Personalized Sentiment Classification
Based on Latent Individuality of Microblog Users

Kaisong Song[1], Shi Feng[1], Wei Gao[2], Daling Wang[1], Ge Yu[1], Kam-Fai Wong[3]

[1]Northeastern University, Shenyang, China

kaisongsong@gmail.com, {fengshi,wangdaling,yuge}@ise.neu.edu.cn

[2]Qatar Computing Research Institute, Qatar Foundation, Doha, Qatar   wgao@qf.org.qa

[3]The Chinese University of Hong Kong, Hong Kong   kfwong@se.cuhk.edu.hk

Abstract

Sentiment expression in microblog posts often reflects user's specific individuality due to different language habit, personal character, opinion bias and so on. Existing sentiment classification algorithms largely ignore such latent personal distinctions among different microblog users. Meanwhile, sentiment data of microblogs are sparse for individual users, making it infeasible to learn effective personalized classifier. In this paper, we propose a novel, extensible personalized sentiment classification method based on a variant of latent factor model to capture personal sentiment variations by mapping users and posts into a low-dimensional factor space. We alleviate the sparsity of personal texts by decomposing the posts into words which are further represented by the weighted sentiment and topic units based on a set of syntactic units of words obtained from dependency parsing results. To strengthen the representation of users, we leverage users following relation to consolidate the individuality of a user fused from other users with similar interests. Results on real-world microblog datasets confirm that our method outperforms state-of-the-art baseline algorithms with large margins.

1 Introduction

With the popular microblogging services such as Twitter and Sina Weibo, users can conveniently express their personal feelings and opinions about all kinds of issues in real time. The enormous microblog data have fueled us with torrents of subjective text. Mining users' sentiment orientation from such vast mount of subjective text has drawn close attention from research communities in recent years.

Much work was done for detecting sentiment orientation from user-generated content including reviews [Pang and Lee, 2005], online discussions [Hassan *et al.*, 2010], blogs [Feng *et al.*, 2011] and microblogs [Hu *et al.*, 2013]. However, most of studies ignore the personal distinctions among different users providing the subjective text. Typically, social media users exhibit various styles when expressing their feelings online. It is hypothesized that such diversity of sentimental manifestation may be pertinent to the latent

aspects of different people including their personality, educational background, current mood and some unknown factors. While using the same wording, people may deliver different sentiment orientations depending on the underlying context, which can be shown by the following example where some users are tweeting about work overtime:

A:	Yoga helps make my body flexible, lean & slim.	
A:	**After work overtime for 3 dys, I lose 3 pounds!**	(+)
B:	*I lose 5!*	(+)
C:	Getting poor feedback on a project where you are getting paid very little money for a lot of work.	
C:	**After work overtime for 3 dys, I lose 3 pounds!**	(-)
D:	*I lose 5!*	(-)

In this example, user **A** is a fan of body fitness who enjoys losing weight by hard work, and user **C** is a complainer who grumbles about hard work causing him loss of health. For the second sentences of both users, if not knowing their background, traditional sentiment models can hardly predict correct polarity without looking into user's identity. Given individual vocabulary choices, it is expected that useful latent information reflecting different individuality could be captured and leveraged to determine the user's sentiment orientation more accurately. Also, if we know that **B** is **A**'s follower in a Yoga class, and **D** follows **C** being **C**'s colleague, the latent individuality of target users could be strengthened further based on such relation among users. Thus, it would be more likely that they may share similar interests and hold consistent sentiment orientation with their respective followees.

In this paper, we aim at catching latent personal distinctions or individuality among different writers of subjective text. We propose a novel, extensible and effective *Personalized Sentiment Classification* method based on a variant of Latent Factor Model (LFM) that realizes sentiment personalization under the sparse distribution of microblog texts which is very common in social media environment. The main contributions of our paper are three-fold:

- We propose a novel, extensible latent-factor-based personalized sentiment classification model, which alleviates the sparsity of training data by decomposing microblog posts into a finer-grained representation via probabilistic matrix factorization and maps users and microblog contents into a shared latent factor space. The latent factors reflect the *interested aspects* of both users

and contents, which correspond to individuality of users.

