**Noname manuscript No.**
(will be inserted by the editor)

# Mixtures of Biased Sentiment Analysers

Received: date / Accepted: date

**Abstract** Modelling bias is an important consideration when dealing with inexpert annotations. We are concerned with training a classifier to perform sentiment analysis on news media articles, some of which have been manually annotated by volunteers. The classifier is trained on the words in the articles and then applied to non-annotated articles. In previous work we found that a joint estimation of the annotator biases and the classifier parameters performed better than estimation of the biases followed by training of the classifier.

An important question follows from this result: can the annotators be usefully clustered into either predetermined or data-driven clusters, based on their biases? If so, such a clustering could be used to select, drop or otherwise categorise the annotators in a crowdsourcing task. This paper presents work on fitting a finite mixture model to the annotators' bias. We develop a model and an algorithm and demonstrate its properties on simulated data. We then demonstrate the clustering that exists in our motivating dataset, namely the analysis of potentially economically relevant news articles from Irish online news sources.

**Keywords** mixture model · sentiment analysis · bias modelling · crowdsourcing · EM algorithm.

## 1 Introduction

According to Pang and Lee (2004)

> "Sentiment analysis seeks to identify the viewpoint(s) underlying a text span; an example application is classifying a movie review as thumbs up or thumbs down".

Address(es) of author(s) should be given

The internet provides a supply of massive amounts of readily available data that is amenable to such analysis. Examples and applications abound including political debates (Thomas et al, 2006), product ratings (Blitzer et al, 2007) and even mining twitter for stock market prediction (Bollen and Mao, 2011).

Brew et al (2010b) examined the economically-relevant sentiment carried by online news articles. These articles were manually annotated by a small group of volunteers as carrying positive, negative or irrelevant sentiment regarding the Irish economy at time of press. Many annotations were missing and the volunteers were, crucially, inexpert. The two goals were to (a) infer the "true" underlying sentiment of the articles and (b) to train a classifier on the words in these annotated articles to be used for sentiment extraction of un-annotated articles. Brew et al (2010b) note that 45% of the articles do not have consensus annotations and that "there is some evidence that the learning process would be better off without them [articles with low consensus]".

However, this is throwing away potentially useful data. We therefore propose keeping these low consensus articles. In Salter-Townshend and Murphy (2012) it was found that the manual annotators were systematically biased. An Expectation-Maximisation algorithm was employed to perform inference on these biases as well as the parameters of a Naive Bayes classifier for labelling the un-annotated articles. In that work it was found that:

1. Modelling the biases lead to better results than just using a majority vote for which articles were positive, negative or irrelevant.
2. Joint estimation of the biases and the classifier is more accurate than learning the biases and then training the classifier.

We now extend that work by exploring a mixture model of sentiment analysers. That is, rather than each annotator having his or her own bias matrix, we impose a finite mixture model with a low number of possible bias matrix configurations. Thus there are clusters of similarly biased annotators. We jointly estimate this mixture model and train a classifier to learn the sentiment carrying terms.

The notation we use in this paper is given in Table 1. Our goal is to use annotations $y$ and words $w$ to simultaneously infer the unobserved types of the articles $\boldsymbol{T}$, the cluster specific bias matrices $\pi$, annotator cluster memberships $\boldsymbol{Z}$ and the parameters of a word based classifier $\theta$ that may then be used to classify un-annotated articles.

## 2 Related Methods

We briefly discuss two related methods for simultaneous clustering of the rows and columns of a matrix for data of form similar to our motivating problem.

**Table 1** Notation used in this paper. $B$ is the number of annotator clusters and $J$ is the number of article types which is 3 in our examples.

| | |
|---|---|
| $K$ | number of annotators |
| $N$ | number of word terms in the articles |
| $A$ | number of articles |
| $y$ | $K \times A$ binary matrix of annotations |
| $w$ | $A \times N$ binary matrix of word term presences |
| $\boldsymbol{T}$ | $A \times J$ length binary indicators for the types of the articles (instances) |
| $\boldsymbol{Z}$ | $K \times B$ length binary indicators for the cluster memberships of the annotators |
| $\pi$ | $B \times J \times J$ cluster specific matrix of annotation error rates |
| $\theta$ | $N \times J$ matrix of word term classifier probabilities |

### 2.1 Co-Clustering and Bayesian Co-clustering

These methods seek to perform joint clustering of the rows and columns of dyadic data connecting two entities. Examples include clustering users into groups with similar tastes and movies into genres, based on a sparse matrix of user ratings of movies. A key difference between this problem and our own is that rather than using ratings to inform clustering of movies the annotations we observe correspond directly to underlying article types. Existing EM algorithms for co-clustering use a variational approximation to the joint likelihood for the row (user) and column (movie) clusterings. This is required as it is impractical to sum over all user and movie cluster assignments. Shan and Banerjee (2008) state that

> "The probability of the entire matrix is, however, not the product of all such marginal probabilities. That is because $\pi_1$ for any row and $\pi_2$ for any column are sampled only once for all entries in this row/column. Therefore, the model introduces a coupling between observations in the same row/column, so they are not statistically independent. Note that this is a crucial departure from most mixture models, which assume the joint probability of all data points to be simply a product of the marginal probabilities of each point."

Our model also samples article type only once for all annotators (rows) and annotator cluster only once for all articles (columns). We can in fact compute the required expected log-likelihood as $J = 3$ and $B$ is also low; thus the expectation is a tractable summation. Furthermore our annotations map directly to the column clusterings. However, we also explore a variational approximation to this computational bottleneck. Our approximation differs from Shan and Banerjee (2008) in that we directly use the product of marginal expectations for the rows and columns rather than formulating a new factorised variational distribution.

### 2.2 Model-based Block Clustering

Govaert and Nadif (2003) propose the Block CEM algorithm for performing simultaneous model-based block clustering of objects and variables. This is closely related to co-clustering but they avoid intractability of the likelihood and use of a variational approximation by focussing on the classification (or complete) likelihood approach. The CEM algorithm is the EM algorithm but with a (hard) classification step added. Thus $\hat{\boldsymbol{T}}$ is estimated based on the maximal assignment of the annotators clusterings $\boldsymbol{Z}$ which is treated as observed data for that step. $\hat{\boldsymbol{Z}}$ is then similarly updated treating the articles assignments $\boldsymbol{T}$ as known. This circumvents the dependence issue between $\hat{\boldsymbol{T}}$ and $\hat{\boldsymbol{Z}}$ as the update on one is conditioned on the other at each E-Step iteration.

Subsequently, Govaert and Nadif (2005) proposed a Block GEM algorithm as an improvement on the block CEM algorithm. The authors note that CEM "is not expected to converge to the ML estimate of the parameters and yields inconsistent estimates". They therefore develop an EM algorithm without a classification step but note that "it is not possible to find the value of $\theta$ that globally maximises the function" and therefore employ the Generalised EM algorithm (GEM; Dempster et al (1977)) which guarantees only to increase the complete log-likelihood as a function on the parameters at each step (rather than maximise it).

