Social Network Analysis and Mining (2018) 8:2
https://doi.org/10.1007/s13278-017-0478-6

ORIGINAL ARTICLE

Learning from noisy label proportions for classifying online social data

Ehsan Mohammady Ardehaly[1] · **Aron Culotta**[1]

Received: 7 March 2017 / Revised: 19 September 2017 / Accepted: 7 November 2017 / Published online: 27 November 2017

Abstract
Inferring latent attributes (e.g., demographics) of social media users is important to improve the accuracy and validity of social media analysis methods. While most existing approaches use either heuristics or supervised classification, recent work has shown that accurate classification models can be trained using supervision from population statistics. These learning with label proportion (LLP) models are fit on bags of instances and then applied to individual accounts. However, it is well known that many social media sites such as Twitter are not a representative sample of the population; thus, there are many sources of noise in these label proportions (e.g., sampling bias). This can in turn degrade the quality of the resulting model. In this paper, we investigate classification algorithms that use population statistical constraints such as demographics, names, and social network followers to fit classifiers to predict individual user attributes. We propose LLP methods that explicitly model the noise inherent in these label proportions. On several real and synthetic datasets, we find that combining these enhancements together can significantly reduce averaged classification error by 7%, resulting in methods that are robust to noise in the provided label proportions.

Keywords Social networks · Text classification · Machine learning

1 Introduction

As social media data are increasingly used to make inferences about the real world, it is important to be able to identify latent attributes of social media users, such as demographics. Doing so will enable more accurate and generalizable inferences to be made in domains such as public health, politics, and marketing (O'Connor et al. 2010; Dredze 2012; Gopinath et al. 2014; Diaz et al. 2016).

Latent attribute inference is often framed as a supervised classification problem, requiring the collection and annotation of many users with the desired classification labels to serve as training data. However, collecting these data at scale is expensive; furthermore, many variables of interest may be difficult or impossible to annotate by manual inspection—e.g., it is rarely possible to guess someone's income level by viewing their Twitter profile. As a result, several recent methods have been proposed that build classifiers using learning from label proportion (LLP) (Schapire et al. 2002; Jin and Liu 2005; Chang et al. 2007; Graca et al. 2007; Quadrianto et al. 2009; Mann and McCallum 2010; Ganchev et al. 2010).

In the LLP setting, training data take the form $\{(X_j, \tilde{p}_j)\}$, where $X_j \in \mathbb{R}^{n_i \times d}$ is a *bag* of n_i feature vectors, and $\tilde{p}_j \in \mathbb{R}^k$ is a distribution over the k class labels in that bag. Thus, rather than requiring labels for individual training instances, LLP only requires a *distribution* over class labels for a bag of instances.

LLP models are very attractive for this task because there are many easily accessible population statistics that can be associated with a set of social media users. For example, Oktay et al. (2014) create bags based on each user's first name and match them with data from the US Social Security Administration to generate bags annotated with age distributions. Similarly, Ardehaly and Culotta (2015) match geotagged Twitter users with US Census statistics by county to fit classifiers for ethnicity.

This prior work in LLP models assumes that the label proportions used for training are accurate. However, this assumption is rarely true in practice. For example, it is generally accepted that social media users are not a representative sample of the population (Watkins 2009). Diaz et al. (2016) investigate such biases in detail, finding not only

✉ Aron Culotta
aculotta@iit.edu

Ehsan Mohammady Ardehaly
emohamm1@hawk.iit.edu

[1] Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA

considerable demographic skew, but also identifying other sources of noise stemming from the fact that user participation on social media changes over time and can shift drastically around important events.

To address this gap, we empirically investigate the effect of label noise in LLP models for social media analysis, and we propose several LLP models that are designed to be robust in the presence of such noise. Our proposed methods build upon binary label regularization (**LR**) (Mann and McCallum 2007), a state-of-the-art LLP model that generalizes logistic regression to the LLP setting. We investigate three enhancements to LR to improve robustness to noise:

1. **Bag bias** We introduce additional free parameters, one per bag, to directly model the noise introduced by each bag; to deal with the noise in the label proportion of bag i, we add a bag bias term (b_i) as a new parameter to the hypothesis.
2. **Correcting label proportions** Our second enhancement aims to correct the bag proportions in the training data using an intermediate model fit on the noisy data. The final model is fit using these adjusted label proportions. The main idea of this approach is to adjust the prior knowledge about the label proportions ($\tilde{y}$) by the inferred posterior label proportions ($\hat{y}$). That is, we fit a model on portions of the noisy training data and then use it to adjust the label proportions on the remainder of the training data.
3. **Matrix factorization** Finally, we investigate matrix factorization methods to identify hidden factors in each bag, which can help reduce overfitting to the noise in the data. This is particularly helpful in our domain, since the feature representation has high dimension (vocabulary size), but is very sparse. (Most examples use only a few features.) Inspired by matrix factorization in recommendation systems, this model identifies common hidden concepts between users and bags.

We conduct experiments to predict age, political preference, and sentiment on five different datasets of Twitter and movie reviews. We consider both natural settings, where the true label noise is unknown, and synthetic settings, where we can directly control the type and amount of noise in the label proportions. Overall, we find that each enhancement described above reduces the classification error rate and that the enhancements appear to be complementary. Averaging results across all tasks, we find that the bag bias enhancement reduces error rate by 2% over the baseline, correcting label proportions reduces error rate by 4%, matrix factorization reduces error rate by 4%, and combining all three enhancements together reduces error by 7% (on average 1.7% absolute improvement in accuracy over the LR baseline).

2 Related work

There is an active body of research that investigates classification models to predict attributes of social media users such as age (Schler et al. 2006; Rosenthal and McKeown 2011; Nguyen et al. 2011; Al Zamal et al. 2012), gender (Rao et al. 2010; Burger et al. 2011; Liu and Ruths 2013), race/ethnicity (Pennacchiotti and Popescu 2011; Rao et al. 2011), personality (Argamon et al. 2005; Schwartz et al. 2013a), political affiliation (Conover et al. 2011; Barberá 2013; Volkova and Van Durme 2015), and occupation (Preotiuc-Pietro et al. 2015). The majority of these approaches rely on hand-annotated training data. More recently, there is growing interest in training such models from label proportions. For example, a number of methods consider name statistics to infer ethnicity and age of social media users (Chang et al. 2010; Oktay et al. 2014; Knowles et al. 2016). Additionally, in prior work we have fit LLP models to predict Twitter user demographics using US Census statistics (Ardehaly and Culotta 2015) as well as using web traffic statistics (Culotta et al. 2016).

The LLP setting is a special case of *lightly supervised learning*, which has been well studied in previous work. Mann and McCallum (2010) introduce semi-supervised method for generalized expectation criteria with weakly labeled data, and Jin and Liu (2005) develop a discriminative framework for learning with class priors. Chang et al. (2012) exploit several kinds of task-specific constraints in semi-supervised algorithms. Quadrianto et al. (2009) propose a model to predict labels of testing set with known label proportions, which has applications in e-commerce, politics, spam filtering, and improper content detection. Wang et al. (2012) work on learning with target prior and propose a probabilistic framework for it. Other researchers propose regression models (Musicant et al. 2007) and Bayesian models (Ganchev et al. 2010; Zhu et al. 2014) for learning from label proportions.

To the best of our knowledge, none of this prior work directly addresses how to handle the noise in label proportions when training LLP models. However, there are numerous works investigating models that are robust to label noise in traditional supervised classification. The majority of these works are either based on eliminating outliers (Fischler and Bolles 1981; Brodley and Friedl 1999; She and Owen 2011) or using shift parameters (She and Owen 2011; Tibshirani and Manning 2014).

Random sample consensus (RANSAC) is an iterative algorithm that tries to find outliers by identifying samples with the highest residual error (Fischler and Bolles 1981). The model iteratively trains a regressor on randomly samples of the training set and estimates the error of the remaining of the training set. By repeating these tasks and scoring the average error of the training data, the algorithm identifies outliers and eliminates

them. We use this method as a baseline; however, our experimental results show that this model does not perform well in LLP settings. This approach extends the classification task of prior work (Brodley and Friedl 1999) by identifying mislabeled training instances. In addition, by adding shift parameters and applying L1 regularization on them, it is possible to identify outliers (She and Owen 2011). The shift parameter is very similar to bag bias in our first enhancement. However, we do not want to remove outliers and reduce our training set. Instead, we propose an enhancement to correct label proportions and retrain a model with estimated label proportions.

The primary contributions of this paper, then, are to (1) empirically investigate how label proportion noise affects LLP models; (2) develop several LLP methods that are robust to such noise; and (3) evaluate these models on several tasks in social media analysis.

3 LLP models

In this section, we first formalize the LLP setting for the task of latent attribute inference and describe two baseline LLP models. Next, we propose three enhancements to these baselines to make them more robust to noise in the label proportions.

3.1 Baselines

Let T be the set of all users and i be a bag. Let T_i be the set of all users who belong to bag i (e.g., the set of users from the same county). Let $X_{u,i} \in \mathbb{R}^d$ be the feature vector for user u in bag i, where we have d features. We do not have the true label for each user, but we want to estimate it as our hypothesis. So, let $h_{u,i} = h(X_{u,i}; \Theta)$ be the label likelihood of the first class for $X_{u,i}$ based on hypothesis function h and model parameters Θ. The function $h_{u,i}$ can be interpreted as the posterior probability of the user. (However, in the linear model baseline below, it is not guaranteed to be between 0 and 1).

Also, let $Z_i \in \mathbb{R}^d$ be the mean of feature vectors for all users in bag i, and $\bar{h}_i$ be the average of hypothesis for all users in the bag i:

$$Z_i = \frac{1}{|T_i|} \sum_{u \in T_i} X_{u,i} \tag{1a}$$

$$\bar{h}_i = \frac{1}{|T_i|} \sum_{u \in T_i} h_{u,i} \tag{1b}$$

Let $\tilde{y}_i$ be the provided bag proportions (prior). Our hypothesis is that $\bar{h}_i$ and $\tilde{y}_i$ are close together. The aim of learning from label proportion (LLP) is to find parameters Θ such that the total error between $\bar{h}_i$ and $\tilde{y}_i$ for all bags is minimized. To control overfitting, we add an L2 regularization term with λ as the regularization strength, and we define the cost function as:

$$J(\Theta) = \sum_i E(\bar{h}_i, \tilde{y}_i) + \frac{\lambda}{2}||\Theta||^2 \tag{2}$$

where E is an error function. As a result, we need to select a hypothesis function h and an error function E to build the cost function $J(\Theta)$. Finally, gradient descent can be used to find model parameters Θ that minimize the error function. In the testing phase, the model parameters Θ are used to infer the label of unlabeled users.

