Uncertainty-aware generative models for inferring document class prevalence

Katherine A. Keith and **Brendan O'Connor**
College of Information and Computer Sciences
University of Massachusetts Amherst
{kkeith,brenocon}@cs.umass.edu

Abstract

Prevalence estimation is the task of inferring the relative frequency of classes of unlabeled examples in a group—for example, the proportion of a document collection with positive sentiment. Previous work has focused on aggregating and adjusting discriminative individual classifiers to obtain prevalence point estimates. But imperfect classifier accuracy ought to be reflected in uncertainty over the predicted prevalence for scientifically valid inference. In this work, we present (1) a generative probabilistic modeling approach to prevalence estimation, and (2) the construction and evaluation of prevalence confidence intervals; in particular, we demonstrate that an off-the-shelf discriminative classifier can be given a *generative re-interpretation*, by backing out an implicit individual-level likelihood function, which can be used to conduct fast and simple group-level Bayesian inference. Empirically, we demonstrate our approach provides better confidence interval coverage than an alternative, and is dramatically more robust to shifts in the class prior between training and testing.[1]

1 Introduction

The goal of *prevalence estimation* is to infer the relative frequency of classes y_i associated with unlabeled examples (e.g. documents) from a group, $x_i \in \mathcal{D}$. For example, one might want to estimate the proportion of blogs with a positive sentiment towards a political candidate (Hopkins and King, 2010), sentiment of responses to natural disasters on social media (Mandel et al., 2012), or prevalence of car types in street photos to infer neighborhood demographics (Gebru et al., 2017). Often, an analyst wants to compare prevalence between multiple groups, such as inferring prevalence variation over time (e.g., changes to online abuse content (Bissias et al., 2016)), or across other covariates (e.g., changes in police officers' "respect" when speaking to minorities (Voigt et al., 2017)). This problem has been re-introduced in many different fields: as "quantification" in data mining (Forman, 2005, 2008), "prevalence estimation" in statistics and epidemiology (Gart and Buck, 1966), and "class prior estimation" in machine learning (Vucetic and Obradovic, 2001; Saerens et al., 2002). In NLP, SemEval 2016 and 2017 included Twitter sentiment class prevalence tasks (Nakov et al., 2016; Rosenthal et al., 2017).

Prevalence estimation assumes access to a (potentially small) set of labeled examples to train a classifier; but unlike the task of individual classification, the goal is to estimate the proportion of a class among examples in a group. If a perfectly accurate classifier is available, it is trivial to construct a perfect prevalence estimate by counting the classification decisions (§3.1). In fact, most application papers in the previous paragraph use this or a similar aggregation rule to conduct their prevalence estimates. However, classifiers often exhibit errors from different sources, including:

- Shifts in the class distribution from training to testing ($P_{train}(y) \neq P_{test}(y)$). A classifier may be biased toward predicting $P_{train}(y)$.
- Difficult classification tasks (such as predicting sentiment or sarcasm) that result in low accuracy classifiers; this can be exacerbated by limited training data, as is common in social science or industry settings that require manual human annotation for labels.

It is typically assumed (and sometimes confirmed) that when an individual classifier has less than 100% accuracy, it can still give reasonable preva-

[1] Code available at http://slanglab.cs.umass.edu/doc_prevalence and https://github.com/slanglab/doc_prevalence.

lence estimates.[2] However, there is relatively little understanding to what extent the quality of the document-level model impacts prevalence estimates. Imperfect classifier accuracy ought to be reflected in uncertainty over the predicted prevalence.

In this work, we tackle both of these challenges simultaneously, using a generative probabilistic modeling approach to prevalence estimation. This model directly parameterizes and conducts inference for the unknown prevalence, naturally accommodating shifts between training and testing, and also allows us to infer confidence intervals for the prevalence. We show that our best model can be seen as an *implicit likelihood* generative re-interpretation of an off-the-shelf discriminative classifier (§4.2); this unifies it with previous work, and also is easy for a practitioner to apply.

We additionally review several types of class prevalence estimators from the literature (§3), and conduct a robust empirical evaluation on sentiment analysis over hundreds of document groups, illustrating the methods' biases and robustness to class prior shift between training and testing. Our method provides better confidence interval coverage and is more robust to class prior shift than previous methods, and is substantially more accurate than an algorithm in widespread use in political science.

2 Problem definition

We consider two prevalence estimation problems: (1) point prediction and (2) confidence interval prediction. In this work, we are most interested in supervised learning for discrete-valued document labels, with access to a small to moderate number (e.g. around 1000) of labeled documents with text x and label y: $(x_i, y_i) \in \mathcal{D}^{train}$. We restrict attention to binary-valued labels $y \in \{0, 1\}$. At test time, there are one or more groups of unlabeled test documents, $\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(G)}$; for example, one group might be a set of tweets sent during a certain month, or a set of online reviews associated with a particular product. For each group $\mathcal{D}$, let $\theta^* \equiv (1/n) \sum_i^n y_i$ be the true proportion of positive labels (where $n = |\mathcal{D}|$).

The *prevalence point prediction* problem is to take an unlabeled document group $\mathcal{D}$ as input and infer an estimated $\hat{\theta} \in [0, 1]$. Ideally, this point estimate should be close to the true prevalence θ^*; we evaluate this by mean absolute error.