More recently, Govaert and Nadif (2008) find that "these methods give encouraging results using simulated binary data, and are better than the other methods" (CEM and decoupled clustering of rows and columns). The block EM algorithm has two steps, each of which is sub-composed of both E and M steps. The first step finds the expected value of $\boldsymbol{Z}$ given the observed data and $\hat{\boldsymbol{T}}$ (E-step) and then finds maximum likelihood values for the parameters (M-step). These E and M parts of the first step are iterated until convergence. Similarly, the second step iterates between finding the expected value of $\boldsymbol{T}$ and the maximum likelihood values of the parameters until convergence. The algorithm iterates over the first and second steps until overall convergence. A variational approximation that $\mathbf{E}[\boldsymbol{T}\boldsymbol{Z}|y] \simeq \mathbf{E}[\boldsymbol{T}|y,\boldsymbol{Z}]\mathbf{E}[\boldsymbol{Z}|y,\boldsymbol{T}]$ is employed as per co-clustering and the authors note that this is equivalent to maximising the fuzzy criterion of Hathaway (1986). We compare an approximate E-step part of our algorithm with this method in Section 4.4.

## 3 Model

Dawid and Skene (1979) modelled the imperfect annotation of patient medical records using error rates of reporting. They introduced the unobserved true response of a patient and modelled these and the observed reported responses as dependent on conditional probability matrices. These bias matrices hold the probabilities of each possible report given a true response. The true responses and the bias matrices are then estimated from the observed data

(the reported responses) using an Expectation-Maximisation (EM) algorithm (Dempster et al, 1977).

Raykar et al (2010) showed that, in the context of training a classifier using such biased annotations, a single algorithm that jointly estimates the parameters of the classifier and the biases outperforms a two stage approach that first estimates the biases and then trains the classifier. Salter-Townshend and Murphy (2012) then demonstrated this approach on the media dataset of Brew et al (2010b). The observed annotations are denoted as $y_{aj}^{(k)}$. This is a binary indicator variable that is 1 iff annotator $k$ has annotated article $a$ as being of type $j$. We now introduce the true (but unobserved) types of the articles, $\boldsymbol{T}$.

**Definition 1** $\boldsymbol{T}_{ai} = 1 \iff$ *article $a$ is of type $i$.*

$\underline{T}_{a,}$ is a vector of length $J = 3$, all zeros but with a 1 in the $i^{th}$ entry indicating that article $a$ is of sentiment $i$. Hence, the complete-data likelihood of the full annotation dataset (including unobserved true types) across all $A$ articles is

$$P(y, \boldsymbol{T}|\pi, p) \propto \prod_{a=1}^{A} \prod_{i=1}^{J} \left\{ p_i \prod_{k=1}^{K} \prod_{j=1}^{J} (\pi_{ij}^{(k)})^{y_{aj}^{(k)}} \right\}^{T_{ai}} \tag{1}$$

where $p_i$ is the marginal probability of type $i$.

The key extension demonstrated in this paper is that rather than model a unique bias matrix for each annotator, we assume that there are a finite (perhaps small) number of bias matrix configurations. Each such matrix then corresponds to an annotator "type" and there are $1 \leq B \leq K$ of these, where $K$ is the number of annotators. We will model such types via a finite mixture model as per McLachlan and Peel (2000). We therefore introduce another set of latent (unobserved) binary indicator vectors, $\boldsymbol{Z}$.

**Definition 2** $\boldsymbol{Z}_{kb} = 1 \iff$ *annotator $k$ is of type $b$.*

$\underline{Z}_{k,}$ is a vector of length $B$ that is all zeros but with a 1 in the $b^{th}$ entry iff annotator $k$ is of type $b$.

Thus, if annotator $k$ is of type $b$ then $\boldsymbol{Z}_{kb} = 1$ and the probability of recording annotation $j$ given a *true* (but unobserved) type $i$ is given by $P(y_{aj}^{(k)}) = \pi_{ij}^{(b)}$. In this case, $\boldsymbol{T}_{ai} = 1$ and so in general

$$P(y_{aj}^{(k)} = 1) = (\pi_{ij}^{(b)})^{\boldsymbol{Z}_{kb}\boldsymbol{T}_{ai}}.$$

Note that these probabilities sum to unity across $j$ for each $i$ and $k$. The observed annotations are thus a probabilistic (Multinomial) function of these $\pi$ matrices. The marginal probability that an annotator is of type $b$ is denoted by $\tau_b$.

The data generation process (depicted in graphical form in Figure 1) given parameters $\pi, \tau$ and $p$, is thus:

1. For each article $a$: select a true type $i$ via: $\underline{T}_{a,} \sim \text{Multinomial}(p, 1)$.

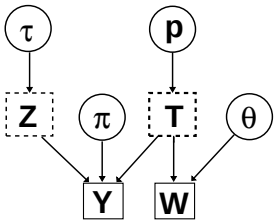


**Fig. 1** Directed Acyclic Graph denoting the data generation process. Observed data is shown in filled-line squares and unobserved data in dotted squares. Parameters of the model are shown in circles.

2. For each annotator: select a bias type $b$ via: $\underline{Z}_{k,} \sim \text{Multinomial}(\tau, 1)$.
3. For **some** $a, k$ pairs: select an annotation: $y_{a,}^{(k)} \sim \text{Multinomial}((\pi_{i,}^{(b)})^{\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}}, 1)$,

where **some** depends on the number of annotations, i.e. we partially observe $y$. Note that the annotators make their decisions independently. The likelihood of the observed data for the cluster model is now given by

$$P(y|\boldsymbol{T}, \boldsymbol{Z}) = \prod_{a=1}^{A}\prod_{i=1}^{J}\left[\prod_{k=1}^{K}\prod_{j=1}^{J}\left(\prod_{b=1}^{B}\left\{\pi_{ij}^{(b)}\right\}^{\boldsymbol{Z}_{kb}}\right)^{y_{aj}^{(k)}}\right]^{\boldsymbol{T}_{ai}}.$$

However, we do not observe $\boldsymbol{T}$ and $\boldsymbol{Z}$; these are treated as missing data and estimated as below.

The complete-data likelihood of the full annotation dataset (including unobserved true types of article and annotator) across all $A$ articles is given by

$$P(y, \boldsymbol{T}, \boldsymbol{Z}) = \prod_{a=1}^{A}\prod_{i=1}^{J}\left[p_i\prod_{k=1}^{K}\prod_{j=1}^{J}\left(\prod_{b=1}^{B}\left\{\tau_b\pi_{ij}^{(b)}\right\}^{\boldsymbol{Z}_{kb}}\right)^{y_{aj}^{(k)}}\right]^{\boldsymbol{T}_{ai}}. \quad (2)$$

where $p_i$ is the marginal probability of article type $i$ and $\tau_b$ is the marginal probability of annotator type $b$.

