Neurocomputing 398 (2020) 55–63

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Document-level emotion detection using graph-based margin regularization

Ruihua Cheng[a], Jie Zhang[b,*], Pengcheng Hu[c,*]

[a] Big Data Statistics Research Center, Tianjin University of Finance and Economics, Tianjin 300222, China
[b] School of Big Data and Software Engineering, Chongqing University, Chongqing 400044, China
[c] Jilin Agricultural University, Changchun, Jilin 130033, China

ARTICLE INFO

Article history:
Received 10 December 2018
Revised 9 January 2020
Accepted 16 January 2020
Available online 23 January 2020

Communicated by Dr Erik Cambria

Keywords:
Sentiment classification
Polarity graph
Unstructured text
Margin-based classifier

ABSTRACT

Sentiment analysis aims to automatically detect the underlying attitudes that users express. For the documents with complex unstructured data, such as reviews, emojis and surveys, it is usually hard to precisely identify the real emotion. Thus, it becomes urgent, yet challenging, to develop a technique that can process and make use of the unstructured information. In this article, we consider sentiment classification for those unstructured features extracted from texts. We propose a regularization-based framework to pursue better classification performance by (1) introducing polarity shifters assembled with sentiment words to create novel bigram features and (2) simultaneously constructing a constraint graph to encode the relative polarity among unstructured features to improve the parameter estimation procedure. Under these settings, our approach can uncover the intrinsic semantic information from the unstructured text data. Theoretically, we justify its underlying equivalent connection with the standard Bayes classifier, which is ideally optimal when the sample distribution is known. Moreover, we show that our new method yields better generalization ability due to the reduced solution search space and the appealing asymptotic consistency. The superior performance from real data experiments demonstrates the robustness and effectiveness of the proposed method.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Recently, sentiment analysis has drawn tremendous attention from both industry and academia. It aims to analyze and detect the emotion information from various textual sources. Sentiment analysis has been applied in different areas, such as helping manufacturers understand customer perceptions of products, providing insights for politicians to make appropriate decisions to improve community life and predicting the trend of financial markets for investors.

In terms of sentiment classification, popular methods usually include two components: a lexicon-based module, which is applied to extract sentiment features [1–7], and a learning-based module, which includes unsupervised learning techniques in word representation [8–10] and supervised classifers [11–20]. Recently, deep learning-based methods were proposed and have been demonstrated with favorable performance for sentiment analysis [21]. Additionally, Ma et al. [22] proposed an augmented LSTM with an attention mechanism and external commonsense knowledge embedding for the aspect-based sentiment classification. Majumder et al. [23] proposed a sophisticated way to model the sentiment and sarcasm with a unified neural network. Ma et al. [24] presented a network with interactive attention. It mainly focuses on both context and target by interactively using two attention networks to learn the significant words.

In this paper, we explore the importance of polarity shifter words, which have not been comprehensively studied in previous works. It should be noted that the polarity shifters are widely used by people to strengthen or weaken the opinion towards a specific topic. Different polarity shifters have very different impacts on the word polarities and could even negate the sentiments. For the documents with complex unstructured data, such as reviews, emojis, and surveys, it is usually hard to precisely locate the real emotion. Therefore, it is necessary, yet challenging, to model the effect of polarity shifter-decorated sentiments and develop a technique that can process and make use of the unstructured information.

In order to solve the above problems, we first divide the polarity shifters into three groups, polarity diminishers, polarity intensifiers and polarity negators, and build novel features based on the sentiment words and polarity shifters. Then, we construct

* Corresponding authors.
E-mail addresses: r.cheng665@gmail.com (R. Cheng), j.zh2099@gmail.com (J. Zhang), hu-xue@163.com (P. Hu).

https://doi.org/10.1016/j.neucom.2020.01.059
0925-2312/© 2020 Elsevier B.V. All rights reserved.

Table 1
Examples of polarity shifters.

Part Of Speech	*Intensify polarity*	*Diminish polarity*
Adverbs	definitely, very, extremely	somewhat, barely, less
Adjectives	bright, authentic	worthless, weak, rough
Verbs	ensure, improve, assure	fail, discourage
Nouns	benefit, favor	disaster, bankruptcy

a constraint graph to fully capture the partial relationships among these features based on their relative sentiment polarities. Finally, the newly extracted polarity shifter-based features and the constructed constraint graph are fed into the margin-based classifier with parallel classification functions to predict the sentiment scores. Theoretically, we justify that the proposed approach can improve predictive capacity by reducing the complexity of parameter space. Moreover, under this design, we see the compelling performance on the real applications.

The contributions of this paper are summarized as follows:

- We introduce polarity shifters into the sentiment study and combine them with opinion words to create novel features. New transformations are proposed to reduce the dimensionality and overcome the data sparsity issue.
- We explore the dependence and structural information for those unstructured features and encode them into a polarity-based partial order graph to impose constraints on the parameters during the model training process and therefore improve the model performance.
- We justify that the proposed approach is equivalent to the standard Bayes classifier in optimality. Moreover, we derive finite-sample error bound and show that adding graph-based constraints into the model training process could improve the prediction accuracy due to the reduced solution search space and the appealing asymptotic consistency.
- We conduct comprehensive experiments to demonstrate the effectiveness and robustness of our proposed method.

The paper is organized as follows. Section 2 introduces the background and related works. Section 3 presents the proposed method. Section 4 demonstrates the theoretical properties and gives the error bound for our proposed method. Section 5 illustrates its superior performance via real data experiments. Finally, Section 6 concludes this article.

2. Background and related works

In this section, we review the existing literature about sentiment analysis including the contextual polarity shifter, semantic word similarity, and some other popular machine learning methods.

2.1. Contextual polarity shifter

The polarity of a word is the affective level and usually represents a positive or negative valence/sentiment of that word. Polarity shifter is the word that can strengthen or weaken the degree of that polarity. Some example words that are used to intensify or diminish polarities are illustrated in Table 1.

2.2. Semantic word similarity

The polarities of two words are usually disparate, as the words may have different sentiment attitudes and are decorated with different polarity shifters. Therefore, it is necessary to apply a metric to quantitatively measure the word similarity. Two popular methods, namely, the Latent Semantic Analysis (LSA) [25,26] and

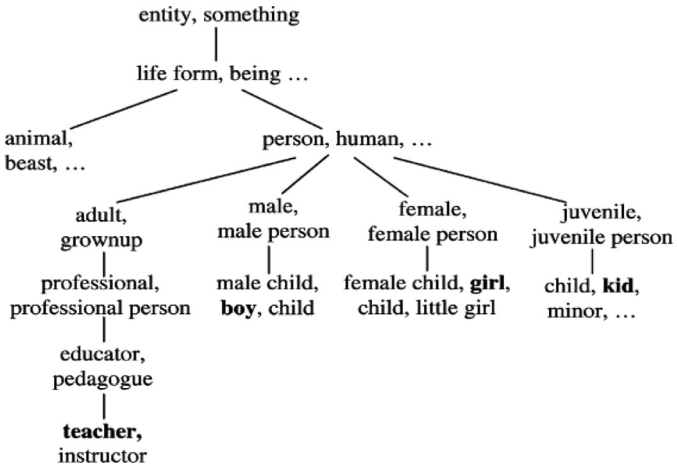


Fig. 1. The semantic hierarchy of WordNet.

the Pointwise Mutual Information (PMI) [27], are considered. The LSA assumes that the senses of words are similar when they are in the same window, while the PMI is a co-occurrence metric computed based on the AltaVista's NEAR operator. Specifically, the PMI between two words is defined as $\mathrm{PMI}(word_1, word_2) = \log(p(word_1, word_2)/p(word_1)p(word_2))$, which aims to normalize the probability of co-occurrence of the two words.

