Balancing Prediction Errors for Robust Sentiment Classification

MOHSIN IQBAL, Information Technology University of the Punjab
ASIM KARIM, Lahore University of Management Sciences
FAISAL KAMIRAN, Information Technology University of the Punjab

Sentiment classification is a popular text mining task in which textual content (e.g., a message) is assigned a polarity label (typically positive or negative) reflecting the sentiment expressed in it. Sentiment classification is used widely in applications like customer feedback analysis where robustness and correctness of results are critical. In this article, we highlight that prediction accuracy alone is not sufficient for assessing the performance of a sentiment classifier; it is also important that the classifier is not biased toward positive or negative polarity, thus distorting the distribution of positive and negative messages in the predictions. We propose a measure, called Polarity Bias Rate, for quantifying this bias in a sentiment classifier. Second, we present two methods for removing this bias in the predictions of unsupervised and supervised sentiment classifiers. Our first method, called Bias-Aware Thresholding (BAT), shifts the decision boundary to control the bias in the predictions. Motivated from cost-sensitive learning, BAT is easily applicable to both lexicon-based unsupervised and supervised classifiers. Our second method, called Balanced Logistic Regression (BLR) introduces a bias-remover constraint into the standard logistic regression model. BLR is an automatic bias-free supervised sentiment classifier.

We evaluate our methods extensively on seven real-world datasets. The experiments involve two lexicon-based and two supervised sentiment classifiers and include evaluation on multiple train-test data sizes. The results show that bias is controlled effectively in predictions. Furthermore, prediction accuracy is also increased in many cases, thus enhancing the robustness of sentiment classification.

CCS Concepts: • **Information systems** → **Sentiment analysis**; • **Computing methodologies** → **Natural language processing**; *Supervised learning by regression*; • **Theory of computation** → Constraint and logic programming;

Additional Key Words and Phrases: Sentiment analysis, bias-aware sentiment analysis, supervised methods, lexicon-based methods, fairness in learning

ACM Reference format:
Mohsin Iqbal, Asim Karim, and Faisal Kamiran. 2019. Balancing Prediction Errors for Robust Sentiment Classification. *ACM Trans. Knowl. Discov. Data* 13, 3, Article 33 (June 2019), 21 pages.
https://doi.org/10.1145/3328795

Authors' addresses: M. Iqbal (corresponding author), Department of Computer Science, The Technical Faculty of IT and Design, Aalborg University, 9220 Aalborg Ø Denmark and Department of Computer Science, Information Technology University, Lahore, Pakistan; emails: mohsin@cs.aau.dk, mi308@itu.edu.pk; A. Karim, Department of Computer Science, Syed Babar Ali School of Science and Engineering, Lahore University of Management Sciences, 54792 Lahore, Pakistan; email: akarim@lums.edu.pk; F. Kamiran, Department of Computer Science, Information Technology University, Lahore, Pakistan; email: faisal.kamiran@itu.edu.pk.

1556-4681/2019/06-ART33 $15.00
https://doi.org/10.1145/3328795

1 INTRODUCTION

Sentiment classification is one of the most widely used tasks in text mining. It involves the labeling of a textual document (e.g., message, sentence, review) with a polarity descriptor (e.g., positive or negative) based on the sentiments and opinions expressed in the document. Sentiment classification, which is a key step in the more general task of sentiment analysis, attempts to summarize and organize the vast amounts of textual information available today in different domains. As such, sentiment classification has a variety of applications such as customer feedback analysis, product ranking, and recommender systems. Oftentimes, mission-critical decisions are based on the results of sentiment classification. Therefore, it is important that sentiment classification's results are not only correct but also reliable.

In general, sentiment classification methods can be unsupervised or supervised in nature. Unsupervised methods use a lexicon of polar words with their valence values to determine the polarity of a textual document. These methods have become very popular in recent years due to their ease of use (no training data is required) and acceptable accuracy. Supervised methods are standard text classification methods that require a labeled training dataset for learning a generative or discriminative model of classification. These methods are generally more accurate but are constrained by the availability of reliable training data. Sentiment classification performance is evaluated by its prediction accuracy—the proportion of correct classifications in a collection of documents.

While prediction accuracy is a sound measure of a classifier's performance it does not provide a complete picture of performance and can be misleading at times (Zliobaite 2015). For example, when labeling 100 documents classifier A produces 90 true positives and 10 false positives (accuracy is 90%) while classifier B produces 5 false positives and 5 false negatives (accuracy is 90% again). Classifier A is giving the misleading impression that all documents are positives, thus distorting the true distribution of 90 positive and 10 negative documents. On the other hand, classifier B's prediction errors are balanced as it maintains the 90:10 distribution in the predictions. Obviously, classifier B will be preferred over classifier A even though both have the same prediction accuracy. Recently, this issue has been identified and a technique for controlling it in lexicon-based methods has been proposed (Iqbal et al. 2015). However, a detailed analysis of the problem and methods for its solution in both unsupervised and supervised classifiers has not been reported. This issue is different from that of class imbalance in classification in two ways: First, it is not necessary that severe class imbalance exists in the training data for imbalanced predictions; biased predictions can still occur due to classifier's inductive bias. Second, biased predictions also occur in unsupervised or lexicon-based sentiment classifiers where training data imbalance is irrelevant.

In this article, we study prediction bias in sentiment classification and present methods for removing it. First, we highlight the limitation of accuracy for assessing the performance of sentiment predictions and the prevalence of imbalanced prediction errors in existing unsupervised and supervised sentiment classifiers. We propose a measure for quantifying this bias and discuss why it is not controlled in standard sentiment classifiers. Second, we present two methods for removing prediction bias (or balancing prediction errors) in unsupervised and supervised classifiers. Our first method is applicable to any lexicon-based or supervised classifier while our second method is a constrained variant of logistic regression (LR). These methods provide an easy way of controlling prediction bias in practice. We evaluate our methods extensively on seven real-world datasets. The results confirm that prediction bias is removed and, in most cases, prediction accuracy is increased over a wide range of train-test data sizes in comparison with standard unsupervised and supervised sentiment classifiers.

In summary, this article makes the following key contributions:

(1) Studies imbalanced prediction errors in lexicon-based and supervised sentiment classifiers, noting that this cannot be attributed to the class imbalance in the training data (for supervised sentiment classification).
(2) Proposes a measure for quantifying prediction bias whose minimization ensures that the distribution of polarities in the predictions is identical to the true distribution.
(3) Presents two easy-to-use methods for removing prediction bias in lexicon-based and supervised sentiment classifiers.
(4) Demonstrates effectiveness of the proposed methods on benchmark datasets.

The rest of the article is organized as follows. We discuss the related work in sentiment analysis and discrimination-aware data mining in Section 2. In Section 3, we discuss the issue of prediction bias in sentiment classification and propose a measure for quantifying it. Section 4 describes our methods for bias-aware sentiment analysis. Section 5 presents the experimental setup and datasets, and Section 6 discusses the experimental evaluation of our methods. We conclude our article in Section 7.

2 RELATED WORK

Sentiment analysis is the task of extracting and summarizing sentiments expressed in a document, while polarity detection or classification is the task of labeling a document as either positive or negative w.r.t. sentiment. Much work has been done on sentiment analysis and several methods have been developed for this purpose. Broadly, these methods can be categorized as either supervised or lexicon based. Lexicon-based sentiment analysis methods use pre-compiled dictionaries of words with their intensity for positive or negative sentiment. These dictionaries are used to ascertain the sentiment of new documents. Lexicon-based sentiment analysis methods, e.g., AFINN (Nielsen 2011) and SentiStrength (Thelwall et al. 2010), have become very popular because of their unsupervised nature and easy-use properties. In the literature, different dictionaries have been presented for different contexts, e.g., PANAS-t and POMS-ex (Bollen et al. 2011) word lists were created for the Web context (informal writing) and LIWC (Tausczik and Pennebaker 2010) was developed for formal English writings. Although lexicon-based methods do not require a labeled dataset for training, their coverage and performance can be affected by the context for which the word list is prepared and the context of the documents. Many lexicon-based methods have been proposed, but the literature suggests that SentiStrength (Thelwall et al. 2010) and AFINN (Nielsen 2011) are the most popular methods. Thus, despite the ease of use of lexicon-based methods they are limited by their generalization performance.

On the other hand, supervised sentiment analysis methods require labeled data to learn a model for predicting the sentiment for unseen documents. The major advantage of supervised methods is their ability to adapt to and learn from the context given in the labeled data. Thus, labeled training data is essential for this type of methods. In Iqbal et al. (2015), it has been shown that supervised methods do have the advantage of better generalization performance but they suffer from the major problem of biased predictions.