3.1.1 Linear baseline (Ridge)

Even though the linear hypothesis is not often used for classification, in this section, we propose a simple linear model for LLP as a baseline and as a starting point for subsequent models. The main motivation of using a linear model is that in our previous work we observed that it sometime performs better than the logistic function (Ardehaly and Culotta 2016). Our experimental results also confirm that ridge model performs better than label regularization for one of our datasets (**Politicians**), and it has comparative accuracy for **Politic-fol** dataset. First, we need to define a hypothesis and an error function. The natural candidate for the linear model is the linear transformation and the residual squared error function; e.g.,

$$h(X_{u,i}) = X_{u,i}^T \theta \tag{3a}$$

$$E\left(\bar{h}_i, \tilde{y}_i\right) = \frac{1}{2}\left(\bar{h}_i - \tilde{y}_i\right)^2 \tag{3b}$$

Now, we can expand the cost function (Eq. 2) for the linear model using definitions in Eq. 1. Also, we can take advantage of the fact that a linear combination of linear functions is still a linear, i.e.,

$$\begin{aligned} J(\theta) &= \sum_i E\left(\bar{h}_i, \tilde{y}_i\right) + \frac{\lambda}{2}||\theta||^2 \\ &= \frac{1}{2}\sum_i \left(\bar{h}_i - \tilde{y}_i\right)^2 + \frac{\lambda}{2}||\theta||^2 \\ &= \frac{1}{2}\sum_i \left(\frac{1}{|T_i|}\sum_{u \in T_i} h_{u,i} - \tilde{y}_i\right)^2 + \frac{\lambda}{2}||\theta||^2 \\ &= \frac{1}{2}\sum_i \left(\frac{1}{|T_i|}\sum_{u \in T_i} X_{u,i}^T\theta - \tilde{y}_i\right)^2 + \frac{\lambda}{2}||\theta||^2 \\ &= \frac{1}{2}\sum_i \left(\left(\frac{1}{|T_i|}\sum_{u \in T_i} X_{u,i}^T\right)\theta - \tilde{y}_i\right)^2 + \frac{\lambda}{2}||\theta||^2 \\ &= \frac{1}{2}\sum_i \left(Z_i^T\theta - \tilde{y}_i\right)^2 + \frac{\lambda}{2}||\theta||^2 \end{aligned} \tag{4}$$

The cost function according to Eq. 4 is known as ridge regression, and by using the linear hypothesis, the problem reduces to the ridge regression over the average of features (Z_i) for each bag.

As a result, in the training step, we train ridge regression for the first class and compute θ. In the testing step, for unlabeled sample with feature vector x, we use the linear hypothesis for the user and classify it as the first class if it is higher than .5, i.e.,

$$\hat{y} = \mathbb{1}\left[x^T\theta \geq .5\right] \tag{5}$$

The main advantage of the linear model is that we do not need individual samples in training step; we only need the average feature vector for each bag. This significantly speeds up the training time—rather than classifying each user for each step of gradient descent, we only need to compute the estimated label proportions for each bag. However, as we will see in experimental results, the lack of information of individual sample reduces the robustness of the linear model to the noise in bags.

3.1.2 Label regularization (LR)

Label regularization, first introduced as a type of expectation regularization by Mann and McCallum (2007), is a semi-supervised learning model that trains on a combination of a small set of labeled data and one bag of unlabeled data with known prior label proportions. The model uses the softmax hypothesis (i.e., multivariate logistic regression), and the cost function is the KL-divergence between the prior and the estimated posterior label proportions of unlabeled data. In our prior work (Ardehaly and Culotta 2015), we applied label regularization to latent attribute inference in social media by (1) removing the requirement of user-annotated training instances, and (2) allowing multiple, overlapping bags. In this section, we first summarize label regularization for multivariate LLP. Given model parameters $\{\theta_{y_1} \ldots \theta_{y_k}\}$ for each class (y), the hypothesis function is defined as follows:

$$h_{u,i}^{(y)} = \frac{\exp\left(X_{u,i}^T\theta_y\right)}{\sum_{y'}\exp\left(X_{u,i}^T\theta_{y'}\right)}$$

To find the model parameters, the error function is defined as the KL-divergence between $\tilde{y}_i$ and $\bar{h}_i$:

$$= -\sum_y \tilde{y}_i^{(y)}\log\tilde{y}_i^{(y)} + \sum_y \tilde{y}_i^{(y)}\log\tilde{y}_i^{(y)}$$
$$= H\left(\tilde{y}_i, \bar{h}_i\right) - H\left(\tilde{y}_i\right)$$

where $H\left(\tilde{y}_i\right)$ is the entropy of the prior and is a constant, and $H\left(\tilde{y}_i, \bar{h}_i\right)$ is the cross-entropy of the posterior and the prior.

Since two of our enhancements require binary classification (matrix factorization and correcting label proportions), we modify multivariate label regularization to binary label regularization by changing the hypothesis function from the softmax to the sigmoid function same as binary logistic regression; i.e.,

$$h_{u,i} = h(X_{u,i}) = \sigma\left(X_{u,i}^T\theta\right) \tag{6}$$

where σ is the logistic (sigmoid) function. Following Mann and McCallum, we define the error function as the KL-divergence between $\tilde{y}_i$ and $\bar{h}_i$:

$$D\left(\tilde{y}_i||\bar{h}_i\right) = H\left(\tilde{y}_i, \bar{h}_i\right) - H\left(\tilde{y}_i\right) \tag{7}$$

As a result, by removing the constant ($H(\tilde{y}_i)$), we define the error function as follows:

$$E\left(\bar{h}_i, \tilde{y}_i\right) = -\sum_i \left(\tilde{y}_i\log\bar{h}_i + \left(1-\tilde{y}_i\right)\log(1-\bar{h}_i)\right) \tag{8}$$

Therefore, by adding L2 regularization, the cost function is:

$$\begin{aligned} J(\theta) &= \sum_i E\left(\bar{h}_i, \tilde{y}_i\right) + \frac{\lambda}{2}||\theta||^2 \\ &= -\sum_i \left(\tilde{y}_i\log\bar{h}_i + (1-\tilde{y}_i)\log(1-\bar{h}_i)\right) + \frac{\lambda}{2}||\theta||^2 \end{aligned} \tag{9}$$

To minimize the error function, we need to compute the gradient of the cost function. We first define:

$$e_{u,i} = \frac{h_{u,i}(1-h_{u,i})\left(\bar{h}_i - \tilde{y}_i\right)}{\bar{h}_i(1-\bar{h}_i)|T_i|} \tag{10}$$

"Appendix" provides the detail of the partial derivative computation, leading to:

$$\frac{\partial}{\partial\theta}J(\Theta) = \sum_{u,i} e_{u,i}X_{u,i} + \lambda\theta \tag{11}$$

In the testing step, for the unlabeled sample with feature vector x, we can estimate the probability of the first class ($y = 1$) as the hypothesis function and infer the first class if this probability is greater than .5, i.e.,

$$P(y = 1|x) = \sigma\left(x^T\theta\right) \tag{12}$$

3.2 Robust LLP models

Below, we describe three enhancements to the baselines above to enable robust learning with LLP. While these enhancements can be applied in any order, in this section, we stack them by increasing order of the enhancement complexity.

We present and validate our approaches assuming binary classification. However, similar to support vector machines,

a one-against-all (OAA) method can be used to extend them to multiclass settings (Vapnik 1995).

3.2.1 Bag bias (LRB)

In this first enhancement, to deal with the noise in the label proportion of bag i, we add a bag bias term (b_i) as a new parameter to the hypothesis, without any modification in the error function. Thus, for k bags, there are k additional model parameters. This approach is inspired by Tibshirani and Manning (2014), who use a similar approach to make logistic regression robust to label noise in traditional supervised classification. The intuition is that these additional "shift parameters" allow the model to shift its prediction separately for each bag in proportion to the amount of noise in its assigned label proportion.

Thus, the hypothesis function is:

$$h_{u,i} = h(X_{u,i}) = \sigma\left(X_{u,i}^T\theta + b_i\right) \tag{13}$$

The cost function is the same as Eq. 9, and the gradient of θ is the same as Eq. 11. The gradient of the new coefficients b_i is:

$$\frac{\partial}{\partial b_i}J(\Theta) = \sum_{u\in T_i} e_{u,i} + \lambda b_i \tag{14}$$

At prediction time, because we do not have the bag that user belongs to, we use the same inference as for the **LR** model.

Note that Tibshirani and Manning (2014) place an L1 penalty on this shift parameter in logistic regression because they assume that most instances are labeled correctly (and thus that most shift parameters should be 0). In our setting, this L1 penalty is not appropriate, as we assume that most if not all of our label proportions are incorrect (supported by experiments below; c.f. Figs. 7 and 8). We refer to this model as **LRB**.

3.2.2 Matrix factorization (LRBF)

Matrix factorization is frequently used in recommendation systems due to its ability to identify latent factors from sparse data (Zhang et al. 2006; Takacs et al. 2008; Salakhutdinov and Mnih 2008; Rendle and Schmidt-Thieme 2008). By analogy to our setting, the user activity can be interpreted as the rating of the latent attribute. Identifying lower-dimensional representations can help make the model more robust to label error by reducing the tendency to overfit label noise. This is particularly helpful in our domain, since the feature representation has high dimension (vocabulary size), but is very sparse. (Most examples use only a few features.) Inspired by kernel matrix factorization (Rendle and Schmidt-Thieme 2008), local collective embedding model (Saveski and Mantrach 2014), and content-based matrix factorization model (Lin et al. 2014), our model identifies common hidden concepts between users and bags. Suppose we have H hidden concepts. The hypothesis is defined as:

$$h_{u,i} = h(X_{u,i}) = \sigma\left(X_{u,i}^T\theta + b_i + X_{u,i}^T F M_i\right) \tag{15}$$

where M_i, a vector of size H, is the hidden factor for bag i, and F is a $d \times H$ matrix (d is the number of features) that maps a user feature vector to a user hidden factor vector, and σ is the logistic function (sigmoid). In terms of recommendation systems, $X_{u,i}^T\theta$ is the user bias, b_i is the item bias, $X_{u,i}^T F$ is the user hidden factor vector, and M_i is the item hidden concept for bag i in recommendation systems (Lin et al. 2014).