Researchers have conducted a series experiments to systematically compare the two metrics from different aspects [28], and the LSA was proved to be better than PMI in many cases. In addition, the performance of LSA could be further improved when combined with the WordNet, which groups nouns, verbs, adjectives and adverbs into sets of synonyms called synsets via semantic relations [29]. The final similarity score between two words is given by

$$M(w_1, w_2) = M_L(w_1, w_2) + 0.5\exp(-\beta S(w_1, w_2)), \tag{1}$$

where M_L is the LSA metric, $S(w_1, w_2)$ is the shortest distance between two words calculated based on the semantic hierarchy from WordNet. Fig. 1 shows an example of the semantic hierarchical structure of WordNet. Note that some words are omitted in Fig. 1 and replaced with "$\cdots$" to save space [30].

2.3. Global vectors for word representation (glove)

Global Vectors for Word Representation (GloVe), proposed by Pennington et al. [31], is an unsupervised learning algorithm for learning vector space representations of words. It could efficiently leverage word distribution information by training only on the nonzero elements in a word-word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus. The word vector produced by the model contains the meaningful word substructure information and could be used to quantify the word similarities, as evidenced by its superior performance in a series of tasks [31].

2.4. Support vector machine (SVM)

Support Vector Machine (SVM) is a popular machine learning method for classification, regression and other learning tasks [32,33]. It aims to find the maximum-margin hyperplane to classify data points into different classes and is believed to outperform classical machine learning techniques especially for high-dimensional datasets [33–35].

Two-class SVM: Let $(X_1, y_1), \ldots, (X_n, y_n)$ denote n observations and $y_i \in \{-1, 1\}$ represents the class label. The SVM is looking for a decision surface that is maximally far away from any data point.

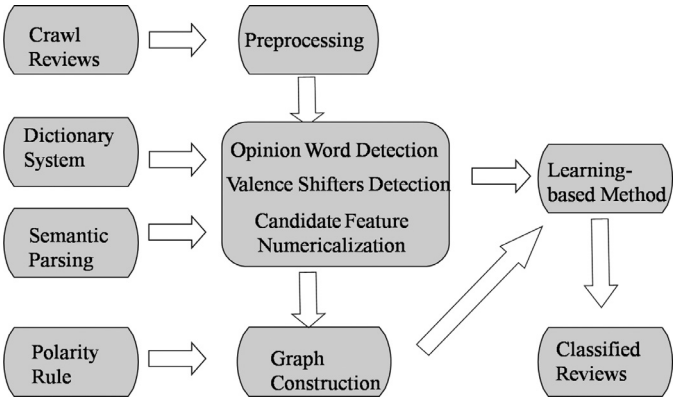


Fig. 2. Sentiment analysis framework.

The decision function is defined as

$$f(x) = sign(\boldsymbol{\beta}^T \phi(x) + \beta_0), \tag{2}$$

where $\phi(.)$ can be a transformed input from kernel trick.

Multi-class SVM: With k classes, there are $k(k-1)/2$ models used for each pair of classes. One popular method is to apply a voting strategy as described in [36]: if the (i, j)th classifier implies that x belongs to ith label rather than the jth label based on Eq. (2), the vote of the ith class is counted up by one; Otherwise, the vote of the jth class is counted up by one. Sum all the votes together and the class with the largest vote number is selected as the prediction.

2.5. Random forest

Random forest (RF) is a combination of tree predictors and has consistently lower generalization error than the decision tree [37]. RF feature selection is a combination of variable subset selection and bootstrapping with variable ranking. The main idea is to generate a vast number of decision trees, which are used to determine the most popular variables based on performance. Similarly, to classify a new sample, the input vector is passed to the trees in the forest. Each tree yields a classification label, referred to as a "vote" for that class. The class with the largest number of votes across all the trees is chosen as the final prediction.

2.6. Long short-term memory

Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture that achieves state-of-the-art performance on important NLP tasks including language modeling [38–40], speech recognition [41,42], and machine translation [38,43]. Compared with traditional recurrent neural networks (RNN), LSTM can overcome the vanishing gradient problem and is more effective at capturing long-term dependencies among words, sentences and documents [41,44]. Recently, bi-directional LSTM models have been applied to model the dependencies between adjacent words within sentences and achieved significant improvements over state-of-the-art recurrent neural network baselines [45,46]. In this paper, the popular LSTM-based methods [45,46] are considered as competing methods and comprehensive experiments are conducted to fully evaluate the proposed method.

3. Methodology

Our proposed sentiment analysis framework is presented in Fig. 2. Feature extraction, feature polarity graph construction and the learning-based classifier are three core components in this framework.

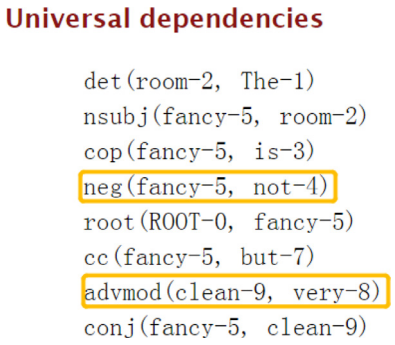


Fig. 3. An example of the parsing result.

3.1. Feature extraction

For each review in the real dataset, it contains a review text and a rating score. We first break those sentences of the review text into individual words without considering the order. Then we use the sentiment dictionary to detect the opinion words, that is, the words with positive or negative sentiment. We also apply a polarity shifter dictionary to detect the polarity shifter words.

Sentiment and Polarity Shifter Dictionaries. We refer to the classical lexicon system[1] collected by Hu et al. [47] as the sentiment dictionary (SD), which contains approximately 6800 sentiment words. General Inquirer system contains both polarity shifter words and sentiment words. We treat the polarity shifter words (i.e., 318 diminishers and 578 intensifiers and negators) from the General Inquirer system as the polarity shifter dictionary (PSD) and the 1915 sentiment words as another sentiment dictionary (GID). The preprocessed reviews are matched with the dictionary system to detect sentiment words as well as polarity shifters related to these words.

Semantic Parsing. We refer to the Stanford parser [48] for semantic parsing. Fig. 3 shows how the parser works on an example sentence: *"The room is not fancy, but very clean"*[2]. The parser detects and assigns *fancy* the *neg* dependency type. Similarly, *very* increases the sentiment degree of *clean*. Therefore, the parser identifies it as the dependency type, *advmod*.

Feature Integration. We define unigrams as the individual opinion words, and bigrams as the combination of polarity shifters with unigrams, such as *very good* or *really good*. Naively treating all the unigrams and bigrams as candidate features will lead to extremely sparse representations and incur the ultra-high dimensionality. In this paper, we propose to conduct the feature integration operation to overcome the explosive feature space problem. Firstly, all unigrams are included as features. Secondly, in contrast to adding all bigrams as new features, for each distinct unigram *word*, we add three extra unified bigrams $plus_{word}$, $minus_{word}$, and not_{word} as features, which represent the sentiment strengthened, weakened, and negated features for that unigram. Note that, it covers all bigrams and, after this transformation, the bigrams are not strict phrases, yet maintaining the general semantic direction. Thirdly, one-hot encoding method is utilized to vectorize the new features. The unigrams are encoded directly and the bigrams are fed into the Stanford Parser for checking the polarity shifter and dependency to determine the feature (i.e., $plus_{word}$, $minus_{word}$, not_{word}) and encoded accordingly.

[1] The lexicon is accessible at https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

[2] A demo is available at http://nlp.stanford.edu:8080/parser/.