For choice of classifier using word terms $w$, Brew et al (2010b) examined $k$-nearest neighbours and support vector machines but settled on Naive Bayes following an assessment of the performance of the methods under cross-validation. We therefore select that classifier for our analysis here but note that any probabilistic classifier is compatible with our mixture model for bias matrices methodology.

Although word-term frequencies are available in the dataset, we model only the presence or absence of these features (word terms). Let $w_{a,} = (w_{a1}, \ldots, w_{aN})$ be a binary vector that indicates the presence of words in document $a$. We employ a Bernoulli likelihood for term $w_{a,}$ given that the article is of type $i$ (that is $\boldsymbol{T}_{ai} = 1$). We train a Naive Bayes classifier to learn the probability of an article type given the words that appear in the article. Although the

Naive Bayes assumption is unlikely to hold exactly in practice, there is much evidence to suggest that it can yield excellent classification results (Domingos and Pazzani, 1997; Hand and Yu, 2001).

The product of Bernoullis likelihood for all $N$ word terms $w_{a,}$ appearing in article $a$ given $\underline{T}_{a,}$ is then

$$P(w_{a,}|\theta, \underline{T}_{a,}) = \prod_{n}^{N} \prod_{i=1}^{J} \left\{ (\theta_{ni})^{w_{an}} (1-\theta_{ni})^{1-w_{an}} \right\}^{\boldsymbol{T}_{ai}}. \tag{3}$$

where $\theta_{ni}$ is the probability that the word term corresponding to $w_{,n}$ appears in an article of type $i$.

The complete data likelihood is the likelihood of both the observed and unobserved data and is a product of (2) and a term in the form of (3) for each article, yielding

$$\begin{aligned} P(w, y, \boldsymbol{T}, \boldsymbol{Z}) &= P(y, \boldsymbol{T}, \boldsymbol{Z}) P(w|\boldsymbol{T}) \\ &= \prod_{k=1}^{K} \prod_{b=1}^{B} \prod_{a=1}^{A} \prod_{i=1}^{J} \prod_{j=1}^{J} \left\{ \pi_{ij}^{(b)} \right\}^{y_{aj}^{(k)} \boldsymbol{T}_{ai} \boldsymbol{Z}_{kb}} \times \prod_{k=1}^{K} \prod_{b=1}^{B} \tau_b^{\boldsymbol{Z}_{kb}} \times \prod_{a=1}^{A} \prod_{i=1}^{J} p_i^{\boldsymbol{T}_{ai}} \\ &\quad \times \prod_{a=1}^{A} \prod_{i=1}^{J} \left[ \prod_{n=1}^{N} \theta_{ni}^{w_{an}} (1-\theta_{ni})^{1-w_{an}} \right]^{\boldsymbol{T}_{ai}}. \end{aligned} \tag{4}$$

Hence, the complete-data log-likelihood is

$$\begin{aligned} \ell_c &= \sum_{k=1}^{K} \sum_{b=1}^{B} \sum_{a=1}^{A} \sum_{i=1}^{J} \sum_{j=1}^{J} y_{aj}^{(k)} \boldsymbol{T}_{ai} \boldsymbol{Z}_{kb} \log(\pi_{ij}^{(b)}) \\ &\quad + \sum_{k=1}^{K} \sum_{b=1}^{B} \boldsymbol{Z}_{kb} \log(\tau_b) + \sum_{a=1}^{A} \sum_{i=1}^{J} \boldsymbol{T}_{ai} \log(p_i) \\ &\quad + \sum_{a=1}^{A} \sum_{i=1}^{J} \boldsymbol{T}_{ai} \left[ \sum_{n=1}^{N} \{ w_{an} \log(\theta_{ni}) + (1-w_{an}) \log(1-\theta_{ni}) \} \right]. \end{aligned} \tag{5}$$

In Section 4 we construct an Expectation-Maximisation (EM) algorithm to find the maximum likelihood estimates of $p$, $\tau$, $\pi$ and $\theta$. The resulting optimisation also yields the expected values of $\boldsymbol{T}_{ai}$ and $\boldsymbol{Z}_{kb}$ and these can be used to classify the articles and cluster the annotators.

## 4 Algorithm

We adopt the EM algorithm of Dempster et al (1977), a common technique in fitting finite mixture models where the cluster assignments are treated as missing data. The algorithm proceeds by iterating between an E-Step and an M-step. The E-step finds the expected value of the complete-data log-likelihood (5) given current estimated parameters ($\hat{p}, \hat{\tau}, \hat{\pi}$ and $\hat{\theta}$) and the M-step finds

updates for the parameters given the expected values $(\hat{\boldsymbol{T}}, \hat{\boldsymbol{Z}})$ of the unobserved data by maximizing the expected complete-data log-likelihood.

Following an initialisation step that assigns starting values to the model parameters $p$, $\tau$, $\pi$ and $\theta$, the algorithm then iterates the E-step and M-step until convergence is achieved. The starting values are computed based on initially assigning articles to the class which has a majority vote of the annotations.

### 4.1 E-step

The E-step of the EM algorithm requires computing the expected complete-data log-likelihood based on the data and the current parameter estimates. Given the linear nature of the complete-data log-likelihood (5) this involves replacing $\boldsymbol{T}_{ai}$, $\boldsymbol{Z}_{kb}$ and $\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}$ by their expected values.

$$\begin{aligned}\hat{\boldsymbol{T}}_{ai} = \mathbf{E}(T_{ai}|y,w) &= P(\boldsymbol{T}_{ai}=1|y,w)\\ &\propto \sum_{\boldsymbol{Z}} P(\boldsymbol{T}_{ai}=1,\boldsymbol{Z},y,w)\\ &\propto \sum_{\boldsymbol{Z}} P(y,w|\boldsymbol{T}_{ai}=1,\boldsymbol{Z})P(\boldsymbol{T}_{ai}=1)P(\boldsymbol{Z}).\end{aligned}$$

This calculation involves summing (4) over the possible values of $\boldsymbol{Z}$ which are matrices where each row contains a single one and the remaining values are zero. Thus, we find that

$$\hat{\boldsymbol{T}}_{ai} = \mathbf{E}[\boldsymbol{T}_{ai}|y,w] \propto \hat{p}_i \prod_{k=1}^{K}\prod_{j=1}^{J}\left\{\sum_{b=1}^{B}\tau_b\pi_{ij}^{(b)}\right\}^{y_{aj}^{(k)}}\prod_{n=1}^{N}\theta_{ni}^{w_{an}}(1-\theta_{ni})^{1-w_{an}}.$$

For ease of notation we now introduce $\lambda_{ai}$ as the probability article $a$ is of type $i$ based on words $w$ only (i.e. the contribution to the joint likelihood from the Naive Bayes classifier).

$$\lambda_{ai} = \frac{\hat{p}_i\prod_{n=1}^{N}\hat{\theta}_{ni}^{w_{an}}(1-\hat{\theta}_{ai})^{1-w_{an}}}{\sum_{i'=1}^{J}\hat{p}_{i'}\prod_{n=1}^{N}\hat{\theta}_{ni'}^{w_{an}}(1-\hat{\theta}_{ni'})^{1-w_{an}}}.$$