- Our sentiment analysis module takes into account different roles of syntactic units (primarily sentiment units and topic units) identified from words' dependency relations, which provide rich emotional and topical clues regarding the interested aspects of different users. Our factorization model achieves more accurate estimation of personalized sentiment scores of the posts by automatically weighing different types of syntactic units.

- We integrate social relation into the model for enhancing the personalized sentiment representation of target users, which is based on the intuition that the following relation between followers and followees can reflect the shared latent individuality of users.

2 Related Work

Research for understanding sentiments of microblogs has been an active research area in recent years [Go *et al.*, 2009; Pak and Paroubek, 2010; Davidov *et al.*, 2010; Wang *et al.*, 2011; Tan *et al.*, 2011; Hu *et al.*, 2013; Hutto and Gilbert, 2014]. Go et al. [2009] and Pack and Paroubek [2010] both acquired sentiment data from Twitter for learning sentiment models. Hutto and Gilbert [2014] presented VADER, a simple yet effective rule-based model for general sentiment analysis of social media text. Speriosu et al. [2011] proposed a semi-supervised approach for polarity annotation by using label propagation over lexical links and follower graph. Wang et al. [2011] automatically generated the overall sentiment polarity for a given hashtag relevant to some coarse-grained topics. Tan et al. [2011] assumed that users that are somehow connected may be more likely to hold similar opinions, therefore, they used social relation to complement a user's viewpoints from their utterances. Hu et al. [2013] studied social relations and proposed a sociological approach to handle microblogging texts for sentiment classification. These studies did not deal with personalization. Our goal is to personalize the sentiment model by differentiating latent individuality of microblog users that are not articulated explicitly and dealing with the sparsity of personal microblog data.

Latent factor model, as a matrix factorization method, has been widely used in online recommendation systems. LFM can capture the hidden elements determining users' preferences which are commonly difficult to analyze. Agarwal et al. [2011] proposed a factor model that incorporates rater-comment and rater-author interactions to rank the comments associated with a given article according to user preference. Chen et al. [2012] proposed a collaborative tweet ranking model for recommending tweets to users. Li et al. [2009] worked on a constrained non-negative trifactorization method based on term-document matrix to learn a sentiment model from lexical priori knowledge, but they failed to personalize the model. Different from these works, we incorporate the factors induced from *social, sentimental and topical evidence* into the matrix factorization process and propose a novel extensible LFM for personalized sentiment modeling on microblogs data.

Personalized sentiment model was rarely explored for microblog users. Calais Guerra et al. [2011] argued that users'

opinion bias always existed in microblogs for some particular entities, and proposed a user bias quality method to transfer user biases into opinion features for specific topics. Li et al. [2011a] proposed a tensor factorization model for review rating, where the reviewer, product and text features were modeled as a three-dimension tensor, to which our work is more closely related. But our work differs significantly from theirs: (1) Their model is particularly customized for product review rating and cannot be directly applied for our microblogging texts independent of specifiable products or objects; (2) Their model is simply based on bag of words which is not as extensible as ours to model fine-grained sentiment-topic expressions and incorporate user connections to strengthen the capturing of useful individuality.

3 Latent Factor Model for Microblog Users

Inspired by the collaborative personalized tweet recommendation [Chen *et al.*, 2012], we try to model the personalized sentiment of microblog users based on LFM. However, we argue that the LFM has poor extensibility for unobserved microblog posts rendering the direct application of it impractical. In this section, we first introduce the basic setting of LFM, then we extend the model by decomposing the post items into word level for alleviating the sparsity of personal microblog data, and also incorporating social relation evidence for personalization modeling. In this section, we give its theoretical derivation, which paves the way for our personalized sentiment model.