Again, we use the same cost function as in Eq. 9, and the gradient of θ and b_i are the same as in the LRB model. The gradient of the new parameters is:

$$\begin{aligned}\frac{\partial}{\partial M_i}J(\Theta) &= \sum_{u\in T_i} e_{u,i}X_{u,i}^T F + \lambda M_i \\ \frac{\partial}{\partial F}J(\Theta) &= \sum_i X_i^T e_i M_i^T + \lambda F\end{aligned} \tag{16}$$

where for bag i, the X_i is the term-to-user matrix. e_i is the vector with size T_i, and each row of it is $e_{u,i}$ (for all users in bag i). The testing phase is same as the **LR** model, and we only use θ to infer the class label.

We refer to the label regularization model that combines both bag bias and matrix factorization as **LRBF**.

3.2.3 Correcting label proportions (LRBFC)

The main idea of this enhancement is to adjust the prior knowledge about the label proportions ($\tilde{y}$) by the inferred posterior label proportions ($\hat{y}$). That is, we fit a model on portions of the noisy training data and then use it to adjust the label proportions on the remainder of the training data. This is inspired by prior work that attempts to identify label errors in supervised classification (Fischler and Bolles 1981; Brodley and Friedl 1999); however, whereas prior work eliminates suspicious instances from the training data, we instead attempt to correct the label proportions, maintaining all the original data.

In this model, we first use tenfold cross-validation on the training set. In each fold, 90% of bags are used to train an LLP classifier. Then, for each bag i in the remaining 10% of bags, we compute the posterior probability of label proportion ($\hat{y}_i$) as the percentage of samples in bag i that are predicted as the first class:

$$\hat{y}_i = \frac{1}{|T_i|}\sum_{u\in T_i} \mathbb{1}\left[\sigma\left(X_{u,i}^T\theta\right) \geq .5\right] \tag{17}$$

After the posterior ($\hat{y}$) is computed for all bags, we adjust the prior. To guarantee that the adjusted prior still belongs to [0, 1], we use a nonlinear weighted average between the prior and the posterior:

$$\tilde{y} \Leftarrow \sigma((1-\gamma)\sigma^{-1}(\tilde{y}) + \gamma\sigma^{-1}(\hat{y})) \tag{18}$$

where σ^{-1} is the logit function (the inverse of the logistic function); i.e.,

$$\sigma^{-1}(p) = logit(p) = \log(p) - \log(1-p) \tag{19}$$

Equation 18 first maps the prior and the posterior to $\mathbb{R}$ with the logit function, then computes the weighted average of them (with a γ factor), and finally maps it back to [0, 1] with the sigmoid function. The hyperparameter γ controls the weighted average strength and can be:

1. *Zero* The prior is not changed (no adjustment).
2. *Small positive* The prior is adjusted toward the posterior.
3. *Small negative* The prior is adjusted away from the posterior.

While using a positive value seems intuitive, the reason that we sometimes need a negative is to neutralize the impact of noise in label proportions. We refer to the label regularization model that combines bag bias, matrix factorization, and corrected label proportions as **LRBFC**.

3.3 Implementation of models

To find model parameters that minimize the cost function, we use the limited memory BFGS (L-BFGS) algorithm (Byrd et al. 1995). Although our cost function is not assured to be convex, the experimental results show L-BFGS works well, as in prior work on label regularization (Mann and McCallum 2010).

Also, to avoid overfitting, we use early stopping regularization, which is extensively used in artificial neural networks and deep learning (Yao et al. 2007; Prechelt 2012; Zhang and Yu 2005). To control early stopping, we use the maximum number of iterations as a hyperparameter.

3.4 Other baseline models

For comparison, we compare our models with these additional baselines:

1. **LRBC** This model is created by adding a label proportion correction step to the **LRB** model. As a result, this model is the **LRBFC** without matrix factorization step. This allows us to isolate the impact of matrix factorization.
2. **RidgeC** This linear model simply adds the label proportion correction step to ridge regression. So, the posterior is computed with the linear hypothesis:

$$\hat{y}_i = \frac{1}{|T_i|} \sum_{u \in T_i} \mathbb{1}\left[X_{u,i}^T \theta > .5\right] \tag{20}$$

3. Random sample consensus (**RANSAC**) An iterative algorithm tries to find outliers by identifying samples with the highest residual error (Fischler and Bolles 1981). In our experiments, we use RANSAC with ridge regression to detect and remove outlier bags, and refit the ridge regression on the remaining bags.

4 Datasets overview

We use two types of datasets in our experiments. The first one comes from Twitter with natural bags created by external sources such as US Census, Social Security Administration, and Quantcast Corporation.[1] For the Twitter data, we consider two tasks: predicting the age and political affiliation of each user.

In the second set of experiments, we consider the task of sentiment classification from both IMDB reviews and tweets. In order to more directly measure accuracy as the type and amount of noise variation, we construct synthetic bags from the data and compare models as we change the quality of the label proportions.

We perform a minimum preprocessing step to extract unigrams, maintain hashtags and follower information, and create a binary document to term matrix.

4.1 Twitter unlabeled dataset

This dataset is collected by the Twitter Streaming API in July 2014 and contains geolocated tweets from the entire USA. To assign the county for each user, the US Census' center of population data for 2010[2] is used. Then we use the *k–d* tree data structure (Maneewongvatana and Mount 2002) to find the nearest center of population to each user and assign its county to the user. This dataset contains 18M geolocated tweets from 2.7 million unique users. We refer to this as the **County-2014** data.

4.2 Twitter labeled datasets

We use labeled data for the **validation/tuning** and **testing** set. We collect three datasets:

[1] http://www.quantcast.com/.

[2] http://www.census.gov/geo/reference/centersofpop.html.

Table 1 Hashtag or Twitter for political follower constraints

Party	Hashtag or Twitter accounts
Democrats	thedemocrats, wegoted, dccc, collegedems, dennis_kucinich, sensanders, repjohnlewis, keithellison, #p2
Republicans	gop, nrsc, the_rga, repronpaul, senrandpaul, senmikelee, repjustinamash, gopleader, #tcot

1. **Politician** Inspired by prior work, we selected official Twitter accounts of several members of US Congress (Cohen and Ruths 2013). We download the most recent 200 tweets for 188 Republican accounts and 189 Democratic accounts. The task is to annotate their political party (Republican or Democrat) based on these tweets.
2. **Politic-followers** Because the **politician** dataset is not a representative sample of all users (politicians talk more about politics than non-politicians), we also collect a separate dataset by identifying a sample of users who follow the official party Twitter account (i.e., "thedemocrats" or "gop," but not both). We randomly select 632 likely Republicans and 598 likely Democrats and download their last 200 activities.
3. **Age** Obtaining Twitter users annotated by age is difficult because it is rarely explicitly mentioned. Thus, inspired by Al Zamal et al. (2012), we use the Twitter search API to collect tweets with phrases like "happy 20th birthday to me." Also, we divide users to below 25 and above 25 years old similar previous works (Rao et al. 2010; Al Zamal et al. 2012). In this way, we collect 1436 users (771 below 25 and 665 above 25), and we download their latest 200 tweets. The task is to classify each user as younger than 25 or older than 25.

4.3 Natural bag constraints

To fit LLP models, we must associate population-level soft constraints with bags of users from the **County-2014** unlabeled data. We use three types of constraints, and for each constraint, we create a bag.

4.3.1 County constraints

This constraint is inspired by previous work using geolocated Twitter activities with population-level statistics to predict the zip-code distribution of demographic attributes of users (Eisenstein et al. 2011) and predicting county health statistics from the Twitter (Schwartz et al. 2013b). The county constraint idea comes from the fact that there are available aggregate-level data for each US county. These data can be found on US Census most recent report (2010). For example, the US Census produces estimates of the age demographics per county. We assume that the population statistics of tweets from a county correlate to US Census estimates. For the political sentiment attribute, we use the 2012 presidential election results. (We did not use the 2016 election, since the unlabeled data are from 2014.) As there are more than 3000 counties in the USA, we select approximately 500 counties with more than 1000 users in our dataset for consideration as bags.

The sample used to compute census statistics undoubtedly differs in systematic ways from the sample of Twitter users identified for each county. This difference motivates the noise robust approaches in this paper.

4.3.2 Name constraints

Previous studies show that a person's first name can be used to infer age (Silver and McCanc 2014; Oktay et al. 2014), based on the fact that some names are more popular than others at different points in history. We build on this idea to create the name constraints. Our aggregate-level data come from the Social Security Administrative (SSA) report. This report indicates the frequency of first names given to babies born in each year.[3] Also, the actuarial table estimates the lifespan of babies born in each year.[4] Combining these data sources, we can associate a distribution over ages for each first name.

However, not all Twitter users reveal their names, and many users use nicknames. In this study, we assume the first term in the name field (if available) as the first name and match it with the list of baby names from SSA. After filtering rare names, we create roughly 175 bags (from 50K total names) that have enough samples in the **County-2014** dataset. For example, there are around 1600 Twitter users in **County-2014** dataset with first name "Katherine," and according to SSA, 86% of people with this name are under 25 years old.

4.3.3 Follower constraints

This constraint type is based on social media activities. The main idea is to create bags according to whether a user tweets a particular hashtag or follows a particular Twitter account. For political sentiment classification, we manually identify 18 hashtags or Twitter accounts based on Table 1 that are strongly associated with political affiliations, and we

[3] http://www.ssa.gov/oact/babynames/.

[4] http://www.ssa.gov/oact/NOTES/as120/LifeTables_Tbl_7.html.

set the label proportions to 90% Republican or 90% Democratic for these bags. For **Politic-followers** dataset, we omit Twitter accounts that are used to construct labeled data (i.e., "thedemocrats" and "gop").

For age prediction, we use population data for 1000 Twitter accounts matched with statistics from Quantcast Corporation.[5] Quantcast Corporation is an audience measurement company that tracks the demographics of users to million of websites (Kamerer 2013). For example, according to their data, 22% of web users who visited "Oprah.com" are younger than 25. As a result, we create a bag for users in **County-2014** dataset who follow "Oprah," and we expect 22% of them are younger than 25. (Quantcast was also used in Culotta et al. (2016), though within a simple linear LLP model.)