Bias in sentiment analysis is a new research area but a substantial amount of work has been done in the related field of discrimination-aware data mining and fairness in learning, first introduced in Pedreshi et al. (2008) and Luong et al. (2011). Discrimination prevention, a key focus area in discrimination-aware data mining, studies techniques for making classifiers learned over biased/discriminatory datasets discrimination aware. Similarly, in sentiment analysis, we are interested in making polarity detection methods bias-free. As such, there are parallels to discrimination

prevention techniques that involve classification algorithm tweaking (Calders and Verwer 2010; Kamishima et al. 2011; Luong et al. 2011). In (Kamiran et al. 2012, 2018), a decision-theoretic framework is presented for making any classifier learned over biased datasets discrimination-aware at run-time. Likewise, methods have been proposed to control discrimination through constraint-based learning, e.g., fair classification (Goh et al. 2016; Zafar et al. 2017) and privacy-preserving analysis (Edwards and Storkey 2016), but little work has been reported on identifying and controlling bias in sentiment analysis.

In machine learning, the problem of class imbalance in general (e.g., see, He and Garcia (2009) for a survey) and in sentiment classification in particular (e.g., Li et al. (2011) and Mountassir et al. (2012)) has been studied extensively. This problem arises when the class imbalance in the training data causes learned classifiers to predict all examples as belonging to the majority class. Methods for handling class imbalance involve data preprocessing to create a more balanced dataset (e.g., Chawla et al. (2002)), specialized classifiers that adjust for class imbalance (e.g., Tang et al. (2009)), and decision-theoretic adjustments at prediction time (e.g., Sun et al. (2007)). However, the class imbalance problem differs from the one addressed in this article in the following ways: (1) Imbalanced prediction errors arise even when class imbalance in the data is not severe (due to predictor bias), while works on handling class imbalance are concerned with severe class imbalance in data (minority class examples are less than 10% of the entire data). (2) Works on handling class imbalance focus on supervised learning while our work considers both supervised and unsupervised models. In particular, to the best of our knowledge, our work is the first to address prediction problems in unsupervised sentiment classification. (3) While different performance measures are adopted in previous works on handling class imbalance we introduce and focus on a new measure whose minimization ensures that the distribution of predicted polarities matches that in the data. Given these differences, a direct empirical comparison with methods for handling class imbalance in classification will not be very meaningful.

3 PREDICTION BIAS IN SENTIMENT CLASSIFICATION

In this section, we highlight the issue of prediction bias in sentiment classification and propose a measure for quantifying it. We start with formalizing the problem setting and notation used in our work. Subsequently, we motivate and present our measure for prediction bias and demonstrate its prevalence in standard sentiment classifiers.

3.1 Problem Setting

We are concerned with a sentiment classifier $C : d \in \mathcal{D} \mapsto \{+, -\}$ that labels a textual document d from a domain of documents $\mathcal{D}$ as either positive (+) or negative (−) according to the sentiment expressed in the document. Without loss of generality, we assume that the binary classification is based on a scoring function $S(\cdot)$ such that $S(d) > \theta$ implies that $C(d) = +$ and $S(d) \leq \theta$ implies that $C(d) = -$. Here, θ is a real number defining the decision boundary between the two classes.

The sentiment classifier $C(\cdot)$ can be supervised or unsupervised in nature. An unsupervised classifier does not have access to labeled examples from $\mathcal{D}$ but it relies upon background resources like a polarity lexicon with polarity scores. A supervised classifier, on the other hand, is trained on a random sample $\mathcal{D}_{tn}$ drawn from $\mathcal{D}$. The performance of the classifier, unsupervised or supervised, is evaluated on a test set $\mathcal{D}_{tt}$ drawn randomly from $\mathcal{D}$. As such, the train and test sets have the same distribution of examples as in the original domain.

Given the above problem setting, the goal is to label the documents in $\mathcal{D}_{tt}$ accurately and without misleading biases. In general, a classifier with higher accuracy and lower bias will be preferred over others with higher biases.

3.2 Sentiment Classifiers

Popularly used sentiment classifiers are either lexicon-based or supervised in nature. We describe these types of classifiers in the following subsections.

3.2.1 Lexicon-based Methods. Unsupervised lexicon-based methods for sentiment classification do not require labeled data for learning but instead rely upon a predefined sentiment lexicon or affective word list to predict the polarity of documents. These methods have become very popular in recent years because of their ease of applicability and strong performances (Gonalves et al. 2013).

Several lexicon-based methods are available with different word lists and word valence values. In general, a typical lexicon-based sentiment classifier defines a list of words and phrases that convey polarity sentiment in documents. Each word and phrase in this sentiment lexicon or affective word list is given a valence value in the interval $[+v, -v]$, where v is a non-negative number that signifies the strength of polarity, and the sign of the valence value indicates the direction of the polarity (positive or negative). Such a lexicon of words/phrases and their valence values is usually developed manually by linguists.

Given a document d to be classified, the lexicon-based classifier computes score $S(d)$ from the words and phrases in d that match those in the lexicon and by applying a combination operation on the valence values of these words/phrases (unmatched words/phrases are given zero valence values). Commonly used operations include maximum valence (e.g., SentiStrength) and weighted average valence (e.g., AFINN). Subsequently, document d is classified as either positive or negative by the following decision rule:

$$C(d) = \begin{cases} + & \text{when } S(d) > \theta \\ - & \text{otherwise} \end{cases}. \tag{1}$$

Here, θ defines the decision boundary. Usually, $\theta = 0$ as positive scores signify positive sentiment in most lexicon based methods.

Lexicon-based sentiment classifiers involve empirically established parameters and as such do not provide any guarantees regarding accuracy or bias in the predictions. Furthermore, these methods incorporate significant inductive bias as they are learned from specific domains using specific techniques and hence their performances do not generalize consistently in other domains.

3.2.2 Supervised Sentiment Classifiers. Supervised sentiment classifiers are standard text classifiers trained on a collection of positive and negative documents. Despite the recent popularity of unsupervised sentiment classifiers, supervised classifiers are still preferred when labeled data is available due to their strong generalization performance. Among the various text classifiers, naive Bayes (NB) classifier and LR are known to produce robust sentiment classification. While both of these classifiers are probabilistic techniques, NB is generative in nature and LR is discriminative in nature.

Supervised classifiers require a labeled train set $\mathcal{D}_{tn}$ for learning. Let $t_i \in \{1, 0\}$ be the label for document $d_i \in \mathcal{D}_{tn}$, where $1 \equiv +$ and $0 \equiv -$. Furthermore, supervised classifiers usually require that documents are represented in a vector space. Let $\mathbf{x}_i = [x_{1i}, x_{2i}, \ldots, x_{Mi}]^{\mathrm{T}}$ be the representation of document d_i such that $x_{ji} \geq 0, \forall i, j$, where M is the vocabulary size (number of distinct words or terms) in $\mathcal{D}_{tn}$. The element x_{ji} quantifies the importance of term j in document d_i and/or document collection according to a term weighting scheme (e.g., term frequency, *TFIDF*).

For NB classifier, the scoring function for a document d_i is defined as

$$S(d_i) = \sum_{j=1}^{M} \log \frac{p(x_{ji}|t_i = 1)}{p(x_{ji}|t_i = 0)} + \log \frac{p(t_i = 1)}{p(t_i = 0)}. \tag{2}$$

Table 1. Truth Table for Sentiment Prediction

Actual↓, Predicted→	Pos (+)	Neg (−)	Sum
Pos (+)	TP	FN	N^+_{tt}
Neg (−)	FP	TN	N^-_{tt}
Sum	$\bar{N}^+_{tt}$	$\bar{N}^-_{tt}$	N_{tt}

Here, $p(x_{ji}|t_i)$ and $p(t_i)$ is the probability of term given class and probability of class, respectively. These probabilities are estimated from the train set $\mathcal{D}_{tn}$. With this scoring function, the classification is given by Equation (1) with $\theta = 0$.

For LR, the scoring function for a document d_i is given by

$$S(d_i) = y_i = \frac{1}{1 - \exp(-w_0 - \sum_{j=1}^{M} w_j x_{ji})}, \tag{3}$$

where w_j ($j = 0, \ldots, M$) are parameters of the model estimated by minimizing the cross-entropy of y and t over the train set:

$$\min_{\mathbf{w}} \sum_{i=1}^{N_{tn}} -t_i \log y_i - (1 - t_i) \log(1 - y_i). \tag{4}$$

With the scoring function given by Equation (3), the label for d_i is again given by Equation (1) but now with $\theta = 0.5$. This is because the scoring function is actually the posterior probability $p(\mathbf{x}_i|t_i = 1)$ which when greater than 0.5 indicates that $\mathbf{x}_i$ corresponds to a positive document.