3.1 Formalization of LFM

Let $U = \{u_1, u_2, \ldots, u_{|U|}\}$ be a set of users, and $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be a set of microblogs posted by U. For each $u \in U$, $I_u \subset I$ is the set of posts by u, and $|I_u| \ll |I|$. The posts with sentiment labels form the set I_u^* ($I_u^* \subset I_u$), and similarly $|I_u^*| \ll |I_u|$. For any post $i \in I_u^*$, we observe a ground-truth polarity x_{ui} ($x_{ui} = 1$ if the observed sentiment of i is positive, or $x_{ui} = 0$ otherwise). Each training instance is represented by a tuple (x_{ui}, u, i), which is organized into a sparse matrix $\mathbf{X}$ of size $|U| \times |I|$, using (u, i) as index and x_{ui} as entry value. The task is to predict the sentiment score $\hat{x}_{ui}$ for the missing items in the set $I_u^+ = I_u - I_u^*$. LFM tries to approximate $\mathbf{X}$ by the product of two low-rank latent factor matrices $\mathbf{W}$: $|U| \times f$ and $\mathbf{H}$: $|I| \times f$, where f is a parameter corresponding to the rank of the approximation, and in general $f \ll min\{|U|, |I|\}$. Therefore, each observed score x_{ui} in $\mathbf{X}$ can be approximated by $\hat{x}_{ui}$ in $\hat{\mathbf{X}}$, which is the product of two components:

$$\hat{x}_{ui} = \mathbf{W}_u \cdot \mathbf{H}_i^{\mathsf{T}}, \tag{1}$$

where $\mathbf{W}_u$ is the user feature vector and $\mathbf{H}_i$ is the post feature vector both in f-dimensional space that is referred to as *interested aspects* of users and contents. For users, interested aspects indicate their latent individuality; in posts, sentiments are expressed towards the interested aspects, which together help model the personalized sentiment scoring for $\hat{x}_{ui}$.

The direct application of LFM has some critical problems: (1) The extensibility is limited since the model has to be retrained when a new post comes in; (2) Each post is treated as
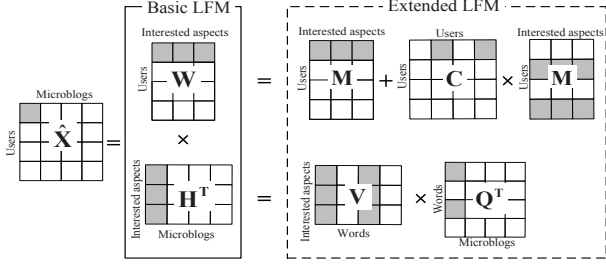
Figure 1: Our extensible Latent Factor Model (LFM)

an entire item, thus its consisted components containing crucial personal sentiment features cannot be utilized; (3) The personal data in matrix $\mathbf{X}$ is generally very sparse, which makes factorization ineffective.

3.2 Our Extension: Word Level Decomposition and Following Relation Incorporation

To resolve these problems, our idea is to derive an extensible model that decomposes posts into some finer-grained level. In this section, we consider words as the basic unit in the first place without complicating the model too much (In the next section, we will consider coupling words of different functions for further enhancement). Meanwhile, it is intuitive that target users may have shared interested aspects with their followees. Therefore, to enhance personalization, we incorporate users' following relations into the factorization model. Figure 1 illustrates our extension which is detailed below.

Suppose there is a vocabulary K containing all the words in the dataset, and then we decompose $\mathbf{H}$ as $\mathbf{H} = \mathbf{Q} \cdot \mathbf{V}^{\mathrm{T}}$, where $\mathbf{Q}$: $|I| \times |K|$ is a post-word matrix whose i-th row is the vector of post i indicating the existence of its component words, and $\mathbf{V}$: $f \times |K|$ is an estimated word-factor matrix projecting each word k into feature vector $\mathbf{V}_k^{\mathrm{T}}$ of size f. Therefore, the estimated matrix $\hat{\mathbf{X}} = \mathbf{W} \cdot (\mathbf{Q} \cdot \mathbf{V}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{W} \cdot (\mathbf{V} \cdot \mathbf{Q}^{\mathrm{T}})$, and formula 1 can be reformulated as:

$$\hat{x}_{ui} = \mathbf{W}_u \cdot (\mathbf{V} \cdot \mathbf{Q}_i^{\mathrm{T}}) \qquad (2)$$