4.4 Synthetic bag constraints

We use two datasets to create synthetic bags, and both of the datasets contain labeled textual instances for the sentiment analysis classification:

1. **IMDB** This dataset provides highly polar movie reviews for 25,000 (12,500 positives and negatives) reviews in the training set and 25,000 (12,500 positives and negatives) reviews in the testing set (Maas et al. 2011).
2. **Replab** This dataset consists of highly polar and neutral tweets referring to a set of 61 topics from different domains: universities, banking, automotive, and music/artists (Amigó et al. 2013). This dataset can be used for sentiment analysis, with the advantage of the existence of neutral samples. The training set has 38K positives, 10K negatives, and 18K neutrals. We remove neutrals from the testing set, and 5K positives and 5K negatives tweets remain.[6]

To understand the behavior of noisy constraints, we synthetically create bags and inject noise to compare the robustness of different models. In the first step, we need to create bags. In our experiments, we create 200 bags by sampling from the training set. We use two types of constraints to create bags:

1. **Random** bags We randomly select 100 samples from the training set such that p percent of them belongs to a specified class.
2. **Chi2** bags χ^2 tests are often used for feature selection in machine learning to identify predictive features (Rogati and Yang 2002; Yang and Pedersen 1997). To better reflect the linguistic cohesion of natural bag constraints, we run χ^2 test on the training set and select features with the highest χ^2 score. Then, we sample 100 samples from the training set that contains the selected term to create each bag.

In all experiments, half of the bags are **random** bags with $p = .70$ i.e., we have:

1. 100 **chi2** bags, each with 100 instances.
2. 50 bags with 70 instances from the first class and 30 samples from the other class.
3. 50 bags with 30 samples from the first class and 70 instances from the second class.Since we know the true label for each instance in this data, the bags initially will have no noise in the label proportions. To simulate noise in label proportions, we use two algorithms to create synthetic noise. We want to explore how algorithms behave as both the type and amount of noise varies. For both algorithms, we sample a random variable n from the normal distribution, i.e., $n \sim \mathcal{N}(\mu, \sigma^2)$. Let $\tilde{y} = \langle \tilde{y}_1, \tilde{y}_2 \rangle$ be the prior label proportions. We define two types of noise:

1. **Sum-noise** We add the noise to the first class proportion and renormalize it:

$$\tilde{y} \leftarrow \frac{1}{Z}\langle \tilde{y}_1 + n, \tilde{y}_2 \rangle \tag{21}$$

where Z is the normalization factor (i.e., $1 + n$). We also set negative values to zero.
2. **Log-noise** We multiply n by the first class proportion and renormalize it:

$$\tilde{y} \leftarrow \frac{1}{Z}(n\tilde{y}_1, \tilde{y}_2) \tag{22}$$

We say **Sum-noise** is **centered** if μ is zero, and **Log-noise** is **centered** when μ is one. Otherwise, we say that the noise is **uncentered**. Also, we define **scale** as the variance of normal distribution (σ^2).

4.5 Experimental settings

To reduce the effect of random variables, in all our experiments we use five different random seeds to:

- Select 20% of the testing set as the **validation** and remaining 80% as the **testing** set.
- Sample instances to generate bags.
- Initialize model parameters for gradient descent.
- Add noise to label proportions (optional).

[5] http://www.quantcast.com/measure/.

[6] We substitute training and testing sets of the original dataset because the training set had lower instances than testing set.

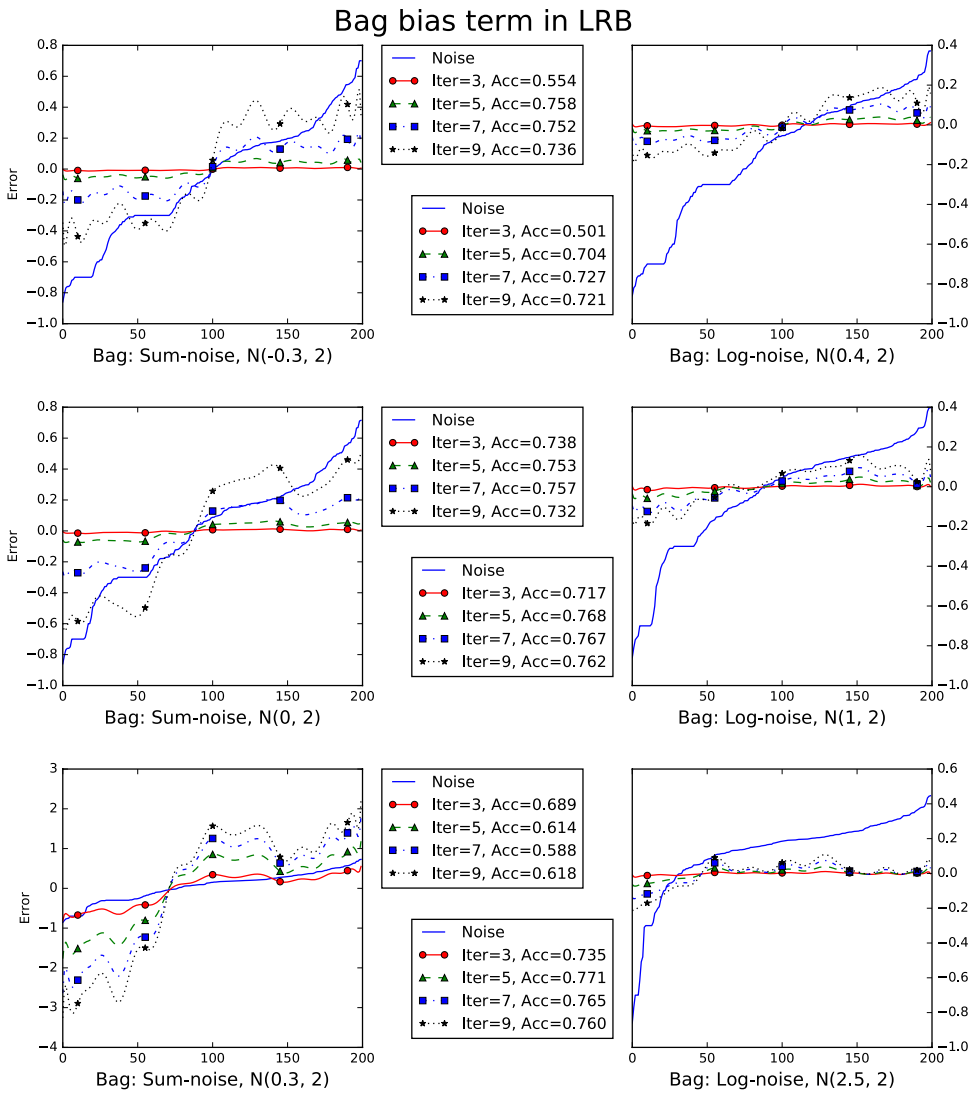


Fig. 1 Bag bias term in LRB model for IMDB dataset

Then we compute the average accuracy of all experiments for both **validation** and **testing** sets to calculate final validation and test scores.

To tune hyperparameters, we apply a simple grid search over validation data for the tuning step. To be fair, in this phase, we only tune one hyperparameter for each model by running models on a predefined range of values (up to 10 tests) and selecting the hyperparameter that results in the highest **validation** accuracy (averaged over five different seeds). In the case of ties, we pick the parameter with the highest F1 score. Finally, we report the **testing** accuracy. Also, in models with matrix factorization, we use only three hidden concepts ($H = 3$) for all experiments. We tune the following hyperparameters:

- L2 regularization (λ) strength for linear models.
- The number of iterations for **LR**, **LRB**, and **LRBF** models (as we showed the importance of the early stopping in Fig. 1).
- The γ in Eq. 18 for **LRBC** and **LRBFC**; we test a range of small negatives and small positives to find the best one.

5 Results: synthetic bags

In this section, we present results on synthetic bags for IMDB and Replab sentiment classification. We investigate several empirical questions:

1. How do the learned bag bias parameters vary with the noise in label proportions?

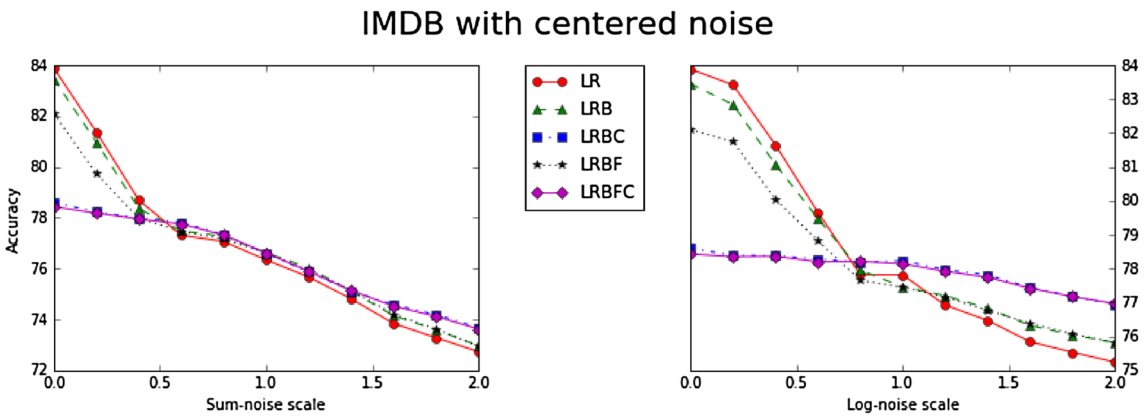


Fig. 2 Comparing different models on IMDB with centered noise: $\mathcal{N}(0, scale)$

2. How does error vary by noise type and size?
3. How does error vary when neutral instances are injected into bags?

We present results for each of these questions below:

How do the bag bias parameters vary with the noise in label proportions? Recall that the LRB model introduces a new parameter for each bag. The intuition is that these parameters will indicate the direction and magnitude of the shift required to compensate for the noise of the label proportions for each bag.

Figure 1 compares different scenarios of noise generation for IMDB dataset. In these plots, the blue line shows the error in bag label proportions. The y-axis indicates the absolute difference between the first class proportion after and before injecting the noise, and bags are sorted in ascending order of the error. We also train the **LRB** model with different iteration numbers and report the accuracy of the testing set in each plot. In addition, we plot the bag bias term in the **LRB** model. The idea is that the bag bias term is expected to be close the noise (the blue line). For example, if a bag has noise injected that increases the proportion of the positive label by 0.4, then we expect the bag bias term for that bag to be close to 0.4. (To improve figure readability, we smooth bag bias using linear regression with RBF kernel.)

According to Fig. 1, the learned bag bias parameter better reflects the noise in each bag after each training iteration. However, at some point, it can jump over the noise. We see a similar effect on the accuracy of the testing set; it increases until 5–7 iterations and then decreases because of the overfitting. This shows how early stopping regularization can prevent overfitting. Also, the figure reveals that the bag bias term is closer to the noise for **Sum-noise** than **Log-noise**. However, even when the bag bias is not close to the noise, it still can achieve a high accuracy (e.g., $\mathcal{N}(1, 2)$ and $\mathcal{N}(2.5, 2)$). We believe that this is in part because there is a nonlinear term in the hypothesis (logistic function), and the bag bias and the noise do not have to be in the same scale.

