certain across values of n that model-based classification yield higher F1-scores than does majority voting. The fact that the advantage of model-based labeling tends to level-off at high n_i shows that it induces performance improvements more quickly, whereas majority voting also achieves high classification performances as n_i approaches 10. Thus, when collecting additional judgments per instance is costly, using model-based aggregation is a better choice in terms of classification performance. To see the significance of this result, consider again that the F1-score takes both precision and recall into account, and hence one can achieve high F1-scores only if the aggregation method of choice performs also reasonable well in correctly classifying minority class instances.

Replicating simulation results in real judgments

Whereas we cannot assess classification performance beyond the gold-standard instances in HAB's data, we can examine how posterior classification uncertainty changes as the number of judgments aggregated per instance is increased. The motivation if this assessment is that, against the background of the simulation study, we know that increasing n_i leads to both average PCU decreases and classification performance improvments. If we were to find PCU decreases as n_i in real judgment data, we would have reason to belief that classification performance improves, too.

Hence, I have replicated HAB's original crowd-sourcing and collected seven additional judgments for a stratified sample of total 300 non-gold-standard instances in HAB data. Specifically, I have stratified instances by posterior class estimate and PCU values, and sampled 12 estimated true and 18 estimated negative instances in each of ten equally sized PCU-percentile folds.

I have then fitted a BBA model to judgments of these 300 instances with $n_i = 3, \ldots, 10$, using the same specifications as for the models fitted to the $n = 300$ simulated judgments.[^123]

[^123] Again, DIC estimates mixed nicely and chains converged quickly. Autocorrelation was neglible using the thinning parameter values shown in Table in the Appendix.

Figure X shows that while average PCU values decrease as n_i is increased, the variation across instances does not decrease substantially. (Distributions are shown in Figure X in the Appendix). This stands in contrast to the results in the results obtained using simulated judgments data, and suggests that there is a set of instances that cannot be classified with high certainty even if additional judgments are collected. This hints at ambiguity in the coding scheme or inherent difficulty of these items (cf. Carpenter 2008), and warrants further investigation.

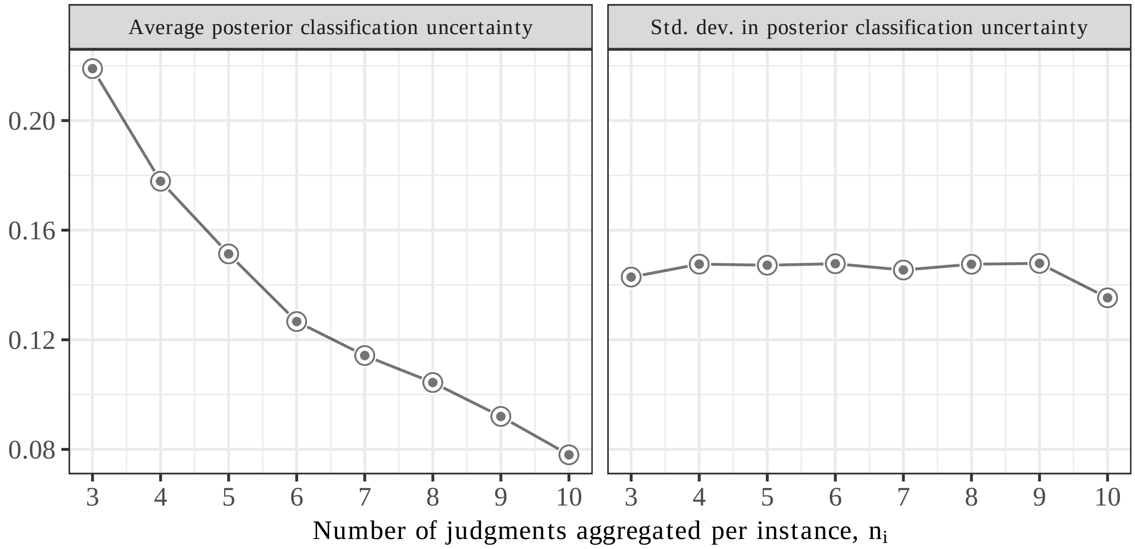

Figure 11: Change in posterior classification uncertainty. Results obtained by fitting BBA model to repeatedly judged elite critique codings.