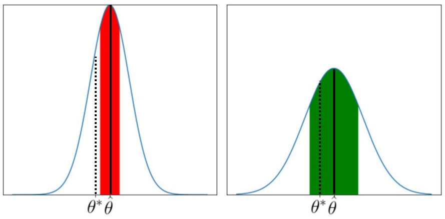


Figure 1: Example posterior distributions with MAP prevalence estimates, $\hat{\theta}$ (solid line) and the true prevalence, θ^* (dashed line). A desirable property is that confidence intervals, technically Bayesian credible intervals, (shaded regions) will be wider for more uncertain models. For example, the wider CI on the right (green) contains θ^* whereas the narrower CI interval on the left (red) does not.

In this work, we are the first (that we know of) to introduce the question of *uncertainty* in prevalence estimation. Since document classifiers are typically far from perfectly accurate, we should expect substantial error in prevalence prediction, and inference methods should quantify such uncertainty. We formalize this as a *prevalence confidence interval* (CI) inference, which takes as input a desired nominal coverage level $(1 - \alpha)$, and predicts a real-valued interval $[\hat{\theta}_{lo}, \hat{\theta}_{hi}] \subseteq [0, 1]$. Ideally, a CI prediction algorithm should have frequentist coverage semantics: over a large number of test groups,[3] $(1-\alpha)\%$ of the predicted intervals ought to contain the true value θ^*. If the problem is hard—for example, the relationship between document features and the label is not captured well by the model—the CI should be wide. We empirically evaluate coverage of CI-aware prevalence inference models. See Fig. 1 for an intuitive example.

3 Review and baselines: Discriminative individual classification aggregation

The most straightforward baseline approach to prevalence estimation is to build on discriminative, supervised learning for individual-level labels, such as binary logistic regression with bag-of-words features, randomized feature hashing

[2] For example, Bissias et al. find a relative mean absolute error of less than 0.01 when the individual classifier has ROC AUC of 0.91.

[3] Or in fact, across many experiments in which the model or algorithm is applied (Wasserman, 2011).

(Weinberger et al., 2009), or neural networks (Goldberg, 2016). Such a model defines an individual document's label probability $p_i \equiv p_\beta(y_i = 1 \mid x_i)$ where parameters β are fit by maximizing regularized likelihood on the labeled training data.

3.1 Classify and Count (CC)

For prevalence point estimation, Forman (2005) defines the "classify and count" (CC) method as simply averaging the most-likely individual label predictions,

$$\hat{\theta}^{CC} = \frac{1}{n}\sum_i 1\{p_i > 0.5\}. \tag{1}$$

This is the most obvious approach for practitioners, but it has at least two weaknesses, which have been addressed in different groups of prior work. First, the class proportions may change between training and test groups, which the Adjusted CC and ReadMe algorithms attempt to fix (§3.2–3.3). Second, it discards probabilistic information, which is remedied by the Probabilistic CC method, and an extension we propose (§3.4–3.5).

3.2 Adjusted Classify and Count (ACC)

CC may encounter problems if the test class distribution is different than the training's. The "adjusted classify-and-count" method (Gart and Buck, 1966; Forman, 2005) treats the classifier output as a proxy variable, and estimates a separate confusion model of classifier output $\hat{y}_i \equiv 1\{p_i > 0.5\}$ conditional on the true label, $p(\hat{y} \mid y)$, from cross-validation within the training set. Assuming the confusion model extends to the test data, a moment-matching approach is then used to infer the true label proportions, by first observing $p_{test}(\hat{y}) = \sum_y p(\hat{y} \mid y) p_{test}(y)$ and solving the linear system for $p_{test}(y)$, the test-time expected class prevalence. Using empirical estimates for the true positive rate TPR $= p(\hat{y} = 1 \mid y = 1)$, and false positive rate FPR $= p(\hat{y} = 1 \mid y = 0)$, and $\hat{\theta}^{CC} = p(\hat{y} = 1)$, it has the closed form

$$\hat{\theta}^{ACC} = \frac{\hat{\theta}^{CC} - \text{FPR}}{\text{TPR} - \text{FPR}}. \tag{2}$$

By design, ACC is more robust to a new test-time prevalence, but it relies on the accuracy of its TPR and FPR estimates, and its lack of probabilistic semantics makes it unclear how to infer confidence intervals.

3.3 ReadMe algorithm

An interesting extension to ACC is to remove the need for a discriminative classifier, by directly modeling text conditional on the latent document class. The ReadMe algorithm, developed in political science (Hopkins and King, 2010), extends ACC's linear system for every term type in a (subsampled and augmented) term vocabulary $\mathcal{V}$, and calculates their class-conditional probabilities from the training data. Assuming these conditional models also hold in the test data, that implies $p_{test}(w) = \sum_y \hat{p}(w \mid y) p_{test}(y)$; the algorithm infers $p_{test}(y)$ by minimizing the squared error of predicted versus empirical term frequencies in the test set. The open-source ReadMe software package[4] has been used in numerous political science studies, including inferring proportions of types of censored Chinese news (King et al., 2013), credit claiming in Congressional press releases (Grimmer et al., 2012), and voter intentions among Twitter messages (Ceron et al., 2015).