How does error vary by noise type and size? Next, we report how testing accuracy varies under different noise conditions. We consider several variants: (1) the **amount** of noise, as quantified by the scale parameter in the Gaussian variable sampled to introduce noise for each bag; (2) the **direction** of noise, as quantified by whether the noise is *centered* or *uncentered*; and (3) the **shape** of the noise, as quantified by whether the noise is **Sum-noise** or **Log-noise**.

We first consider **centered** noise. Figure 2 shows results of IMDB dataset. The left plot displays the **Sum-noise** and the right plot presents **Log-noise**, and x-axis is the noise scale. The first observation is that all models are more robust to the **Log-noise** than the **Sum-noise** for the same scale. For example for scale two, the accuracy of all models with the **Sum-noise** is roughly between 73 and 74%, and for the **Log-noise**, the accuracy is between 75 and 77%.

Also, models with the cross-validation step (**LRBC** and **LRBFC**) have lower accuracy with the smaller injected noise and are more robust to the higher noise. We believe that is because we only tune γ for these models, and according to Fig. 1 the number of iterations needs to change for different noise scales. As a result, these models can achieve higher accuracy with lower injected noise by using more iterations in the L-BFGS algorithm.

Furthermore, after some point, **LRB** and **LRBF** have close results together and have slightly higher accuracy than the **LR** model. Finally, we can observe that in most cases, matrix factorization step does not have any improvement. Thus, it shows that there are no hidden concepts in the synthetic bags.

Figure 3 shows the **centered** noise for the Replab dataset and indicates the similar behavior. Again, all models are more robust to the **Log-noise** than the **Sum-noise**, and with the higher noise **LRBFC** and **LRBC** are close together with highest accuracy, and then **LRB** and **LRBF** are close

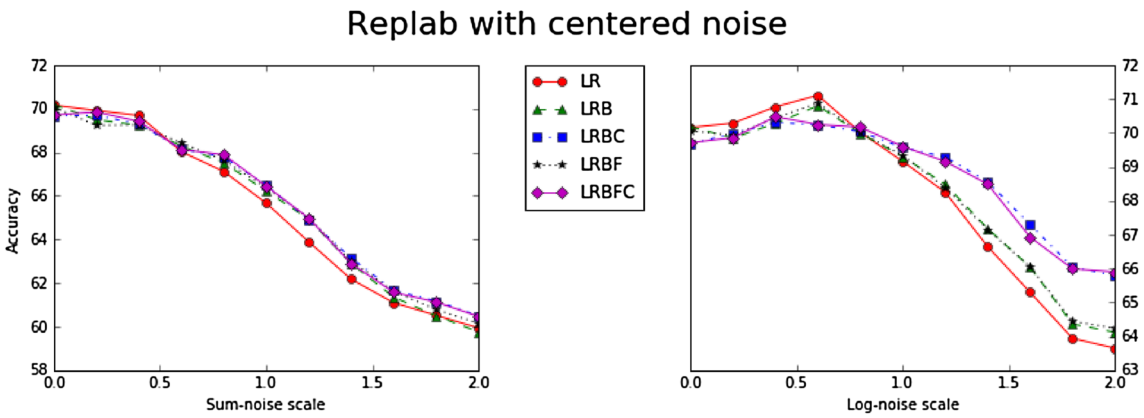


Fig. 3 Comparing different models on Replab with centered noise: $\mathcal{N}(0, scale)$

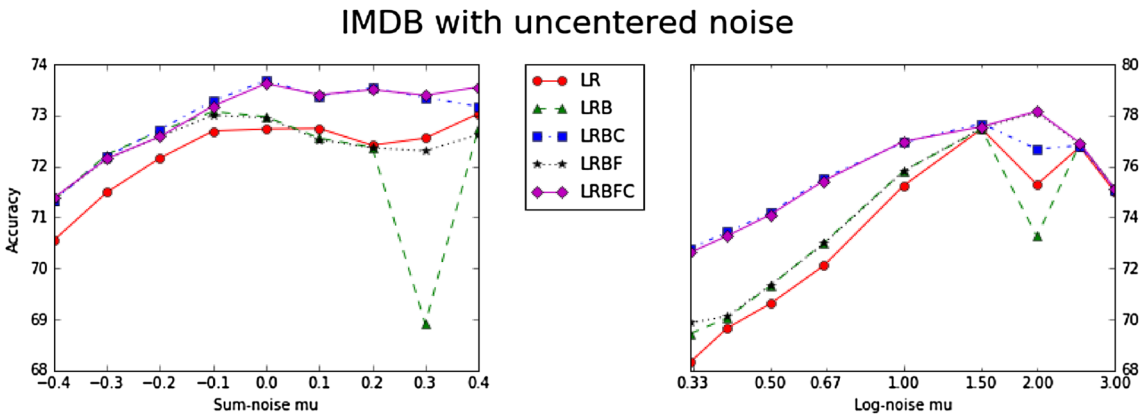


Fig. 4 Comparing different models on IMDB with uncentered noise: $\mathcal{N}(\mu, 2)$

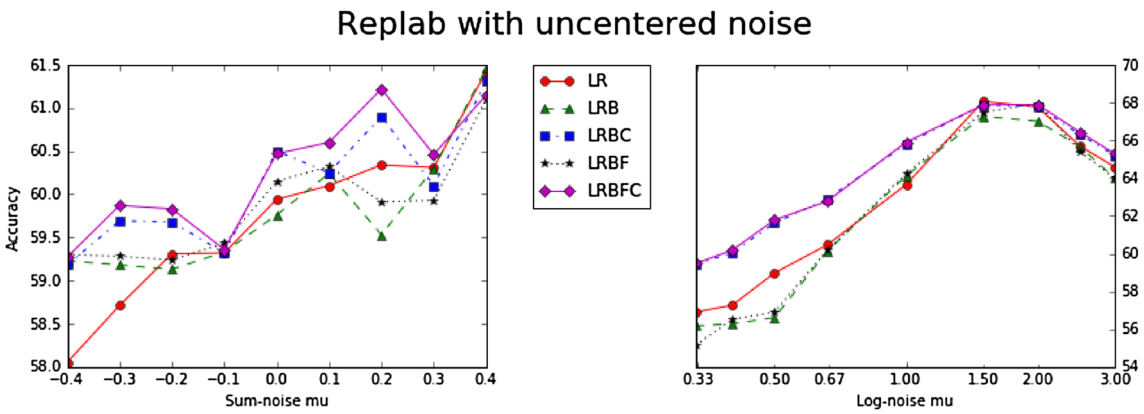


Fig. 5 Comparing different models on Replab with uncentered noise: $\mathcal{N}(\mu, 2)$

together with slightly lower accuracy, and finally **LR** has the lowest accuracy.

In addition, other observations include: for the Log-noise, adding noise can even increase accuracy (e.g., for scale .6); all models are very close together for the **Sum-noise**; and the average difference between **Sum-noise** and **Log-noise** is higher than the IMDB experiment in Fig. 2.

Next, we consider **uncentered** noise. Figure 4 illustrates results of the IMDB dataset with scale 2. The x-axis on this plot indicates different μ values. Clearly, **LRBC** and

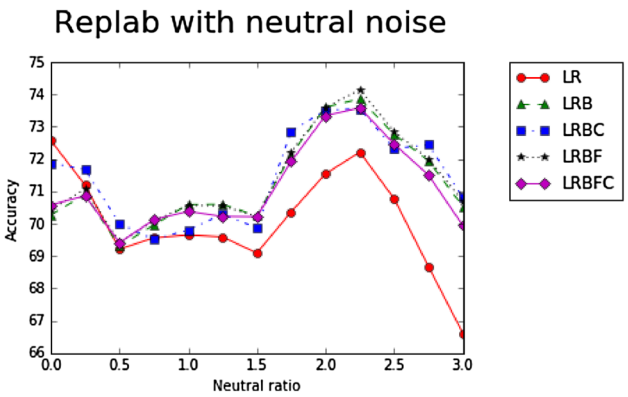


Fig. 6 Comparing different models on REPLAB with injected neutrals

Table 2 Summary of datasets and tasks

Task name	Bag types	Testing dataset	Testing labels
Politician	472 counties	Politician	200 Republican
	18 followers		200 Democrat
Politic-fol	472 counties	Politic-followers	600 Republican
	16 followers		600 Democrat
Age	700 followers	Age	800 below 25
	175 names		700 above 25
IMDB	200 synthetics	IMDB-test	25K positive
			25K negative
Replab	200 synthetics	Replab-test	5K positive
			5K negative

LRBFC are very close together and outperform in almost all cases. **LRB** and **LRBF** models are almost identical and outperform **LR**, except $\mu = .3$ in **Sum-noise** and non-positive μ (except $\mu = 2$) for the **Log-noise**.

For Replab with the **uncentered** noise with scale 2, as shown in Fig. 5, we find that **LRBC** and **LRBFC** are close together and outperform other models for the **Log-noise**. For **Sum-noise**, **LRBFC** is slightly better than **LRBC**, and similarly, **LRBF** is slightly better than **LRB**. That indicates in this case, matrix factorization has a slight improvement in accuracy.

The next observation for both IMDB and Replab experiments is that accuracy increases with the higher value of μ, and even it can be more than the **centered** noise. However, at some point, it decreases. It shows that in some cases, models are more robust to the **uncentered** noise than the **centered** noise.

Finally, we consider an alternative noise setting in which bags may contain neutral instances. In social media, many users have an indifferent sentiment or the attribute cannot be induced from their activities (e.g., for demographic characteristics such as age). We investigate how the number of neutral examples influences the error rate. Of our data sources, only the Replab dataset has neutral instances annotated, so we use that data to simulate neutral noise. Rather than adding noise to prior label proportions, we inject some neutral instances into bags. Figure 6 displays results of the Replab dataset as the number of neutral instances increases. The x-axis in this plot shows the ratio of injected neutrals, e.g., neutral ratio three means we insert 300 neutral examples to each bag (since each original bag has 100 samples). In addition, for **chi2** bags, we inject neutral samples with the corresponding χ^2 feature.