3.3 Measuring Prediction Bias

As stated earlier, the performance of a sentiment classifier $C(\cdot)$ is evaluated by predicting the labels for all documents in $\mathcal{D}_{tt}$. Let N^+_{tt} and N^-_{tt} be the number of positive and negative documents, respectively, in $\mathcal{D}_{tt}$, and $N_{tt} = N^+_{tt} + N^-_{tt}$ is the total number of documents in the test set. These numbers reflect the distribution of polarities in the domain of analysis.

Table 1 shows the truth table for the sentiment classifier over the test set. Each cell in this table gives the number of examples correctly or incorrectly predicted by the classifier. For example, TP (true positives) and FN (false negatives) are the numbers of examples correctly predicted as positive and incorrectly predicted as negative, respectively, by the classifier. The row and column sums are given in the respective rightmost and bottom cells of the table.

Naturally, we prefer that the classifier makes as few errors as possible, i.e., $(FP + FN)$ is as small as possible (or accuracy is as high as possible). Although error rate $((FN + FP)/N_{tt})$ or accuracy $((TP + TN)/N_{tt})$ is a sound measure of performance it does not provide a complete picture and may hide biases in the predictions (Zliobaite 2015). For example, consider a domain whose test set has $N_{tt} = 100$, $N^+_{tt} = 70$, and $N^-_{tt} = 30$ examples. A classifier that labels all examples as positive (i.e., $TP = 70$, $FN = 0$, $FP = 30$, and $TN = 0$) will have an accuracy of 70%. The same accuracy can be produced by another classifier with $TP = 55$, $FN = 15$, $FP = 15$, and $TN = 15$. The latter classifier, however, maintains the same distribution of 70 positives and 30 negatives in the prediction, i.e., $\bar{N}^+_{tt} = N^+_{tt}$ and $\bar{N}^-_{tt} = N^-_{tt}$). The first classifier, on the other hand, gives the misleading impression that all documents are positive in the domain, i.e., $\bar{N}^+_{tt} = 100$ and $\bar{N}^-_{tt} = 0$.

This bias results from the imbalance in prediction errors, i.e., when FP is not equal to FN. This leads to the following definition for *Error Imbalance (EI)* or *Polarity Bias Rate (PBR)*.

Definition 1 (*PBR*, *EI*). The *PBR* of a sentiment classifier on N_{tt} examples is defined as

$$PBR = \frac{FP - FN}{N_{tt}}. \tag{5}$$

PBR ranges from −1 to +1, where positive values indicate bias toward incorrect positive predictions. *PBR* can also be expressed as the difference between the proportion of predicted positive examples $(TP + FP)/N_{tt} = \bar{N}^{+}_{tt}/N_{tt}$) and the proportion of actual positive examples $((TP + FN)/N_{tt} = N^{+}_{tt}/N_{tt})$. For convenience, *PBR* can also be given as a percentage. Considering the example introduced earlier, the bias of the first classifier is $PBR = 0.3$ or it produces 30% more positives than in the actual distribution in the domain.

It is worth noting that in most applications of sentiment analysis errors in labeling positive and negative documents (i.e., false positives and false negatives) are equal. In other words, the cost of misclassifying positive and negative documents is the same. Hence, minimizing the error rate (or maximizing the accuracy) is a goal of sentiment classification. Nonetheless, to provide a broader perspective of performance, especially when the distribution of positive and negative documents differ greatly, the average recall can also be computed. The average recall is defined as $0.5(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$ (i.e., average of sensitivity and specificity). Also, note that when PBR is equal to zero, recall for positive class (and negative class) become equal to precision for positive class, thus corresponding to the break-even point in the ROC space. Therefore, it is redundant to provide the values for average precision.

Numerous measures have been proposed for text classification problems. Different measures have their pros and cons that make them less appropriate for certain tasks. The F1-score is the harmonic mean of precision and recall of a selected class (e.g., positive sentiment) while in sentiment analysis classification performance on both classes are equally important (Sokolova and Lapalme 2009). The AUC is meant for evaluating and comparing classifiers rather than quantifying performance for a single operating condition. More precisely, the AUC is obtained by sweeping through all operating conditions (e.g., classification thresholds) for a classifier while in practice classification is done using a single operating condition.

Sokolova and Lapalme (2009) provide a detailed analysis of different measures for text classification. They note that for sentiment analysis and other human-annotated data classification tasks accuracy is the most appropriate performance measure.

3.4 Prediction Bias in Sentiment Classifiers

In this section, we highlight that standard supervised and unsupervised sentiment classifiers are often biased. We discuss the performance characteristics of two lexicon-based (SentiStrength and AFINN) and two supervised (NB and LR) sentiment classifiers on seven benchmark datasets. Table 2 shows the key characteristic of these datasets,[1] including their positive–negative example distribution. In addition to reporting accuracy and *PBR* (bias), we also report average recall—mean of recall for positive and recall for negative documents.

3.4.1 Lexicon-based Methods. Table 3 gives the accuracy, *PBR*, and average recall of AFINN and SentiStrength on benchmark datasets. These results are on 40% samples of the respective datasets. The distribution of positive and negative examples of the respective datasets is maintained in the samples used for testing.

These results confirm that significant bias can exist in the predictions of lexicon-based classifiers. In all but one case (SentiStrength on Movie Review), this bias is toward positive predictions.

[1] Thelwall (2013), Pang and Lee (2004), and Maas et al. (2011).

Table 2. Datasets and Their Key Characteristics

Data	# Docs	% Pos	% Neg
Movie Review	2,000	50	50
BBC	1,000	9.9	90.1
Digg	1,084	19.5	80.5
Runner World	1,046	46.3	53.7
Twitter	4,242	31.6	68.4
YouTube	3,407	48.9	51.1
Large Movie Review	25,000	50	50

Table 3. Accuracy, *PBR* (Bias), and Average Recall in Existing Lexicon-based Sentiment Classifiers—Results on 40% Test Set

	AFINN			SentiStrength		
Data	Accuracy	*PBR*	Avg. Recall	Accuracy	*PBR*	Avg. Recall
Movie Review	63.13	15.38	63.13	54.50	−39.50	54.50
BBC	69.50	23.00	65.95	84.25	7.75	72.98
Digg	74.48	12.53	71.52	79.12	7.42	73.95
Runner World	64.20	16.71	65.25	68.02	0.48	67.88
Twitter	69.30	11.02	69.18	72.89	12.14	73.81
YouTube	73.44	1.47	73.46	77.04	4.18	77.12

Table 4. Accuracy, *PBR* (Bias), and Average Recall in Existing Supervised Sentiment Classifiers—Results on 40% Test Set

	Logistic Regression			Naive Bayes		
Data	Accuracy	*PBR*	Avg. Recall	Accuracy	*PBR*	Avg. Recall
Movie Review	78.88	6.375	78.88	57.75	29.50	57.75
BBC	60.00	32	59.54	57.25	31.25	50.01
Digg	55.68	26.22	54.88	62.41	16.71	56.36
Runner World	59.19	−9.79	58.24	58.23	−6.92	57.49
Twitter	69.95	−8.25	61.72	64.64	−0.24	58.99
YouTube	68.09	0.22	68.07	60.53	12.47	60.79

Also, there appears to be no relation between bias and the underlying distribution of positive and negative documents. Since it is impossible to determine *a priori* the direction and magnitude of bias in lexicon-based classifiers, their results should be verified before using them for important decision-making tasks.

3.4.2 Supervised Sentiment Classifiers. Table 4 shows the accuracy, *PBR*, and average recall of LR and NB classifier on 40% test sets after learning on the respective remaining 60% training sets of the benchmark datasets. The test sets are identical to those used in evaluating lexicon-based methods (see Table 3).

These results confirm that significant bias also exists in supervised sentiment classifiers. The bias is primarily positive but again no general patterns between bias and dataset characteristics are obvious. Nonetheless, the presence of bias implies that supervised classifiers should also be used with care as they can distort the polarity mix of the predictions.

Unlike unsupervised methods, supervised classifiers learn from labeled documents from the domain of analysis. They, therefore, have the opportunity to tune their performance accordingly. The NB classifier is heavily dependent on the veracity of its underlying assumptions (e.g., independence of terms given class) and probability estimation errors. Thus, its performance is hard to characterize in practice on real-world datasets. On the other hand, LR optimizes a performance criterion—the cross-entropy—directly over the train set. However, minimizing the cross-entropy does not guarantee that the errors are balanced in the predictions. In general, different inductive biases in classifiers produce varying performances on test sets that do not ensure balanced prediction errors.