3.2. Feature polarity graph construction

We construct a directed graph in which nodes are the extracted features (i.e., unigrams and unified bigrams obtained via feature integration) and the edges encode the polarity strength relationship between each pair of nodes. Specifically, edge $A \rightarrow B$ means that the feature B has a higher polarity degree than feature A. With this setup, we are able to introduce appropriate constraints on the features during the coefficient estimation procedures.

The key to construct the polarity graph is to calculate the polarity for each unigram and unified bigram and determine the direction of each edge. We propose a novel procedure to accurately estimate the polarity for each feature. We define the polarity of each phrase/word as the normalized semantic distance between this phrase/word with the word *good* and with the word *bad*, that is,

$$Pol(w) = (M(good, w) - M(bad, w))/M(good, bad), \tag{3}$$

where $M(.,.)$ is calculated via Eq. (1). For the unigram node in the graph, we calculate its polarity score directly via Eq. (3). As it is hard to directly measure the polarity of unified bigrams $plus_{word}$ and $minus_{word}$, we propose to compute the values by averaging the polarities of the intensified and diminished phrases, respectively. Let *ins* and *dim* denote a polarity intensifier and diminisher, respectively, from PSD. Taking the sentiment word *nice* as an example, the polarity of unified bigram $plus_{nice}$ is computed by taking the expectation of the polarities of those related phrases which are composed by polarity intensifiers and word *nice*. That is, $Pol(plus_{nice}) = E(Pol(ins_{nice}))$. Likewise, we have $Pol(minus_{nice}) = E(Pol(dim_{nice}))$. Because negating a word simply reverses the original sentiment, we have $Pol(not_{nice}) = -Pol(nice)$. With above calculations, both unigram and unified bigram nodes can be scored and we can determine the edge direction based on their partial relationships. For example, we have $not_{nice} \rightarrow minus_{nice} \rightarrow nice \rightarrow plus_{nice}$.

3.3. Learning-based classifier

We propose to combine our new feature-extraction procedure with the state-of-the-art margin-based ordinal classifier (MBC) to obtain better performance. We call the proposed method fMBC.

Margin-based Ordinal Classifier. MBC pursues the large margin hyperplane and could separate the classes more precisely than other methods [49]. More importantly, MBC takes a directed graph as input during the optimization procedure to impose a partial ordering constraint on the feature coefficients. It could reduce the solution search space, and therefore optimize the estimated coefficients.

Suppose that there are M ordinal categories and $y_i \in \{1, \ldots, M\}$. Let $\mathbf{g} = (g_1, \ldots, g_{M-1})^T$ be $M-1$ classification functions, where $g_m(\boldsymbol{x}) = \boldsymbol{\omega}^T\boldsymbol{x} + \omega_{0m}$. Note that $\boldsymbol{\omega}$ is shared across the $M-1$ prediction functions, while the intercepts ω_{0m} are different. Let $S = \{1, \ldots, M-1\}$. The decision function is defined as follows

$$\psi(x) = \begin{cases} M & \text{if } g_m(\boldsymbol{x}) < 0 \text{ for all m} \in S \\ \min\{m : g_m(\boldsymbol{x}) \geq 0\} & \text{otherwise.} \end{cases} \tag{4}$$

The objective function can be formulated as

$$\min_{\boldsymbol{\omega}} \left(\frac{||\boldsymbol{\omega}||^2}{2} + \lambda \sum_{i=1}^{n} \sum_{m=1}^{M-1} \nu_{im} \right),$$
$$\text{such that } \boldsymbol{\omega} \in \mathcal{W}, 1 - (\boldsymbol{x}_i^T\boldsymbol{\omega} + \omega_{0m})\text{sign}(m - y_i) \leq \nu_{im},$$
$$\nu_{im} \geq 0, \omega_{01} \leq \omega_{02} \leq \ldots \leq \omega_{0,M-1}, \tag{5}$$

where λ is the tuning parameter, ν quantifies the misclassification degree. In addition, $\mathcal{W} = \{\boldsymbol{\omega} : W\boldsymbol{\omega} \geq 0\}$ specifies the coefficient constraints, where W is a matrix carrying the partial ordering information of features. For example, if the feature $x^{(j)}$ directs to the feature $x^{(j')}$ in the constructed polarity graph (Section 3.2), it indicates $\omega_{j'} \geq \omega_j$. Correspondingly, a row in W is assigned with the j'th entry being 1, the jth entry being -1 and the rest being 0 to ensure that the dot product of this row and vector $\boldsymbol{\omega}$ is $\omega_{j'} - \omega_j$. Therefore, $W\boldsymbol{\omega} \geq 0$ enforces the polarity constraints among the features.

For the optimization, we derive the dual form of the original objective function by adding Lagrange multipliers and apply a quadratic programming procedure to find the optimal parameters. In the following paragraphs, we explore the theoretical properties of the proposed method.

4. Properties of the proposed procedure

Inspired by support vector machine to estimate $\text{sign}\{p(\boldsymbol{x}) - 1/2\} > 0$ by $\{\hat{g}(\boldsymbol{x}) > 0\}$, we study the optimal performance of fMBC. The general form is to minimize the objective criterion

$$O(\boldsymbol{\omega}) = \lambda_n J(\boldsymbol{\omega}) + \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(M-1)} \sum_{m=1}^{M-1} L(\text{sign}(m - y_i) g_m(\boldsymbol{x}_i))$$
$$\text{subject to } \boldsymbol{\omega} \in \mathcal{W}, \omega_{01} < \omega_{02} < \cdots < \omega_{0,M-1}, \tag{6}$$

where $J(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_2^2/2$ denotes the geometric margin and $L(s) = \min((1-s)_+, 1)$ is the classification loss.

Let $Q(\boldsymbol{\omega}, \boldsymbol{Y})$ denote the cost function $(M-1)^{-1}\sum_{m=1}^{M-1} L(\text{sign}(m - \boldsymbol{Y})g_m(\boldsymbol{X}))$ in (6). Considering that each class might have different cost in M classes, we define the cost corresponding to each class as $\boldsymbol{\tau} = \{\tau_m\}$ with $0 \leq \tau_m \leq 1, m = 1, \ldots, M$. In that case, (6) is the reduced version of weighted objective function when $\tau_m = 1/2$. Seeking the optimal $\boldsymbol{g}$ to minimize (6) can obtain the sign of $\{\hat{\boldsymbol{g}}_{\boldsymbol{\tau}}(\boldsymbol{X})\}$, which actually is a Bayes estimator, denoted by $\bar{\boldsymbol{g}}_{\boldsymbol{\tau}}(\boldsymbol{X})$. Define $U(\boldsymbol{Y}) = \tau_m$ if $\boldsymbol{Y} = m$. It is known that the second term of the weighted objective function goes to $E[U(\boldsymbol{Y})Q(\boldsymbol{\omega}, \boldsymbol{Y})]$. Theorem 1 gives the optimization basis of fMBC.

Theorem 1. *Seeking the optimal $\boldsymbol{g}$ to minimize (6)with $L(s)$ outputs the Bayes decision rule, i.e., for $m = 1, \ldots, M-1$,*

$$\bar{\boldsymbol{g}}_{\boldsymbol{\tau}}(\boldsymbol{X}) = \text{sign}\left\{ \sum_{k=1}^{m} P(\boldsymbol{Y} = k)\tau_k - \sum_{k=m+1}^{M-1} P(\boldsymbol{Y} = k)\tau_k \right\}.$$

Theorem 1 shows that our method can estimate Bayes classifier $\bar{\boldsymbol{g}}_m(\boldsymbol{x})$ by using $\text{sign}\{\hat{\boldsymbol{g}}_m(\boldsymbol{x})\}$ estimated by fMBC. Bayes classifier is (ideally) optimal and available when the sample distribution is known [50]. The proposed method is believed to be able to replace the Bayes classifier since it is feasible as the sample distribution is unknown.