ReadMe is theoretically appealing in that it infers latent class prevalences to explain the test group's textual evidence; but as a non-probabilistic model, it does not directly imply a method for confidence intervals (Hopkins and King use the bootstrap). Furthermore, our experiments (§5), contra the original paper, show its implementation exhibits poor performance.

3.4 Probabilistic Classify and Count (PCC)

Both the CC and ACC methods discard uncertainty information from the classification model. In a difficult classification setting, for example, we might expect many probabilities to be near, say, 0.6, in which case the CC method may undercount the negative class. This suggests an alternative method, "probabilistic classify and count" (PCC):

$$\hat{\theta}^{PCC} = \frac{1}{n}\sum_i p_i \tag{3}$$

which is the expected prevalence, $(1/n)\sum_i y_i$, assuming each y_i is distributed according to the original probabilistic classifier.

3.5 PCC Poisson-Binomial distribution (PB-PCC)

If we assume each y_i is conditionally independent given text x_i and model parameters β, this

[4] https://gking.harvard.edu/readme

defines a fully probabilistic model for the class prevalence. Let the latent variable $S = \sum_i y_i$; its distribution is thus Poisson-Binomial (Chen and Liu, 1997). The modeled prevalence distribution $p(\frac{S}{n} \mid \mathcal{D})$ can be exactly inferred by Monte Carlo inference: each iteration samples every y_i and sums for an S sample. The S/n distribution over many iterations can be used to construct a Monte Carlo CDF $\hat{F}$, from which any $[\hat{F}(t), \hat{F}(t{+}1{-}\alpha)]$ is an $(1-\alpha)$-sized credible interval (where $0 \leq t \leq t+1-\alpha \leq 1$). This model has prevalence expectation $E[\frac{S}{n}] = \hat{\theta}^{PCC}$, and variance

$$\text{Var}\left[\frac{S}{n}\right] = \frac{1}{n^2}\sum_i p_i(1-p_i). \tag{4}$$

To a certain degree, this model captures uncertainty in the classifier since per-document variance, $p_i(1-p_i)$, is high when $p_i = 0.5$ and low when near 0 or 1. However, it also has a major weakness—the variance concentrates with a large test group size n, which is the wrong behavior when a classifier is truly noisy, for example, when a classifier is genuinely uncertain and predicts the same constant $p_i = q$ for each document. In this case, the correct behavior would be to maintain a flat, wide posterior belief about θ, which is better accomplished by the generative model we introduce in the subsequent section.

4 Our approach: generative probabilistic modeling

We turn to generative modeling, that seeks to to jointly model the probability of labels and text in both the training and test groups, by assuming a document's text is generated conditional on the document label. Language models have widespread use in natural language processing, and class-conditional models have been used for document classification (e.g. multinomial Naive Bayes; McCallum and Nigam (1998)). We use a similar generative setup to explicitly model a class prevalence for test group g, with a generative story for each (bag-of-words) document i in the group:

$$\theta_g \sim \text{Dist}(\alpha) \tag{5}$$

$$y_{i,g} \sim \text{Bernoulli}(\theta_g) \tag{6}$$

$$x_{i,g} \sim \text{Multinomial}(\phi_{y_{i,g}}) \tag{7}$$

The test group is assumed to have a latent class prior θ_g, which itself has a prior distribution (we assume $\text{Dist}(\alpha) = \text{Unif}(0,1)$ in this work). For each class k, ϕ_k is a class-conditional unigram language model, which is learned from the training data but fixed at test time. We then perform inference to find θ_g that gives a high probability to text data $\{x_i \in \mathcal{D}^{(g)}\}$. Figure 2 shows the probabilistic graphical model.

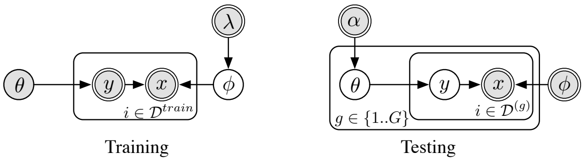


Figure 2: Our generative model for prevalence estimation. **Left:** Class-conditional language models (ϕ) are learned at training time. **Right:** Test-time inference for multiple groups' latent prevalences (θ).

4.1 MNB and Loglin language models

We experiment with two explicit language models in this generative framework: (1) multinomial Naive Bayes (**MNB**), using a training-time symmetric Dirichlet prior $\phi_y \sim \text{Dir}(\lambda/V)$ for vocabulary size V and "pseudocount" λ, and (2) an additive log linear model (**Loglin**, a.k.a. SAGE (Eisenstein et al., 2011)). Loglin estimates words' probabilities as deviations from a background log-probability m,

$$\eta_{y,w} \sim \text{Laplace}(\lambda) \tag{8}$$

$$\phi_{y,w} = \exp(m_w + \eta_{y,w}) / \sum_j \exp(m_j + \eta_{y,j})$$

where m_w is the empirical log probability of a word w among all training documents, and $\eta_{y,w}$ denotes class-specific deviations of the log-probability of a word w, MAP estimated under a sparsity-inducing L1 penalty. Such sparse additive models have been used in both supervised and unsupervised document modeling; for example, as a document-level posterior classifier it outperforms MNB (Eisenstein et al., 2011), or even discriminative models (Taddy, 2013), and its sparsity helps interpretability for analyzing political, literary, and legal texts (Monroe et al., 2008; Sim et al., 2013; Bamman et al., 2014; Wang et al., 2012).

