MOMRESP: A Bayesian Model for Multi-Annotator Document Labeling

Paul Felt*, **Robbie Haertel***, **Eric K. Ringger**, **Kevin D. Seppi**
Department of Computer Science
Brigham Young University
Provo, Utah 84602 USA
paul_felt@byu.edu, robbie.haertel@gmail.com, {ringger, kseppi}@cs.byu.edu

Abstract

Data annotation in modern practice often involves multiple, imperfect human annotators. Multiple annotations can be used to infer estimates of the ground-truth labels and to estimate individual annotator error characteristics (or reliability). We introduce MOMRESP, a model that improves upon item response models to incorporate information from both natural data clusters as well as annotations from multiple annotators to infer ground-truth labels for the document classification task. We implement this model and show that MOMRESP can use unlabeled data to improve estimates of the ground-truth labels over a majority vote baseline dramatically in situations where both annotations are scarce and annotation quality is low as well as in situations where annotators disagree consistently. Correspondingly, in those same situations, estimates of annotator reliability are also stronger than the majority vote baseline. Because MOMRESP predictions are subject to label switching, we introduce a solution that finds nearly optimal predicted class reassignments in a variety of settings using only information available to the model at inference time. Although MOMRESP does not perform well in annotation-rich situations, we show evidence suggesting how this shortcoming may be overcome in future work.

Keywords: Bayesian models, corpus annotation, crowd-sourcing, identifiability

1. Introduction

To build labeled corpora and to train NLP models using supervised learning methods we rely heavily on imperfect annotators, ranging from nearly perfect experts to very imperfect workers in crowd-sourcing settings. Often we rely on multiple annotators providing redundant annotations to improve the quality of the resulting labeled data. Although annotations produced by fallible annotators are not individually trustworthy, they can be used collectively to infer the correct labels for data and also to estimate annotator error characteristics. To be clear, in this paper, we use the term *annotation* to refer to human input and *label* to refer either to gold-standard reference labels or to model-predicted labels.

In the machine learning literature, annotations are useful first and foremost as a means of inducing a model that can be used to predict the labels of new data. In other fields, corpus labels are interesting in their own right because they facilitate meaningful data analysis. For example, corpus linguists employ corpora labeled with linguistic categories to aid in the analysis of diachronic trends and patterns in language (Kucera and Francis, 1967; Gardner and Davies, 2007). Estimating ground-truth labels is closely related to the task of learning individual annotator error characteristics, since these can be used to upweight trustworthy annotations and downweight others. Annotator error profiles can also be used to elucidate opportunities to retain, train, and advise annotators.

In this paper we introduce, implement, and evaluate a model for the inference of ground-truth labels and annotator reliability that utilizes features of the data as well as annotations from multiple annotators; the two sources of information reinforce one another and make possible ground truth inference that is superior to the available baselines.

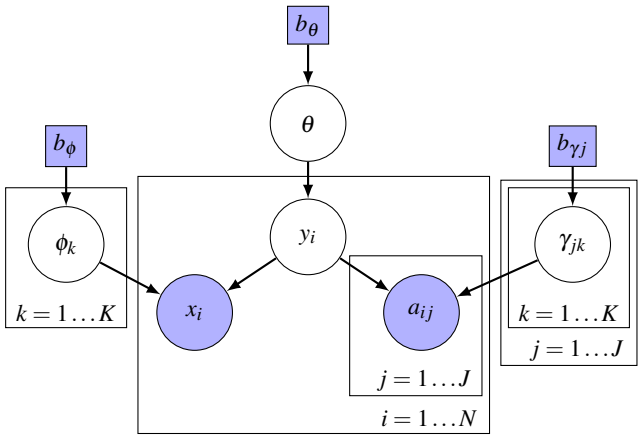
*The first and second authors contributed equally to this work. The second author is now affiliated with Google.

2. Previous Work

Dawid and Skene (1979) laid the groundwork for inferring ground-truth labels and estimating individual annotator accuracy by proposing a statistical model known as the "item-response" model. Carpenter (2008) and Pasternack and Roth (2013) describe models that are essentially Bayesian versions of the same model. There is a growing body of variations and extensions to this simple item-response model to account for correlations among annotators or item difficulty (Carpenter, 2008; Hovy et al., 2013; Lin et al., 2012; Raykar and Yu, 2012; Smyth et al., 1995; Weld et al., 2011; Whitehill et al., 2009; Zhou et al., 2012). Only human annotations are leveraged in the previously described approaches. However, Lam and Stork (2005), in effect, extend the item-response model such that the inferred label of a data item depends not only on annotations, but also on the features of the data instance itself. Carroll et al. (2007) propose a model that similarly takes advantage of data features. However, neither of these proposed models is implemented or evaluated in previous work.

Inferring labels and estimating annotator trustworthiness can be seen as special cases of the fact-finding task (Han, 2009; Pasternack and Roth, 2010) in which the annotators are sources of information and their annotations are claims. However, the general task of fact-finding is far more complex than the special case of data annotation.