Thus, we get

$$\hat{\boldsymbol{T}}_{ai} \propto \lambda_{ai}\prod_{k=1}^{K}\prod_{j=1}^{J}\left\{\sum_{b=1}^{B}\hat{\tau}_b\hat{\pi}_{ij}^{(b)}\right\}^{y_{aj}^{(k)}}.$$

Similarly, we find that

$$\hat{\boldsymbol{Z}}_{kb} = \mathbf{E}[\boldsymbol{Z}_{kb}|y,w] \propto \hat{\tau}_b\prod_{a=1}^{A}\prod_{j=1}^{J}\left\{\sum_{i=1}^{J}\lambda_{ai}\hat{\pi}_{ij}^{(b)}\right\}^{y_{aj}^{(k)}}.$$

It is important to note that $\mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w] \neq \mathbf{E}[\boldsymbol{T}_{ai}|y,w]\mathbf{E}[\boldsymbol{Z}_{kb}|y,w]$ and we need to compute the expected value of this product for the M-step updates. We can compute $\mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w]$ by noting that $\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb} = 1$ if and only if both $\boldsymbol{T}_{ai} = 1$ and $\boldsymbol{Z}_{kb} = 1$, thus we have

$$\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb} = \mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w] = \frac{\hat{\tau}_b\lambda_{ai}\prod_{j=1}^{J}(\hat{\pi}_{ij}^{b})^{y_{aj}^{(k)}}}{\sum_{b'=1}^{B}\sum_{i'=1}^{J}\hat{\tau}_{b'}\lambda_{ai'}\prod_{j=1}^{J}(\hat{\pi}_{i'j}^{b'})^{y_{aj}^{(k)}}}.$$

It is worth noting that we need to compute and store this expected value for all $(a,i,k,b)$ and thus for $A \times J \times K \times B$ elements. This creates a computational bottleneck in our algorithm. We discuss this issue in the context of related models and propose a solution to this bottleneck in Section 4.4.

### 4.2 M-step

The estimate of the *a priori* probability of article type $p$ and bias type $\tau$ are given by

$$\hat{p}_i = \frac{\sum_{a=1}^{A}\hat{\boldsymbol{T}}_{ai}}{A}. \qquad\qquad \hat{\tau}_b = \frac{\sum_{k=1}^{K}\hat{\boldsymbol{Z}}_{kb}}{K}.$$

The estimates for the word frequencies within each article type $\theta_{ni}$ are

$$\hat{\theta}_{ni} = \frac{\sum_{a=1}^{A} w_{an}\hat{\boldsymbol{T}}_{ai}}{\sum_{a=1}^{A}\hat{\boldsymbol{T}}_{ai}}.$$

The estimate of the bias matrix terms of each type $\pi_{ij}^{(b)}$ are

$$\hat{\pi}_{ij}^{(b)} = \frac{\sum_{a=1}^{A}\sum_{k=1}^{K} y_{aj}^{(k)}\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb}}{\sum_{j=1}^{J}\sum_{a=1}^{A}\sum_{k=1}^{K} y_{aj}^{(k)}\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb}}.$$

### 4.3 Observed Data Likelihood Estimate

The EM algorithm guarantees that the value of the observed log-likelihoood ($\ell$) always increases at each EM iteration. However, the observed data log-likelihood is not directly calculable in our example. We can estimate it as:

$$\ell = \ell_c - \sum_{a=1}^{A}\sum_{i=1}^{J}\sum_{k=1}^{K}\sum_{b=1}^{B}(\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb})\log(\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb}).$$

where $\ell_c$ is the complete data log-likelihood as given in (5).

### 4.4 Approximate E-step

In Section 4.1, we noted that the calculation of $\mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w]$ for all $(a,i,k,b)$ is a computational bottleneck in the EM algorithm. We could create an iterative algorithm to approximate $\mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w]$ by using available expressions for $\mathbf{E}[\boldsymbol{T}_{ai}|\hat{\boldsymbol{Z}},y,w]$ and $\mathbf{E}[\boldsymbol{Z}_{kb}|\hat{\boldsymbol{T}},y,w]$, that is,

$$\mathbf{E}[\boldsymbol{Z}_{kb}|\hat{\boldsymbol{T}},y,w] = \frac{\hat{\tau}_b \prod_{i=1}^{J}\prod_{j=1}^{J}\left\{\hat{\pi}_{ij}^{(b)}\right\}^{\sum_{a=1}^{A} y_{aj}^{(k)}\hat{\boldsymbol{T}}_{ai}}}{\sum_{b'=1}^{B}\hat{\tau}_{b'}\prod_{i=1}^{J}\prod_{j=1}^{J}\left\{\hat{\pi}_{ij}^{(b')}\right\}^{\sum_{a=1}^{A} y_{aj}^{(k)}\hat{\boldsymbol{T}}_{ai}}}.$$

$$\mathbf{E}[\boldsymbol{T}_{ai}|\hat{\boldsymbol{Z}},y,w] = \frac{\lambda_{ai}\prod_{b=1}^{B}\prod_{j=1}^{J}\left\{\hat{\pi}_{ij}^{(b)}\right\}^{\sum_{k=1}^{K} y_{aj}^{(k)}\hat{\boldsymbol{Z}}_{kb}}}{\sum_{i'=1}^{J}\lambda_{ai'}\prod_{b=1}^{B}\prod_{j=1}^{J}\left\{\hat{\pi}_{i'j}^{(b)}\right\}^{\sum_{k=1}^{K} y_{aj}^{(k)}\hat{\boldsymbol{Z}}_{kb}}}.$$

The calculation of these two expressions can be iterated until convergence, thus yielding an approximation of $\mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w]$. This is essentially the approach advocated in Govaert and Nadif (2005) and Govaert and Nadif (2008), as outlined in Section 2. However, this would not avoid the storage issue associated with requiring this expected value for all possible $(a,i,k,b)$ and thus for $A \times J \times K \times B$ values.

Instead, we propose a simple variational approach based on the approximation $\widehat{\boldsymbol{T}\boldsymbol{Z}}_{aikb} = \mathbf{E}[\boldsymbol{T}_{ai}\boldsymbol{Z}_{kb}|y,w] \approx \mathbf{E}[\boldsymbol{T}_{ai}|y,w]\mathbf{E}[\boldsymbol{Z}_{kb}|y,w] = \hat{\boldsymbol{T}}_{ai}\hat{\boldsymbol{Z}}_{kb}$; this approximation is studied in Section 6.1 and is shown to be accurate. Note that unlike the conditional update equations above, this approximation does not require iteration until convergence. It is similar to the approximation used in Shan and Banerjee (2008), however they find that there are no closed form maximisations for the lower bound in the variational approximation for their model and so they iterate over numerical optimisation equations. Thus there is an additional round of iterations within their E-step compared to our method. This is due to the fact that they introduce a new set of factorised distributions with variational parameters whereas we simply use the existing and already computed marginal expectations.