4.2 Implicit likelihoods from discriminative classifiers (LR-Implicit)

This generative formulation has a major advantage over the discriminative, CC-style aggregation models because it sets up a likelihood and posterior distribution over θ. But in terms of document modeling for classification purposes, the independence assumptions of the generative model are typically too strong, and for document-level classification, discriminative models tend to outperform similarly parameterized generative ones, especially when the training set is sufficiently large (Ng and Jordan, 2002). Thus, discriminative models may have information better suited to class prevalence inference. Also, since the most common practice for document classification is to use discriminative models, it would be helpful to more effectively use discriminative posteriors within our generative context.

In Naive Bayes-style generative document classification, the model defines $p_{gen}(x \mid y)$ and class prior $p(y)$, which are combined to calculate the posterior $p_{gen}(y \mid x) \propto p_{gen}(x \mid y)p(y)$. Discriminative models, by contrast, directly define a $p_{disc}(y \mid x)$. We can, however, expand this quantity via Bayes Rule:

$$p_{\text{disc}}(y \mid x) = p_{\text{implicit}'}(x \mid y) p_{\text{train}}(y)/p(x). \quad (9)$$

The "implicit document likelihood" $p_{\text{implicit}'}(x \mid y)$ is a likelihood function that, combined with a particular class prior $p(y)$, would have resulted in the same posterior predicted by the discriminative model. Given the discriminative posterior predictions and the training-time class prior $p_{\text{train}}(y) = \hat{\theta}_{\text{train}}$, an implicit likelihood function can be backed out for any particular document x; we define the "simple implicit" likelihood for document x to be:

$$p_{\text{implicit}}(x \mid y) = p_{\text{disc}}(y \mid x)/\hat{\theta}_{\text{train}}. \quad (10)$$

This takes the form of a correction of the discriminative posterior, by dividing out the training-time class prevalence.[5]

Our **LR-Implicit** generative model uses the same class prevalence and document label generation setup as before, but to calculate the individual documents' $p(x \mid y)$ probabilities, it uses p_{implicit} based on a logistic regression p_{disc}.[6]

This model is inspired by Saerens et al. (2002)'s EM algorithm for adjusting a classifier for a test set's class prior; they derive it differently by applying the assumption $p_{\text{train}}(x \mid y) = p_{\text{test}}(x \mid y)$, expanding each side with Bayes' Rule, solving for $p_{\text{test}}(y \mid x)$, then estimating $p_{\text{test}}(y)$ via EM. This in fact optimizes the same marginal likelihood function in the next section under the implicit-discriminative generative model; our formulation broadens it as a fully Bayesian or likelihood-based model.

4.3 Inference

To estimate class prevalence, we use the marginal log likelihood over θ to obtain a posterior over θ. For each each test group g, we have the marginal log probability of all document texts,

$$\begin{aligned} \text{MLL}_g(\theta) &\equiv \log p(\mathcal{D}^{(g)} \mid \theta) \quad (11) \\ &= \sum_{i \in \mathcal{D}^{(g)}} \log \sum_{y \in \{0,1\}} p(x_i, y_i = y \mid \theta) \\ &= \sum_{i \in \mathcal{D}^{(g)}} \log \left(\theta L_i^+ + (1-\theta) L_i^- \right), \end{aligned}$$

where we denote the class-conditional document text likelihoods $L_i^+ \equiv p(x_i \mid y_i = 1)$ and $L_i^- \equiv p(x_i \mid y_i = 0)$. The gradient for an individual document is $(L_i^+ - L_i^-)/(\theta L_i^+ + (1-\theta)L_i^-)$; intuitively, the sign of the numerator says that documents that are more likely under the positive than negative class encourage higher likelihood for larger values of θ. When the model is uncertain about a document—that is, when $L_i^+ \approx L_i^-$—that document contributes a relatively flat likelihood curve, expressing little preference for likely values of θ. If a model is more heavily regularized—for example, when the log-linear additive model is more dominated by the background language model—this condition tends to hold for the documents, leading to a flat, highly uncertain likelihood curve.

The marginal log likelihood is unimodal over $\theta \in [0,1]$, since it is concave, being a sum of concave log-linear functions, and having negative curvature:

$$\frac{\partial^2 \text{MLL}_g}{\partial \theta^2} = - \sum_{i \in \mathcal{D}^{(g)}} \left(\frac{L_i^+ - L_i^-}{\theta L_i^+ + (1-\theta) L_i^-} \right)^2 . \quad (12)$$

Since it is concave and there is only one parameter, a very wide variety of techniques could be

[5] Technically, $p_{\text{implicit}'}$ is retrievable only up to a constant, and p_{implicit} is one particular compatible implicit likelihood, since it can be multiplied by any constant and is still consistent with Eq. 9, and would give rise to the same document- and group-level posteriors.