We now study the complexity of space $\mathcal{G}$ through measuring its bracketing metric entropy, $H(\cdot, \mathcal{G})$. We denote it by the cardinality logarithm of the minimum size of ϵ-bracketing. Define $N(\epsilon, m) = \{g_1^l, g_1^u, \ldots, g_m^l, g_m^u\} \subset \mathcal{L}_2$, which qualifies $\max_{1 \leq j \leq m} \|g_j^l - g_j^u\|_2 \leq \epsilon$, where $\|g\|_2 = (\int g^2 d_\mu)^{1/2}$ and for $g \in \mathcal{G}$, there exists a j such that $g_j^l \leq f \leq g_j^u$ almost everywhere with probability 1. $H(\epsilon, \mathcal{G})$ can be denoted by $\log(\min\{m: N(\epsilon, m)\})$.

Let $\mathcal{G}_{\mathcal{W}} = \{\mathbf{g} = (g_1, \ldots, g_{M-1}) : g_m(\boldsymbol{X}(s)) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{X} + \omega_{0m}, \boldsymbol{\omega} \in \mathcal{W}, \omega_{01} < \omega_{02} < \cdots < \omega_{0,M-1}\}$ represent the parameter space. Define the distance of margin loss between parameters as $R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) = E(Q(\boldsymbol{\omega}, \boldsymbol{Y})) - E(Q(\boldsymbol{\omega}_0, \boldsymbol{Y}))$, and the variation as $D(\boldsymbol{\omega}, \boldsymbol{\omega}_0) = Var(Q(\boldsymbol{\omega}, \boldsymbol{Y}) - Q(\boldsymbol{\omega}_0, \boldsymbol{Y}))$. For positive constant d_1, let $\mathcal{G}_{\mathcal{W}}(\mu, \eta) = \{\boldsymbol{g} \in \mathcal{G}_{\mathcal{W}} : Q(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \mu, J(\boldsymbol{\omega}) \leq d_1\eta\}$, where $Q(\boldsymbol{\omega}, \boldsymbol{\omega}_0)$ measures the cost distance between two parameters. We next give the complexity bound of local parameter space.

Theorem 2. *For any $0 < t < \epsilon \leq 1$, $H(t, \mathcal{V}_{\mathcal{W}}(\epsilon)) \leq \boldsymbol{O}(|A|M\log(c'\epsilon/t))$ for some constant c', where $\mathcal{V}_{\mathcal{W}}(\epsilon) =$*

$\{\mathcal{G}_{\mathcal{W}} \cap \{R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \le \epsilon^2\}\}$ *is a local parameter space and* $|A|$ *is the number of nonzero* $\boldsymbol{\omega}$.

As the complexity of parameter space can be upper bounded, we investigate the asymptotic property of the proposed method. Based on the measured complexity, neighbour smoothness and bounded moment generating function, we develop the following assumptions.

Assumption 1 (Neighbour Smoothness)**.** For some positive constants θ, d_2, assume that

$$\sup_{\mathbf{g} \in \mathcal{G}_{\mathcal{W}}(\mu,\eta)} D(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \le d_2 \mu^2 \eta^{\theta}.$$

Assumption 2 (Moment Generating)**.** Assume that

$$\frac{Q(\boldsymbol{\omega}, y_i) - Q(\boldsymbol{\omega}_0, y_i)}{T(y_i)} \le |\boldsymbol{\omega}(\boldsymbol{x}_i) - \boldsymbol{\omega}_0(\boldsymbol{x}_i))|,$$

where $\sup(\mathrm{E}_i \exp(T(y_i)))$ is finite.

Assumption 3 (Complexity)**.** Assume that there are positive constants d_3 and d_4, satisfying that for $\delta > 0$

$$\sup_{i \ge 1} \phi(\delta, \eta) \le d_4 \sqrt{n},$$

where, $\phi(\delta, \eta) = \int_{V_1}^{V_2} H^{1/2}(\kappa, \mathcal{G}_{\mathcal{W}}(\infty, \eta)) d\kappa / V_1$ with $V_1 = d_3(\delta^2 + \eta)^{1+\theta/2}$ and $V_2 = d_4 \lambda_n (\delta^2 + \eta)$.

Theorem 3. *Based on Assumption 1–3, there are positive constants* d_5, d_6, *satisfying that for any positive* δ,

$$P^*\left(\sup_{R(\boldsymbol{\omega},\boldsymbol{\omega}_0) \ge \delta^2} (O(\boldsymbol{\omega}) - O(\boldsymbol{\omega}_0)) \ge -\delta^2/2\right) \le 7exp(-d_5 n \min(\lambda_n^2/\delta^2, \lambda_n)).$$

From Theorem 3, with more algebra, we can obtain

$$P^*\left(\sup_{R(\boldsymbol{\omega},\boldsymbol{\omega}_0) \ge \delta^2} (O(\boldsymbol{\omega}_0) - O(\boldsymbol{\omega})) \ge \delta^2/2\right) \ge 1 - 7exp(-d_5 n \min(\lambda_n^2/\delta^2, \lambda_n)),$$

which implies that, compared with true parameters $\boldsymbol{\omega}$'s, those incorrectly specified $\boldsymbol{\omega}_0$'s yields larger overall loss and the difference is greater than a threshold in probability 1 as n goes infinite. With this information, the upper error bound of the estimates for the proposed method can be obtained as follows.

Corollary 1. *Under Assumptions 1–3, the estimated* $\hat{\boldsymbol{\omega}}$ *satisfies that for* $\xi_n > 0$ *with* $\xi_n^2 = \max(\lambda_n, \delta_n^2)$,

$$P(R(\boldsymbol{\omega}_0, \hat{\boldsymbol{\omega}}) \ge \xi_n^2) \le 7\exp(-d_5 \xi_n^2 n)$$

For the proof of Theorem 3 and Corollary 1, as the assumptions above satisfy the conditions of Theorem 2 and its Corollary in the paper [51], it is straightforward to obtain the theory through clear logic demonstration, which thus is not detailed here. As observed, the reduction of complexity of parameter space $\mathcal{G}_{\mathcal{W}}$ reduces the error bound, which reflects the significant impact of coefficient constraints. By Corollary 1 and assumption 3, the best possible rate for the estimate can be achieved in probability when δ_n is minimized. Actually, balancing λ_n and optimization power is a trade-off. Corollary 1 gives insights into choosing λ_n.

The analytical method we use to obtain the probability of upper error bound is different from previous studies. Classically, there are two schemes in learning theory mainly applied for classification. First, Vapnic–Chervonenkis size of the decision functions and the practical training error are used to quantify the bound of generalization error. Assuming that the real labels are given in a finite set C with $card(C) = K$, the fundamental upper bound form is described as follows,

$$e(\hat{g}) \le \inf_{\epsilon > 0} \frac{(K(2K-1))}{n} \left(\sum_i^n I(\hat{g}(\boldsymbol{x}_i) y_i \le \epsilon) + \frac{\lambda(\mathcal{G}, \epsilon)}{n^{1/2}} \right),$$

where $\lambda(\mathcal{G}, \epsilon)$ is associated to the ϵ-entropy of the observations misclassified. The $e(\hat{g})$ for a given training sample size is upper bounded by properly opting ϵ [52]. But the classifier localization is resulted from random bound, failing to discriminate the classifiers' disparity. Mammen et al. [53] introduce another study, which is intended to balance between the training sample error and the space size of classification functions, and conclude that these two studies are reciprocal with each other. However this study is not appropriate for our method, since it only focuses on those classifiers in minimizing the practical error upon closed sets.