Furthermore, there is a good deal of work focused on the machine learning problem of training a useful model from multiple error-prone annotations, especially in light of recent interest in crowd-sourcing. However, much of this work relies more on heavy redundancy than sophisticated aggregation techniques (Snow et al., 2008). The *de facto* standard approach is to infer ground-truth corpus labels using a simple majority vote, assess inter-annotator agreement to gain some confidence in the quality of the labels, and then pass the resulting labeled corpus to a ma-



$J :=$ number of annotators
$K :=$ number of labels
$N :=$ number of instances
$F :=$ number of features(word types)

$$\theta \sim SymDir(b_\theta), \qquad dim(\theta) = K$$
$$\forall k: \ \phi_k \sim SymDir(b_\phi), \qquad dim(\phi_k) = F$$
$$\forall j,k: \ \gamma_{jk} \sim Dir(b_{\gamma_{jk}}), \qquad dim(\gamma_{jk}) = K$$
$$dim(\gamma_j) = K \times K$$
$$\forall i: \ y_i|\theta \sim Cat(\theta), \qquad y_i \in \{1 \ldots K\}$$
$$\forall i: \ x_i|y_i,\phi \sim Multinom(|x_i|_1, \phi_{y_i}), \qquad dim(x_i) = F$$
$$\forall i,j: \ a_{ij}|y_i,\gamma_j \sim Multinom(|a_{ij}|_1, \gamma_{j,y_i}), \qquad dim(a_{ij}) = K$$

Figure 1: MOMRESP: a generative Bayesian model for inferring ground-truth labels y from the annotations a of multiple annotators and from data x while modeling individual annotator accuracies γ. ($|\cdot|_1$ denotes the L_1-norm.)

chine learning algorithm in order to train the desired model in the traditional batch manner. Sheng et al. (2008) construct training corpora in which each instance, annotated n times, is replicated with each annotation and weighted by $\frac{1}{n}$. Models trained from this 'soft' labeling are shown to always be at least as good as, and usually better than, those trained by majority vote. Jurgens (2013) experiments with constructing similar weighted datasets using explicit annotator input.

3. Methods

To take advantage of the opportunities presented by natural data clusters and multiple, potentially sparse annotations, we now present MOMRESP, a generative Bayesian model that can infer ground-truth labels from multiple, noisy annotators and simultaneously estimate the error characteristics of each annotator. We also present an MCMC inference algorithm for the model.

3.1. Model

MOMRESP is inspired by Bayesian models described, but not implemented or evaluated, in previous work (Carroll et al., 2007; Carroll, 2010; Haertel, 2013). It is called MOMRESP because it adds a mixture-of-multinomials (MOM) data component to a Bayesian item-response model. The model is based on three main principles:

1. Ground-truth labels y are unobservable.
2. All annotations a may carry useful information, even when incorrect.
3. A document's words x can help in determining a document's label y.

Figure 1 presents the model as a directed graphical model. The model assigns probability to variables as though they were generated according to the following process. First, label class proportions θ and word proportions ϕ_k for each label class $k \in \{1 \ldots K\}$ are drawn. (K is the number of label classes.) For each annotator $j \in \{1 \ldots J\}$, a probability vector γ_{jk} is drawn specifying the probabilities of the annotations annotator j is likely to produce in the presence of the label class k. (J is the number of annotators.) Thus, γ_j can be seen as an annotator j-specific confusion matrix (alternatively: a contingency table or error matrix), where each row sums to 1. For each of the N documents, the ith annotated document is generated by drawing a document label y_i from the categorical distribution $Cat(\theta)$ and then drawing words x_i from a multinomial distribution with parameters ϕ_{y_i} and drawing annotations a_{ij} from a multinomial distribution with parameters γ_{jy_i}.

Notice that in the absence of annotations a, this model reduces to a mixture-of-multinomials document clustering model. In the absence of any data x, this model becomes a multinomial item-response model—a Bayesian version of the approach of Dawid and Skene (1979).

3.2. Inference

Having formulated a generative model and specified a distribution over every variable in the model means that we can apply standard Bayesian statistical machinery such as Markov Chain Monte Carlo inference to the task of inferring the values of hidden labels y and annotator error characteristics γ given some observed set of documents x and annotations a, however incomplete the annotations may be. We derive an efficient collapsed Gibbs sampler for y. Then the per-annotator confusion matrix γ is constructed from these samples after-the-fact, as described below.

Liu (1994) provides empirical and theoretical evidence that analytically integrating out parameters where possible (i.e., 'collapsing' the model) improves Gibbs sampling. Accordingly, we derive a collapsed sampler by analytically integrating out the parameters of the model (θ, ϕ, γ).

Let $p(y_i = c|y_{i' \neq i})$ be the full conditional distribution for y_i. It represents the conditional distribution over possible values c for y_i given the data and annotations and sample values for all other latent class labels $y_{i'}$ where $i' \neq i$. We omit its derivation due to space constraints, merely noting that it is very similar to the derivation for a mixture-of-multinomials model (Walker, 2012). For notational simplicity, we define the following count variables, which are disambiguated by the letters of their superscripts. For the purpose of the full conditionals, each count variable excludes the counts associated with the instance i being sampled (in-

dicated by sums over $i' \neq i$):

$$\forall k: \; n_k^{(\theta)} = \sum_{i' \neq i}^{N} \mathbb{1}(y_{i'} = k)$$

$$\forall j,k,k': \; n_{jkk'}^{(\gamma)} = \sum_{i' \neq i}^{N} a_{i'jk'}^{\mathbb{1}(y_{i'}=k)}$$

$$\forall k,f: \; n_{kf}^{(\phi)} = \sum_{i' \neq i}^{N} x_{i'f}^{\mathbb{1}(y_{i'}=k)}.$$

That is, $n_k^{(\theta)}$ is the number of instances currently labeled k; $n_{jkk'}^{(\gamma)}$ is the number of times that annotator j chose annotation k' on instances labeled k; and $n_{kf}^{(\phi)}$ is the number of times word (or feature) f occurs with instances having an inferred label value of k.