[6] The implicit likelihood still has the form of a logistic regression, adjusting its bias term: if $p_{\text{disc}}(y \mid x) = \sigma(\beta' x + \beta_0)$, then $p_{\text{implicit}}(x \mid y) = \sigma(\beta' x + \beta_0 - \log(\theta_{\text{train}}/(1-\theta_{\text{train}})))$.

used to reliably find a mode, including EM or first- or second-order methods. At least two approaches to inferring confidence intervals are possible. One is to use a central limit theorem-style approximation, assuming the sampling distribution is approximated by a normal with mean θ^{MLE} and variance $-[\partial^2 \text{MLL}_g / \partial \theta^2]^{-1}$. The second, which we focus on, is Bayesian estimation for $\log p(\theta_g \mid \mathcal{D}^{(g)}) \propto \log p(\theta_g) + \text{MLL}_g(\theta_g)$ by simply using a grid search over values $\theta \in \{0.001, 0.002, ...0.999\}$ to infer both the posterior mode θ^{MAP} as well as a 90% highest posterior density interval.[7] In small-scale experiments, this model had very similar results to the central limit theorem (with EM for θ^{MLE}).

5 Experiments

5.1 Data

In order to compare document class prevalence estimators, we desire datasets that (1) have natural document groups that correspond to realistic, real-world applications, (2) have a large number of test groups (hundreds or more), and (3) are freely available for academic research. It has been a challenge to fulfill these criteria in previous work. Nakov et al. (2016) conduct large-scale manual annotation of Twitter sentiment for SemEval 2016 Task 4, with topic-based test groups; unfortunately, redistribution is restricted to message IDs, making the original dataset difficult to reconstruct under Twitter's terms of service if messages have since been deleted. Bella et al. (2010) and Esuli and Sebastiani (2015) use large, pre-existing labeled document corpora, but they do not contain natural groups; evaluations utilize randomly sampled synthetic groups.

To better fulfill these criteria, we select the task of business review sentiment prevalence, where the goal is to estimate the proportion of reviews that are positive for one particular business; specifically, we use labeled data from the Yelp Dataset Challenge Round Nine[8] corpus, which consists of 4.1M reviews by 1M users for 144K businesses. We sample 500 businesses with at least 200 reviews each as the test groups. We treat the task as binary classification, and assign $y_i = 1$ to reviews with 3 or more stars. This task seems reasonably representative of real-world sentiment analysis problems, and this type of dataset can easily be collected and reproduced from Yelp or other widely available review data.

For training, we simulate a small-scale annotation project by sampling 2000 labeled documents from the rest of the corpus. This is a **natural** prevalence that on average is about the same as the test groups, though individual test groups may have a much different prevalence (ranging from 0.096 to 0.997, mean (stdev) 0.823 (0.136)). We also construct a **synthetic** training setting with a highly skewed class prior, selecting 2000 documents with a 0.1 class prevalence (i.e. 200 positive documents in the group). In each case, for every model, we re-run and average results over 10 different samples of the training set. For preprocessing, we tokenize with NLTK[9] and lowercase.

5.2 Model training

We use L1 regularization for logistic regression based on the vector of a documents' word counts, to be most directly comparable to the generative models; for each model, we select its hyperparameter (LR and Loglin's λ, or MNB's pseudocount) by minimizing cross-validated cross-entropy of individual document posteriors (within the labeled training set), over a grid search of powers of 2. The log-linear additive model is trained with OWL-QN (Andrew and Gao, 2007)[10] and the logistic regression model is trained with the default implementation in scikit-learn (Pedregosa et al., 2011).[11] We used ReadMe with its default parameters.[12]

5.3 Results

For each of the 500 test groups, we calculate a prevalence point estimate $\hat{\theta}$ with each method, and evaluate by averaging across groups for mean absolute error $\sum_g |\hat{\theta}_g - \theta^*_g|$ and bias $\sum_g (\hat{\theta}_g - \theta^*_g)$.[13] For the models that allow for confidence interval

[7] Since we use a uniform prior, this is just the MLE. Technically, we used a prior of Beta$(1.0001, 1.0001)$ to avoid certain issues with tie-breaking, but it was not necessary.

[8] Downloaded June 2017 from https://www.yelp.com/dataset_challenge.

[9] http://www.nltk.org/

[10] Via github.com/larsmans/pylbfgs

[11] Version 0.18.2

[12] Version 0.99837 from https://gking.harvard.edu/readme, with default parameters features=15, n.subset=300, prob.wt=1. We bypass the ReadMe software's text preprocessing pipeline, and instead have it use nearly the same document-term matrices as the other models. Since it only handles binary document-term matrices, we transformed counts to indicators; with other models this change only made a minor difference in results.

[13] For the generative (MLL) models, $\hat{\theta}$ is the MAP estimate; the posterior mean gives similar results.