5. Experiment

In this section, we first consider three learning-based classifiers, namely, margin-based ordinal classifier (MBC) [49], Support Vector Machine (SVM) [34] and Random Forest (RF) [37]. We systematically compare the usefulness and effectiveness of the proposed feature-integrated method with the original version for these methods. Specifically, we denote them as {MBC, SVM, RF} and {fMBC, fSVM, fRF} for the traditional methods and the proposed feature-integrated methods, respectively. In addition, we also apply the popular deep learning-based methods as competing methods to fully evaluate the performance of proposed method. The detailed setup for each method is described as follows:

- **MBC:** MBC method only uses the unigrams as features and is integrated with the polarity graph constructed based on the individual words.
- **SVM:** Similar to the setup of MBC, the individual words are used as features, but without using the feature polarity graph.
- **RF:** RF uses the same setup as SVM.
- **fMBC:** For fMBC, both the unigrams and the extended unified bigrams are fully leveraged, together with the partial ordering constraint graph, which is constructed based on the relative polarities[3].
- **fSVM:** fSVM takes both the unigrams and the extended unified bigrams as input features. The constarint graph is ignored as it is not applicable to the SVM classifier.
- **fRF:** fRF uses the same setup as fSVM.
- **GloVe-SSA**: GloVe model was trained on a large corpus, Wikipedia 2018[4], to obtain word-level embeddings (200-dimensional vectors). The word embeddings are used as initial inputs to the structured self-attentive sentence model, proposed by Lin et al. [46], for sentiment analysis.
- **GloVe-CLNS**: GloVe-CLNS uses the same pre-trained word embeddings as GloVe-SSA. The context-sensitive lexicon-based method, proposed by Teng et al. [45], is utilized for sentiment analysis.

The testing error is calculated as follows:

$$T = \frac{1}{n_t} \sum_{j=1}^{n_t} l(y_j, g(\boldsymbol{x}_j)),$$

where n_t is the sample size and $l(y_j, g(\boldsymbol{x}_j))$ represents the classification error function. For MBC, it can be expressed in a specific

[3] The constraints is described as $\boldsymbol{\omega} \in \{\boldsymbol{\omega}: W\boldsymbol{\omega} \ge 0\}$, with matrix W representing the relational polarity graph.

[4] https://meta.wikimedia.org/wiki/Data_dump_torrents#English_Wikipedia.

Table 2
Average testing error and its standard deviation (in parentheses) on 100 replications for competing methods on different datasets. MBC, SVM, RF, fMBC, fSVM and fRF are trained under two different dictionaries (SD and GID), while GloVe-SSA and GloVe-CLNS use the GloVe model to generate word embeddings.

Method	IMDB		Yelp		MPQA	
	SD	GID	SD	GID	SD	GID
MBC	0.401 (.0003)	0.427 (.0005)	0.258 (.0002)	0.261 (.0004)	0.378 (.0003)	0.391 (.0002)
SVM	0.468 (.0006)	0.484 (.0008)	0.349 (.0005)	0.362 (.0005)	0.462 (.0004)	0.472 (.0003)
RF	0.462 (.0007)	0.489 (.0010)	0.351 (.0002)	0.360 (.0003)	0.471 (.0004)	0.479 (.0002)
fMBC	**0.379** (.0005)	**0.405** (.0007)	**0.221** (.0004)	**0.249** (.0003)	**0.352** (.0005)	**0.366** (.0004)
fSVM	0.426 (.0004)	0.451 (.0008)	0.303 (.0004)	0.325 (.0008)	0.410 (.0004)	0.425 (.0006)
fRF	0.429 (.0005)	0.457 (.0003)	0.313 (.0007)	0.327 (.0005)	0.415 (.0003)	0.438 (.0003)
GloVe-SSA	0.430 (.0005)		0.329 (.0004)		0.405 (.0005)	
GloVe-CLNS	0.428 (.0004)		0.336 (.0004)		0.408 (.0003)	

form

$$l(y_j, g(\boldsymbol{x}_j)) = \frac{1}{M-1}\sum_{m=1}^{M-1} I\{g_m(\boldsymbol{x}_j)\text{sign}(m - y_j) \le 0\},$$

where $g_m(\boldsymbol{x}_j)\text{sign}(m - y_j) \le 0$ indicates that the function g_m misclassifies the sample $\boldsymbol{x}_j$.

The experiments are conducted based on three data sets: (1) 84,919 movie reviews with 10 different labels from IMDB [54]; (2) 78,966 product reviews with 5 rating scores from Yelp Data Challenge in 2013; (3) 1984 sentences with positive/negative labels obtained from news and documents stored in MPQA corpus.

To comprehensively compare the competing methods, we try to explore different dictionaries to leverage the sentiment words from the data sets. Specifically, we consider both the **SD** dictionary, which is from English opinion lexicon system [47] and the **GID** dictionary, which is from General Inquirer system [55]. Note that SD and GID contain 6790 and 1915 sentiment words, respectively.

We conduct the experiment 100 times by randomly splitting the data into training and testing datasets. We use 10-fold cross-validation to choose the hyper-parameters. Table 2 shows the average testing error for different methods on different datasets with different sentiment dictionaries. The number in the parentheses is the standard deviation of the testing error under 100 random replications.

5.1. Effect of feature

As shown in Table 2, the classifiers integrated with the newly extracted unified bigram features provide much better results than the corresponding competing methods (e.g., fMBC vs MBC, fSVM vs SVM and fRF vs RF). The testing error decreases by at least 5% for these methods. Therefore, our newly extracted bigram features are informative and could enable significant improvements. For the deep learning-based methods, GloVe-SSA and GloVe-CLNS, their results are comparable to fSVM and fRF, but are much lower than fMBC. We also notice that the overall performance of methods using the SD sentiment dictionary is better than the same method using GID dictionary. It is expected because the SD dictionary contains more sentiment words than GID dictionary.

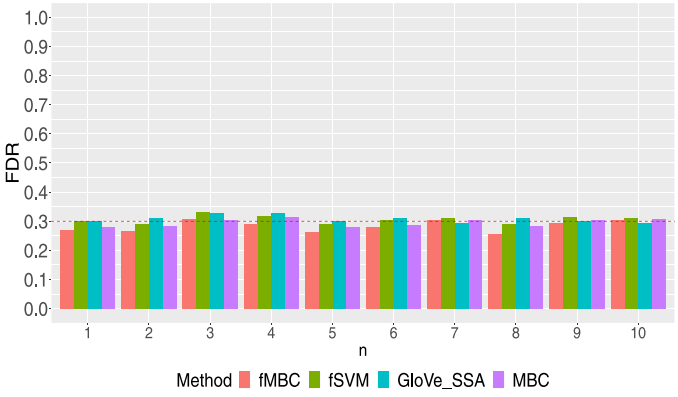


Fig. 4. FDR result for each experiment on MPQA corpus.

5.2. Effect of graph

The relational polarity information, which is encoded in the constraint graph, is useful for the sentiment prediction. This is strongly supported by the observation that fMBC demonstrates much better performance than fSVM and fRF, as shown in Table 2. We argue that the improvements mainly come from the integration of relational polarity graph information, as fMBC, fSVM and fRF share the same features.