4 BALANCING ERRORS IN SENTIMENT CLASSIFICATION

We present two methods for balancing the errors in the predictions of sentiment classifiers. The first method, called Bias-Aware Thresholding (BAT), can eliminate bias in any lexicon-based or supervised sentiment classifier while our second method, called Balanced Logistic Regression (BLR), modifies the standard LR optimization task by introducing an error-balancing constraint.

4.1 Bias-Aware Thresholding (BAT)

BAT removes prediction bias in a sentiment classifier by moving its decision boundary until its prediction errors (false positives and false negatives) are balanced. BAT can be applied to both unsupervised and supervised sentiment classifiers without requiring modifications to the respective methods; it only modifies the threshold of the respective decision rule. As such, BAT is widely applicable and easy-to-use.

Consider a sentiment classifier $C(\cdot)$ having scoring function $S(\cdot)$. For a given document d, BAT classifies the document according to the following decision rule:

$$C_{BAT}(d) = \begin{cases} + & \text{when } S(d) > \theta + \delta \\ - & \text{otherwise} \end{cases}, \tag{6}$$

where $\delta \in \mathfrak{R}$ is a boundary shift parameter. A positive value of δ will shift the boundary toward the positive region potentially reducing false-positive errors and increasing false-negative errors in the prediction while a negative value for δ will have the opposite effect.

For probabilistic sentiment classifiers like NB and LR, BAT can be related to cost-sensitive learning in which specific errors (false positives or false negatives) are reduced by shifting the decision boundary (Elkan 2001; Kamiran et al. 2012). Specifically, shifting the decision boundary by increasing δ increases the cost for false positives while decreasing the value of δ increases the cost for false negatives, thus forcing fewer errors of the respective type in the predictions.

Some labeled documents are needed to estimate the value of δ. For supervised sentiment classifiers, the value of δ is fixed over the train set. For unsupervised lexicon-based classifiers, a small number of labeled documents is sufficient to estimate the value of δ accuracy (as verified in our experiments).

Algorithm 1 outlines the working of the BAT.

4.2 Balanced Logistic Regression (BLR)

BLR is a constrained LR model for bias-free supervised sentiment classification. BLR incorporates a constraint for balancing prediction errors in the standard LR optimization formulation. As such, BLR learns a bias-free model automatically without requiring any user input.

Consider a standard LR model $C(\cdot)$ with the scoring function $S(\cdot)$ defined in Equation (3). The parameters $\mathbf{w}$ of the standard LR model are found by minimizing the unconstrained optimization problem defined in Equation (4). On the other hand, BLR estimates the parameters $\mathbf{w}$ by solving

ALGORITHM 1: BAT—Bias-Aware Thresholding

Required: Standard sentiment classifier $C(d)$ with scoring function $S(d)$ and parameter θ, Documents $d \in \mathcal{D}$, Threshold δ
Output: Label (+ or −) for documents $d \in \mathcal{D}$
for all $d \in \mathcal{D}$ **do**
  if $S(d) > \theta + \delta$ **then**
    $C_{BAT}(d) = +$
  else
    $C_{BAT}(d) = -$
  end if
end for

the following constrained optimization problem:

$$\min_{\mathbf{w}} \sum_{i=1}^{N_{tn}} -t_i \log y_i - (1 - t_i) \log(1 - y_i), \tag{7}$$

$$\text{Subject to } \frac{1}{N} \left| \sum_{i=1}^{N_{tn}} t_i I(y_i \leq 0.5) - (1 - t_i) I(y_i > 0.5) \right| \leq \epsilon. \tag{8}$$

Here, $I(\cdot)$ is an indicator function that returns 1 when its argument is true and 0 otherwise, and $\epsilon \geq 0$ is a small number. The objective of this optimization problem is still the minimization of the cross-entropy between y and t over the train set but now a minimum is desired that also satisfies the constraint that *PBR* is less or equal to the small number ϵ.

Unlike standard LR, BLR solves a non-linear constrained optimization problem with the added complexity that the constraint is not continuous as it involves counts of errors. Such an optimization problem can be solved by a general non-linear constraint optimization technique like interior point. However, such techniques can be slow. But, its computational time is not a deciding factor since optimization is done off-line and once for a given dataset; there is really no practical difference between a few seconds and a few minutes. The BLR algorithm is shown in Algorithm 2. In our implementation, we solved the optimization problem using MATLAB's fmincon.[2]

5 EXPERIMENTAL SETUP AND DATASETS

In this section, we introduce the datasets and outline our experimental setup.

5.1 Datasets

Table 2 summarizes the key characteristics of our evaluation datasets. In all, we conduct experiments on seven datasets described below.

The Movie Review dataset (Pang and Lee 2004) is a collection of comments on movies obtained from Internet Movie Database (IMDB). The dataset contains 1,000 positive and 1,000 negative comments.

The Web 2.0 dataset is a human-labeled dataset made available by the SentiStrength research group (Thelwall 2013). It includes a wide range of messages, tweets, reviews, and comments from different sources. We use collections from the following: (1) BBC forum, (2) Digg, (3) Runners World forum, (4) Twitter, and (5) YouTube. Each document in the collection is labeled with positive and negative sentiment scores. To assign a binary polarity label to a document we subtract its negative

[2] https://www.mathworks.com/help/optim/ug/fmincon.html.

score from its positive score and label it as positive if this difference is greater than zero and label it as negative otherwise.

The Large Movie Review dataset (Maas et al. 2011) is a large collection of movie reviews. It contains a total of 50,000 reviews with an equal percentage of positive and negative reviews. The original dataset is divided into test and training sets of 25,000 reviews each. We performed experiments using the training dataset only.

ALGORITHM 2: BLR—Balanced logistic regression

Required: Train documents $\mathcal{D}_{tn}$, Test documents $\mathcal{D}_{tt}$
Output: Label (+ or −) for documents $d \in \mathcal{D}_{tt}$
$S(\cdot) \leftarrow$ Scoring function for BLR by solving constrained optimization problem for $\mathcal{D}_{tn}$
for all $d \in \mathcal{D}_{tt}$ **do**
  if $S(d) > 0.5$ **then**
    $C_{\mathcal{BLR}}(d) = +$
  else
    $C_{\mathcal{BLR}}(d) = -$
  end if
end for

5.2 Experimental Setup

We compare BAT with two popular lexicon-based sentiment classifiers (AFINN and SentiStrength). Since BAT requires the selection of parameter or threshold δ, we evaluate BAT's performance after fixing the threshold on varying sizes of labeled data. As this setting becomes identical to that of supervised classification, we also compare BAT with supervised classifiers when they are learned over the same set of labeled data. We compare BLR with standard LR on varying training data sizes.

For each experiment of the BAT, the value of threshold δ that produces the smallest absolute *PBR* on the labeled data is selected. This is done through iterative line search. More precisely, when PBR is positive then the value of δ is increased by a fixed proportion and when PBR is negative then the value of δ is decreased by a fixed proportion. These steps are repeated until PBR becomes close to zero. Subsequently, this value is used while predicting the labels of new documents.

We use Python implementation of AFINN and Java implementation of SentiStrength available from the respective websites. While using these implementations no additional preprocessing of the text documents is done. For NB and LR classifiers, we use the implementation provided in RapidMiner. Standard text preprocessing of tokenization, stop word removal, and stemming is performed on the text documents. We report prediction accuracy and *PBR* on different sizes of the datasets. We also report average recall to understand its tradeoff with accuracy and *PBR*.

6 RESULTS AND DISCUSSION

We divide the presentation of experimental results into two parts starting with unsupervised sentiment classification followed by supervised sentiment classification.

6.1 Unsupervised Sentiment Classification

We conduct two categories of experiments. First, we evaluate the performance of BAT with changing values of its threshold δ. We do this for both BAT combined with AFINN and BAT combined with SentiStrength. Second, we compare the performance of BAT/AFINN with NB classifier and

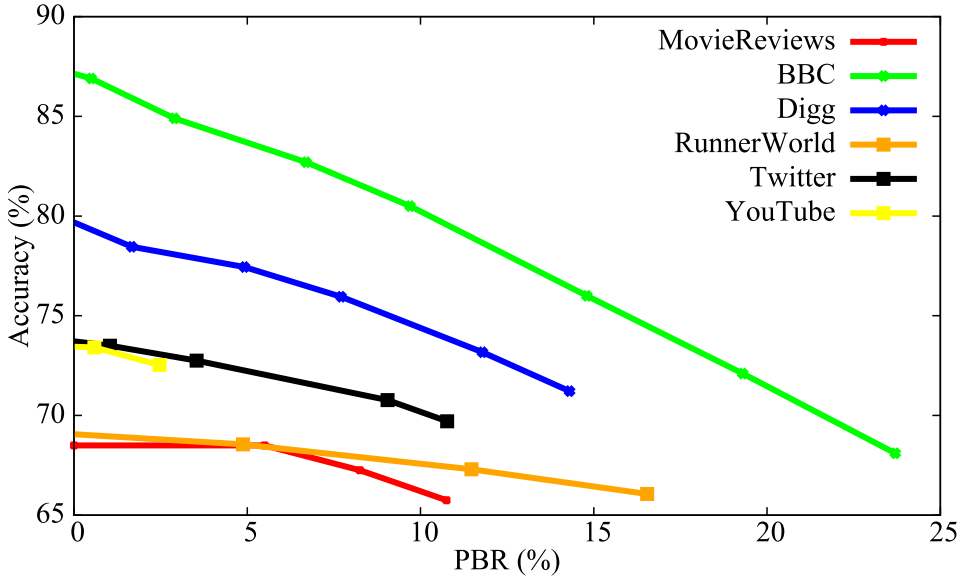

Fig. 1. Performance of BAT when combined with AFINN (BAT-AFINN).