To incorporate social connections, we project the user-factor matrix $\mathbf{W}$ into $\mathbf{M} + \mathbf{C}\mathbf{M}$ with a newly estimated user-factor matrix $\mathbf{M}$: $|U| \times f$ and an observed followee-follower connection matrix $\mathbf{C}$: $|U| \times |U|$ where each entry C_{uv} indicates whether user u follows user v (its value is normalized instead of binary). As a result, formula 2 is extended as:

$$\hat{x}_{ui} = (\mathbf{M}_u + \mathbf{C}_u\mathbf{M}) \cdot (\mathbf{V} \cdot \mathbf{Q}_i^{\mathrm{T}}) \qquad (3)$$

For estimating $\hat{\mathbf{X}}$, we derive an optimization objective based on probabilistic matrix factorization [Salakhutdinov *et al.*, 2007]. A linear model with Gaussian observation noise can be adopted to generate the observed variables. We can obtain the probability of generating $\mathbf{X}$ as $P(\mathbf{X}|\mathbf{M}, \mathbf{V}) = \prod_{u=1}^{|U|} \prod_{i=1}^{|I|} [\mathcal{N}(x_{ui}|\hat{x}_{ui}, \delta^2)]^{\mathbf{1}_{ui}}$, where $\mathcal{N}(x|\mu, \delta^2)$ is the Gaussian distribution with mean $\mu = \hat{x}_{ui}$ and variance δ^2, and $\mathbf{1}_{ui}$ is an indicator function which takes 1 if u and i are observed in the training set or takes 0 otherwise.

Then we can place a zero-mean Gaussian prior distribution on user and word feature vectors as follows: $P(\mathbf{M}) = \prod_{u=1}^{|U|} \mathcal{N}(\mathbf{M}_u|0, \delta_U^2\mathbf{E})$ and $P(\mathbf{V}) = \prod_{k=1}^{|K|} \mathcal{N}(\mathbf{V}_k|0, \delta_K^2\mathbf{E})$, where $\mathbf{E}$ is an identity matrix, δ_U^2 is the variance of user features, and δ_K^2 is the variance of word features.

Based on Bayesian formula, we can obtain the posterior $P(\mathbf{M}, \mathbf{V}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{M},\mathbf{V}) \times P(\mathbf{M}) \times P(\mathbf{V})}{P(\mathbf{X})}$. Then the parameters (i.e., features in latent space) $\mathbf{\Theta} = (\mathbf{M}, \mathbf{V})$ of the model can be found by taking the logarithm of the posterior and using maximum a posterior estimation. Finally, we derive the objective function by minimizing the sum of squared errors with a regularization term:

$$\min_{\mathbf{\Theta}} \left\{ \sum_{u=1}^{|U|} \sum_{i=1}^{|I|} \mathbf{1}_{ui}(x_{ui} - \hat{x}_{ui})^2 + Regularizer \right\} \qquad (4)$$

where $Regularizer = \lambda_U \sum_u \|\mathbf{M}_u\|_2^2 + \lambda_K \sum_k \|\mathbf{V}_k\|_2^2$, $\lambda_U = \frac{\delta^2}{\delta_U^2}$ and $\lambda_K = \frac{\delta^2}{\delta_K^2}$ are the fixed coefficients corresponding to users and words, respectively. The first term strives to fit the given sentiment scores, and the regularizer avoids overfitting by penalizing the magnitudes of the parameters. After estimating the parameters $\mathbf{\Theta}$, we can predict the sentiment score of any unobserved post by using formula 3.