From Table 2, we can find that fMBC, MBC, fSVM under SD and GloVe-SSA have better performance than other competing methods on the three datasets. Next we further evaluate these representative methods through introducing more metrics. To save space, they are experimented on MPQA dataset, as we observe similar results for the IMDB and Yelp datasets (see Table 2). We first explore their false discovery rate (FDR) at the level of 0.3 for the positive sentences classification on ten experiments. The result is shown in Fig. 4. We find that both of these methods can dominate FDR roughly under the specified level. Simultaneously, we also show their sensitivities and the amount of improvement of the proposed method through Table 3. As observed, the increase can reach 3.4%, 5.2% and 5.0% on average.

Finally, we explore their performance through precision and recall metrics. As shown in Fig. 5, fSVM and GloVe-SSA yield lower precision and recall than other methods, while the proposed fMBC method has the best performance. fMBC achieves an overall

Table 3
Sensitivities of classifiers fMBC, MBC and fSVM on ten experiments ($Diff = \frac{Senti_{(fMBC)}}{Senti_{(other)}} - 1$). Due to the limitation of space, we represent GloVe-SSA by G-SSA.

Method	1	2	3	4	5	6	7	8	9	10
fMBC	0.743	0.734	0.725	0.736	0.741	0.729	0.738	0.740	0.749	0.732
MBC	0.715	0.716	0.713	0.720	0.701	0.707	0.721	0.713	0.703	0.711
fSVM	0.701	0.708	0.685	0.704	0.689	0.702	0.707	0.704	0.697	0.705
G-SSA	0.704	0.711	0.698	0.709	0.692	0.696	0.703	0.697	0.693	0.712
Diff(1)	3.9%	2.5%	1.6%	2.2%	5.7%	3.1%	2.3%	3.7%	6.5%	2.9%
Diff(2)	6.1%	3.6%	5.8%	4.5%	7.5%	3.8%	4.3%	5.1%	7.5%	3.8%
Diff(3)	5.5%	3.2%	3.9%	3.8%	7.1%	4.7%	5.0%	6.2%	8.1%	2.8%

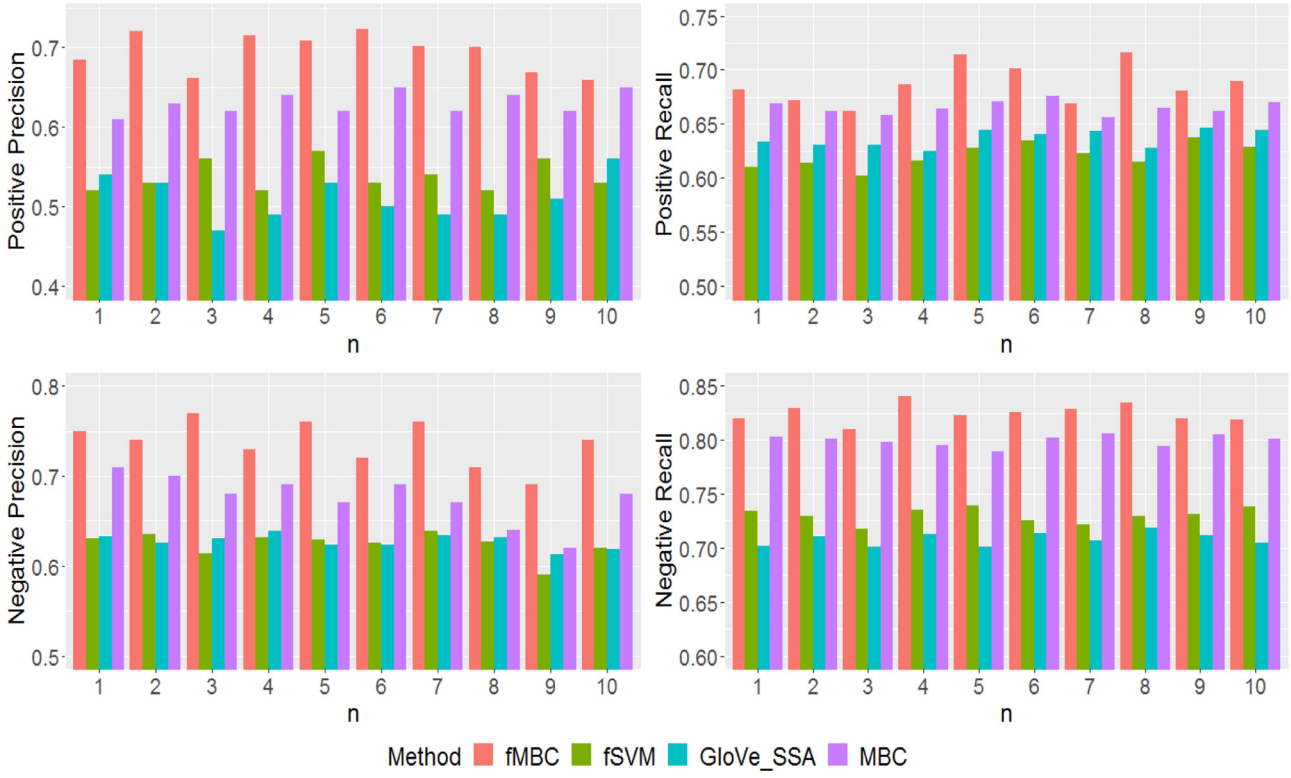


Fig. 5. Precision and Recall for fMBC, fSVM, GloVe_SSA and MBC on MPQA dataset.

accuracy of 74%, followed by MBC with an overall accuracy of 71%. The above results further verify the usefulness and efficiency of the proposed polarity-graph-based method.

6. Conclusion

We propose an extended regularization framework, with the polarity shifter embedded into features and the relative polarities encoded into a constraint graph, to predict the sentiment levels. Theoretically, we explain its connection with the Bayes classifier and show the asymptotic consistency for the proposed method. The decent performance on the real application demonstrates the usefulness and robustness of our proposed method.

With the favorable performance on real data sets, the applications of the proposed method to other aspects, such as social topic, product reviews [56,57], and different purposes, such as recommendation products, anomaly detection [58,59] can be further explored.

Declaration of Competing Interest

The authors declare that they do not have any financial or nonfinancial conflict of interests.

CRediT authorship contribution statement

Ruihua Cheng: Conceptualization, Methodology, Software. **Jie Zhang:** Supervision, Validation, Writing - review & editing. **Pengcheng Hu:** Data curation, Investigation, Visualization.

Acknowledgment

The authors thank Dr. Bin Zhu for proofreading and all reviewers and the editor for their constructive comments, which have greatly helped to improve the presentation of the article. The research was supported by the Natural Science Foundation of China (NSFC) (No.71771163)

Appendix A

A1. The proof of Theorem 1

Proof. As minimizing $E[U(\boldsymbol{Y})Q(\boldsymbol{\omega},\boldsymbol{Y})|\boldsymbol{X}=\boldsymbol{x}]$ for any given fixed $\boldsymbol{x}$, can yield the minimum of $E[U(\boldsymbol{Y})Q(\boldsymbol{\omega},\boldsymbol{Y})]$, we have

$$E[U(\boldsymbol{Y})Q(\boldsymbol{\omega},\boldsymbol{Y})|\boldsymbol{X}=\boldsymbol{x}]=\sum_{m=1}^{M-1}\tau_m I(\boldsymbol{Y}=m)Q(\boldsymbol{\omega},\boldsymbol{Y})),$$

where $I(\cdot)$ is the indicator function. The goal is to find $\tilde{g}_1,\ldots,\tilde{g}_{M-1}$, that minimizes $A(g_1,\ldots,g_{M-1})=\sum_{m=1}^{M-1}\tau_m I(\boldsymbol{Y}=m)Q(\boldsymbol{\omega},\boldsymbol{Y})$. The best candidate interval for the minimizer of $A(g_1,\ldots,g_{M-1})$ must in the range $[-1,1]^{M-1}$. Since for $m\in\{1,\ldots,M-1\}$, any $g_m\geq 1$ or $g_m\leq -1$, denote $g_m^{'}=\mathrm{sign}(g_m)$, then $g_m^{'}\in[-1,1]$, we can obtain $A(g_1^{'},\ldots,g_{M-1}^{'})<A(g_1,\ldots,g_{M-1})$ with little bit of comparison. The problem thus can be turned into the search in $[-1,1]^{M-1}$.