LR. In these experiments, the NB, LR, and the threshold δ of BAT are learned from labeled datasets and performance is measured over the respective remaining test datasets. The baselines for our evaluations are standard AFINN and SentiStrength (for the first category of experiments) and standard NB and LR (for the second category of experiments).

Figure 1 shows the performance of BAT when combined with lexicon-based method AFINN (BAT-AFINN) on different datasets. In this figure, prediction accuracy is given on the y-axis and prediction bias measured via *PBR* is given on the x-axis. Each line gives performances for a different dataset, and each point on the line shows the accuracy and *PBR* values for a specific threshold δ. The threshold $\delta = 0$ for the rightmost point on each line and δ increases for points on the left. It is clear from this figure that standard AFINN (the rightmost point on each line) exhibits a strong bias toward positive sentiment. When AFINN is combined with BAT bias reduces gradually to zero and there is also a gradual increase in accuracy, with an increase in threshold δ. This observation confirms that AFINN's predictions have a systematic bias and changing the prediction threshold not only reduces this bias but also increases prediction accuracy.

Figure 2 shows the performance of BAT when combined with SentiStrength (BATSS). We do not report results of SentiStrength on the five Web 2.0 datasets because these datasets were used in the development of SentiStrength lexicon. Unlike the observation made for AFINN in Figure 1, SentiStrength exhibits a systematic bias toward negative sentiment (the leftmost point on each line). When SentiStrength is combined with BAT and threshold δ is decreased from zero bias reduces and accuracy increases. Again, this is a beneficial trend that is similar to that observed for BAT-AFINN.

The preceding experiments highlight that different lexicon-based methods can have different biases on different datasets. To apply BAT combined with a lexicon-based method on a given dataset, it is necessary to find the parameter/threshold δ that reduces the bias to the desired level (ideally to zero). For this purpose, some labeled examples for the dataset are needed. Once an appropriate threshold has been selected using the labeled examples, this can then be used for predicting the polarity of new examples. Given this procedure, the following two questions arise:

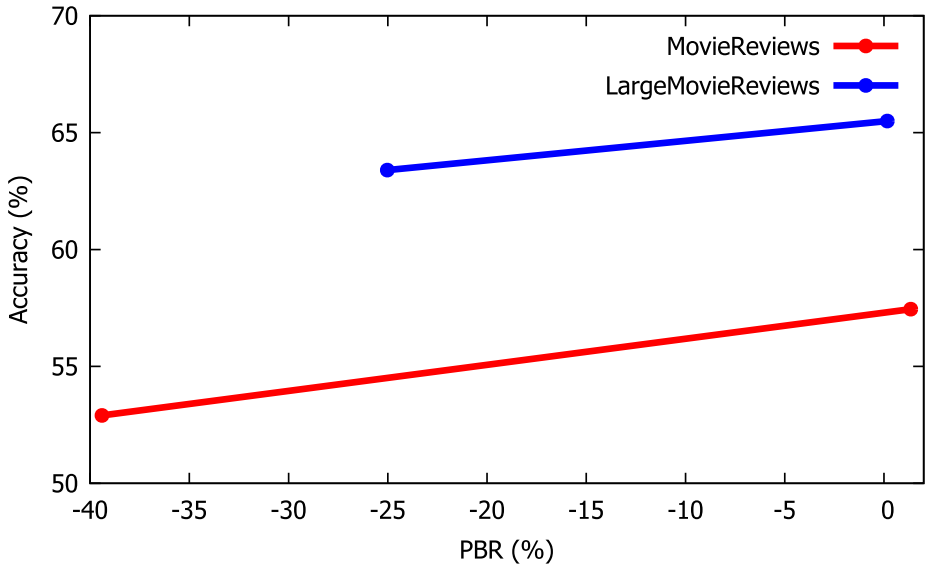

Fig. 2. Performance of BAT when combined with SentiStrength (BAT-SS).

(1) how large should the labeled dataset be to reliably tune the threshold? (2) How well would a supervised method perform when learned on such a labeled dataset?

To address these questions, we conduct additional experiments. We hold-out varying sizes of each dataset and, for each size, we learn the following: (a) the threshold δ that reduces the bias on this dataset to zero, and (b) a standard NB and LR classifier on this dataset. We then compare the performances of BAT-AFINN using the learned threshold with the learned NB and LR classifiers on the remaining portion of the respective datasets.

Figures 3 and 4 show the performance of BAT combined with AFINN and NB classifier and LR, respectively, over the test portions after learning from varying sizes of the datasets (each sub-figure shows results for one dataset). The x-axis in each sub-figure gives the training data size as a percentage of total data size and the y-axis gives the percent accuracy or *PBR*.

The following observations can be made from Figures 3 and 4:

(1) The threshold learned over the training data translates nicely to the test data by producing *PBR* values close to zero on the test data.
(2) Even when a very small training data size is used (2.5%) the performance of BAT-AFINN remains strong. In fact, there is no practically noticeable difference in *PBR* between 2.5% and 20% sizes of training data.
(3) NB and LR classifiers, on the other hand, produce varying non-zero biases. For some datasets, this bias is positive while for others it is negative.
(4) More interestingly, in the vast majority of cases, the accuracy of BAT combined with AFINN beats that for NB and LR classifiers.

6.2 Supervised Sentiment Classification

Supervised sentiment classification can use standard text classifiers like NB classifier and LR. As discussed earlier, BAT can be combined with any sentiment classifier to produce bias-aware predictions while BLR learns a bias-aware LR model for sentiment classification. We conduct two categories of experiments for evaluating supervised sentiment classification.