Using these count variables, the full conditional distribution for the MOMRESP model has the following form:

$$\begin{aligned} p(y_i = c | y_{i' \neq i}) &\propto (b_\theta + n_c^{(\theta)}) \\ &\cdot \prod_{j=1}^{J} \left(\sum_{k'=1}^{K} \left(b_{\gamma_j} + n_{jck'}^{(\gamma)} \right) \right)^{-\overline{\sum_{k'=1}^{K} a_{ijk'}}} \prod_{k'=1}^{K} \left(b_{\gamma_j} + n_{jck'}^{(\gamma)} \right)^{\overline{a_{ijk'}}} \\ &\cdot \left(\sum_{f=1}^{F} \left(b_\phi + n_{cf}^{(\phi)} \right) \right)^{-\overline{\sum_{f=1}^{F} x_{if}}} \prod_{f=1}^{F} \left(b_\phi + n_{cf}^{(\phi)} \right)^{\overline{x_{if}}} \end{aligned} \tag{1}$$

where the notation $x^{\bar{n}}$ represents the rising factorial defined as $x^{\bar{n}} := x(x+1)(x+2)\ldots(x+n-1)$ and $x^{-\bar{n}} := \frac{1}{x^{\bar{n}}}$.

Gibbs sampling yields samples that consist of label values for each y_i. In our sampling experiments, we use the label values y in the single last sample of the Markov chain as the predicted corpus labels. Walker (2012)'s experiments with a sampler for the mixture-of-multinomials model reveals that taking the last sample is an efficient way to summarize the Markov chain without sacrificing the quality of the samples. In order to estimate the accuracy vector γ_{jk} for annotator j on label k, we compute the mean of the Dirichlet distribution for γ_{jk} using the sufficient statistics $n_{jkk'}^{(\gamma)}$. The mean values of θ and ϕ are similarly estimated using $n_k^{(\theta)}$ and $n_{kf}^{(\phi)}$, respectively. In order encourage our sampler to explore effectively, we anneal our sampler with 250 samples at each temperature of the following schedule: $1000, 500, 200, 100, 50, 20, 10, 5, 2, 1$. This is followed by an additional 500 samples without annealing.

3.3. Class Correspondence Correction

The problem of class correspondence, or label switching, arises in unsupervised and semi-supervised mixture models (Stephens, 2000). The model described in Section 3.1. assigns the same probability to any permutation of the inferred class label types assignable to y. That is, we could relabel every y whose value is class A to class B and vice versa as long as we also swapped the rows in γ and ϕ that correspond to class A and class B. The model has no reason (aside from weak prior preferences) to prefer one of these solutions over the other. However, inferred labels y are only useful when they align with true gold-standard labels; therefore, we must solve the class correspondence problem.

Stephens (2000) points out that the problem of finding an optimal (with respect to some loss function) reassignment of inferred label classes can be posed and efficiently solved as an instance of the well-known assignment problem. In this formulation, we have K 'source' classes as a result of sampling, and each source class will be assigned to one of K 'destination' classes. Let $L(c', c'')$ be some loss function that defines the cost of (re-)assigning source class c' to destination class c''. Each possible assignment from c' to c'' is represented with a boolean variable $\pi_{c'c''}$, where $\pi_{c'c''} = 1$ indicates the presence of an assignment in the final solution, and $\pi_{c'c''} = 0$ indicates absence. Our objective is to discover the solution π that minimizes

$$\sum_{c'=1}^{K} \sum_{c''=1}^{K} \pi_{c'c''} L(c', c'')$$

subject to the following constraints

$$\forall c' \in \{1 \ldots K\}: \; \sum_{c''=1}^{K} \pi_{c'c''} = 1$$

$$\forall c'' \in \{1 \ldots K\}: \; \sum_{c'=1}^{K} \pi_{c'c''} = 1$$

$$\forall c', c'': \; \pi_{c'c''} \geq 0$$

The first set of constraints ensures that each source class is assigned once. The second set ensures that each destination class receives a single assignment. The final constraints ensure non-negativity. Although this formulation allows for fractional assignments, the optimal solution is guaranteed to have integer values because the constraint matrix is totally unimodular (Burkard et al., 2009).

3.4. Loss Functions

The problem remains of choosing a loss function L. Intuitively, L's purpose is to penalize decisions that assign inferred label classes to the wrong gold-standard label classes. We experiment with three loss functions. The first relies on gold-standard labels; the second two do not.