Figure X indeed shows that there is a small portion of instances whose PCU values tend to increase as more judgments per instance are collected. Though these instances cannot be classified with high certainty, the model-based varying-n_i approach I use here allows to separate these instances from others, whose PCU values decrease on average as more judgments are aggregated.

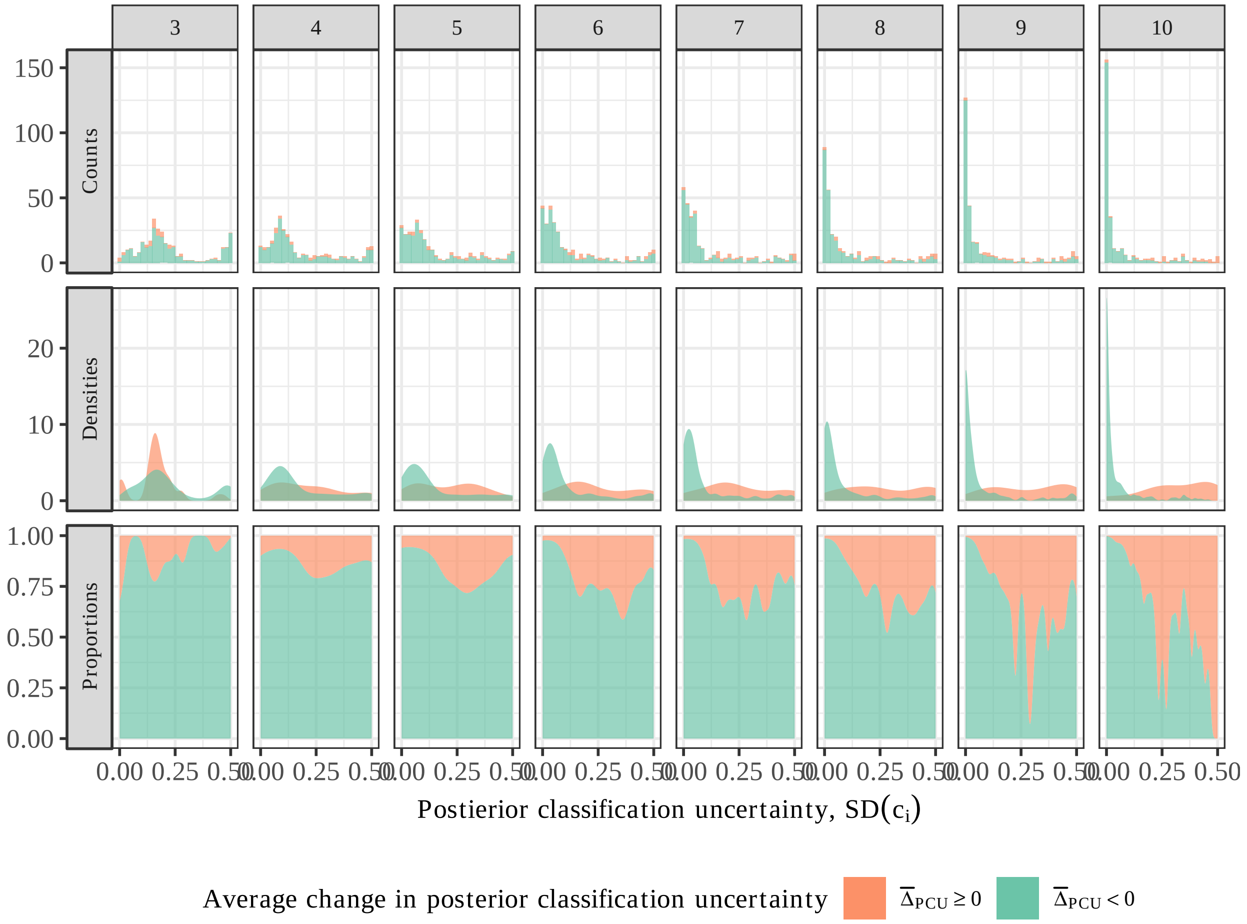

Figure 12: Empirical distribution of instance-level posterior classification uncertainty as function of n_i. Counts, densities and relative proportions of instances with an average increase in (or no) PCU change ($\bar{\Delta}_{\text{PCU}} > 0$) vs. others. Change in PCUs (Δ_{PCU}) computed for each instance by substracting PCU value obtained for n_i from value obtained for n_i. This difference is negative if PCU decreases. Average in Δ_{PCU} computed by aggregating over values $n_i = 3, \ldots, 9$ at instance level. Non-negative average change flags instances for which collecting more judgments did not lead to reduction in posterior classification uncertainty.