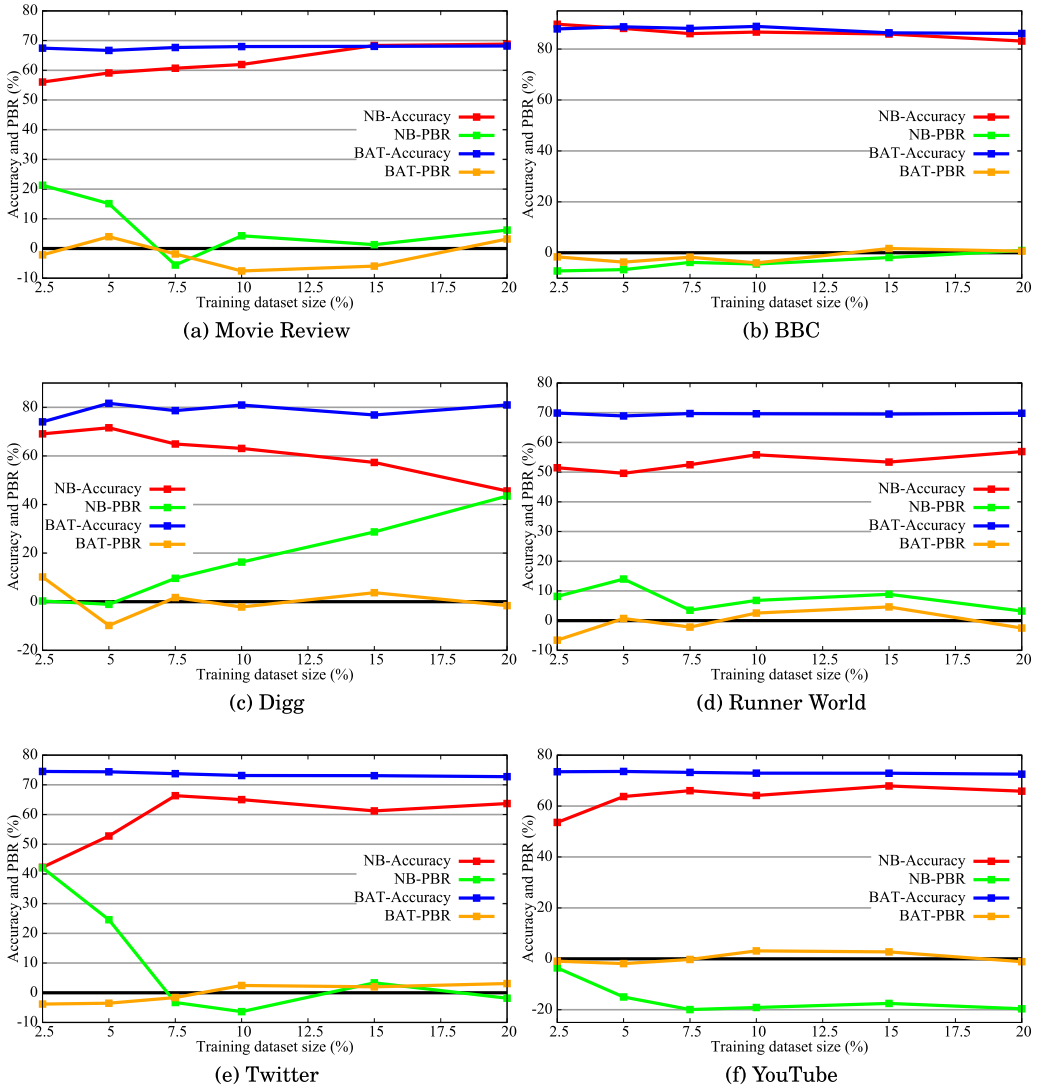

Fig. 3. Comparison of BAT-AFINN with naive Bayes classifier on training dataset of varying sizes.

First, we evaluate the performance of BAT when combined with standard logistic regression (BAT-LR) and standard NB classifier. In these experiments, NB and LR are learned over training datasets while the threshold of BAT combined with NB and LR is selected over the same datasets; performance is measured over the respective test datasets. Second, we compare the performance of BLR with standard LR. In these experiments, LR and BLR are learned from same labeled datasets and performance is measured over the respective remaining test datasets. In all cases, the test dataset is unexposed during learning, and we conduct experiments with varying training-test sizes. For performance, we report both *PBR* and accuracy percentages. The baselines for our evaluations are standard LR and standard NB classifier. In the end, we also evaluate the tradeoff between accuracy/bias and average recall and compare it with that of standard supervised sentiment classifiers.

Figures 5 and 6 show the performance comparison of the BAT-LR and BAT when combined with standard naive Bayes (BAT-NB) with standard LR and NB, respectively. The x-axis in each

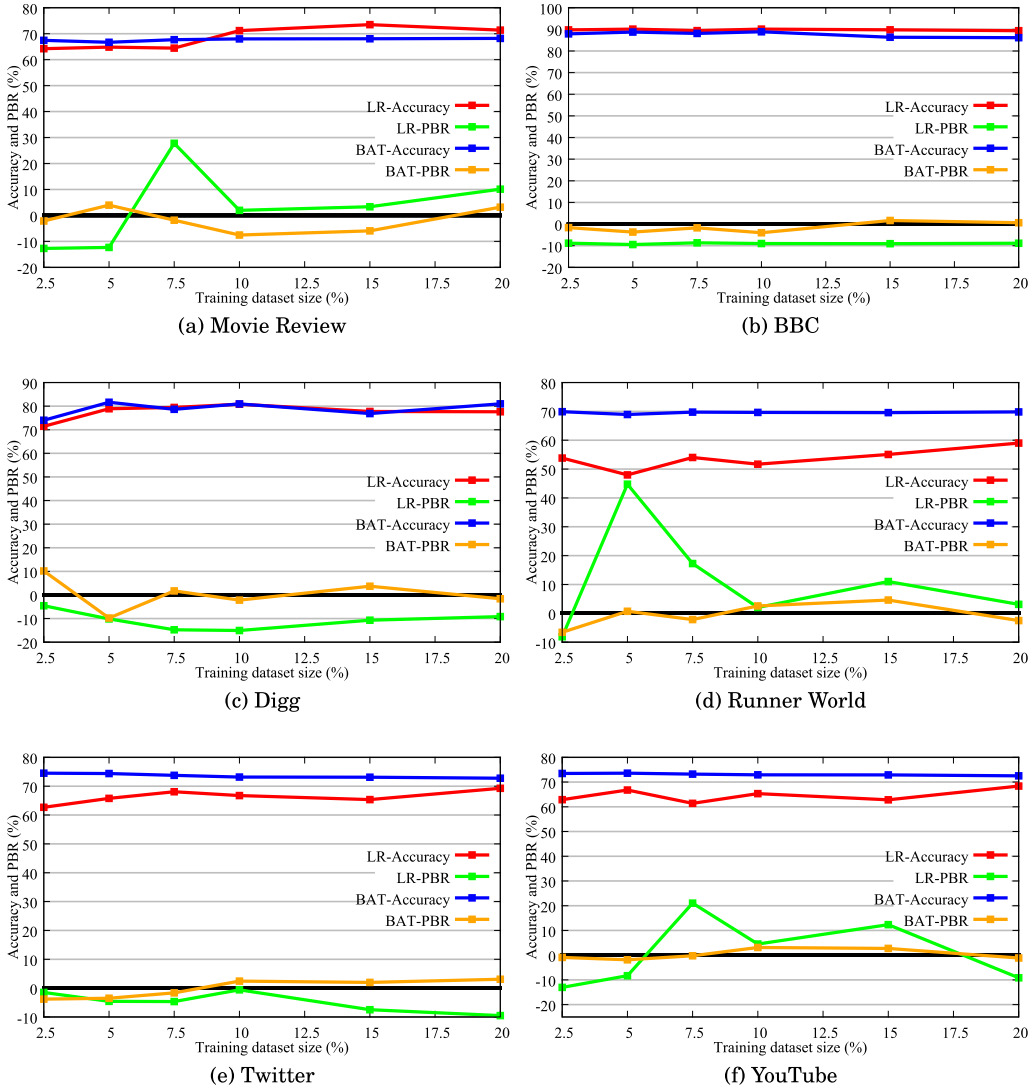

Fig. 4. Comparison of BAT-AFINN with logistic regression on training dataset of varying sizes.

sub-figure gives the training data size as a percentage of total data size and the y-axis gives the percent accuracy or *PBR*.

It is clear from these figures that BAT-NB and BAT-LR consistently outperform standard NB and standard LR w.r.t. bias and accuracy. That is, the shift in decision boundary learned by BAT over the training data (the threshold δ) makes the predictions over the test data less biased (*PBR* is close to zero) and more accurate. This result is similar to that observed for BAT combined with lexicon-based methods, hence confirming the effectiveness and generality of BAT for bias control in sentiment classification.

We now compare BLR with standard LR. Figure 7 shows the performance of standard LR with BLR over the respective remaining test portions after learning from varying training dataset sizes (shown on the x-axis of each sub-figure). In each sub-figure, prediction accuracy and prediction bias measured via *PBR* is given on the y-axis.