According to Fig. 6, **LR** has the lowest accuracy in most cases, and the difference among other models is not significant, but on average **LRBC** is slightly better than others. Most importantly, for the neutral ratio in [1.75, 2.5], the accuracy is even higher than without any neutral instances. And at peak for 2.25 (225% neutrals into bags), the accuracy of the best model is roughly 1.5% higher than the score of the best model without any neutral instances. We believe that is in part because neutral samples may result in a better separation in bags, and classifiers achieve a higher accuracy. This result is promising, because in social media we have many neutral instances; these results suggest that we may still learn accurate classifiers despite this.

To summarize these experiments with synthetic noise, we examined several types of prior and neutral noise and we found that:

1. **LRBC** and **LRBFC** have similar accuracy and are the most robust to noise.
2. **LRB** and **LRBF** have a similar accuracy and are more accurate than **LR**.
3. There are no hidden concepts in synthetic bags, and as a result matrix factorization does not provide a significant improvement.
4. Injecting a proper amount of neutral instances into bags can increase accuracy of models.
5. Models are often more robust to **uncentered** noise than **centered** noise.

6 Twitter experiments

With the promising results on synthetic noise experiments, we expand our experiments to social media data with natural noise. Table 2 summarizes the data used in these experiments:

1. **Politicians** We use follower (18 bags) and county (472 bags) constraints, and the testing set is the **Politicians** dataset.
2. **Politic-fol** 16 follower bags with 472 county bags, and the testing set is **Politic-followers** dataset. The two bags

Table 3 Classification accuracy on the testing set for competing LLP models

Dataset	LR	LRB	LRBC	LRBF	LRBFC	RANSAC	Ridge	RidgeC
Politician	87.3 ± 3.7	89.9 ± 2.1	90.7 ± 1.7	90.4 ± 1.0	91.0 ± 1.0	78.2 ± 6.1	88.6 ± 0.6	88.5 ± 0.7
Politic-fol	71.7 ± 1.2	71.2 ± 2.5	71.6 ± 2.1	72.5 ± 2.0	73.3 ± 1.0	70.1 ± 0.5	71.4 ± 0.8	71.6 ± 0.9
Age	76.0 ± 1.6	75.5 ± 4.7	76.1 ± 0.9	77.1 ± 1.9	77.5 ± 1.7	72.0 ± 1.7	75.4 ± 0.6	75.5 ± 0.6
Imdb	75.2 ± 2.1	75.4 ± 2.1	76.0 ± 1.6	75.5 ± 1.8	76.0 ± 1.6	72.4 ± 3.3	70.1 ± 3.4	70.6 ± 3.6
Replab	66.8 ± 4.1	66.9 ± 4.3	67.5 ± 4.0	66.9 ± 4.3	67.6 ± 4.0	65.1 ± 4.1	65.3 ± 4.0	65.5 ± 4.4
Average	75.4	75.8	76.4	76.5	**77.1**	71.6	74.2	74.4

The method with the highest average accuracy is in bold

Table 4 Mann–Whitney *U* test between **LR** and **LRBFC**

Dataset	*p* value
Politician	0.0375
Politic-fol	0.0361
Age	0.0873
Imdb	0.0046
Replab	0.2381

Table 5 Impact of each enhancement on accuracy of Twitter datasets

Dataset	Bag bias	Matrix factor	Correcting label prop
Politician	**89.9**	87.3	89.3
Politic-fol	71.2	70.4	**71.9**
Age	75.9	**79.5**	78.0

The method with the highest accuracy per dataset is in bold

("thedemocrats" and "gop") used to generate the labeled dataset are removed from the set of training bags.

3. **Age** We use name (175 bags) and follower (700 bags) constraints and use the **age** dataset as the testing set.
4. **IMDB** For comparison, we include the average of all IMDB experiments in the previous section.
5. **Replab** Average of all Replab experiments.

To make a fair comparison, similar to the synthetic noise, we use five different seeds to train models and select 20% of the testing set as the **validation** set, and then we tune our models on the average of validation accuracies for each seed and report the average and the standard deviation of the testing accuracies. The comparison of all different LLP models is summarized in Table 3. On average across all datasets, LRB reduces the error rate by 2% over the LR baseline, LRBC reduces error rate by 4%, LRBF reduces error rate by 4%, and combining all three enhancements together into LRBFC reduces error by 7%. Also, the model with all enhancements (i.e., LRBFC), on average, has 1.7% absolute improvement than the LR baseline. (Results are similar when evaluating with F1.)

According to this table, **LRBFC** outperforms other models for Twitter and ties with **LRBC** for synthetic experiments. Also, **LRBF** has a higher accuracy than **LRB** for the Twitter, but ties in the synthetic noise. That confirms that there are hidden concepts in Twitter bags, while there are not in synthetic bags. Also, the linear models have lower accuracy than nonlinear models in most cases; we believe that is because linear models train on the average of features per bag and omit individual features of each instance. Furthermore, we find that RANSAC performs poorly for this task—in this case, removing noisy bags from training is worse than the Ridge baseline.

To evaluate the significance of enhancements, we perform Mann–Whitney *U* test between **LR** and **LRBFC** models. In this test, we want to find an evidence that difference between two models is statistically significant, and we report *p* value in Table 4. According to this table, with threshold .1, the accuracy improvement in **LRBFC** is statistically significant in all tasks except for **Replab** dataset. That is, in part, because we use all results in Sect. 5 to perform this test for **Replab**, and due to the tuning, in some of them **LR** works better than **LRBFC**. Since the only large dataset is **IMDB** and we have a very small *p* value for that, that is an strong evidence of statistically significance of enhancements.

According to Table 3, while bag bias and matrix factorization do not have a big impact on robustness to noise for synthetic experiments, correcting label proportions has the highest impact on accuracy. However, it is not clear which enhancement has the highest impact for Twitter experiments. Therefore, we add only one enhancement to LR baseline and report the accuracy of Twitter datasets in Table 5. According to this table, the bag bias has the highest impact on **Politician**, correcting label proportions has the highest impact on **Politic-fol**, and matrix factorization has the highest impact on **age** dataset. Since each enhancement has different impact, making it hard to select the best enhancement by greedy approach.

To further investigate these results, we sought to characterize the type of noise present in the Twitter data. Since we do not have access to user-level labels in the **County-2014** data, we must estimate them. So, we used the following procedure: (1) fit a supervised logistic

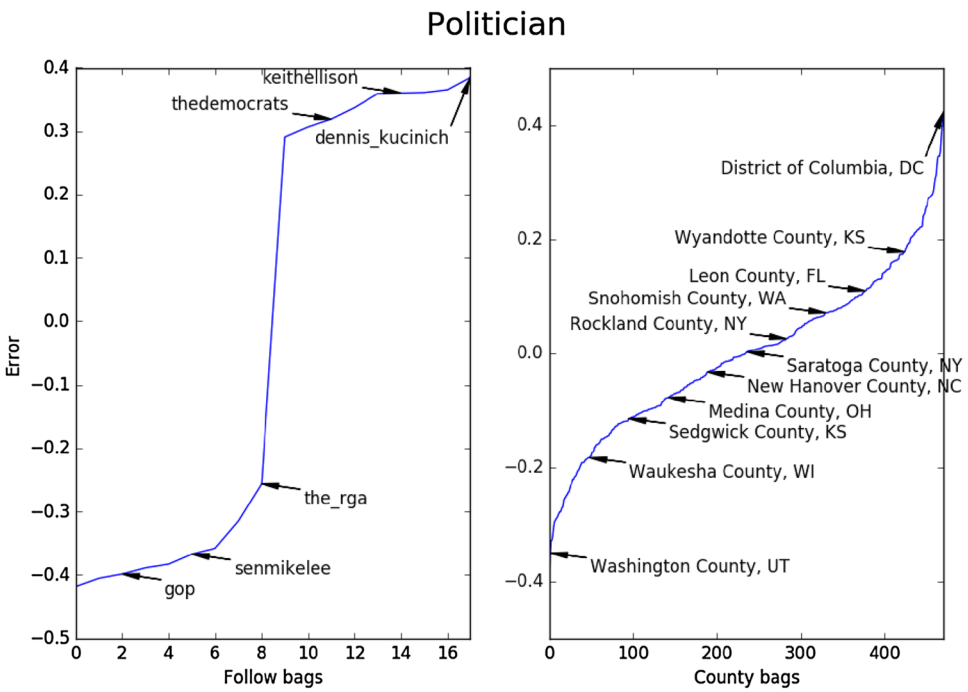


Fig. 7 Noise in **Politicians** bags

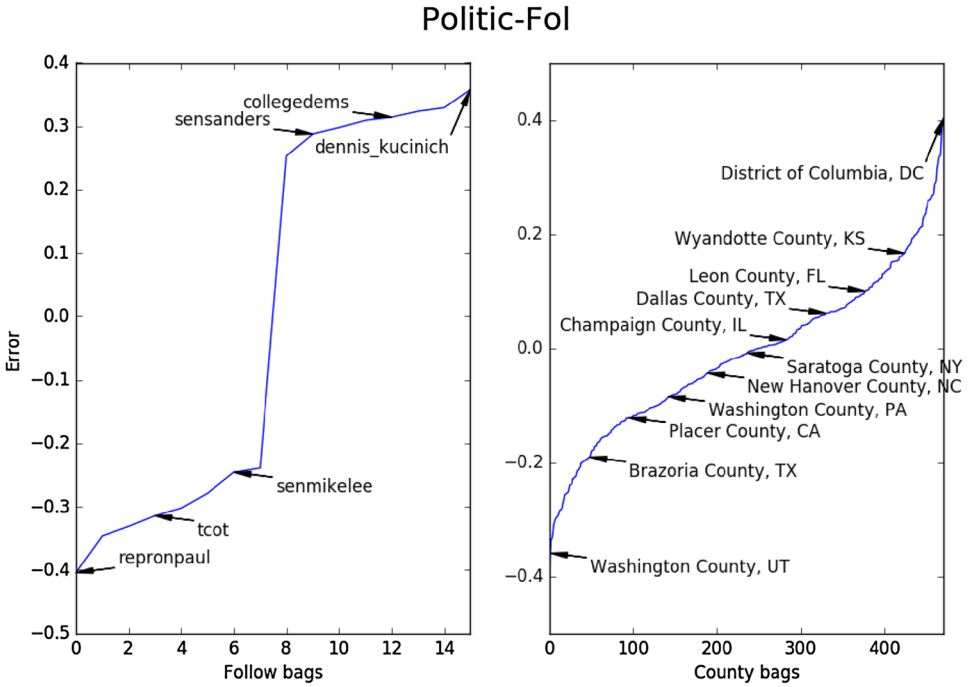
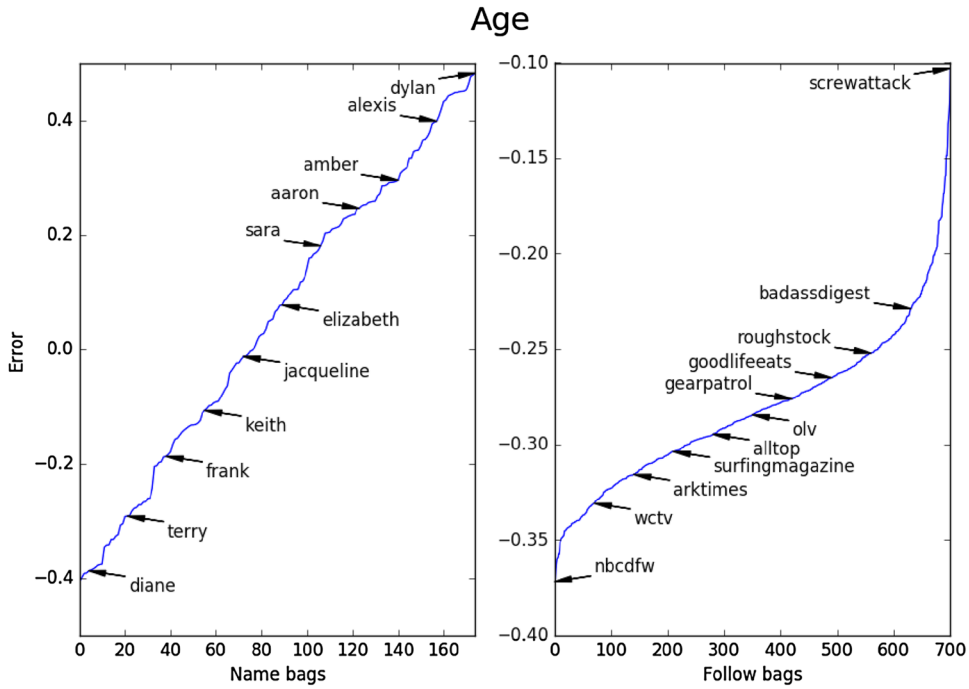