When $-1\leq g_m(\boldsymbol{x})\leq 1$, for $m\in\{1,\ldots,M-1\}$, we have $\{1-\mathrm{sign}(m-\boldsymbol{Y})g_m(\boldsymbol{x})\}_+=1-\mathrm{sign}(m-\boldsymbol{Y})g_m(\boldsymbol{x})$. Minimizing $E[U(Y)Q(\boldsymbol{\omega},\boldsymbol{Y})]$ then becomes

$$\begin{aligned}&\min E\left[\frac{1}{M-1}U(\boldsymbol{Y})\sum_{m=1}^{M-1}\min(1,(1-\mathrm{sign}(m-\boldsymbol{Y})g_m(\boldsymbol{x}))_+)\right]\\&=E(U(\boldsymbol{Y}))-\max\left(\frac{\sum_{m=1}^{M-1}E(U(\boldsymbol{Y}))(\mathrm{sign}(m-\boldsymbol{Y})g_m(\boldsymbol{x}))}{M-1}\right)\\&=E(U(\boldsymbol{Y}))-\max\left(\frac{\sum_{m=1}^{M-1}E[E\{U(\boldsymbol{Y})\mathrm{sign}(m-\boldsymbol{Y})|\boldsymbol{x}\}g_m(\boldsymbol{x})]}{M-1}\right)\end{aligned}\quad(7)$$

Furthermore, since the $M-1$ terms for the second component of (7) are independent, fixing other g_k, $k\in\{1,\ldots,m-1,m+$

$1, \ldots, M-1\}$, the optimization problem becomes finding $\hat{g}_m$ to maximize $E\{U(\boldsymbol{Y})\text{sign}(m-\boldsymbol{Y})|\boldsymbol{x}\}g_m(\boldsymbol{x})$. It is easy to have

$$E\{U(\boldsymbol{Y})\text{sign}(m-\boldsymbol{Y})|\boldsymbol{x}\} = \sum_{k=1}^{m} P(\boldsymbol{Y}=k)\tau_k - \sum_{k=m+1}^{M-1} P(\boldsymbol{Y}=k)\tau_k,$$

outputing the minimizer of $E[U(\boldsymbol{Y})Q(\boldsymbol{\omega}, \boldsymbol{Y})]$ to be

$$\text{sign}\left(\sum_{k=1}^{m} P(\boldsymbol{Y}=k)\tau_k - \sum_{k=m+1}^{M-1} P(\boldsymbol{Y}=k)\tau_k\right), \text{ for } m = 1, \ldots, M-1. \tag{8}$$

Thus we finish this proof. □

A2. The proof of Theorem 2

Proof. Since, for functions $\boldsymbol{g}_m(\boldsymbol{\omega})$ and $\boldsymbol{g}_m(\boldsymbol{\omega}_0)$, $\|Q(\boldsymbol{\omega}, \cdot) - Q(\boldsymbol{\omega}_0, \cdot)\|_2 \leq \|\boldsymbol{g}_m(\boldsymbol{\omega}) - \boldsymbol{g}_m(\boldsymbol{\omega}_0)\|_2$. And $\|\boldsymbol{g}_m(\boldsymbol{\omega}) - \boldsymbol{g}_m(\boldsymbol{\omega}_0)\|_2^2 = \|\boldsymbol{X}\boldsymbol{\omega} - \boldsymbol{X}\boldsymbol{\omega}_0\|_2^2 \leq \|\boldsymbol{X}\|^2\|\boldsymbol{\omega} - \boldsymbol{\omega}_0\|_2^2 + (\omega_{0m} - \omega_{0.0m})^2$. Hence $R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \epsilon^2$ implies that $\|\boldsymbol{\omega} - \boldsymbol{\omega}_0\| \leq c'\epsilon$ for some constant $c' > 0$. Then to bracket $\mathcal{G}_m$, the necessary count of grouping is less than the number of balls with diameter t to comprise the set $\{\boldsymbol{\omega}: R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \epsilon^2\}$. Therefore, $H(t, \mathcal{G}_m \cap R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \epsilon^2\}) \leq O((P-1)\log(\frac{c'\epsilon}{t}))$, where P is the feature size. As the relationship between $\mathcal{G}_{\mathcal{W}}$ and $\mathcal{G}_m$ is with $\mathcal{G}_{\mathcal{W}} = \{\boldsymbol{g}: \boldsymbol{g} \in \Pi_{m=1}^{M}\mathcal{G}_m, \boldsymbol{\omega} \in \mathcal{W}\}$, we have $H(t, \mathcal{G}_{\mathcal{W}} \cap \{R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \epsilon^2\}) \leq O((P-1)M\log(\frac{c'\epsilon}{t}))$. Define the number of nonzero $\boldsymbol{\omega}$ as $|A|$. Then $H(t, \mathcal{G}_{\mathcal{W}} \cap \{R(\boldsymbol{\omega}, \boldsymbol{\omega}_0) \leq \epsilon^2\}) \leq O(|A|M\log(\frac{c'\epsilon}{t}))$. Therefore, for some constant $c' > 0$, $H(t, \mathcal{V}_{\mathcal{W}}(\epsilon)) \leq \boldsymbol{O}(|A|M\log(c'\epsilon/t))$. □

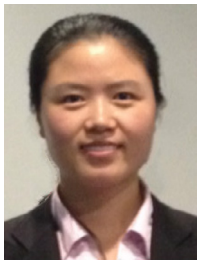
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neucom.2020.01.059