4 Personalized Sentiment Modeling

Although the word-level decomposition can make post-word matrix $\mathbf{Q}$ denser, the bag-of-words model is still rather coarse for sentiment analysis, which might drag the parameter estimation with respect to words in $\mathbf{V}$. Some basic problems should be considered: (1) Negation often modifies sentiment dramatically, e.g., from positive to negative; (2) It is necessary to differentiate sentiment words and non-sentiment words while some non-sentiment words may need special treatment. For example, intensifiers (e.g., 'extremely', 'very', etc.) are sentiment shifters and should be modeled together with sentiment words they modify; (3) Prior knowledge, e.g., sentiment values in a sentiment lexicon, is important. We can treat (3) straightforwardly by letting the entry of $\mathbf{Q}$ take the sentiment value for sentiment word (sentiment values are usually larger than 1).

Other than sentimental evidence, topic-related words may reflect users' individuality which need to be modeled appropriately. In fact, the hidden topics can naturally result from the interested aspects in the factorized matrix $\mathbf{V}$ which interact with users via the user-factor matrix. We just need to specify topical words and treat them specially in the model.

To compromise sparsity alleviation and model's effectiveness, we resort to the syntactic units in the posts derived from dependency parsing as intermediary for better estimating the word features in the latent space, which is detailed below.

4.1 Syntactic Units from Dependency Parsing

Words are not independent of each other in a sentence because usually there exists dependency relation among them. Syntactic units based on dependency can capture long-distance word relation and convey finer-grained sentiment and topic information than bag of words, bigrams or trigrams. When estimating word features, we will consider such dependency relation to improve the accuracy of estimation
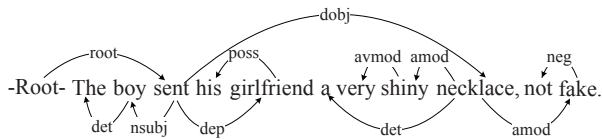
Figure 2: Typed dependencies representation of a post

(see Section 4.2). Here we describe how the syntactic units can be extracted.

Figure 2 shows an example of typed dependencies represented as a directed graph, where a *head* word points to a *dependent* word on each edge denoting that the dependent somehow modifies the head. The dependencies are all binary relations, so we can obtain a group of word pairs such as {sent, necklace}, {shinny, very}, {fake, not}, etc., where we do not differentiate word order. While microblog text is typically informal and noisy, we still resort to the Stanford neural network parser[1], which is trained on formal text, to parse the posts. However, we lexically normalize the posts and correct ill-formed words during preprocessing by referring to [Han *et al.*, 2013]. This saves the intensive labor cost to annotate data for training a tailored microblog parser while we can still obtain reasonable parsing result.

One issue is that the obtained pairwise units from dependency parsing are not all useful. Therefore, we resort to sentiment lexicon and part-of-speech (POS) of the words to extract those units that provide sentimental and topical evidence, referred to as *sentiment units* and *topic units*, respectively. We adopt the following rules for extraction:

1. Units containing words in sentiment lexicon are kept as sentiment units; Other units which contain noun or verb are kept as topic units because typically nouns and verbs can describe the topics concerned;

2. Adverbs in the units obtained after rule 1 are kept because they are indispensible in expressing the intensity (e.g., very) or shift (e.g., not) of emotions, or the accomplishment of some actions (e.g., already);

3. Words with other types of POS are removed from the units which are generally not sentimental nor topical;

4. Due to the removal of some words from the pairs after rule 3, it can result in singletons. A resulted singleton is either just left as it is or merged into larger units that contain that word.

4.2 Integrating Sentiment Units and Topic Units

Given the syntactic units extracted, we need to specify the entries' values in $\mathbf{Q}$ for better estimating the word feature values in $\mathbf{V}$. Note that we use words rather than the identified syntactic units as the ultimate representation in matrices because the chance of having unseen words is generally lower than having unseen units in test data. But we will consider the extracted word relation for parameter estimation.