This result is quite significant as only model-based aggregationg allows to separate these instances. As Table X shows, majority voting is not as cautious when it comes to classifying these instances, and thus the correlation between model-based and majority winner labeling decreases in instances with non-negative average PCU change.

Summary and discussion

In this analysis, I have address three questions pertaining to the crowd-sourced measurement of elite critique. First, I have assessed crowd coders' abilities to classify elite critique as well

Table 3: Correlation between model-based classifications and majority voting.

	Number of judgments per instance							
Δ_{PCU}	3	4	5	6	7	8	9	10
≥ 0	0.961	0.817	0.653	0.569	0.472	0.499	0.368	0.452
< 0	0.876	0.885	0.842	0.869	0.833	0.776	0.823	0.842

as the variability in abilities across coders by fitting a Beta-Binomial by Annotator model to codings data collected by Hua, Abou-Chadi, and Barberá (2018). I find that crowd coders exhibit rather low true-positive detection abilities compared to their performances in true-negative detection, and that there is substantial variation in their true-positive detection abilities (see Figures 3 and 4). Our posterior belief about how sensitive elite-critique coders crowd workers are is thus more uncertain than our belief in their specificity in this task, but on average we would expect lower true-positive than true-negative detection abilities (see Figure 5).

Second, I have implemented a simulation study designed to assess how increasing the number of coded instances (sample size) and the number of judgments per instance affects model-based labeling quality. I conclude that increasing the number of judgement aggregated per instance reduces instance-level uncertainty of model-based classifications, whereas the gains achieved by increasing the number of coded instance are negligible (see Figure 9). Specifically, as the number of judgments aggregated per instance is increased, (i) posterior classification uncertainty in instance labels decreases in the overwhelming share of instances, and (ii) the variation in posterior classification uncertainty across instances decreases. What is more, the veracity of model-based classifications with respect to simulated true labels, measured by accuracy and F1-scores, improves substantially, largely due to decreasing amounts of true-positive instances that are classified wrongly (see Figure 10).

Third, using the same simulated data, I have compared model-based labels to labels induced by majority voting. I find strong correlations between these labels (see Table 2), largely independent of the number of coded instances and the number of judgments aggregated per instance. However, I also find that while majority voting tends to outperform model-based labeling in terms of classification performance (F1-scores) when only four or fewer judgments are aggregated per instances, this relationship reverses in favor of model-based labeling if this number is increased—particularly so as the number of coded instances is increased (see Figure 14)

With regard to the results of the simulation study, an open question is whether we would find similar patterns in real codings data. As in HAB's original dataset each instance that is not in the gold-standard dataset was judged by only three coders, the second and third questions cannot feasibly be answered with this data. Collecting more judgments for a subset of instances in their original dataset would allow to do so. How to obtain ‘true’ labels for instances in this subset would be another question, however. Submitting them to trained coders or experts to determine their true labels seems to be the most viable option.