CorrectCount

Confronted with a similar semi-supervised class correspondence problem, Nigam and McCallum (2006) note that it would be a relatively simple matter to align a small number of latent classes *manually* with true classes by inspecting a few documents assigned to each class. The CORRECTCOUNT loss function automates this insight by assuming that some number of gold-standard labels are available for the purposes of empirical class alignment. Predicted labels y are compared with gold-standard labels to compute an error matrix E where $E_{kk'}$ is the number of times a document with the gold-standard label k was predicted by the model to have label k'. The ideal E is diagonal because a diagonal error matrix represents no errors in the predicted labels. Transposed columns in E indicate that the model is "calling things by the wrong name," a symptom of label switching in the inference process. Correcting transposed columns involves changing the predicted label class at index c' to some better position c''. CORRECTCOUNT measures the

magnitude of the diagonal entry of the moved column in its destination position:

$$L_{\text{CC}}(c', c'') = -E_{c''c'}$$

This loss function favors column assignments with strong diagonal entries (i.e., good label accuracy) in a given error matrix E. It is implicitly parameterized by the number of gold-standard labels used to generate E. Were this number to equal the size of the corpus (an untenable situation), using the resulting CORRECTCOUNT loss function in conjunction with the LP solver would yield the highest possible label accuracy achievable with any solution to the class correspondence problem.

BestAnnotator

Assembling gold-standard labels is cumbersome, and it begs the question which MOMRESP aims to address in the first place. The BESTANNOTATOR loss function instead only uses parameter estimates found in the model. Recall that inference yields estimates of error characteristics γ_j for each annotator j, and each γ_j matrix can be viewed as a normalized confusion matrix. When inferred label classes are permuted, the rows of matrix γ_j are permuted. Suppose there is an annotator $\hat{j}$ whose annotations we believe to be mostly in accord with the truth; i.e., for no class k is she more likely to choose some other class k'. Unfortunately, the model's predicted $\gamma_{\hat{j}}$ is subject to label switching, manifested as permutations of the rows of $\hat{j}$'s true confusion matrix. Then our belief is that with the correct row ordering, $\gamma_{\hat{j}}$ is strongly diagonal. Thus, the intuition underlying the BESTANNOTATOR loss function is that the overall divergence of a given γ_j from the identity matrix — assuming γ_j's rows are in the proper order — should be small. Consequently, we define this loss function using KL Divergence from the indicated row of the identity matrix I:

$$L_{\text{BA}}(c', c'') = KL(I_{c''} || \gamma_{\hat{j}c'})$$

AggregateAnnotator

Rather than relying on a single annotator, we might try aggregating across annotators. AGGREGATEANN assumes that for no class k are all annotators more likely to choose some other class k'. Thus, we aggregate the error characteristics γ of all annotators. Summed rows would not constitute probability vectors, so rather than normalizing the summed probability vectors and employing KL divergence, this loss function measures the aggregated diagonal entry of the row in its proposed destination c'':

$$L_{\text{AA}}(c', c'') = -\sum_{j=1}^{J} \gamma_{jc'c''}$$

4. Experiments

Models dealing with multiple error-prone annotations can be challenging to evaluate in a controlled way because multiply-annotated benchmark datasets for classification have not been established. Sheng et al. (2008) simulate annotations for an annotator by corrupting the ground-truth labels according to an accuracy parameter associated with that annotator. They use a strategy they term Generalized Round Robin (GRR) for determining which data instances are annotated and with how many annotations. In GRR, an instance is selected at random (without replacement) to be annotated d separate times by annotators selected randomly with replacement. After all instances have been annotated, the process is repeated. Thus, an instance can be annotated more than d times if revisited. GRR simulation has two parameters of interest: the number d of annotations per instance per round and annotator quality.

	A1	A2	A3	A4	A5
HIGH	90	85	80	75	70
MED	70	65	60	55	50
LOW	50	40	30	20	10
CONFLICT	50†	40†	30†	20†	10†

Table 1: Annotator (A1-A5) accuracies for each quality level (HIGH, MED, LOW, CONFLICT). † indicates that errors are systematic (see text for details).

We use GRR with annotators from the quality pools (one named pool per row) in Table 1. Each pool lists the accuracy of five annotators, A1-A5 (five is an arbitrary choice for the experiments). In the quality settings HIGH, MED, and LOW, annotator errors are distributed uniformly across the incorrect classes. Because there are no patterns among errors, these settings approximate situations in which annotators are ultimately in agreement about the task they are doing, although some are better at it than others.

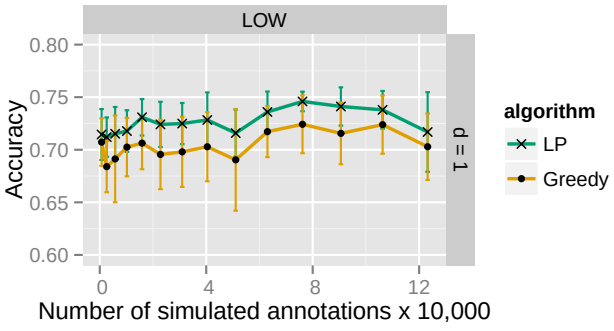


Figure 2: Linear Programming (LP) versus Greedy search (Greedy) for class correspondence correction. Both approaches use the same CorrectCount loss function.

The CONFLICT quality setting in Table 1 is special in that annotator errors are systematic rather than uniform random. Systematic errors are produced at simulation time by constructing a confusion matrix (similar to 'γ') for each simulated annotator with diagonals set to the desired accuracy, and with off-diagonals sampled from a symmetric Dirichlet distribution with parameter 0.1 for sparsity and then scaled so that each row sums to 1. These draws from a sparse Dirichlet yield error patterns that are quite self-consistent. For example, annotator A5 in the CONFLICT setting will label class B as B only 10% of the time, but might label B as C 85% of the time. CONFLICT approximates an annotation project where annotators understand the annotation guidelines differently from one another.

