**Accounting for Confirmation Bias in Crowdsourced Label Aggregation (Supplementary Materials)**

Meric Altug Gemalmaz and Ming Yin, Purdue University

In this document, we provide the full derivations for our paper and also provide the political statements we used in our experiment.

Full code and the data can be found at the directory.

**1 Introduction:**

Consider the scenario that N annotators are asked to complete M binary labeling tasks. An annotator i’s label on task j is denoted as lij ∈ {0, 1}, with 0 representing the preferable label (e.g., “true news”, “neutral statement”). Our goal is to determine the true label, zj ∈ {0, 1}, for each task j using all the labels collected on it. To model annotators’ possible confirmation bias during their label generation processes, we assume the observed labels lij depend on several causal factors: (1) the values implied by the information in the task; (2) the annotator’s values; (3) the annotator’s degree of bias characterizing how much the annotator is subject to confirmation bias, (4) annotator’s inherent tendency to provide the preferable label, and (5) the true label of the task. Our entire label generation model for the crowdsourced annotators is shown in Figure X.

Diagram

Description automatically generated

Figure X: The probabilistic graphical model of annotators’ label generation process. The shaded node is observed.

The values of annotator i are captured by the parameter ci ∈ [0, 1], while the values of the information contained in task j are captured by the parameter sj ∈ [0, 1]. For example, when considering the left–right political spectrum, ci = 1 (or sj = 1) could mean the values of annotator i (or the values implied by information in task j) are extremely conservative, while ci = 0 (or sj = 0) means the values of annotator i (or the values implied by information in task j) are extremely liberal. Annotators’ confirmation bias is captured via the *distance* between ci and sj—holding all other variables equal, the closer ci and sj are to each other, the more likely annotator i will provide the preferable label in task j (i.e., P (lij = 0) is larger).

We further use the parameter pi ∈ [0, 1] to characterize the extent to which annotator i is subject to confirmation bias. Here, pi = 0 means that annotator i is heavily influenced by her confirmation bias, such that she decides her label on tasks (almost) entirely based on how much the information contained in the task aligns with her values. Conversely, when pi = 1, annotator i is not influenced by her confirmation bias at all, such that she decides her label on tasks (almost) entirely based on the ground truth label zj of the task, and zj ∼ Bernoulli(1 − π) (i.e., the prior probability for a task to have the preferable label as its ground truth is π, P(zj = 0) = π). When 0 < pi < 1, the annotator is influenced by her confirmation bias to some degree, and the smaller pi is, the more she is subject to the confirmation bias.

Finally, we use a global parameter a ∈ [0, +∞) to represent annotators’ inherent tendency to provide the preferable label on any task, or in other words, annotators’ base rate of providing the preferable label in tasks. When a = 0, the base rate for annotators to provide the preferable label in tasks is very high, while a = +∞ means the base rate for annotators to provide the preferable label in tasks is very low.

Under our model, the chance for annotator i to provide the preferable label on task j (i.e., lij = 0) is characterized as:

P(lij = 0 | zj, sj, ci, pi, a) = (1)

**Note:** In our code, to make computations simpler, we refer to parameter as

(i.e., ) and learn the parameter .

**2 Inference (Full EM Derivation):**

We use the Expectation-Maximization (EM) algorithm to estimate the maximum likelihood estimates of the hidden parameters and infer the values of the hidden variables zj.

**Expectation Step:**

In particular, in the Expectation step, we compute the posterior probabilities for each hidden variable zj based on the current estimates of parameters and the observed labels:

p (zj | **L**, **s**, **c, p,** a, π) = p(zj | **L**j, sj, **c**, **p**, a, π)

µ p(zj | sj, **c**, **p**, a, π) p(**L**j | zj, sj, **c**, **p**, a)

µ p(zj | π)

**Note:** We used p(zj | sj, **c**, **p**, a, π) = p(zj | π), this equality assumption comes from the conditional independence assumptions from the probabilistic graphical model.

Here, we use Wj to denote the set of all annotators who have provided labels on task j.

When lij = 0, p(lij | zj, sj, ci, pi, a) can be computed using Equation 1; otherwise, p(lij | zj, sj, ci, pi, a) = 1 − P(lij = 0 | zj, sj, ci, pi, a).

**Maximization Step:**

For the Maximization step, we search for optimal parameter values to maximize the auxiliary function Q, i.e., the expectation of the complete data log-likelihood. The standard auxiliary function is defined as the expectation of the joint log-likelihood of the observed (**L**) and the hidden (**Z**) variables given the parameters (**s**, **c**, **p**, a, π), with respect to the posterior probabilities of the **Z** values computed during the last Expectation step.

Since lij are conditionally independent given **z**, **s**, **c**, **p**, a

Expectation is taken over the posterior distribution of **z** that is .

**Note:** The parameter values , are from the previous Maximization step.

Expanding the expectation:

=

Note that the label probability equation will change according to the label. If the label is ‘0’ the equation (1) will be used and if the label is ‘1’ then 1 - equation (1) will be used.

In order to find the optimal parameter values, we take the derivative of the function with respective to parameters . When we take the derivatives with respect to the parameters , the first summation vanishes. Focusing on the second summation;

**Note:** We calculate the gradient using equation in our code, here we provide the gradients of that equation.

When the label is ‘0’ (using equation (1)):

= = =

=

When the label is ‘1’ (using 1 – equation (1)):

= =

= =

Finding the locally optimal values of the parameters requires setting these gradients to zero. The resulting equations needs to be solved using iterative methods. We use gradient descent to find the locally optimal values of the parameters.

When we take the derivative with respect to , the second summation vanishes. Focusing on the first summation; (posterior probabilities are blue and prior probabilities are red)

Taking the derivative with respect to ;

=

**3 Political Statements:**

|  |  |
| --- | --- |
| Statement | Label |
| (S1) Outlawing guns is the best solution to prevent wars. | Liberal Opinion |
| (S2) Most of the problematic shooting events were led by mentally ill people. | Conservative Opinion |
| (S3) Gun bans alleviate intimate partner homicide. | Liberal Fact |
| (S4) USA always has more devastating gun violence than any other first world nation. | Liberal Opinion |
| (S5) Most of the murders in US were led by people who didn’t have the right mental state at the moment. | Conservative Opinion |
| (S6) Chicago still had many shooting victims even though it had gun ban. | Conservative Fact |
| (S7) Active shooter events in the U.S. is sometimes associated with mental illness. | Conservative Fact |
| (S8) In the past years, gun related deaths covered a significant portion of deaths in USA. | Liberal Fact |
| (S9) Easy usage of the guns increases firearm related deaths. | Liberal Opinion |
| (S10) Many criminals obtain guns from illegal sources. | Conservative Fact |
| (S11) The share of Americans supporting gun control increased. | Liberal Fact |
| (S12) Guns easily freed USA from British Forces. | Conservative Opinion |