Apendix

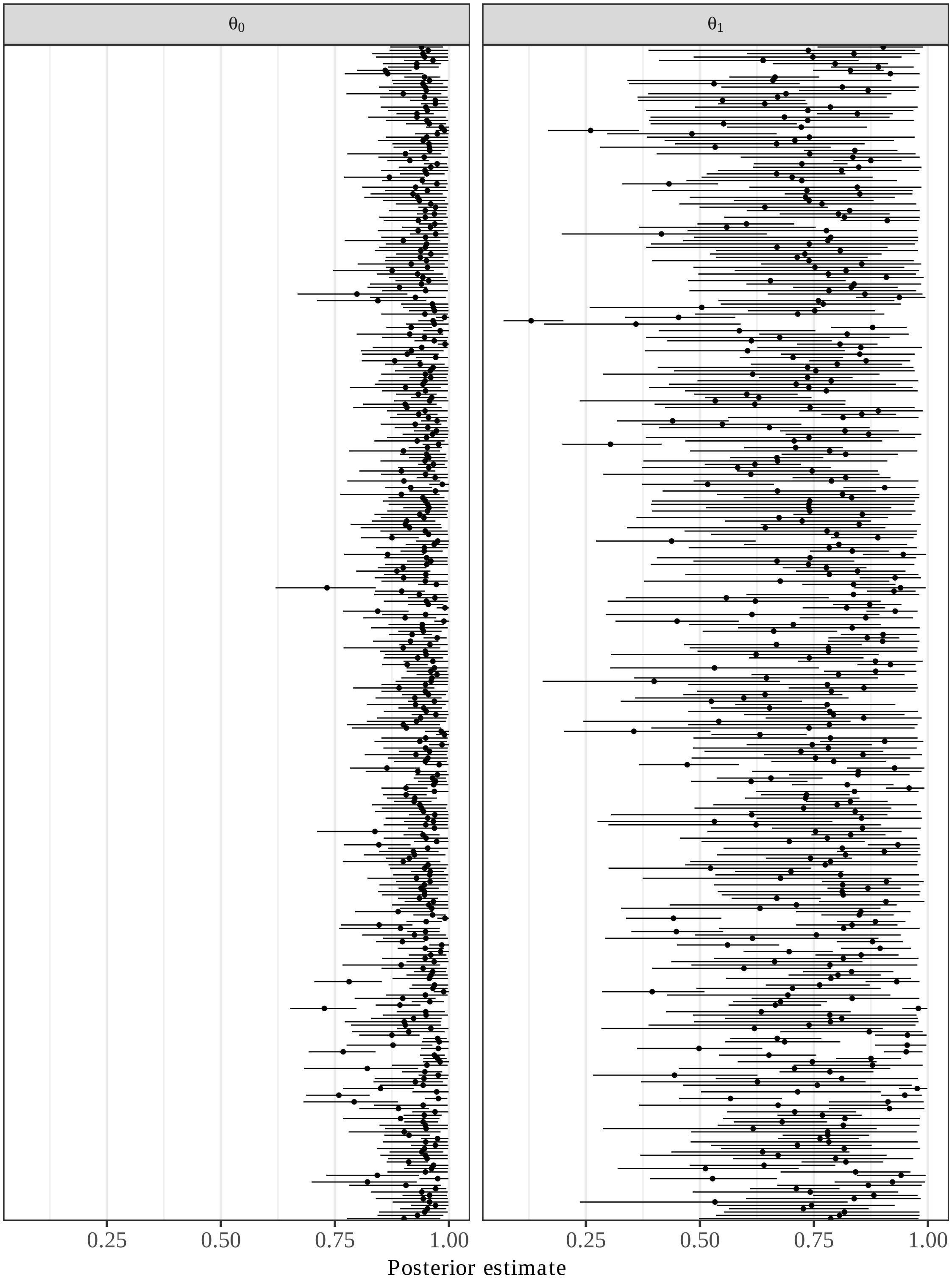

Figure 13: Mean posterior estimates and 90% credibility intervals of coders' abilities for anti-elitism labeling. θ_0 and θ_1 refer to specificity and sensitivity, respectively. Estimates obtained from three MCMC chains of 10K iterations each, retaining only every 10th estimate.

Table 4: Number of judgments in simulated codings datasets with varying n and n_i.

	Number of judgments per instance							
n	3	4	5	6	7	8	9	10
200	600	800	1000	1200	1400	1600	1800	2000
250	750	1000	1250	1500	1750	2000	2250	2500
300	900	1200	1500	1800	2100	2400	2700	3000
350	1050	1400	1750	2100	2450	2800	3150	3500
400	1200	1600	2000	2400	2800	3200	3600	4000
450	1350	1800	2250	2700	3150	3600	4050	4500
500	1500	2000	2500	3000	3500	4000	4500	5000

Table 5: Number of burn-in iterations of BBA models fitted to simulated $nimesn_i$-sized datasets.