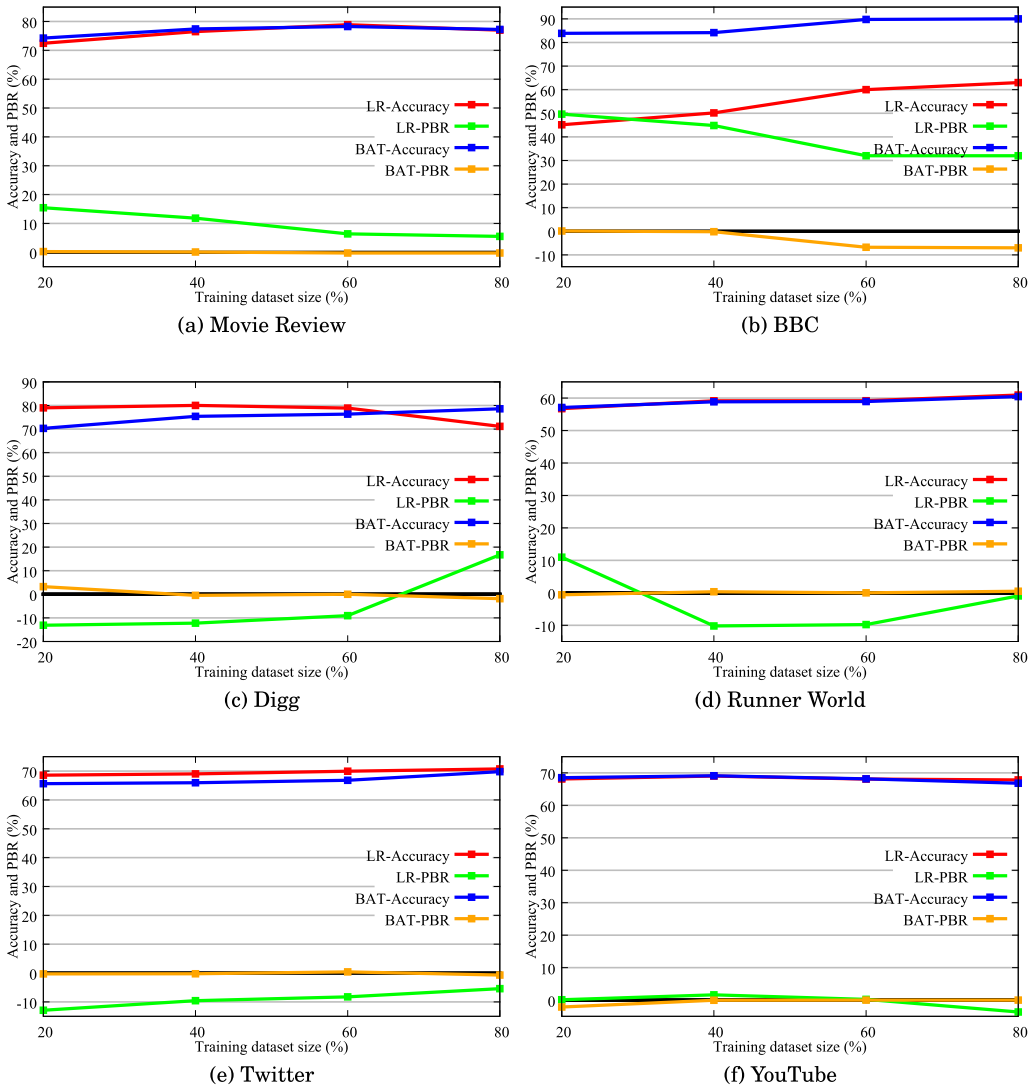

Fig. 5. Comparison of BAT-LR with standard logistic regression on training dataset of varying sizes.

The following observations can be made from Figure 7:

(1) LR produces varying non-zero biases. For some datasets, this bias is positive while for others it is negative.
(2) In some cases, LR performs extremely poorly and produces higher absolute *PBR* than accuracy, e.g., in BBC dataset.
(3) BLR consistently produces *PBR* close to zero, thus confirming that its constraint-based learning over the training data translates nicely to the test data.
(4) BLR not only reduces prediction bias measured via *PBR* toward zero but also improves the prediction accuracy by a significant amount.
(5) More importantly, in the vast majority of cases, BLR outperforms standard LR w.r.t. to both bias and accuracy.

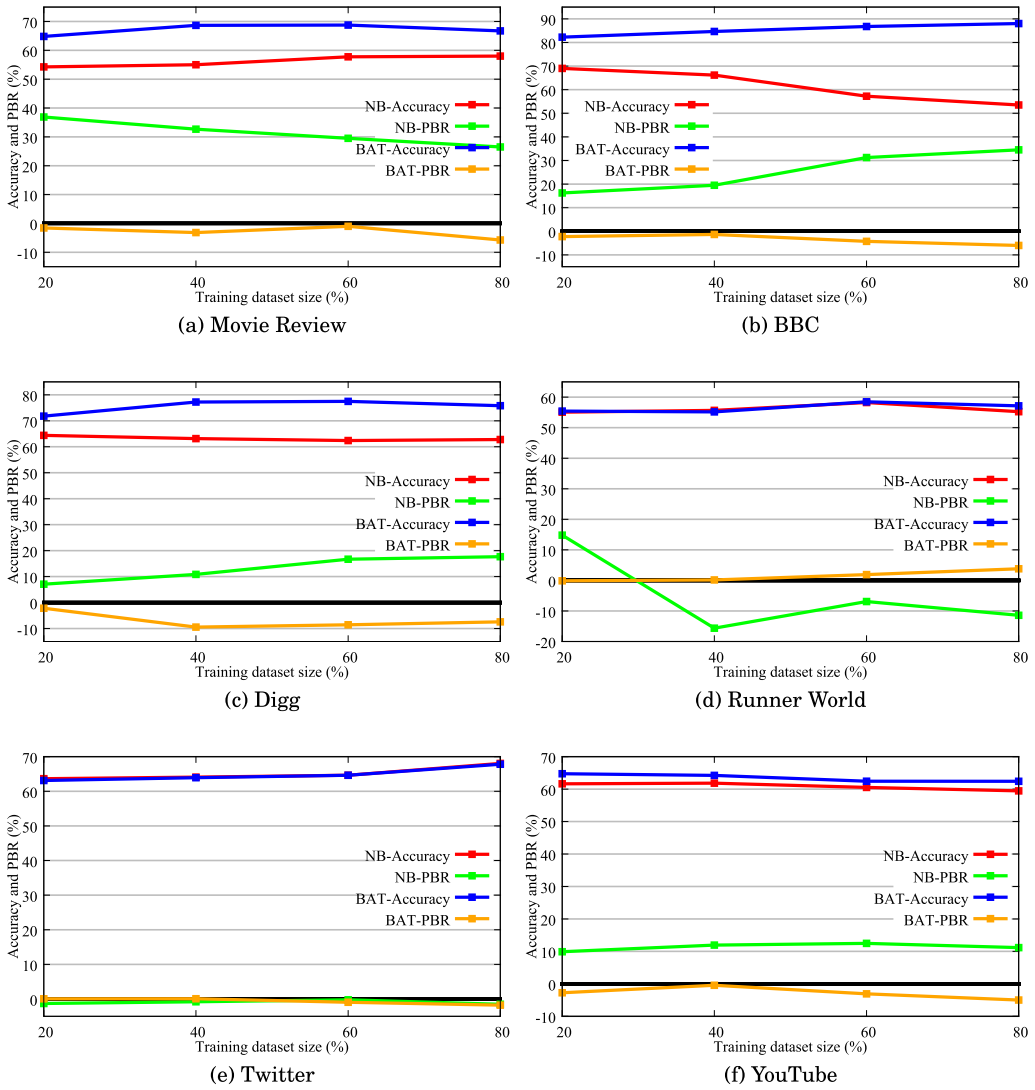

Fig. 6. Comparison of BAT-NB with standard naive Bayes classifier on training dataset of varying sizes.

Tables 5 and 6 show accuracy, *PBR*, and average recall of BAT combined with AFINN and SentiStrength and BLR on the benchmark datasets. These results are based on the same 60%–40% training-test sets used to produce results for standard supervised sentiment classifiers shown in Table 4. The key observation from these tables is that average recall remains relatively unaffected after the shift in decision boundary (BAT) or incorporation of bias constraint (BLR) in datasets having little to no class imbalance (i.e., the proportion of positive and negative documents is almost identical). On the other hand, average recall drops (although accuracy remains high or even increases) for datasets having a significant class imbalance (e.g., BBC dataset). This is because the recall for the minority class drops significantly as its errors are increased to achieve balance in errors from the minority and majority classes. However, since the cost of misclassifying positive and negative documents in sentiment analysis is usually identical higher accuracy and lower prediction bias are sufficient measures of performance for practical sentiment analysis.