## 5 Model Selection

Our model for a finite mixture of biased annotators requires a choice of $B$, the number of annotator bias types. The Bayesian Information Criterion (BIC) (Schwarz, 1978) is a common choice for selecting the number of clusters in a finite mixture model. This criterion may be thought of as a penalized log-likelihood or as an approximation to a Bayes Factor (Kass and Raftery, 1995). We run our EM algorithm for various values of $B$ until convergence and we calculate the BIC as:

$$BIC(B) = -2\ell(B) + (\#\text{free parameters}) \times \log(\#\text{observations}),$$

where $\ell(B)$ is the maximized likelihood for $B$ clusters.

We consider each parameter $(\tau, \pi, p, \theta)$ in turn and count the number of observations that contribute to its estimation when penalizing for the parameter. The number of free parameters associated with each of these, along with the number of observations is given by Table 2. Thus the BIC is given by:

**Table 2** Table of numbers of free parameters associated with each set of parameters.

| Name | #free parameters | #observations |
|---|---|---|
| $\tau$ | $B-1$ | $\sum_{j=1}^{J}\sum_{a=1}^{A}\sum_{k=1}^{K} y_{aj}^{(k)}$ |
| $\pi$ | $J$ | $\sum_{j=1}^{J}\sum_{b=1}^{B}\sum_{a=1}^{A}\sum_{k=1}^{K} \hat{\boldsymbol{Z}}_{kb}^{y_{aj}^{(k)}}$ |
| $p$ | $J-1$ | $A$ |
| $\theta$ | $N(J-1)$ | $A$ |

$$BIC(B) = -2\ell(B) + (B-1)\log\sum_{j=1}^{J}\sum_{a=1}^{A}\sum_{k=1}^{K} y_{aj}^{(k)} + J\sum_{j=1}^{J}\sum_{b=1}^{B}\sum_{a=1}^{A}\sum_{k=1}^{K} \hat{\boldsymbol{Z}}_{kb}^{y_{aj}^{(k)}} + (J-1)\log(A) + N(J-1)\log(A).$$

The value of $B$ with the lowest corresponding BIC is then chosen as the number of annotator clusters. We also examine the Normalized Entropy Criterion (NEC) of Celeux and Soromenho (1996):

$$NEC(B) = \frac{\ell(B) - \ell_c(B)}{\ell(B) - \ell(1)} = \frac{\sum \widehat{\boldsymbol{TZ}}\log(\widehat{\boldsymbol{TZ}})}{\ell(B) - \ell(1)},$$

where $\ell(1)$ is the observed data log-likelihood for a single cluster. Again, the value of $B$ for which this value is lowest is selected as the number of clusters. As per Biernacki et al (1999) we set $NEC(1) = 1$, so if the minimum value is not less than 1 no clustering is chosen (i.e. a single bias matrix is used). We examine this criterion along with BIC as Celeux and Soromenho (1996) note that the standard asymptotic theory that BIC relies on "does not hold in the mixture context". However, there are some theoretical results that support the use of BIC (Leroux, 1992; Keribin, 2000) and BIC has been shown to yield excellent results for choosing the number of mixture components (eg. Fraley and Raftery, 2002; McNicholas and Murphy, 2008).

## 6 Results

We report first on a simulation study to determine whether our algorithm returns results that are comparable with the known values used to generate the data. We examine the clusterings of both annotators and articles and test the performance of our model selection approach based on BIC and NEC. We then apply the method to the motivating media dataset and report the results we find.

### 6.1 Simulated Data

We assess our algorithm on a set of simulated data for which we know the ground truth $\boldsymbol{Z}$ and $\boldsymbol{T}$ values, along with the cluster specific $\pi$ bias matrices. We simulated data with $B = 5$ clusters and 6 members of each cluster so that $K = B \times 6 = 30$; we set $A = 100$ and $N = 20$. We then assessed how well our algorithm performed at recovering the correct annotator clusterings, the correct article types and the correct number of annotator types. The algorithm successfully recovered the underlying $p, \tau, \pi$ and $\theta$ parameters. A Pearson correlation test was applied to a Central Logratio Transformation (Aitchison, 1986) of the ground truth and estimated values of $\pi$. This transformation takes the parameters from the compositional space to $\mathbb{R}^3$. The correlation was found to be 0.86 with a p-value of $4.53 \times 10^{-14}$. Similarly, the correlation between ground truth and estimated $\theta$ was found to be 0.97 with a p-value less than $2.2 \times 10^{-16}$.

Now that we have established that the algorithm can recover the correct clusterings and parameter values for known $B$ we next explore the model selection criteria used to choose $B$. We simulated 100 datasets for each value of $B$ from 1 to 5. We then fit our model to each of these 500 datasets for each value 1 to 5. Thus we fit 2500 models to 500 datasets. We compute both BIC and NEC for each of these 2500 fits and compare the two model selection criteria by checking what proportion of times they select the correct number of clusters $B$. Figure 2 summarises the results as a heatmap plot of the BIC and NEC criteria. The x-axis shows the true value of $B = B_T$ used to simulate data and the y-axis shows the value of $B = B_m$ used to fit the model. The lattice is coloured in grayscale with higher values denoting a higher proportion of times the criterion choose $B_m$ when the data was simulated using $B_T$. The diagonal from bottom left to top right of each heatmap represents $B_m = B_T$, the correct model choice. We see that BIC performed markedly better at recovering the correct number of underlying annotator clusterings as there is more weight on the diagonal. However we note that BIC sometimes over-penalizes.

We now wish to explore the E-step approximation that $\widehat{\boldsymbol{TZ}}_{aikb} \simeq \hat{\boldsymbol{T}}_{ai}\hat{\boldsymbol{Z}}_{kb}$ for use in the $\pi$ updates in the E-step of the EM algorithm. Such an approximation leads to a large speed-up of the EM algorithm as it eliminates the need to index over a $A \times J \times B \times K$ array. This calculation represents the largest bottleneck in the optimisation. We ran 500 repeated simulations for randomly varying values of $B$ and $\pi$ and simulated corresponding random $\boldsymbol{T}$ and $y$. We then fit our model both with and without the approximation in the E-step (and in the log-likelihood calculation). The results are surprising in that the approximation version of the EM algorithm performed better in terms of mean squared error of $\boldsymbol{T}, \boldsymbol{Z}, \pi$ and $\theta$. The top of Table 3 summarises these results. The change in accuracy was found to be strongly statistically significant for all 4 variables.