	Number of judgments per instance							
n	3	4	5	6	7	8	9	10
200	5000	4500	4000	3500	3000	2500	2000	1500
250	4500	4000	3500	3000	2500	2000	1500	1000
300	4000	3500	3000	2500	2000	1500	1000	1000
350	3500	3000	2500	2000	1500	1000	1000	1000
400	3000	2500	2000	1500	1000	1000	1000	1000
450	2500	2000	1500	1000	1000	1000	1000	1000
500	2000	1500	1000	1000	1000	1000	1000	1000

Table 6: Thinning parameter of BBA models fitted to simulated $n \times n_i$-sized datasets.

	Number of judgments per instance							
n	3	4	5	6	7	8	9	10
200	50	40	40	20	20	10	10	10
250	50	40	40	20	20	10	10	10
300	40	30	30	20	20	10	10	10
350	40	20	20	20	10	10	10	10
400	40	20	20	10	10	10	5	5
450	40	20	20	10	10	10	5	5
500	40	20	15	10	10	5	5	5

Table 7: Proportion of instances with positive change in posterior classification uncertainty.

		Number of judgments successively added						
n	n_i	1	2	3	4	5	6	7
200	3	0.095^{+}	0.08^{+}	0.065^{+}	0.025^{*}	0.015^{*}	0.005^{**}	0^{***}
200	4	0.13	0.115	0.11	0.065^{+}	0.02^{*}	0^{***}	
200	5	0.215	0.12	0.1^{+}	0.06^{+}	0.005^{**}		
200	6	0.16	0.115	0.055^{+}	0.02^{*}			
200	7	0.13	0.085^{+}	0.045^{*}				
200	8	0.105	0.055^{+}					
200	9	0.065^{+}						
250	3	0.104	0.064^{+}	0.06^{+}	0.032^{*}	0.024^{*}	0.004^{**}	0.004^{**}
250	4	0.124	0.112	0.1^{+}	0.06^{+}	0.024^{*}	0.004^{**}	
250	5	0.252	0.148	0.1^{+}	0.04^{*}	0.024^{*}		
250	6	0.184	0.112	0.056^{+}	0.016^{*}			
250	7	0.132	0.088^{+}	0.04^{*}				
250	8	0.084^{+}	0.056^{+}					
250	9	0.06^{+}						
300	3	0.093^{+}	0.087^{+}	0.057^{+}	0.04^{*}	0.023^{*}	0.02^{*}	0.007^{**}
300	4	0.137	0.133	0.11	0.077^{+}	0.033^{*}	0.027^{*}	
300	5	0.233	0.153	0.147	0.073^{+}	0.043^{*}		
300	6	0.187	0.127	0.067^{+}	0.053^{+}			
300	7	0.157	0.087^{+}	0.047^{*}				
300	8	0.107	0.047^{*}					
300	9	0.067^{+}						
350	3	0.117	0.08^{+}	0.063^{+}	0.046^{*}	0.026^{*}	0.011^{*}	0.006^{**}
350	4	0.134	0.143	0.129	0.08^{+}	0.029^{*}	0.017^{*}	
350	5	0.231	0.166	0.137	0.06^{+}	0.037^{*}		
350	6	0.18	0.129	0.083^{+}	0.043^{*}			
350	7	0.129	0.086^{+}	0.046^{*}				
350	8	0.074^{+}	0.043^{*}					
350	9	0.049^{*}						
400	3	0.128	0.115	0.085^{+}	0.052^{+}	0.022^{*}	0.015^{*}	0.008^{**}
400	4	0.15	0.122	0.112	0.065^{+}	0.035^{*}	0.018^{*}	

Table 7: Proportion of instances with positive change in posterior classification uncertainty. *(continued)*