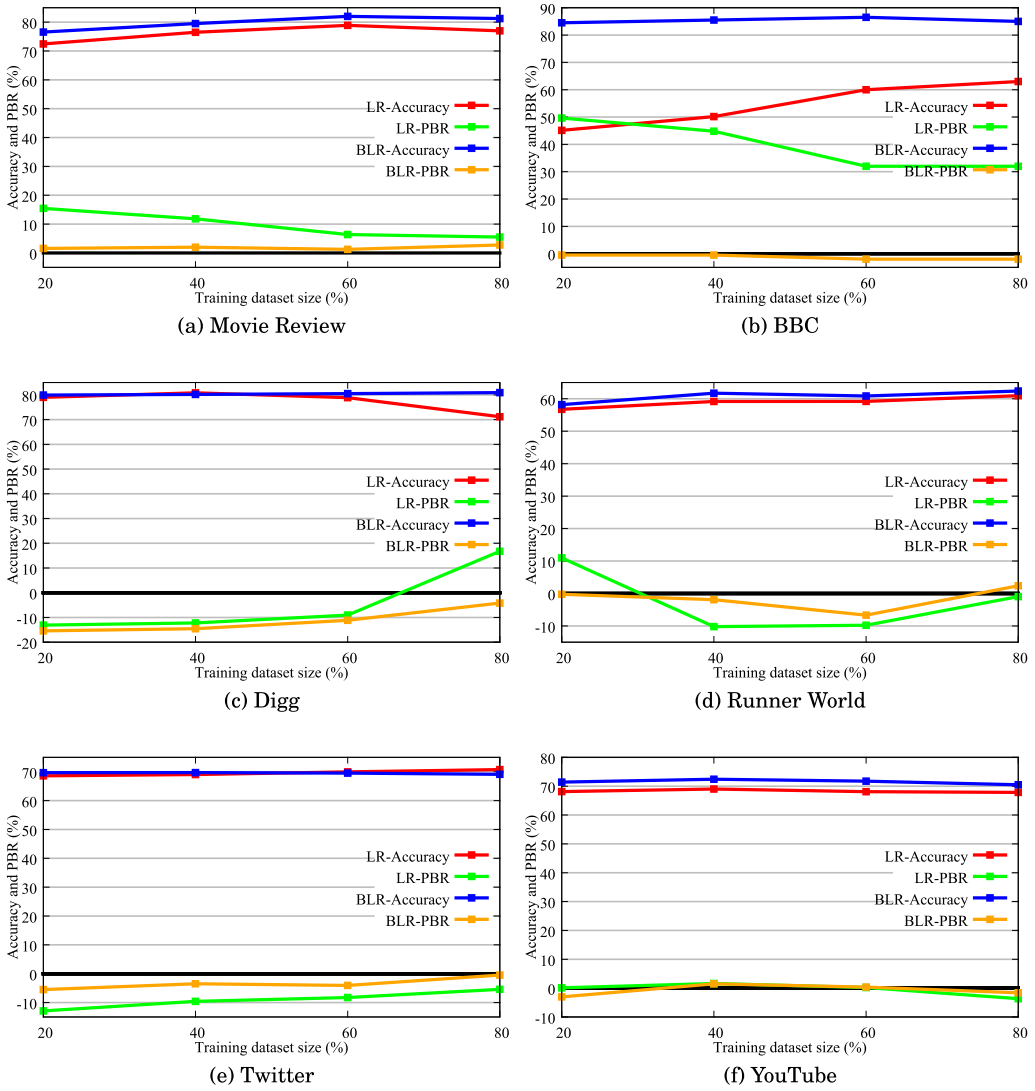

Fig. 7. Comparison of BLR with standard logistic regression on training dataset of varying sizes.

6.3 Key Points

Our evaluation of standard and bias-aware sentiment classifiers has highlighted the following key points:

(1) Imbalanced prediction errors can distort the true distribution of sentiments in the predictions. This is an important issue in practice that has not received much attention before.
(2) Existing lexicon-based and supervised sentiment classifiers can produce varying polarity biases in their predictions. This is because these methods do not control prediction bias explicitly, and in general, incorporate differing inductive biases that produce varying performances in practice.

Table 5. Accuracy, *PBR* (Bias), and Average Recall in BAT When Combined with Supervised Sentiment Classifiers—Results on 40% Test Set

	Logistic Regression			Naive Bayes		
Data	Accuracy	*PBR*	Avg. Recall	Accuracy	*PBR*	Avg. Recall
Movie Review	78.25	−0.25	78.25	68.75	−1	68.75
BBC	89.75	−6.75	55.44	86.75	−4.25	52.64
Digg	76.33	0	62.29	77.49	−8.59	55.8
Runner World	58.95	0	58.72	58.47	1.91	58.39
Twitter	66.82	0.41	61.79	64.64	−0.94	58.69
YouTube	68.15	0	68.14	62.44	−3.08	62.35

Table 6. Accuracy, *PBR* (Bias), and Average Recall in BLR—Results on 40% Test Set

	BLR		
Data	Accuracy	*PBR*	Avg. Recall
Movie Review	82.00	01.25	82.00
BBC	86.50	−02.00	57.07
Digg	79.12	−18.10	49.14
Runner World	60.86	−06.68	60.15
Twitter	69.54	−04.07	63.02
YouTube	71.75	00.37	71.75

(3) Imbalance in prediction errors does not appear to be related to the class imbalance in the dataset; existing sentiment classifiers produce varying imbalanced prediction errors on datasets with no class imbalance.
(4) Since false positives and false negatives have equal cost, accuracy and *PBR* provide a complete picture of a sentiment classifier's performance; average recall becomes useful when severe class imbalance costs in the dataset.
(5) BAT is an effective, easy-to-use method for balancing prediction errors in lexicon-based and supervised sentiment classifiers. BLR is a parameter-free balanced variant of LR for supervised sentiment classification.

7 CONCLUSION

Systematic bias in polarity predictions can jeopardize decisions based on automatic sentiment analysis of textual content. Prediction polarity bias can produce excessive false positives or false negatives that distorts the true sentiment distribution of the dataset. In this article, we study the problem of bias in polarity prediction in detail focusing on both supervised and lexicon-based methods. We define a measure, named PBR, for quantifying this bias. Subsequently, we develop two approaches for controlling bias in supervised and lexicon-based sentiment classifiers. Our first approach, called BAT, combines with any lexicon-based or supervised sentiment classifier to make it bias aware. Specifically, BAT introduces a prediction threshold that penalizes systematic errors to reduce bias and improve accuracy. BAT is simple yet effective, and can be readily used in practice. Our second approach, called BLR, is a constrained variant of standard LR that enforces

that prediction errors are balanced. BLR is an automatic bias-free supervised sentiment classifier that requires no user-specifiable parameters.

We evaluate our approaches on seven real-world datasets. BAT is combined and compared with lexicon-based methods AFINN and SentiStrength, and supervised classifiers NB and LR. BLR is compared with standard Logistic Regression. The experimental results confirm that our approaches control bias effectively while maintaining (usually improving) prediction accuracy.

This topic has much potential for further research with significant implications for practitioners. The reasons for biases and more effective solutions for their control need to be investigated. Moreover, extensive experimental evaluations and their relation to linguistics may yield additional insights into the problem.

REFERENCES

Johan Bollen, Alberto Pepe, and Huina Mao. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*.

Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

Nitesh Chawla, Kevin Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

Harrison Edwards and Amos J. Storkey. 2016. Censoring representations with an adversary. In *Proceedings of the International Conference on Learning Representations*.

Charles Elkan. 2001. The foundations of cost-sensitive learning. *In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01)*. 973–978.

Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael Friedlander. 2016. Satisfying real-world goals with dataset constraints. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2423–2431.

Pollyanna Gonalves, Matheus Arajo, Fabrcio Benevenuto, and Meeyoung Cha. 2013. Comparing and combining sentiment analysis methods. In *Proceedings of the 1st ACM Conference on Online Social Networks*. 27–38.

H. He and E. A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284.

Mohsin Iqbal, Asim Karim, and Faisal Kamiran. 2015. Bias-aware lexicon-based sentiment analysis. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. 845–850. DOI: https://doi.org/10.1145/2695664.2695759

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. 2012. Decision theory for discrimination-aware classification. In *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM'12)*. 924–929.

Faisal Kamiran, Sameen Mansha, Asim Karim, and Xiangliang Zhang. 2018. Exploiting reject option in classification for social discrimination control. *Information Sciences* 425 (2018), 18–33.

Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *Proceedings of the 3rd IEEE International Workshop on Privacy Aspects of Data Mining*. 643–650.

Shoushan Li, Guodong Zhou, Zhongqing Wang, Sophia Yat Mei Lee, and Rangyang Wang. 2011. Imbalanced sentiment classification. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. ACM, 2469–2472.

Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. 2011. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 502–510.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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, OR, 142–150. DOI: http://www.aclweb.org/anthology/P11-1015

A. Mountassir, H. Benbrahim, and I. Berrada. 2012. An empirical study to address the problem of unbalanced data sets in sentiment classification. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC'12)*. 3298–3303.

F. Å. Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages*. 93–98.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*. 271–278.

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 560–568.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management* 4, 4 (2009), 427–437.

Yanmin Sun, Mohamed S. Kamel, Andrew K. C. Wong, and Yang Wang. 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* 40, 12 (2007), 3358–3378.

Y. Tang, Y. Q. Zhang, N. V. Chawla, and S. Krasser. 2009. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 1 (2009), 281–288.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology* 29, 1 (March 2010), 24–54.

Mike Thelwall. 2013. Heart and Soul: Sentiment Strength Detection in the Social Web with Sentistrength. Retrieved from http://sentistrength.wlv.ac.uk/documentation/SentiStrengthChapter.pdf.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in short strength detection informal text. *Journal of the Association for Information Science and Technology* 61, 12 (December 2010), 2544–2558. DOI: https://doi.org/10.1002/asi.v61:12

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. 2017. Fairness constraints: Mechanism for fair classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. 962–970.

Indre Zliobaite. 2015. On the relation between accuracy and fairness in binary classification. In *Proceedings of the 2nd Workshop on Fairness, Accountability, and Transparency in Machine Learning*.

Received August 2016; revised December 2018; accepted April 2019