		Natural training prevalence ≈ 0.8				Synthetic training prevalence = 0.1			
		Point est.		CIs		Point est.		CIs	
		MAE	Bias	Cover.	Width	MAE	Bias	Cover.	Width
Const.	Pred. train mean	0.114	-0.045	—	—	0.723	-0.723	—	—
	Pred. 100%	0.177	0.177	—	—	0.177	0.177	—	—
	ReadMe	0.233	-0.222	—	—	0.383	-0.382	—	—
Disc. (LR)	CC	0.048	0.042	—	—	0.503	-0.503	—	—
	ACC	0.048	-0.001	—	—	0.132	-0.015	—	—
	PB-PCC	0.049	-0.017	0.283	0.044	0.464	-0.464	0.001	0.054
Gen. (MLL)	MNB	0.078	0.058	0.120	0.046	0.199	-0.199	0.022	0.073
	Loglin	0.089	-0.070	0.410	0.100	0.140	-0.036	0.510	0.273
	LR-Implicit	0.050	0.001	0.454	0.074	0.069	-0.051	0.439	0.082

Table 1: Mean absolute error (MAE), bias, nominal 90% confidence interval coverage, and average CI width for the 500 Yelp data test groups, averaged over 10 simulations of resampled training (2000 document) sets. We examine both the natural positive class training prevalence ($E[\theta_{train}] = 0.7783$), and a synthetic fixed prevalence of 0.1. Dashes indicate the methods that are not able to calculate confidence intervals.

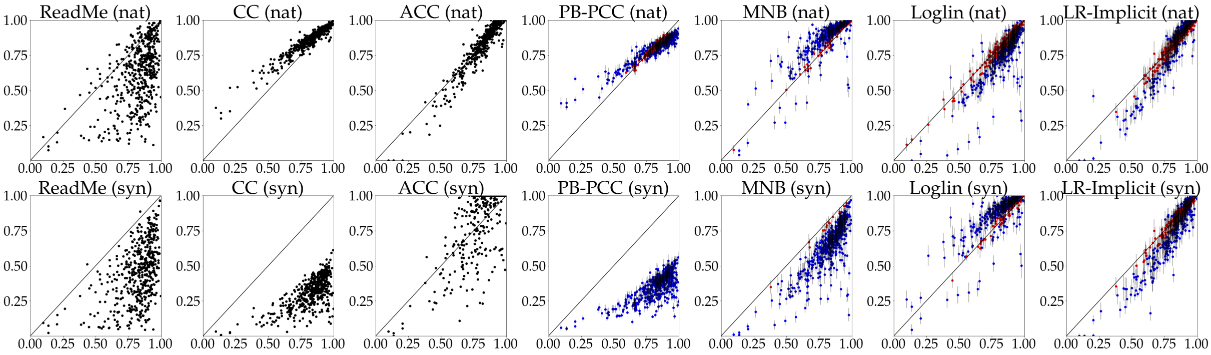


Figure 3: Gold prevalence θ^* (x-axis) versus predicted prevalence $\hat{\theta}$ (y-axis) for each of the 500 test groups with **natural** (nat) training prevalence (top row) and **synthetic** (syn) 0.1 training prevalence (bottom row). A black $y = x$ line is plotted for visualization. For the models that allow for confidence intervals, 90% CIs for each group are given by the faint grey lines. Blue dots indicate the CI does not contain θ^* and red dots indicate the CI does contain θ^*. For each setting, we show the the model with median MAE across training resamplings.

prediction, we infer 90% intervals and calculate coverage, which is best if it is 0.90. We also report average CI width; a narrower interval indicates more confidence (even if misplaced). Results are in Table 1; every result is averaged over 10 resamplings of the training set.

The ReadMe software did not have competitive performance; we hope in follow-up work to understand why Hopkins and King found it had considerably stronger performance than SVM-based CC.

For the natural training class prevalence setting (first column, Table 1), the discriminative-based models (CC, PCC and the adjusted variants ACC and LR-Implicit) all have very similar point estimate performance, outperforming the purely generative models (MNB and Loglin). For CI coverage, the log-linear and LR-Implicit generative models have significantly better coverage than the discriminative model (PB-PCC) or MNB. Future work is required to improve coverage to be closer to the nominal ideal of 90%.

By contrast, when the class prevalences are mismatched (second column, Table 1), the non-adjusted CC and PCC methods give extremely poor and biased point estimates, and PB-PCC has incredibly poor CI coverage. ACC and the generative models do much better, presumably because their models directly allow for variability in the test class prior. While Loglin has somewhat higher coverage in this setting, overall, LR-Implicit has

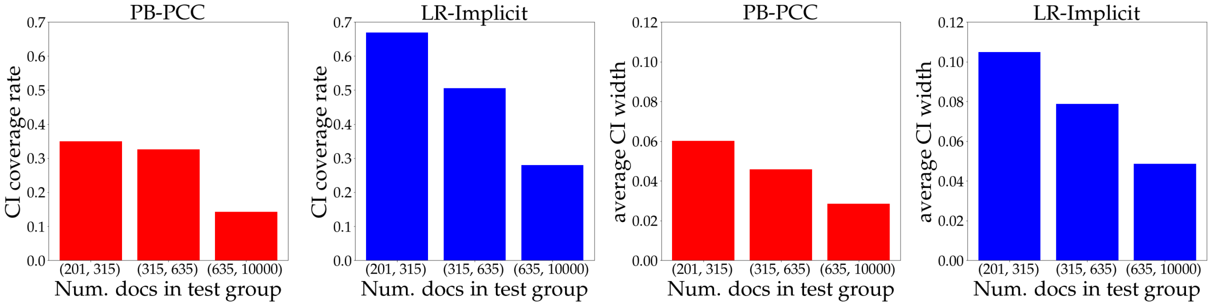
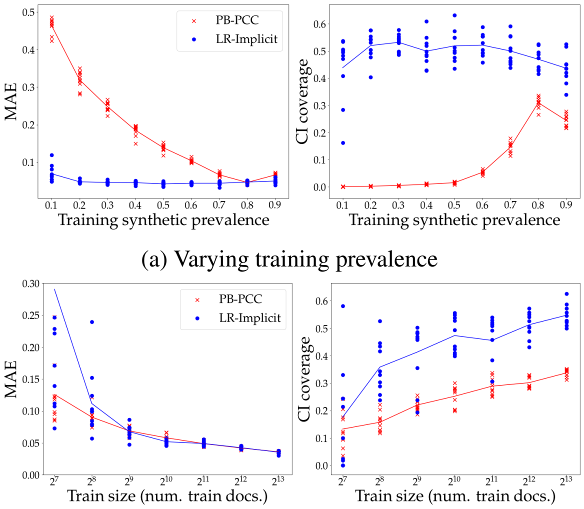