		Number of judgments successively added						
n	n_i	1	2	3	4	5	6	7
400	5	0.195	0.14	0.12	0.065^{+}	0.028^{*}		
400	6	0.175	0.125	0.06^{+}	0.04^{*}			
400	7	0.142	0.078^{+}	0.042^{*}				
400	8	0.102	0.055^{+}					
400	9	0.068^{+}						
450	3	0.118	0.1^{+}	0.056^{+}	0.04^{*}	0.024^{*}	0.018^{*}	0.009^{**}
450	4	0.158	0.131	0.104	0.064^{+}	0.036^{*}	0.022^{*}	
450	5	0.187	0.129	0.102	0.051^{+}	0.024^{*}		
450	6	0.162	0.12	0.076^{+}	0.04^{*}			
450	7	0.158	0.078^{+}	0.031^{*}				
450	8	0.084^{+}	0.04^{*}					
450	9	0.058^{+}						
500	3	0.142	0.106	0.064^{+}	0.038^{*}	0.032^{*}	0.016^{*}	0.006^{**}
500	4	0.16	0.13	0.092^{+}	0.064^{+}	0.038^{*}	0.016^{*}	
500	5	0.178	0.12	0.116	0.06^{+}	0.028^{*}		
500	6	0.164	0.13	0.082^{+}	0.032^{*}			
500	7	0.14	0.086^{+}	0.046^{*}				
500	8	0.078^{+}	0.034^{*}					
500	9	0.048^{*}						

Table 8: Proportion of iterations with negative change in F1-score

		Number of judgments successively added						
n	n_i	1	2	3	4	5	6	7
200	3	0.007^{**}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
200	4	0.083^{+}	0.004^{**}	0^{***}	0^{***}	0^{***}	0^{***}	
200	5	0.088^{+}	0.03^{*}	0.002^{**}	0^{***}	0^{***}		
200	6	0.268	0.067^{+}	0.018^{*}	0.018^{*}			
200	7	0.151	0.051^{+}	0.081^{+}				
200	8	0.158	0.307					
200	9	0.519						
250	3	0.001^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
250	4	0.045^{*}	0.003^{**}	0^{***}	0^{***}	0^{***}	0^{***}	
250	5	0.086^{+}	0.016^{*}	0.002^{**}	0^{***}	0^{***}		
250	6	0.174	0.05^{*}	0.021^{*}	0.025^{*}			
250	7	0.256	0.184	0.23				
250	8	0.304	0.421					
250	9	0.502						
300	3	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
300	4	0.05^{*}	0.002^{**}	0^{***}	0^{***}	0^{***}	0^{***}	
300	5	0.098^{+}	0.015^{*}	0.004^{**}	0^{***}	0^{***}		
300	6	0.188	0.085^{+}	0.027^{*}	0.005^{**}			
300	7	0.28	0.127	0.063^{+}				
300	8	0.261	0.143					
300	9	0.24						
350	3	0.001^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
350	4	0.013^{*}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	
350	5	0.097^{+}	0.02^{*}	0.003^{**}	0^{***}	0^{***}		
350	6	0.224	0.06^{+}	0.02^{*}	0.004^{**}			
350	7	0.219	0.104	0.051^{+}				
350	8	0.268	0.159					
350	9	0.259						
400	3	0.001^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
400	4	0.061^{+}	0.001^{***}	0^{***}	0^{***}	0^{***}	0^{***}	
400	5	0.058^{+}	0.003^{**}	0^{***}	0^{***}	0^{***}		

Table 8: Proportion of iterations with negative change in F1-score *(continued)*

		Number of judgments successively added						
n	n_i	1	2	3	4	5	6	7
400	6	0.087^{+}	0.009^{**}	0.001^{***}	0^{***}			
400	7	0.155	0.056^{+}	0.02^{*}				
400	8	0.287	0.112					
400	9	0.192						
450	3	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
450	4	0.085^{+}	0.001^{***}	0^{***}	0^{***}	0^{***}	0^{***}	
450	5	0.048^{*}	0.001^{***}	0^{***}	0^{***}	0^{***}		
450	6	0.082^{+}	0.005^{**}	0^{***}	0^{***}			
450	7	0.142	0.045^{*}	0.012^{*}				
450	8	0.249	0.114					
450	9	0.256						
500	3	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}
500	4	0.052^{+}	0^{***}	0^{***}	0^{***}	0^{***}	0^{***}	
500	5	0.044^{*}	0.001^{***}	0^{***}	0^{***}	0^{***}		
500	6	0.06^{+}	0.004^{**}	0.001^{***}	0^{***}			
500	7	0.178	0.065^{+}	0.023^{*}				
500	8	0.259	0.123					
500	9	0.26						