We then plotted the measures of accuracy for the exact versus the approximate E-step update. We found that there were a small number of outliers for the exact version. These were due to convergence of the algorithm to local

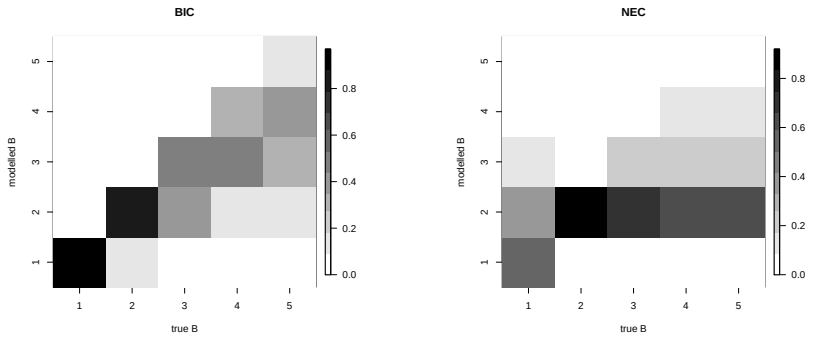


**Fig. 2** Heatmap plots of the probabilities of the model selection criteria selecting a model with $B_m$ clusters (y-axis) when simulating data using $B_T$ clusters (x-axis). 100 random datasets were simulated for each of $B_T = 1, \ldots, 5$ and the model was fit using $B_m = 1, \ldots, 5$ for each of these 500 datasets. The BIC and NEC were calculated in each case and the figure summarises the proportion of times each $B_m$ value was lowest and thus selected for each value of $B_T$. For a given cell corresponding to a unique combination of $B_T$ and $B_M$ values, the degree of shading denotes the proportion of times the model choose $B_m$ clusters for that value of $B_T$.

**Table 3** Mean values for the difference in MSE for exact E-step and Approximate E-step under two initialisation methods. The Mean Squared Error for 4 terms of interest is found under an exact and an approximate E-step. The approximation is $\widehat{\boldsymbol{TZ}}_{aikb} \simeq \hat{\boldsymbol{T}}_{ai}\hat{\boldsymbol{Z}}_{kb}$. The result for the approximation is subtracted from the result for the exact E-step under each randomly simulated dataset. All values have a positive mean showing that the MSE is larger on average for the exact E-step algorithm. We also report the p-value of a one-tailed paired t-test for these differences.

Majority Vote Initialisation

| | $\boldsymbol{T}$ | $\boldsymbol{Z}$ | $\pi$ | $\theta$ |
|---|---|---|---|---|
| MSE(exact)-MSE(approx) | 0.00312 | 0.0169 | 0.00288 | 0.000984 |
| p-value | 0.00863 | $6.96 \times 10^{-10}$ | $2.49 \times 10^{-16}$ | 0.00586 |

Ground Truth Initialisation

| | $\boldsymbol{T}$ | $\boldsymbol{Z}$ | $\pi$ | $\theta$ |
|---|---|---|---|---|
| MSE(exact)-MSE(approx) | $-8.49\times10^{-5}$ | -0.000714 | -0.000803 | $-3.81\times10^{-6}$ |
| p-value | 0.00459 | 0.0311 | $3.32\times10^{-158}$ | 0.00354 |

maxima. As is the case in any iterative optimisation our algorithm may become stuck in such local modes. It is easy to show that local modes exist in our model as label switching of both the annotator clusters and the article types may occur. Furthermore, local minima occur as annotators are moved from one cluster to another with small changes to the cluster specific parameters. Shan and Banerjee (2008) state that they must "use simulated annealing in the inference step to avoid bad local minima" (minima of the negative log-likelihood, i.e. maxima of the log-likelihood). We attempt to avoid these local maxima by supplying the algorithm with starting values as close as possible to the global maximum; specifically, we first fit the model of Salter-Townshend and Murphy

(2012) and then apply a hierarchical clustering to these maximimum likelihood estimates to obtain a hard clustering. Averages across members of each cluster provide starting values for our algorithm. This method avoids local maxima that are far from the global maximum. Repeated random restarts in theory allow us to find a higher global maximum, however we found in practice that an impracticably large number were required to beat the values obtained following the above initialisation procedure. The random initialised algorithms typically become stuck in bad local maxima far from the global maximum.

We then re-ran the above comparisons with both the exact and approximate EM algorithms initialised to the true values of all parameters. The results are shown in the lower half of Table 3. The exact E-step update now performs better in terms of mean squared error of $\boldsymbol{T}, \boldsymbol{Z}, \pi$ and $\theta$. Again, the change in accuracy was found to be strongly statistically significant for all 4 variables. As we don't know ground truth values for real data, we report results using the approximate E-step updates (i.e. we make use of the approximation $\widehat{\boldsymbol{TZ}}_{aikb} \simeq \hat{\boldsymbol{T}}_{ai}\hat{\boldsymbol{Z}}_{kb}$). This approximate E-step update is faster, less prone to becoming stuck at local maxima and requires far less memory.

### 6.2 Economic Sentiment in Media Dataset

The Irish economic sentiment in media dataset that we analyze is a subset of the data described in detail in Brew et al (2010a,b). For brevity, we will simply refer to it as the media dataset in this paper. The media dataset is comprised of $A = 1306$ articles collected from 3 online Irish news services (`rte.ie`, `irishtimes.com` and `independent.ie`), collected from July to October 2009. $K = 31$ non-expert volunteers have annotated an average of 222.9 of these articles as having either negative, positive or irrelevant impact on the Irish economy at time of press. Note that a classifier was used to pre-screen the articles to reduce the number of irrelevant articles as low as possible. Thus each article has an average of just 5.29 annotations. There are $N = 70873$ word terms appearing in these articles.

In order to reduce the impact of words that are too common we eliminate words that appear in more than $A - 50$ articles where $A$ is the total number of articles. [1] We also eliminate words that appeared in less than 30 articles. To further reduce the dimensionality of the data, we selected the top 300 most negative words (as indicated by a majority vote classifier), the top 300 most positive words and the top 300 most irrelevant words only.

To apply our methodology to the media dataset we eliminated annotators who annotated fewer than 50 articles leaving a set of 22 annotators. It takes 41 minutes to run the exact $\widehat{\boldsymbol{TZ}}_{aikb}$ calculation and 17 minutes to run using the approximation $\widehat{\boldsymbol{TZ}}_{aikb} \simeq \hat{\boldsymbol{T}}_{ai}\hat{\boldsymbol{Z}}_{kb}$ on the media dataset. We report results

[1] These words are: "said", "its", "year", "a", "s", "to", "by", "irish", "has", "and", "that", "at", "as", "an", "they", "for", "of", "are", "on", "not", "but", "last", "this", "have", "from", "was", "with", "it", "the", "in", "he", "would", "be", "will", "is", "their", "mr", "were", "had", "which", "we", "ireland", "been", "his", "2009", "per", "cent".

using this approximation due to the result from Section 6.1 that showed that it is in fact more accurate on average in the absence of ground truth starting values.

Figure 3 shows a plot of the BIC and NEC against $B$. BIC favours a 7 cluster model with 3 annotators being in a cluster on their own. A model with 2 clusters is at a local minimum of the BIC curve. In contrast, NEC selects a 5 cluster model for the media dataset.