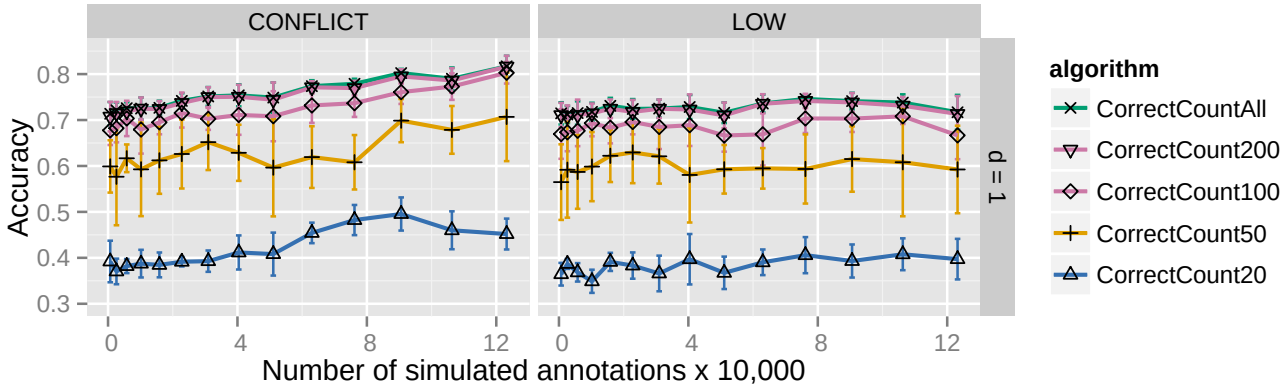


Figure 3: Correcting the class correspondence problem using the CORRECTCOUNT loss function with various amounts of manually labeled data to create its error matrix. Notice that performance does not plateau until about 200 items have been manually labeled, or 10 per class.

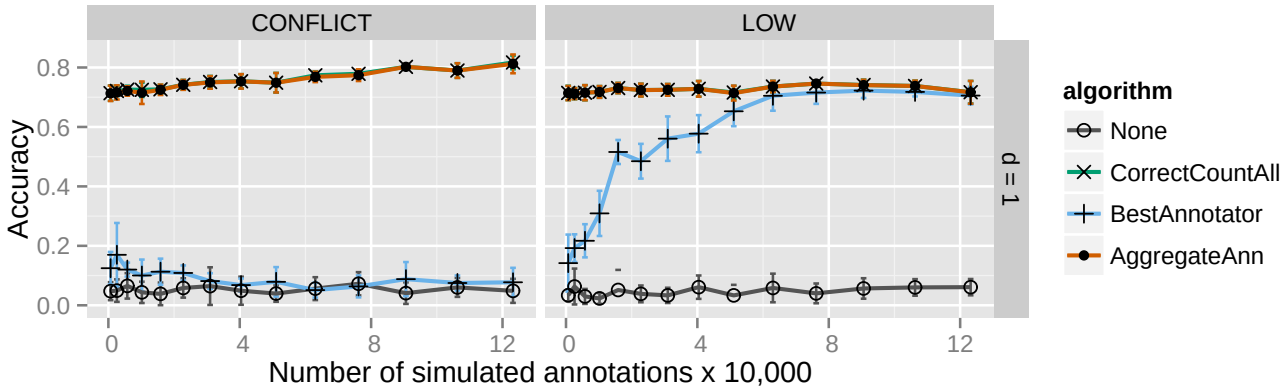


Figure 4: Class correspondence correction strategies. NONE takes the raw sampling results in which classes are essentially randomly permuted. CORRECTCOUNT uses information from gold-standard labels for all the data and is an upper bound on what is possible. BESTANNOTATOR uses the γ_j of an arbitrarily selected annotator j. AGGREGATEANN uses the aggregated γ matrices of all annotators, and performs at nearly the level of the upper bound in all settings tested (including those not shown).

We use GRR to simulate the annotators in Table 1 annotating the 20 Newsgroups data set (Joachims, 1997), which consists of approximately 20,000 documents evenly divided among 20 classes. For each experiment, we randomly select a subset of 17,000 documents to serve as our corpus. 20 Newsgroups is an appropriate dataset because it is a well-known text classification benchmark and has been used by previous work in evaluating related semi-supervised mixture-of-multinomials models (Nigam et al., 2006). In all plots, all graphed lines represent at least 20 randomized runs on different simulations (the corpus and annotations change in every simulation), with error bars—too small to see in most cases—representing one standard deviation.

4.1. Class Correspondence Correction

In Section 3.3. we explained that label switching among predicted label classes could be rectified by defining a loss function over potential label class reassignments and using linear programming to find the label class permutation incurring the smallest possible loss overall. To confirm that linear programming is worth the time required to implement, Figure 2 demonstrates that in practice, using linear programming to find the optimal alignment solution yields gains over a simple greedy solution. Figure 2 graphs the model's inferred label accuracy after sampling and *post hoc* class correspondence correction. Greedy and LP both use the CORRECTCOUNT loss function with access to a full confusion matrix, and both are given the same set of sampled model parameters to correct. Greedy assembles a solution by iteratively selecting the source class k corresponding to the error matrix column with the largest value, and then assigning it to the unclaimed destination class k' that will result in the largest value along the diagonal of the error matrix. The optimal linear programming search, although slower in practice, yields answers with a higher mean and less variance than the greedy search.