Fig. 8 Noise in **Politic-fol** bags

regression classifier on the testing set; (2) for each bag, estimate the posterior bag probability ($\bar{h}$) as the mean of the posterior likelihood of instances in each bag same as Eq. 1. We can then plot the difference between the estimated posterior and the prior ($\tilde{y} - \bar{h}$), sorted by increasing order of error, to characterize how the label proportion noise varies by bag.

Figures 7 and 8 illustrate the estimated noise in **Politicians** and **Politic-fol** bags. The first observation is that they are very similar and resemble the **Log-noise** plots in Fig. 1. Even though the plots seem to exhibit centered noise, we suspect that is because of the impact of too many neutral instances in bags.

Fig. 9 Noise in **Age** bags



Also, the county constraints come from the 2012 presidential election, which Democrats won, but our bags were sampled from Twitter activities in 2014, when Democrats lost the majority in both the House of Representatives[7] and the Senate.[8] Therefore, we expect that the county's prior overestimates Democrats, and the actual zero in these plots is likely to be slightly lower than what is shown in plots. As a result, we anticipate that the noise is somewhat **uncentered**.

For example, according to these plots, District of Columbia (DC) has the highest error in bags; this means DC has around 40% fewer Democrats than expected. This may in part be due to the fact that, as the seat of government, many people live in D.C. but do not vote in D.C. Thus, the distribution of political preference expressed by Twitter users in D.C. may be expected to deviate from the observed vote distribution in D.C. elections.

Figure 9 estimates noise type for the **age** experiment. According to this figure, name bags have **Sum-noise**, while follower bags have **Log-noise**. Even though this plot shows centered noise for name bags, same as the politics experiments, we suspect that our bags have younger users than the prior label proportions, because younger users tend to be overrepresented on Twitter (Mislove et al. 2011; Lenhart and Fox 2009). As a result, name bags has **uncentered** noise, and the actual zero in this figure is likely to be lower that what is shown in the plot. The follow bags in this figure have **uncentered** noise, and because the actual zero is higher, its μ is even lower than what is plot shown, and the noise is seemingly **uncentered** with very low μ.

7 Conclusion

Our results indicate that LLP models fit on social media data and population statistics can be used to classify individual user attributes, despite the sampling bias inherent in the training data. We have proposed three enhancements to label regularization to make it more robust to noise in the provided labeled proportions:

1. *Bag bias* By estimating the bag noise it reduces the classification error by 2% over a label regularization baseline.
2. *Matrix factorization* By learning hidden concepts in users it reduces error by another 3%.
3. *Cross-validation* Adjusting the prior label proportion with cross-validation achieves 3% reduction in classification error.

Together, these enhancements reduce error by 7% on average across all tasks.

We also find that while there are no hidden concepts in synthetic bags, there are latent concepts in social media (e.g., Twitter), and a matrix factorization model inspired by

[7] https://en.wikipedia.org/wiki/United_States_House_of_Representatives_elections,_2014.

[8] https://en.wikipedia.org/wiki/United_States_Senate_elections,_2014.

recommendation systems can discover such latent concepts to reduce generalization error. Finally, our synthetic experiments suggest that there is considerable variation in accuracy depending on the type of noise in label proportions, though our proposed methods outperform the baseline on average under all types of noise we investigated.

In future work, we will further investigate methods that are tailored to different noise types, enabling a hybrid approach that first estimates the type of noise and then applies the appropriate adjustment that is most suited to that noise type.

Acknowledgements We thank the anonymous reviewers for helpful feedback. This research was funded in part by National Science Foundation under Grants #IIS-1526674 and #IIS-1618244. Any opinions, findings, and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

Appendix: Partial derivatives of LR cost function

We use logistic function derivative, i.e.,

$$\frac{\partial}{\partial\theta}\sigma(f) = \sigma(f)(1-\sigma(f))\frac{\partial}{\partial\theta}f \tag{23}$$

to compute the derivative of hypothesis as:

$$\frac{\partial}{\partial\theta}h_{u,i} = h_{u,i}(1-h_{u,i})X_{u,i} \tag{24}$$

Now we can compute the partial derivative of cost function:

$$\begin{aligned}
\frac{\partial}{\partial\theta}J(\Theta) &= -\sum_i \left(\tilde{h}_i\frac{\partial}{\partial\theta}\log\bar{h}_i + (1-\tilde{h}_i)\frac{\partial}{\partial\theta}\log(1-\bar{h}_i)\right) + \lambda\theta \\
&= -\sum_i \left(\frac{\tilde{y}_i}{\bar{h}_i}\frac{\partial\bar{h}_i}{\partial\theta} - \frac{1-\tilde{y}_i}{1-\bar{h}_i}\frac{\partial\bar{h}_i}{\partial\theta}\right) + \lambda\theta \\
&= \sum_i \frac{\bar{h}_i-\tilde{y}_i}{\bar{h}_i(1-\bar{h}_i)}\frac{\partial\bar{h}_i}{\partial\theta} + \lambda\theta \\
&= \sum_i \frac{\bar{h}_i-\tilde{y}_i}{\bar{h}_i(1-\bar{h}_i)}\frac{\partial}{\partial\theta}\frac{1}{|T_i|}\sum_{u\in T_i}h_{u,i} + \lambda\theta \\
&= \sum_i \frac{\bar{h}_i-\tilde{y}_i}{\bar{h}_i(1-\bar{h}_i)|T_i|}\sum_{u\in T_i}\frac{\partial}{\partial\theta}h_{u,i} + \lambda\theta \\
&= \sum_i \frac{\bar{h}_i-\tilde{y}_i}{\bar{h}_i(1-\bar{h}_i)|T_i|}\sum_{u\in T_i}h_{u,i}(1-h_{u,i})X_{u,i} + \lambda\theta \\
&= \sum_{u,i}\frac{(\bar{h}_i-\tilde{y}_i)h_{u,i}(1-h_{u,i})}{\bar{h}_i(1-\bar{h}_i)|T_i|}X_{u,i} + \lambda\theta \\
&= \sum_{u,i} e_{u,i}X_{u,i} + \lambda\theta
\end{aligned} \tag{25}$$

where $e_{u,i}$ is defined in Eq. 10. The partial derivative of other variables in LRBF model is computed similarly.

References

Al Zamal F, Liu W, Ruths D (2012) Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In: ICWSM

Amigó E, Carrillo de Albornoz J, Chugur I, Corujo A, Gonzalo J, Martín T, Meij E, de Rijke M, Spina D (2013) Overview of RepLab 2013: evaluating online reputation monitoring systems. In: Proceedings of the fourth international conference of the CLEF initiative, pp 333–352

Ardehaly E Mohammady, Culotta A (2015) Inferring latent attributes of twitter users with label regularization. In: Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies, Association for Computational Linguistics, Denver, Colorado, pp 185–195. http://www.aclweb.org/anthology/N15-1019

Ardehaly EM, Culotta A (2016) Domain adaptation for learning from label proportions using self-training. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence, IJCAI 2016, New York, NY, USA, pp 3670–3676, 9-15 July 2016. http://www.ijcai.org/Abstract/16/516

Argamon S, Dhawle S, Koppel M, Pennebaker JW (2005) Lexical predictors of personality type. In: Proceedings of the joint annual meeting of the interface and the classification society of North America

Barberá P (2013) Birds of the same feather tweet together. Bayesian ideal point estimation using twitter data. In: Proceedings of the social media and political participation, Florence, Italy, pp 10–11

Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167

Burger JD, Henderson J, Kim G, Zarrella G (2011) Discriminating gender on twitter. In: Proceedings of the conference on empirical methods in natural language processing, association for computational linguistics, Stroudsburg, PA, USA, EMNLP '11, p 13011309. http://dl.acm.org/citation.cfm?id=2145432.2145568

Byrd RH, Lu P, Nocedal J, Zhu C (1995) A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput 16(5):1190–1208

Chang MW, Ratinov L, Roth D (2012) Structured learning with constrained conditional models. Mach Learn 88(3):399–431

Chang M, Ratinov L, Roth D (2007) Guiding semi-supervision with constraint-driven learning. In: ACL, association for computational linguistics, Prague, Czech Republic, pp 280–287. http://cogcomp.cs.illinois.edu/papers/ChangRaRo07.pdf

Chang J, Rosenn I, Backstrom L, Marlow C (2010) Epluribus: ethnicity on social networks. In: ICWSM

Cohen R, Ruths D (2013) Classifying political orientation on twitter: it's not easy! In: ICWSM