Figure 4: CI coverage rate (left two graphs) and average CI width (right two graphs) for three bins of the test groups, binned by number of documents.



(a) Varying training prevalence

(b) Varying training size

Figure 5: MAE and 90% CI coverage for PB-PCC while varying **(a)** training prevalence (the proportion of the 2000 training documents with positive reviews) and **(b)** training size (number of documents in the training data) with natural prevalence. Lines are the averages over 10 resamplings of training sets and points represent one resampling.

consistently strong performance in both training settings, and for both point estimation and (relatively, at leas) confidence intervals.

Figure 3 shows θ^* versus $\hat{\theta}$ for each of the 500 test groups for each of the models, including predicted CIs. CC's and PCC's erroneous assumptions are directly viewable: in the natural prevalence setting, the slope shallower than 1, indicating a persistent under-sensitivity to the true class prevalence—unlike ACC and the generative models. In the synthetic training case, CC and PCC wildly underpredict, presumably because they are biased by the low training-time prevalence $\theta_{\text{train}} = 0.1$.

5.4 Comparison of PB-PCC and LR-Implicit

Since PB-PCC and LR-Implicit represent the strongest members of non-adjusted classification aggregation and generative modeling, respectively, we further compare their results. When varying synthetic training prevalence across 0.1 to 0.9 (Figure 5a), LR-Implicit has much better MAE in all settings except near the natural prevalence (the test groups have, on average, 0.82 positive prevalence), and consistently stronger CI coverage.

Figure 5b shows results for natural class prevalence when varying the training set size. Unfortunately, LR-Implicit is disadvantaged at very small test sizes—its MAE is higher when there are only a few hundred training documents ($\leq 2^8 = 256$), though performance converges after that. We suspect this may occur because, when textual evidence is weak, the classifier learns to more heavily rely on its bias term, which can be a useful form of bias when the training class prevalence matches the test groups (on average). However, at all levels, LR-Implicit's coverage is better.

Since we hypothesized that PB-PCC may be overconfident for large test groups (§3.5), we test this by binning test groups by the number of documents per group. Figure 4 confirms that PB-PCC exhibits overconfidence for larger groups (smaller CI width alongside lower CI coverage), but LR-Implicit suffers from the same problem as well.

6 Additional Related Work

González et al. (2017a) reviews the class prevalence estimation literature, and we note a few threads of work here. Bella et al. (2010) propose a probabilistic variant of ACC, and Esuli and Sebastiani (2015) compare many methods on news article topics (RCV1) and medical record subject heading (OHSUMED-S) class prevalence tasks, finding varying results among CC, ACC, and PCC. A number of other empirical evaluations were conducted in two SemEval Twitter sentiment prevalence shared tasks, with varying results among these and other methods with a range of classifiers (Nakov et al., 2016; Rosenthal et al., 2017); Nakov et al. note that CC was often one of the strongest methods. Esuli and Sebastiani as well as Xue and Weiss (2009) present semi-supervised loss-augmented classifier training methods to improve prevalence estimation. Tasche (2017) presents theoretical results for ACC and Saerens et al.'s EM method (what we call the LR-Implicit MLE), arguing they correctly predict θ^* under class prior shift; we confirm that those two methods are indeed better than many alternatives in our empirical evaluation. While we focus on inference of the test-time class prior as a class prevalence estimate, Saerens et al. (2002) also show their method can improve individual-level classification accuracy, which Sulc and Matas (2018) use for image classification. (From the viewpoint of individual classification, this phenomenon is known as prior probability shift (Moreno-Torres et al., 2012).) González et al. (2017b) and Card and Smith (2018), similarly to our results, find that CC is much poorer than ACC under class shift. Card and Smith also show that PCC can be sensitive to properties of the classifier, finding that well-calibrated classifiers can give strong performance. They argue that discriminative aggregation models are appropriate for tasks where humans respond to text. Jerzak et al. (2018) analyze issues in class prevalence estimation and propose the ReadMe2 algorithm, which adds external word embeddings, optimization-based dimension reduction, and similarity matching to ReadMe's moment-matching framework.