In Figure 3 we explore how much manually labeled data the CORRECTCOUNT loss function requires in order to be effective. CorrectCount20 has access to 20 randomly selected labeled instances, CorrectCount50 has access to 50, and so on. CorrectCountALL has access to the full error matrix and should therefore be regarded as an upper bound on improvements to be gained from realigning in-

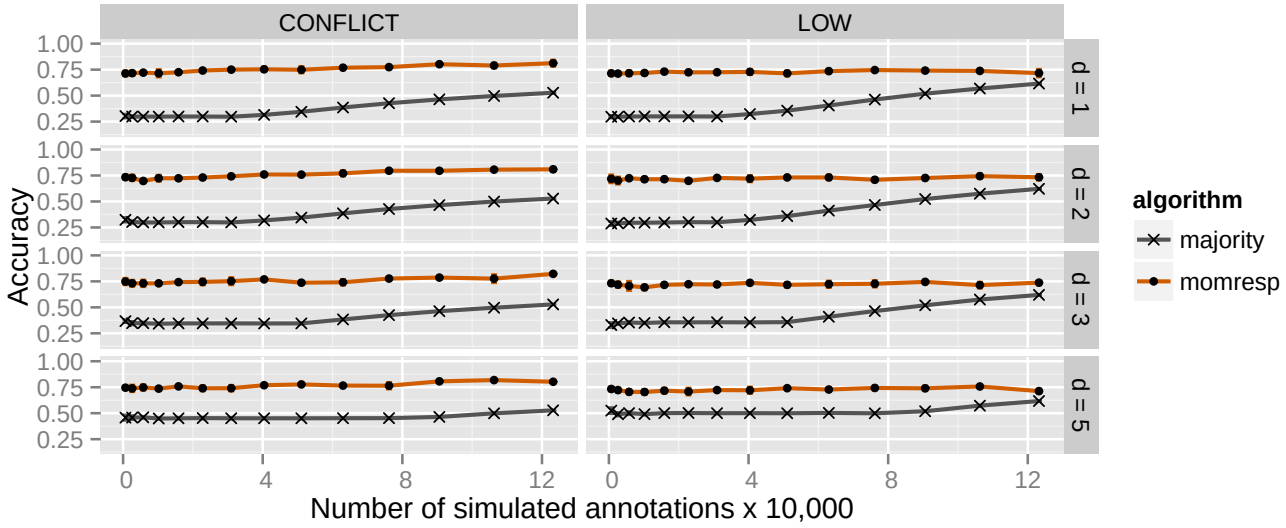


Figure 5: Inferred label accuracy on the 20 Newsgroups dataset

ferred label classes. Notice that once approximately 200 gold-standard labels are available then CORRECTCOUNT's performance reaches its peak. That is roughly 10 labels per class, and represents a non-negligible amount of work.

So as not to be dependent on any gold-standard labels, the loss functions BESTANNOTATOR and AGGREGATEANN use only inferred model parameters γ. Figure 4 compares their behavior with the upper bound CORRECTCOUNT and 'None,' which indicates the performance of samples whose latent class correspondence has not been corrected. BESTANNOTATOR struggles badly in the CONFLICT setting in which annotators make systematic errors. This is unsurprising, since CONFLICT violates the assumption made by BESTANNOTATOR that an annotator exists who is basically aligned with the truth. BESTANNOTATOR does better in LOW where errors are made uniformly randomly, however it still takes some time to overcome data sparsity. AGGREGATEANN is robust to the systematic errors in CONFLICT because the patterns in the errors are washed out in aggregate. AGGREGATEANN is also more robust to data sparsity since it does not limit itself to drawing on information from a single individual's annotations. AGGREGATEANN follows the upper bound CORRECTCOUNT so closely that we use it for all subsequent experiments.

Although because of space constraints we have focused on select experimental conditions, the patterns seen in Figures 2, 3, and 4 hold for all other unshown experimental settings, including for the MED and HIGH annotation settings.

4.2. Inferred Label Accuracy

We now compare MOMRESP with majority vote, dubbed MAJORITY. We run simulations that sweep the annotator quality pools from Table 1 and annotations $d \in \{1,2,3,5,10\}$ per instance per round. Accuracy is computed by comparing the withheld gold-standard labels to the label inferred by the model with corrected class correspondences. In order to examine the question of inferred label quality, this section examines accuracy over all items in the dataset with one or more annotations. In Figure 5 we chart accuracy as a function of the number of simulated annotations. Because there may be multiple annotations per document, these learning curves extend beyond the length of the corpus. Curves end when every document has roughly 7 annotations.

Majority vote (MAJORITY) suffers from a number of shortcomings. Because it shares no information among instances, accuracy is constant until every item in the dataset has been selected for annotation at least once (at x-coordinate $17,000 \cdot d$) and instances begin to be re-annotated. Not surprisingly, when annotation quality is high and there are enough annotations per instance, majority vote is sufficient to always select the correct label; e.g., when $d = 10$ and annotator quality is HIGH (not shown). However, when annotations are sparse or of low quality, there is significant room for improvement above the baseline. Also notice that because majority vote implicitly assumes that annotation errors are uniform random, MAJORITY particularly struggles to deal with the systematic errors encountered in CONFLICT.