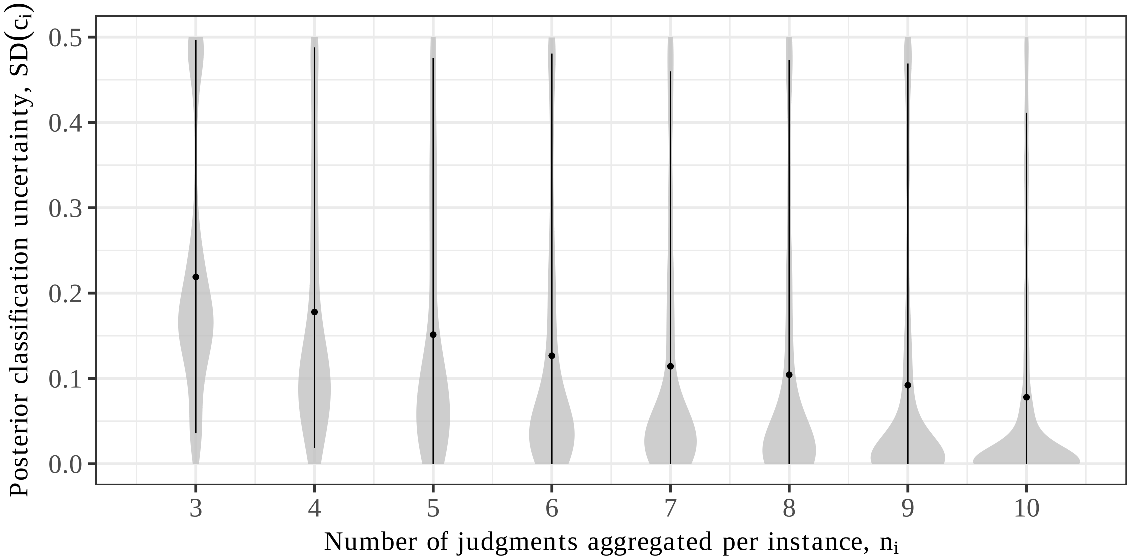

Figure 14: Distributions of posterior classification uncertainty by n and n_i. Results obtained by fitting BBA model to repeatedly judged elite critique codings.

References

Abedi, Amir. 2004. *Anti-Political-Establishment Parties: A Comparative Analysis*. Routledge Studies in Extremism and Democracy. London ; New York: Routledge.

Barr, Robert R. 2009. "Populists, Outsiders and Anti-Establishment Politics." *Party Politics* 15 (1): 29–48. https://doi.org/10.1177/1354068808097890.

Benoit, Kenneth, Drew Conway, Benjamin E. Lauderdale, Michael Laver, and Slava Mikhaylov. 2016. "Crowd-Sourced Text Analysis: Reproducible and Agile Production of Political Data." *American Political Science Review* 110 (02): 278–95. https://doi.org/10.1017/S0003055416000058.

Bergman, Matthew E. 2018. "Quantitative Measures of Populism: A Survey." SSRN Scholarly Paper ID 3175536. Rochester, NY: Social Science Research Network. https://papers.ssrn.com/abstract=3175536.

Berinsky, Adam J, Michele F Margolis, and Michael W Sances. 2014. "Separating the Shirkers from the Workers? Making Sure Respondents Pay Attention on Self-Administered Surveys." *American Journal of Political Science* 58 (3): 739–53.

Carpenter, Bob. 2008. "Multilevel Bayesian Models of Categorical Data Annotation." Unpublished manuscript. unpublished manuscript. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.174.1374&rep=rep1&type=pdf.

Dawid, Alexander Philip, and Allan M Skene. 1979. "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm." *Applied Statistics* 28 (1): 20–28. https:

//doi.org/10.2307/2346806.