Conover MD, Gonçalves B, Ratkiewicz J, Flammini A, Menczer F (2011) Predicting the political alignment of twitter users. In: 2011 IEEE third international conference on Privacy, security, risk and trust (passat) and 2011 IEEE third international conference on social computing (socialcom). IEEE, pp 192–199

Culotta A, Kumar NR, Cutler J (2016) Predicting twitter user demographics using distant supervision from website traffic data. J Artif Intell Res (JAIR) 55:389–408

Diaz F, Gamon M, Hofman JM, Kıcıman E, Rothschild D (2016) Online and social media data as an imperfect continuous panel survey. PloS ONE 11(1):e0145406

Dredze M (2012) How social media will change public health. IEEE Intell Syst 27(4):81–84. https://doi.org/10.1109/MIS.2012.76

Eisenstein J, Smith NA, Xing EP (2011) Discovering sociolinguistic associations with structured sparsity. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: Volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, p 13651374. http://dl.acm.org/citation.cfm?id=2002472.2002641

Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395. https://doi.org/10.1145/358669.358692

Ganchev K, Graca J, Gillenwater J, Taskar B (2010) Posterior regularization for structured latent variable models. J Mach Learn Res 11:20012049. http://dl.acm.org/citation.cfm?id=1756006.1859918

Gopinath S, Thomas JS, Krishnamurthi L (2014) Investigating the relationship between the content of online word of mouth, advertising, and brand performance. Market Sci 33(2):241–258

Graca J, Ganchev K, Taskar B (2007) Expectation maximization and posterior constraints. NIPS 20:569–576

Jin R, Liu Y (2005) A framework for incorporating class priors into discriminative classification. In: Ho TB, Cheung D, Liu H (eds) Advances in knowledge discovery and data Mining. PAKDD 2005. Lecture Notes in Computer Science, vol 3518. Springer, Berlin

Kamerer D (2013) Estimating online audiences: understanding the limitations of competitive intelligence services. First Monday 18(5). https://dx.doi.org/10.5210/fm.v18i5.3986

Knowles R, Carroll J, Dredze M (2016) Demographer: extremely simple name demographics. In: NLP+ CSS 2016, p 108

Lenhart A, Fox S (2009) Twitter and status updating. PEW Internet & American Life Project, Washington DC

Lin CJ, Kuo TT, Lin SD (2014) A content-based matrix factorization model for recipe recommendation. In: Tseng V, Ho T, Zhou ZH, Chen A, Kao HY (eds) Advances in knowledge discovery and data mining, lecture notes in computer science, vol 8444. Springer International Publishing, pp 560–571. https://dx.doi.org/10.1007/978-3-319-06605-9_46

Liu W, Ruths D (2013) What's in a name? Using first names as features for gender inference in twitter. In: AAAI spring symposium on analyzing microtext. http://dblp.uni-trier.de/rec/bibtex/conf/aaaiss/LiuR13

Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, Association for Computational Linguistics, Portland, Oregon, USA, pp 142–150. http://www.aclweb.org/anthology/P11-1015

Maneewongvatana S, Mount DM (2002) Analysis of approximate nearest neighbor searching with clustered point sets. Data Struct Near Neighb Search Methodol 59:105–123

Mann GS, McCallum A (2007) Simple, robust, scalable semi-supervised learning via expectation regularization. In: Proceedings of the 24th international conference on machine learning, ACM, New York, NY, USA, ICML '07, p 593600. https://doi.org/10.1145/1273496.1273571

Mann GS, McCallum A (2010) Generalized expectation criteria for semi-supervised learning with weakly labeled data. J Mach Learn Res 11:955984. http://dl.acm.org/citation.cfm?id=1756006.1756038

Mislove A, Lehmann S, Ahn YY, Onnela JP, Rosenquist JN (2011) Understanding the demographics of twitter users. In: Proceedings of the fifth international AAAI conference on weblogs and social media (ICWSM'11), Barcelona, Spain

Musicant D, Christensen J, Olson J (2007) Supervised learning by training on aggregate outputs. In: Seventh IEEE international conference on data mining, 2007. ICDM 2007, pp 252–261. https://doi.org/10.1109/ICDM.2007.50

Nguyen D, Smith NA, Ros CP (2011) Author age prediction from text using linear regression. In: Proceedings of the 5th ACL-HLT workshop on language technology for cultural heritage, social sciences, and humanities, Association for Computational Linguistics, Stroudsburg, PA, USA, LaTeCH '11, p 115123. http://dl.acm.org/citation.cfm?id=2107636.2107651

O'Connor B, Balasubramanyan R, Routledge BR, Smith NA (2010) From tweets to polls: linking text sentiment to public opinion time series. ICWSM 11:122–129

Oktay H, Firat A, Ertem Z (2014) Demographic breakdown of twitter users: an analysis based on names. In: ASE Bigdata/Socialcom/Cyber Security Conference, Academy of Science and Engineering (ASE), Los Angeles. http://www.merl.com/publications/TR2014-042

Pennacchiotti M, Popescu AM (2011) A machine learning approach to twitter user classification. In: Adamic LA, Baeza-Yates RA, Counts S (eds) ICWSM. The AAAI Press. http://dblp.uni-trier.de/db/conf/icwsm/icwsm2011.html

Prechelt L (2012) Early stopping — But When?. In: Montavon G, Orr GB, Müller KR (eds) Neural networks: tricks of the trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin. https://doi.org/10.1007/978-3-642-35289-8_5

Preotiuc-Pietro D, Lampos V, Aletras N (2015) An analysis of the user occupational class through twitter content. In: ACL

Quadrianto N, Smola AJ, Caetano TS, Le QV (2009) Estimating labels from label proportions. J Mach Learn Res 10:23492374. http://dl.acm.org/citation.cfm?id=1577069.1755865

Rao D, Paul MJ, Fink C, Yarowsky D, Oates T, Coppersmith G (2011) Hierarchical Bayesian models for latent attribute detection in social media. In: Adamic LA, Baeza-Yates RA, Counts S (eds) ICWSM. The AAAI Press

Rao D, Yarowsky D, Shreevats A, Gupta M (2010) Classifying latent user attributes in twitter. In: Proceedings of the 2nd international workshop on search and mining user-generated contents, ACM, New York, NY, USA, SMUC '10, p 3744. https://doi.org/10.1145/1871985.1871993

Rendle S, Schmidt-Thieme L (2008) Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In: Proceedings of the 2008 ACM conference on recommender systems, ACM, New York, NY, USA, RecSys '08, pp 251–258. https://doi.org/10.1145/1454008.1454047

Rogati M, Yang Y (2002) High-performing feature selection for text classification. In: Proceedings of the eleventh international conference on information and knowledge management, ACM, New York, NY, USA, CIKM '02, pp 659–661. https://doi.org/10.1145/584792.584911

Rosenthal S, McKeown K (2011) Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: Volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, p 763772. http://dl.acm.org/citation.cfm?id=2002472.2002569

Salakhutdinov R, Mnih A (2008) Probabilistic matrix factorization. In: Platt JC, Koller D, Singer Y, Roweis ST (eds) Advances in neural information processing systems, Curran Associates, Inc., Red Hook, vol 20, pp 1257–1264. http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf

Saveski M, Mantrach A (2014) Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM conference on recommender systems, ACM, New York, NY, USA, RecSys '14, pp 89–96. https://doi.org/10.1145/2645710.2645751

Schapire RE, Rochery M, Rahim MG, Gupta NK (2002) Incorporating prior knowledge into boosting. In: Proceedings of the nineteenth international conference on machine learning, pp 538–545

Schler J, Koppel M, Argamon S, Pennebaker J (2006) Effects of age and gender on blogging. In: AAAI 2006 spring symposium on computational approaches to analysing weblogs (AAAI-CAAW), pp 06–03

Schwartz HA, Eichstaedt JC, Kern ML, Dziurzynski L, Lucas RE, Agrawal M, Park GJ, Lakshmikanth SK, Jha S, Seligman MEP, Ungar LH (2013a) Characterizing geographic variation in well-being using tweets. In: Seventh international AAAI conference on weblogs and social media (ICWSM)

Schwartz HA, Eichstaedt JC, Kern ML, Dziurzynski L, Ramones SM, Agrawal M, Shah A, Kosinski M, Stillwell D, Seligman MEP, Ungar LH (2013) Personality, gender, and age in the language of social media: the open-vocabulary approach. PloS ONE 8(9):e73791. https://doi.org/10.1371/journal.pone.0073791

She Y, Owen AB (2011) Outlier detection using nonconvex penalized regression. J Am Stat Assoc 106(494):626–639

Silver N, McCanc A (2014) How to tell someone's age when all you know is her name. Retrieved from http://fivethirtyeight.com/features/how-to-tell-someones-age-when-all-you-know-is-her-name/

Takacs G, Pilaszy I, Nemeth B, Tikk D (2008) Investigation of various matrix factorization methods for large recommender systems. In: IEEE international conference on data mining workshops, 2008. ICDMW '08, pp 553–562. https://doi.org/10.1109/ICDMW.2008.86

Tibshirani J, Manning CD (2014) Robust logistic regression using shift parameters. In: ACL, pp 124–129

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Volkova S, Van Durme B (2015) Online bayesian models for personal analytics in social media. In: Proceedings of the twenty-ninth conference on artificial intelligence (AAAI), Austin, TX

Wang Z, Lyu S, Schalk G, Ji Q (2012) Learning with target prior. In: Pereira F, Burges C, Bottou L, Weinberger K (eds) Advances in neural information processing systems, vol 25. Curran Associates, Inc., New York, pp 2231–2239. http://papers.nips.cc/paper/4849-learning-with-target-prior.pdf

Watkins SC (2009) The young and the digital: what the migration to social-network sites, games, and anytime, anywhere media means for our future. Beacon Press, Boston

Yang Y, Pedersen JO (1997) A comparative study on feature selection in text categorization. In: Proceedings of the fourteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '97, pp 412–420. http://dl.acm.org/citation.cfm?id=645526.657137

Yao Y, Rosasco L, Caponnetto A (2007) On early stopping in gradient descent learning. Constr Approx 26(2):289–315. https://doi.org/10.1007/s00365-006-0663-2

Zhang S, Wang W, Ford J, Makedon F (2006) Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the 6th SIAM conference on data mining, SDM, pp 549–553

Zhang T, Yu B (2005) Boosting with early stopping: Convergence and consistency. Ann Stat 33(4):1538–1579. http://projecteuclid.org/euclid.aos/1123250222

Zhu J, Chen N, Xing EP (2014) Bayesian inference with posterior regularization and applications to infinite latent svms. J Mach Learn Res 15:1799–1847