MOMRESP is superior to MAJORITY in the annotation settings considered in Figures 5 and 6. MOMRESP allows information from the features (word counts) to be used when selecting a label; in effect, the features get a vote alongside the annotations. Furthermore, class-conditional word probabilities ϕ are shared among both annotated and unannotated instances. Thus, MOMRESP is able to leverage information from all instances when inferring labels for instances with annotations. This gives MOMRESP a tremendous advantage in the early stages of corpus annotation when annotations are too sparse or uncertain to trust, such as where $d = 1$ and annotator quality is LOW.

4.3. Annotator Error Estimation

We now measure the effectiveness of the model in learning annotator accuracy. Because we simulated annotator error characteristics, we can compare model predictions with the truth. We compute expected annotator accuracy accord-

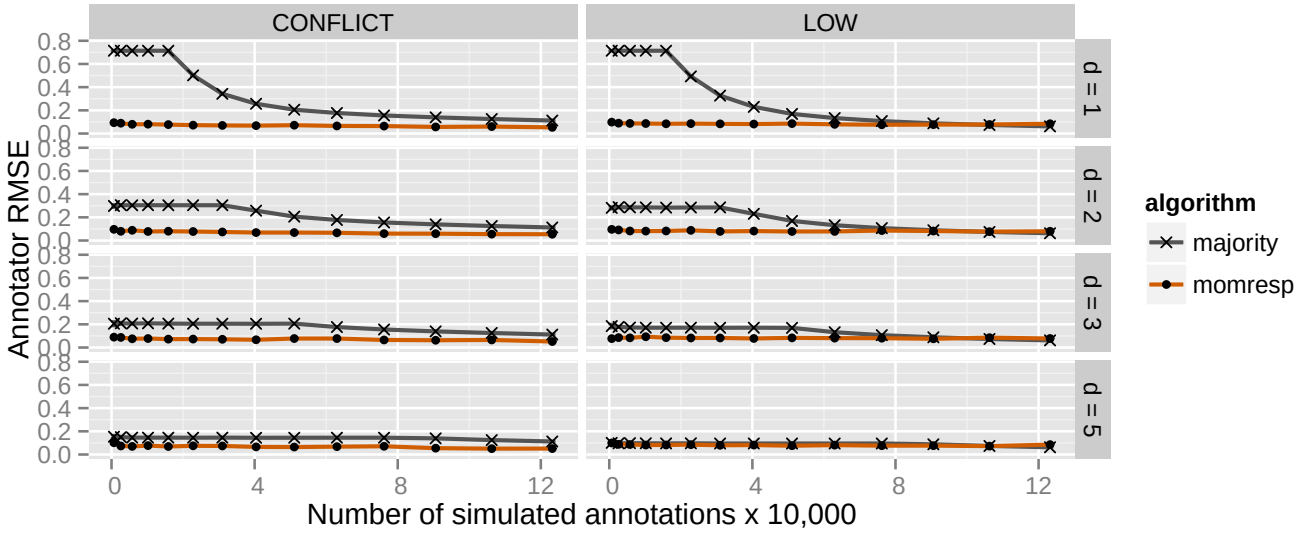


Figure 6: Root mean squared error (RMSE) between true and estimated annotator accuracy.

ing to each model and compare those predictions with the simulation parameters used to generate the dataset. These differences are aggregated across all annotators using root mean-squared error. When RMSE is high, estimates of annotator accuracy are poor; when it is low, estimates are good. For MOMRESP, expected annotator accuracy is computed by inspecting the diagonals of the γ matrices. We compare with a baseline approach (MAJORITY) defined to be the percentage of times that each annotator agreed with the majority vote label.

Figure 6 shows that a model's ability to accurately learn annotator error characteristics is closely related to its ability to infer correct corpus labels. MOMRESP enjoys an advantage in settings stages when it can use data evidence to assess the likely quality of sparse annotations. MAJORITY overtakes MOMRESP as redundant annotations accumulate. MAJORITY provides poor estimates when faced with systematic annotator error in CONFLICT.

4.4. Failure Cases

The modeling assumptions made in MOMRESP cause it to perform very poorly in settings where annotations are far more informative than the natural data clusters. MOMRESP assumes that documents were generated by selecting a class y and subsequently selecting words x and annotations a based on class y. Accordingly, during inference words and annotations are given equal consideration. However, when annotations are highly accurate and sufficiently abundant, they contain far more information than the average word. For example, in Figure 7 annotators are highly accurate, and each document has at least 3 annotations at all times.

To test our hypothesized explanation of MOMRESP's bad behavior in these situations, we tried removing the model's data component, effectively turning it into a Bayesian item-response model. This data-insensitive model is called ITEMRESP in Figure 7. Notice that ITEMRESP performs as well as MAJORITY in annotation-rich settings, but lacks the advantages of MOMRESP in annotation-poor settings such as at the beginning of the MED column. These complementary strengths and weaknesses suggest that were a model able to give appropriate weight to the evidence coming from document words, it could enjoy the complementary strengths of MOMRESP and ITEMRESP. Preliminary tests using ad-hoc weighting methods are promising. A better solution will build the data component weighting into the model in a principled way.