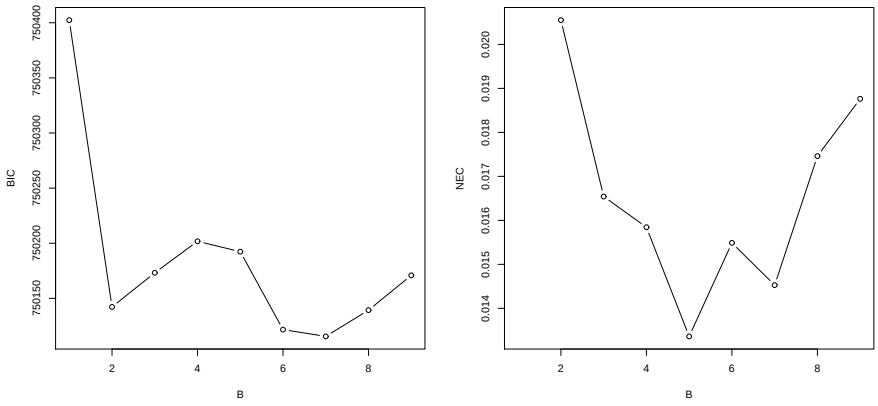


**Fig. 3** The BIC and NEC values plotted against the number of clusters for the media dataset. BIC selects a 7 cluster model whereas NEC selects a model with 5 types of bias matrix.

The marginal probability of belonging to each cluster for the 7 cluster model was found to be:

$$\hat{\tau} = \{0.136, 0.194, 0.157, 0.044, 0.378, 0.045, 0.045\}.$$

Table 4 shows the cluster specific bias matrices for the 7 cluster model. We summarise the clusters as:

1. Most accurate discrimination between positive and negative but rarest use of irrelevant label. Also most negative overall.
2. Similar to 1, but much better at discerning irrelevant articles.
3. Closest to a spammer as they are the least able to distinguish between any article types.
4. Most accurate at correctly identifying negative articles but low accuracy on irrelevant articles.
5. Most consistent and least directed bias towards one particular label. However, not particularly good at correctly identifying any particular type.
6. Most positive overall and most accurate at correctly identifying positive articles.

7. Least negative and least positive overall; this is due to a far higher probability of labelling all article types as irrelevant. As a consequence, also the most accurate at correctly identifying irrelevant articles.

It is worth noting that the two most accurate clusters at distinguishing between positive and negative articles were the worst at spotting irrelevant articles.

**Table 4** Cluster specific bias matrices $\hat{\pi}^{(b)}$ values with Jackknife standard errors in brackets for the 7 cluster model. Where columns in a matrix do not exactly sum to unity this is due to rounding to 2 decimal places. Each bias matrix has the marginal probability $\tau_b$ of occurring on the top. The last box shows the key to interpreting these conditional probabilities. e.g. *pos*|*neg* is the cluster specific probability an annotator labels a negative article as positive. For example, the first cluster has a probability of 0.65 of recording a negative article as negative, 0.31 of recording it as positive and 0.04 of recording it as irrelevant.

| | $\tau_1 = 0.136$ | | | $\tau_2 = 0.194$ | |
|---|---|---|---|---|---|
| 0.65 (0.01) | 0.31 (0.01) | 0.04 (0.01) | 0.63 (0.01) | 0.23 (0.01) | 0.14 (0.02) |
| 0.36 (0.02) | 0.62 (0.02) | 0.02 (0.01) | 0.34 (0.06) | 0.48 (0.05) | 0.18 (0.10) |
| 0.60 (0.06) | 0.32 (0.03) | 0.08 (0.04) | 0.51 (0.02) | 0.26 (0.00) | 0.23 (0.02) |
| | $\tau_3 = 0.157$ | | | $\tau_4 = 0.044$ | |
| 0.40 (0.04) | 0.29 (0.06) | 0.31 (0.02) | 0.72 (0.02) | 0.08 (0.01) | 0.20 (0.01) |
| 0.23 (0.06) | 0.40 (0.04) | 0.37 (0.03) | 0.38 (0.17) | 0.43 (0.23) | 0.19 (0.06) |
| 0.29 (0.06) | 0.20 (0.03) | 0.51 (0.03) | 0.35 (0.01) | 0.41 (0.18) | 0.24 (0.17) |
| | $\tau_5 = 0.378$ | | | $\tau_6 = 0.045$ | |
| 0.58 (0.02) | 0.16 (0.01) | 0.26 (0.01) | 0.48 (0.24) | 0.32 (0.24) | 0.19 (0.01) |
| 0.38 (0.01) | 0.42 (0.07) | 0.20 (0.07) | 0.22 (0.15) | 0.70 (0.26) | 0.07 (0.11) |
| 0.42 (0.01) | 0.18 (0.02) | 0.41 (0.02) | 0.29 (0.05) | 0.47 (0.05) | 0.24 (0.00) |
| | $\tau_7 = 0.045$ | | | | |
| 0.44 (0.01) | 0.02 (0.22) | 0.54 (0.23) | *neg*\|*neg* | *pos*\|*neg* | *irr*\|*neg* |
| 0.26 (0.12) | 0.32 (0.01) | 0.42 (0.13) | *neg*\|*pos* | *pos*\|*pos* | *irr*\|*pos* |
| 0.27 (0.01) | 0.02 (0.12) | 0.72 (0.12) | *neg*\|*irr* | *pos*\|*irr* | *irr*\|*irr* |

We used the Jackknife (Quenouille (1949), Tukey (1958); see Miller (1974) for a review) method to estimate the standard errors of the entries in the cluster specific bias matrices shown in Table 4. A single annotator was removed for each Jackknife iteration and the model refit to the remaining data. This created $K = 22$ Jackknife estimates of each $\hat{\pi}_{ij}^{(b)}$ which we denote ${}_k\hat{\pi}_{ij}^{(b)}$. The standard errors reported in Table 4 are then calculated as

$$se\left(\hat{\pi}_{ij}^{(b)}\right) = \sqrt{\frac{K-1}{K}\sum_{k=1}^{K}\left(\left({}_k\hat{\pi}_{ij}^{(b)} - \hat{\pi}_{ij}^{(b)}\right)^2\right)} \tag{6}$$

It can be seen that the standard errors are smaller for the clusters with more members. This is unsurprising as removal of a single member of a small cluster will have a larger effect on parameter estimates than removal of a member of a large cluster.

We next examine the estimates for the word-terms classifier. We re-normalize $\hat{\theta}$ so that the weights across the three types of sentiment sum to unity and

call this $\phi$. It is worth noting that 98.85% of word terms do not change maximal sentiment categorisation when moving from a model with a bias matrix specified for each annotator to a mixture model with just 7 annotator types.

Figure 4 depicts tag clouds for the three types of sentiment. Tag clouds are a popular method for summarising the sentiment carried by word terms. Random layouts of the words are created with both boldness and font size weighted according to the sentiment attributed to them. We simply pass $\phi$ to a tag cloud generator.