Engler, Sarah. 2018. "The Survival of New Centrist Anti-Establishment Parties: The Interplay of Anti-Corruption Discourse and Ideology over Time." Dissertation, University of Bern.

Engler, Sarah, Bartek Pytlas, and Kevin Deegan-Krause. 2019. "Assessing the Diversity of Anti-Establishment and Populist Politics in Central and Eastern Europe." *West European Politics* 0 (0): 1–27. https://doi.org/10.1080/01402382.2019.1596696.

Hobolt, Sara B., and James Tilley. 2016. "Fleeing the Centre: The Rise of Challenger Parties in the Aftermath of the Euro Crisis." *West European Politics* 39 (5): 971–91. https://doi.org/10.1080/01402382.2016.1181871.

Hua, Whitney, Tarik Abou-Chadi, and Pablo Barberá. 2018. "Networked Populism: Characterizing the Public Rhetoric of Populist Parties in Europe." Paper prepared for the 2018 MPSA Conference. Paper prepared for the 2018 MPSA Conference.

———. 2019. "Measuring Anti-Elite Rhetoric of Political Parties in Europe."

Ipeirotis, Panagiotis G, Foster Provost, and Jing Wang. 2010. "Quality Management on Amazon Mechanical Turk." In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 64–67. ACM.

Jagers, Jan, and Stefaan Walgrave. 2007. "Populism as Political Communication Style: An Empirical Study of Political Parties' Discourse in Belgium." *European Journal of Political Research* 46 (3): 319–45. https://doi.org/10.1111/j.1475-6765.2006.00690.x.

Krippendorff, Klaus. 2004. *Content Analysis: An Introduction to Its Methodology.* 2nd ed. Thousand Oaks, Calif: Sage.

Mudde, Cas, and Cristóbal Rovira Kaltwasser. 2013. "Populism." In *The Oxford Handbook of Political Ideologies*.

Oliver, J Eric, and Wendy M Rahn. 2016. "Rise of the Trumpenvolk: Populism in the 2016 Election." *The ANNALS of the American Academy of Political and Social Science* 667 (1): 189–206.

Passonneau, Rebecca J, and Bob Carpenter. 2014. "The Benefits of a Model of Annotation." *Transactions of the Association for Computational Linguistics* 2: 311–26.

Pauwels, Teun. 2011. "Measuring Populism: A Quantitative Text Analysis of Party Literature in Belgium." *Journal of Elections, Public Opinion & Parties* 21 (1): 97–119. https://doi.org/10.1080/17457289.2011.539483.

———. 2017. "Measuring Populism: A Review of Current Approaches." In *Political Populism: A Handbook*, edited by Reinhard C. Heinisch, Christina Holtz-Bacha, and Oscar Mazzoleni, 123–36. Nomos.

Plummer, Martyn. 2003. "JAGS: A Program for Analysis of Bayesian Graphical Models

Using Gibbs Sampling." In. Vol. 124. Vienna, Austria.

Polk, Jonathan, Jan Rovny, Ryan Bakker, Erica Edwards, Liesbet Hooghe, Seth Jolly, Jelle Koedam, et al. 2017. "Explaining the Salience of Anti-Elitism and Reducing Political Corruption for Political Parties in Europe with the 2014 Chapel Hill Expert Survey Data." *Research & Politics* 4 (1): 1–9. https://doi.org/10.1177/2053168016686915.

Sheng, Victor S, Foster Provost, and Panagiotis G Ipeirotis. 2008. "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers." In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 614–22. ACM.

Sikk, Allan. 2012. "Newness as a Winning Formula for New Political Parties." *Party Politics* 18 (4): 465–86. https://doi.org/10.1177/1354068810389631.

Taggart, Paul A. 1996. "The New Populism and the New Politics." In *The New Populism and the New Politics*, by Paul A. Taggart, 11–46. London: Palgrave Macmillan UK. https://doi.org/10.1007/978-1-349-13920-0_2.

Zulianello, Mattia. 2018. "Anti-System Parties Revisited: Concept Formation and Guidelines for Empirical Research." *Government and Opposition* 53 (04): 653–81. https://doi.org/10.1017/gov.2017.12.