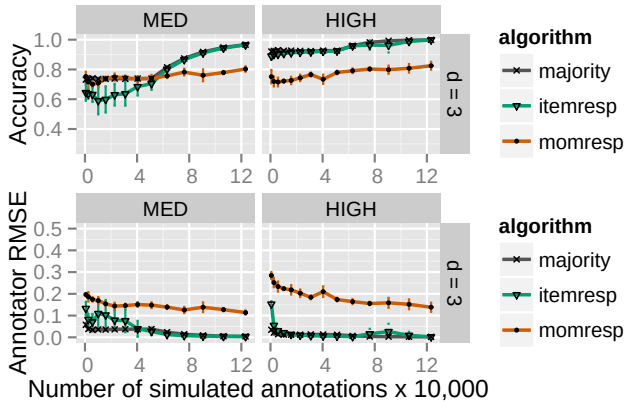


Figure 7: Label accuracy (top) and RMSE between true and estimated annotator accuracy (bottom). d=3 indicates that each document is labeled 3 times before moving to the next document, and MED and HIGH indicate that annotations are highly reliable. MOMRESP's data component becomes a liability in these situations.

5. Conclusions and Future Work

We have presented MOMRESP, a model that incorporates information from both natural data clusters as well as annotations from multiple annotators to infer ground-truth estimates for the document classification task. We have demonstrated that MOMRESP can use unlabeled data to improve estimates of the ground-truth labels over a majority vote baseline dramatically in situations where both annotations are scarce and annotation quality is low as well as in situations where annotators disagree consistently. Corre-

spondingly, in those same situations, estimates of annotator reliability are also stronger than the baseline. Because MOMRESP predictions are subject to label switching, we identified a solution that found nearly optimal predicted class reassignments in a variety of settings using only information available to the model at inference time. MOMRESP does not perform well in cases where annotations are more informative than natural data clusters. We showed that when the data component is removed from the model — turning it into a Bayesian item-response model — it performs well in annotation-rich settings at the expense of performance in annotation-poor settings. Future work will focus on combining the complementary strengths of MOMRESP and data-ignorant item-response models by refining the structure of the model to weight the information from the model's data component in a principled way.

Acknowledgments

We express our appreciation to the BYU Foulton Supercomputing Lab for computing resources that made this work possible, and also to Kristian Heal, Deryle Lonsdale, and Kevin Black for valuable input and feedback.

6. References

Burkard, R. E., Dell'Amico, M., and Martello, S. (2009). *Assignment Problems, Revised Reprint*. Siam.

Carpenter, B. (2008). Multilevel bayesian models of categorical data annotation. *Unpublished manuscript*.

Carroll, J., Haertel, R., McClanahan, P., Ringger, E., and Seppi, K. (2007). Modeling the annotation process for ancient corpus creation. In *Proceedings of ECAL 2007*, pages 25–42. Charles University.

Carroll, J. (2010). *A Bayesian decision theoretical approach to supervised learning, selective sampling, and empirical function optimization*. Ph.D. thesis, Brigham Young University.

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

Gardner, D. and Davies, M. (2007). Pointing out frequent phrasal verbs: A corpus-based analysis. *Tesol Quarterly*, 41(2):339–359.

Haertel, R. A. (2013). *Practical Cost-Conscious Active Learning for Data Annotation in Annotator-Initiated Environments*. Ph.D. thesis, Brigham Young University.

Han, J. (2009). Mining heterogeneous information networks by exploring the power of links. In *Discovery Science*, pages 13–30. Springer.

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

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on ML*, pages 143–151. Morgan Kaufmann Publishers Inc.

Jurgens, D. (2013). Embracing ambiguity: A comparison of annotation methodologies for crowdsourcing word sense labels. In *Proceedings of NAACL-HLT*, pages 556–562.

Kucera, H. and Francis, W. N. (1967). *Computational Analysis of Present-day American English*. Brown University Press, Providence, RI.

Lam, C. P. and Stork, D. G. (2005). Toward optimal labeling strategy under multiple unreliable labelers. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 42–47.

Lin, C., Mausam, and Weld, D. (2012). Dynamically switching between synergistic workflows for crowdsourcing. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Liu, J. S. (1994). The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):pp. 958–966.

Nigam, K., McCallum, A., and Mitchell, T. (2006). Semi-supervised text classification using EM. *Semi-Supervised Learning*, pages 33–56.

Pasternack, J. and Roth, D. (2010). Knowing what to believe (when you already know something). In *COLING*, Beijing, China, 8.

Pasternack, J. and Roth, D. (2013). Latent credibility analysis. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1009–1020.

Raykar, V. C. and Yu, S. (2012). Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *The Journal of Machine Learning Research*, 13:491–518.

Sheng, V., Provost, F., and Ipeirotis, P. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD*, pages 614–622. 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*, pages 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 *Proceedings of EMNLP*, pages 254–263. Association for Computational Linguistics.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.

Walker, D. D. (2012). *Bayesian Text Analytics for Document Collections*. Ph.D. thesis, Brigham Young University.

Weld, D., Mausam, and Dai, P. (2011). Human intelligence needs artificial intelligence. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22:2035–2043.

Zhou, D., Platt, J., Basu, S., and Mao, Y. (2012). Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, volume 25, pages 2204–2212.