7 Conclusion

Document class prevalence estimation is a widespread and much understudied task. We show that simple and obvious classifier aggregation methods display consistent biases, especially under class prior shift. Given how widely some of the less effective methods are used, machine learning and natural language processing research could have real impact in this space.

We also call attention to the need for *uncertainty aware* inference—methods that give confidence intervals to summarize their uncertainty. While our method is a first step, future work is necessary to better understand the problem and develop methods with improved coverage. Also, our framework can accommodate a wide array of document and language models—while we focus on bag-of-words models, recent advances in sequence, neural, and attention-based document models could be added directly to our generative model, or used as a discriminative-implicit component. The overall framework could also be extended to multiclass, and potentially, structured prediction settings.

References

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Antonio Bella, Cesar Ferri, José Hernández-Orallo, and Maria Jose Ramirez-Quintana. 2010. Quantification via probability estimators. In *IEEE 10th International Conference on Data Mining (ICDM)*.

George Bissias, Brian Levine, Marc Liberatore, Brian Lynn, Juston Moore, Hanna Wallach, and Janis Wolak. 2016. Characterization of contact offenders and child exploitation material trafficking on five peer-to-peer networks. *Child Abuse & Neglect*, 52:185 – 199.

Dallas Card and Noah A Smith. 2018. The importance of calibration for estimating proportions from annotations. In *Proceedings of Empirical Methods in Natural Language Processing*.

Andrea Ceron, Luigi Curini, and Stefano M Iacus. 2015. Using sentiment analysis to monitor electoral campaigns: Method matters—evidence from the United States and Italy. *Social Science Computer Review*, 33(1):3–20.

Sean X. Chen and Jun S. Liu. 1997. Statistical applications of the Poisson-binomial and conditional Bernoulli distributions. *Statistica Sinica*, pages 875–892.

Jacob Eisenstein, Amr Ahmed, and Eric P. Xing. 2011. Sparse additive generative models of text. In *Proceedings of ICML*.

Andrea Esuli and Fabrizio Sebastiani. 2015. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):27.

George Forman. 2005. Counting positives accurately despite inaccurate classification. In *European Conference on Machine Learning*.

George Forman. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.

John J. Gart and Alfred A. Buck. 1966. Comparison of a screening test and a reference test in epidemiologic studies. II. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology*, 83(3):593–602.

Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. 2017. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *Proceedings of the National Academy of Sciences*.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Pablo González, Alberto Castaño, Nitesh V. Chawla, and Juan José Del Coz. 2017a. A review on quantification learning. *ACM Computing Surveys*, 50(5):74:1–74:40.

Pablo González, Jorge Díez, Nitesh Chawla, and Juan José del Coz. 2017b. Why is quantification an interesting learning problem? *Progress in Artificial Intelligence*, 6(1):53–58.

Justin Grimmer, Solomon Messing, and Sean J Westwood. 2012. How words and money cultivate a personal vote: The effect of legislator credit claiming on constituent credit allocation. *American Political Science Review*, 106(4):703–719.

Daniel J. Hopkins and Gary King. 2010. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247.

Connor T. Jerzak, Gary King, and Anton Strezhnev. 2018. An improved method of automated nonparametric content analysis for social science. *Working paper*.

Gary King, Jennifer Pan, and Margaret E Roberts. 2013. How censorship in china allows government criticism but silences collective expression. *American Political Science Review*, 107(2):326–343.

Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during Hurricane Irene. In *Proceedings of the Second Workshop on Language in Social Media*, pages 27–36. Association for Computational Linguistics.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*.

Burt L. Monroe, Michael P. Colaresi, and Kevin M. Quinn. 2008. Fightin' Words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372.

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 Task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*.

Andrew Ng and Michael Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems (NIPS)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 Task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. 2002. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41.

Yanchuan Sim, Brice Acree, Justin H. Gross, and Noah A. Smith. 2013. Measuring ideological proportions in political speeches. In *Proceedings of EMNLP*.

Milan Sulc and Jiri Matas. 2018. Improving cnn classifiers by estimating test-time priors. *arXiv preprint arXiv:1805.08235*.

Matt Taddy. 2013. Multinomial inverse regression for text analysis. *Journal of the American Statistical Association*, 108(503):755–770.

Dirk Tasche. 2017. Fisher consistency for prior probability shift. *Journal of Machine Learning Research*, 18(95):1–32.

Rob Voigt, Nicholas P Camp, Vinodkumar Prabhakaran, William L Hamilton, Rebecca C Hetey, Camilla M Griffiths, David Jurgens, Dan Jurafsky, and Jennifer L Eberhardt. 2017. Language from police body camera footage shows racial disparities in officer respect. *Proceedings of the National Academy of Sciences*.

Slobodan Vucetic and Zoran Obradovic. 2001. Classification on data with biased class distribution. In *European Conference on Machine Learning*, pages 527–538. Springer.

William Yang Wang, Elijah Mayfield, Suresh Naidu, and Jeremiah Dittmar. 2012. Historical analysis of legal opinions with a sparse mixed-effects latent variable model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Larry Wasserman. 2011. *All of statistics*. Springer Science & Business Media.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*.

Jack Chongjie Xue and Gary M Weiss. 2009. Quantification and semi-supervised classification methods for handling changes in class distribution. In *Proceedings of KDD*.