References

[1] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[2] M. Taboada, C. Anthony, K. Voll, Creating semantic orientation dictionaries, in: Proceedings of 5th International Conference on Language Resources and Evaluation (LREC), 2006, pp. 427–432.
[3] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, Comput. Linguist. 37(2) (2011) 267–307.
[4] A. Bandhakavi, N. Wiratunga, S. Massie, D. Padmanabhan, Lexicon generation for emotion detection from text, IEEE Intell. Syst. 32 (1) (2017) 102–108.
[5] A. Weichselbraun, S. Gindl, F. Fischer, S. Vakulenko, A. Scharl, Aspect-based extraction and analysis of affective knowledge from social media streams, IEEE Intell. Syst. 32 (3) (2017) 80–88.
[6] A. Dey, M. Jenamani, J.J. Thakkar, Senti-n-gram: an n-gram lexicon for sentiment analysis, Expert Syst. Appl. 103 (2018) 92–105.
[7] X. Fu, W. Liu, Y. Xu, L. Cui, Combine hownet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis, Neurocomputing 241 (2017) 18–27.
[8] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, E. Cambria, Learning word representations for sentiment analysis, Cognit. Comput. 9 (6) (2017) 843–851.
[9] A. García-Pablos, M. Cuadros, G. Rigau, W2vlda: almost unsupervised system for aspect based sentiment analysis, Expert Syst. Appl. 91 (2018) 127–137.
[10] Y. Zhang, D. Song, X. Li, P. Zhang, Unsupervised sentiment analysis of twitter posts using density matrix representation, in: European Conference on Information Retrieval, 2018, pp. 316–329.
[11] M. Ebrahimi, A.H. Yazdavar, A. Sheth, Challenges of sentiment analysis for dynamic events, IEEE Intell. Syst. 32 (5) (2017) 70–75.
[12] S. Poria, N. Majumder, D. Hazarika, E. Cambria, A. Gelbukh, A. Hussain, Multimodal sentiment analysis: addressing key issues and setting up the baselines, IEEE Intell. Syst. 33 (6) (2018) 17–25.
[13] F. Xu, J. Yu, R. Xia, Instance-based domain adaptation via multiclustering logistic approximation, IEEE Intell. Syst. 33 (1) (2018) 78–88.
[14] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? sentiment classification using machine learning techniques., in: Proceedings of the Conference on Empirical Methods in NLP, 2002, pp. 79–86.
[15] F. Salvetti, R. Christoph, S. Lewis, Opinion polarity identification of movie reviews, in: Computing Attitude and Affect in Text: Theory and Applications, Springer, 2006, pp. 303–316.
[16] J. Wang, B. Peng, X. Zhang, Using a stacked residual LSTM model for sentiment intensity prediction, Neurocomputing 322 (2018) 93–101.
[17] F. Huang, S. Zhang, J. Zhang, G. Yu, Multimodal learning for topic sentiment analysis in microblogging, Neurocomputing 253 (2017) 144–153.
[18] W. Gao, J.L. Guirao, B. Basavanagoud, J. Wu, Partial multi-dividing ontology learning algorithm, Inf. Sci. (Ny) 467 (2018) 35–58.
[19] W. Gao, L. Yan, M. Saeedi, H.S. Nik, Ultimate bound estimation set and chaos synchronization for a financial risk system, Math. Comput. Simul. 154 (2018) 19–33.
[20] W. Gao, W. Wang, Analysis of k-partite ranking algorithm in area under the receiver operating characteristic curve criterion, Int. J. Comput. Math. 95 (8) (2018) 1527–1547.
[21] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75.
[22] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM, Thirty-Second AAAI Conference on Artificial Intelligence (2018) 5876–5883.
[23] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A.F. Gelbukh, Sentiment and sarcasm classification with multitask learning, IEEE Intell. Syst. 34 (3) (2019) 38–43.
[24] D. Ma, S. Li, X. Zhang, H. Wang, Interactive Attention Networks for Aspect-level Sentiment Classification, AAAI Press, 2017, pp. 4068–4074.
[25] T.K. Landauer, S.T. Dumais, A solution to platoâs problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge, Psychol. Rev. 104 (1997) 211–240.
[26] S.T. Dumais, Latent semantic analysis, Annu. Rev. Inf. Sci. Technol. 38 (2005) 188.
[27] P. Turney, Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews, in: Proceedings of 40th Meeting of the Association for Computational Linguistics, 2002, pp. 417–424.
[28] L. Han, A. Kashyap, T. Finin, J. Mayfield, J. Weese, Umbc ebiquity-core: semantic textual similarity systems, in: Proceedings of the Second Joint Conference on Lexical and Computational Semantics, 1, 2013, pp. 44–52.
[29] C. Fellbaum, Wordnet, Wiley Online Library, 1998.
[30] Y. Li, Z.A. Bandar, D. McLean, An approach for measuring semantic similarity between words using multiple information sources, IEEE Trans. Knowl. Data Eng. 15 (4) (2003) 871–882.
[31] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[32] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (Aug) (2008) 1871–1874.
[34] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[35] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[36] J. Friedman, Another approach to polychotomous classification, Technical Report, Department of Statistics, Stanford University, 1996.
[37] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[38] S. Merity, N.S. Keskar, R. Socher, Regularizing and optimizing lstm language models, CoRR abs/1708.02182 (2017).
[39] T. Mikolov, Statistical language models based on neural networks, Presentation at Google, Mountain View, 2nd April, 80, 2012.
[40] T. Mikolov, G. Zweig, Context dependent recurrent neural network language model, in: 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 234–239.
[41] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, Lstm: a search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2016) 2222–2232.
[42] S. Han, J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang, et al., Ese: efficient speech recognition engine with sparse lstm on fpga, ACM, 2017, pp. 75–84.
[43] S. Venugopalan, L.A. Hendricks, R. Mooney, K. Saenko, Improving lstm-based video description with linguistic knowledge mined from text, Empirical Methods in Natural Language Processing (2016) 1961–1966.
[44] Y. Goldberg, A primer on neural network models for natural language processing, J. Artif. Intell. Res. 57 (2016) 345–420.
[45] Z. Teng, D.-T. Vo, Y. Zhang, Context-Sensitive Lexicon Features for Neural Sentiment Analysis, Association for Computational Linguistics, 2016, pp. 1629–1638.
[46] Z. Lin, M. Feng, C.N. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, CoRR abs/1703.03130 (2017).
[47] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[48] D. Chen, C. Manning, A fast and accurate dependency parser using neural networks, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 740–750.

[49] J. Wang, X. Shen, Y. Sun, A. Qu, Classification with unstructured predictors and an application to sentiment analysis, J. Am. Stat. Assoc. 111 (2016) 1242–1253.
[50] Y. Lin, Support vector machines and the bayes rule in classification, Data Min. Knowl. Discov. 6 (3) (2002) 259–275.
[51] X. Shen, On the method of penalization, Stat. Sin. 8 (1998) 337–357.
[52] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[53] E. Mammen, A.B. Tsybakov, et al., Smooth discrimination analysis, Ann. Stat. 27 (6) (1999) 1808–1829.
[54] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars), ACM, 2014, pp. 193–202.
[55] P.J. Stone, E.B. Hunt, A computer approach to content analysis: studies using the general inquirer system, ACM, 1963, pp. 241–256.
[56] K. Wang, W. Meng, S. Li, S. Yang, Multi-modal mention topic model for mentioned recommendation, Neurocomputing 325 (2019) 190–199.
[57] S. Gao, X. Li, Z. Yu, Y. Qin, Y. Zhang, Combining paper cooperative network and topic model for expert topic analysis and extraction, Neurocomputing 257 (2017) 136–143.
[58] D. Hou, Y. Cong, G. Sun, J. Liu, X. Xu, Anomaly detection via adaptive greedy model, Neurocomputing 330 (2019) 369–379.
[59] E. Tonnelier, N. Baskiotis, V. Guigue, P. Gallinari, Anomaly detection in smart card logs and distant evaluation with twitter: a robust framework, Neurocomputing 298 (2018) 109–121.



Ruihua Cheng received the Ph.D. degree in Mathematical Sciences from the New Jersey Institute of Technology, Newark, NJ, USA, in 2016. From 2017 to 2019, she was a Data Scientist with Innovation Institute, JD.com, Beijing, China. She is currently a researcher with Big Data Statistics Research Center, Tianjin University of Finance and Economics, Tianjin, China. Her current research interests include statistical modeling, sentiment analysis, and machine learning. Her researches have been published in conference proceedings including ICDM, SDM, etc.

Jie Zhang (S5) received the B.E. degree in software engineering from Nanjing University, Nanjing, China, in 2012, and the Ph.D. degree in computer science from the New Jersey Institute of Technology, Newark, NJ, USA, in 2017. His research interests include data mining, bioinformatics, statistical modeling, and machine learning.

Pengcheng Hu received the B.E. and M.S. degree from Jilin Agricultural University, Changchun, China, and M.S. degree from university of Camerino, Camerino, Italy. His research interest includes microbiological analysis.