The results are interesting in that the negative tag cloud strongly depicts many governmental terms such as the surnames of the then current ("cowen") and most recent ("ahern") Taoiseach (Prime Minister) of Ireland during the time of data collection, along with the word "taoiseach". The top ten most negative words were: "reason", "don't", "fail", "nothing", "editor", "example", "political", "ministers", "cowen" and "reality". Note that we did not distinguish between "fail" and "Fáil" which is part of the name of the majority party in the ruling coalition government at the time of the data collection. The minority partner in this coalition were the "greens", whose name also appears.

Looking at the positive tag cloud, the top ten words are: "index", "pretax", "stocks", "rose", "analysts", "fell", "outlook", "weak", "forecast" and "bloomberg". Interestingly, many of these terms are related to movement on the stock market. Indeed, both rose and fell appear. Examination of the articles shows that both rising and falling oil prices were marked as positive by various annotators. Other terms of interest in the positive word cloud include "germany" and "oil".

The irrelevant tag cloud top ten words are: "redundancy", "premises", "relations", "dispute", "strike", "spokesperson", "court", "protest", "siptu" and "signed". SIPTU (Services, Industrial, Professional and Technical Union) is the largest trade union in Ireland. The data collection mechanism included the use of a trained classifier to pre-filter the three sources of news for articles that had high probability of carrying economic sentiment. Thus these words may be thought of as terms that relate to the economic situation but that may not carry a strong sentiment signal.

## 7 Conclusions

We have developed a finite mixture model for clustering biased sentiment annotators and demonstrated inference using an EM algorithm. Our algorithm may be used to estimate the bias-type memberships of the annotators and infer the parameters of the bias matrices. We have shown that this may be done simultaneously with training a probabilistic classifier to learn the word-term associations with sentiment. Although we have focussed on data with three levels of sentiment (negative, positive and irrelevant) our approach extends readily to arbitrary numbers of levels. Indeed, we speculate that an interesting


**Fig. 4** Negative (left), positive (middle) and irrelevant (right) tag clouds for the word-terms.

extension is to the case where we model either more or fewer levels than are reported in the annotations.

Under a series of simulation studies we found that BIC usually identified the correct number of annotator clusters but NEC did not. We found that both criteria performed well in the case of well separated clusters. We also found the surprising result that ignoring the *a posteriori* dependency of the annotator and article clusterings in the E-step updates led to an improvement in accuracy. This is similar to the variational approximation used in the literature on co-clustering but has not to our knowledge been examined in terms of classification accuracy. The result reverses when we initialise at ground truth values for parameters rather than initialising using a majority vote. We advocate using the approximation when applying the model to real data for which knowledge of the ground truth parameters is not available. In addition, the approximation is several times faster and requires far less storage in RAM.

We expect that such a model will have the potential to generate savings in the context of crowdsourcing. We have demonstrated the clustering of annotators into clusters discovered automatically by the algorithm. Such clustering allows for fewer bias-matrices to be estimated; this means more accurate estimation and more tractable reporting of results as only a small number of bias types need be interpreted. However, clusters could also be based on predefined bias-matrices to cluster the annotators into e.g. spammers, oracles, etc. This may be thought of as an extension and generalisation of Raykar and Yu (2012).

We have observed interesting cluster behaviours detected in the media dataset. BIC and NEC disagree on how many clusters there are in this data. We have reported results based on the 7 cluster model selected by BIC. The clusters correspond to differing annotator behaviour types and the advantage of clustering to a low number of bias types is that we can more readily explain and interpret our results.

## References

Aitchison J (1986) The Statistical Analysis of Compositional Data. Monographs on statistics and applied probability, Chapman and Hall, London

Biernacki C, Celeux G, Govaert G (1999) An improvement of the NEC criterion for assessing the number of clusters in a mixture model. Pattern Recognition Letters 20(3):267–272

Blitzer J, Dredze M, Pereira F (2007) Biographies, Bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In: Carroll JA, van den Bosch A, Zaenen A (eds) Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), pp 187–205

Bollen J, Mao H (2011) Twitter mood as a stock market predictor. Computer 44(10):91–94

Brew A, Greene D, Cunningham P (2010a) The interaction between supervised learning and crowdsourcing. In: NIPS Workshop on Computational Social Science and the Wisdom of Crowds

Brew A, Greene D, Cunningham P (2010b) Using crowdsourcing and active learning to track sentiment in online media. In: Coelho H, Studer R, Wooldridge M (eds) Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), IOS Press, pp 1–11

Celeux G, Soromenho G (1996) An entropy criterion for assessing the number of clusters in a mixture model. Journal of Classification 13(2):195–212

Dawid A, Skene A (1979) Maximum likelihood estimation of observer errorrates using the EM algorithm. Journal of the Royal Statistical Society Series C (Applied Statistics) 28(1):20–28

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39(1):1–38

Domingos P, Pazzani M (1997) On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29:103–130

Fraley C, Raftery AE (2002) Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association 97(458):611–631

Govaert G, Nadif M (2003) Clustering with block mixture models. Pattern Recognition 36(2):463–473

Govaert G, Nadif M (2005) An EM algorithm for the block mixture model. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(4):643–647

Govaert G, Nadif M (2008) Block clustering with Bernoulli mixture models: Comparison of different approaches. Computational Statistics & Data Analysis 52(6):3233–3245

Hand DJ, Yu K (2001) Idiot's Bayes — Not so stupid after all? International Statistical Review 69(3):385–398

Hathaway R (1986) Another interpretation of the EM algorithm for mixture distributions. Statistics & Probability Letters 4(2):53–56

Kass R, Raftery AE (1995) Bayes factors and model uncertainty. Journal of the American Statistical Association 90:773–795

Keribin C (2000) Consistent estimation of the order of mixture models. Sankhyā Series A 62(1):49–66

Leroux BG (1992) Consistent estimation of a mixing distribution. The Annals of Statistics 20:1350–1360

McLachlan G, Peel D (2000) Finite mixture models. Wiley-Interscience

McNicholas PD, Murphy TB (2008) Parsimonious Gaussian mixture models. Statistics and Computing 18(3):285–296

Miller RG (1974) The jackknife-a review. Biometrika 61(1):1–15

Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Scott D, Daelemans W, Walker MA (eds) Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL 2004), pp 271–278

Quenouille M (1949) Approximate tests of correlation in time-series. Journal of the Royal Statistical Society Series B (Methodological) 11(1):68–84

Raykar V, Yu S, Zhao L, Valadez G, Florin C, Bogoni L, Moy L (2010) Learning from crowds. Journal of Machine Learning Research 11:1297–1322

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

Salter-Townshend M, Murphy T (2012) Sentiment analysis of online media. In: Lausen B, van del Poel D, Ultsch A (eds) Algorithms from and for Nature and Life, Springer, Studies in Classification, Data Analysis, and Knowledge Organization

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6(2):461–464

Shan H, Banerjee A (2008) Bayesian co-clustering. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), IEEE, pp 530–539

Thomas M, Pang B, Lee L (2006) Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pp 327–335

Tukey JW (1958) Bias and confidence in not quite large samples. Annals of Mathematical Statistics 29(2):614