Under bag-of-words setting, the entry of $\mathbf{Q}$ in formula 2 can take $\frac{1}{|K_i|}$ for the observed words in post i and 0 for the

[1]nlp.stanford.edu/software/nndep.shtml

unobserved ones, where $|K_i|$ is size of vocabulary of i. Under the new setting, we need to differentiate words in different types of units by giving them different values. We thus obtain the prediction formula by using the weighted combination of words occurring in the two types of syntactic units:

$$\hat{x}_{ui} = (\mathbf{M}_u + \mathbf{C}_u\mathbf{M}) \cdot \left(\sum_{s \in S_i} \frac{1}{|S_i|} \sum_{k \in s} \mathbf{V}_k + \sum_{t \in T_i} \frac{1}{|T_i|} \sum_{k \in t} \mathbf{V}_k \right) \quad (5)$$

where S_i is the set of sentiment units and T_i is the set of topic units in post i, s is any sentiment unit in S_i and t is any topic unit in T_i, $\mathbf{V}_k$ is the feature vector of component word k, and $\frac{1}{|S_i|}$ and $\frac{1}{|T_i|}$ are the weights of words in sentiment units and topic units, respectively.

4.3 Integrating Baseline Predictor

Koren [2009] suggested a baseline predictor be incorporated into LFM for better generalization. Different from the term of user-item interaction, the baseline predictor embodies general properties of users and words, which takes the first-order form $\mathbf{b} = \omega + b_u + \sum_{k \in K_i} b_k$, where constant ω represents overall sentiment orientation of all the posts in corpus, variables b_u and b_k indicate the observed deviations of user u and word k in post i, respectively, from the value of ω. The prediction formula by integrating the baseline predictor becomes:

$$\hat{x}_{ui} = \mathbf{b} + (\mathbf{M}_u + \mathbf{C}_u\mathbf{M}) \cdot \left(\sum_{s \in S_i} \frac{1}{|S_i|} \sum_{k \in s} \mathbf{V}_k + \sum_{t \in T_i} \frac{1}{|T_i|} \sum_{k \in t} \mathbf{V}_k \right)$$
$$(6)$$

4.4 Model Training and Inference

Based on formula 6, we rewrite formula 4 to obtain the final objective function as follows:

$$\min_{\Theta} \left\{ \sum_{u,i} (x_{ui} - \hat{x}_{ui})^2 + Regularizer^+ \right\} \quad (7)$$

where $Regularizer^+ = \lambda \sum_u b_u^2 + \lambda \sum_k b_k^2 + Regularizer$.

The objective function is convex. We search for its minimum using stochastic gradient descent [Bottou, 2004]. All the variables in the parameter space $\Theta = (\mathbf{b}_U, \mathbf{b}_K, \mathbf{M}, \mathbf{V})$ can be estimated automatically, where $\mathbf{b}_U = \{b_u\}$ and $\mathbf{b}_K = \{b_k\}$ are bias sets, $\mathbf{M} = \{\mathbf{M}_u\}$ are user latent features, and $\mathbf{V} = \{\mathbf{V}_k\}$ are word latent features. The set $\lambda_\Theta = \{\lambda, \lambda_U, \lambda_K\}$ contains the regularization coefficients, for which λ is tuned using the development set, λ_U and λ_K are fixed via $\frac{\delta^2}{\delta_U^2}$ and $\frac{\delta^2}{\delta_K^2}$ which are set empirically.

Given a new post i of a user u, its personalized sentiment score $\hat{x}_{ui}$ is predicted by formula 6. Then, we use sigmoid function $\sigma(x) = \frac{1}{1+\exp(-x)}$ to transform it into a distribution. If $\sigma(\hat{x}_{ui}) \in [0, 0.5)$, we classify the sentiment of the post as negative, or as positive if $\sigma(\hat{x}_{ui}) \in [0.5, 1]$.

5 Experimental Evaluation

5.1 Datasets and Setup

Large scale microblog corpus annotated manually is not available. Since emoticons are personal tags given by writers,

Statistics items	Weibo dataset	Twitter dataset
# of posts	43,250	48,563
# of positive posts	32,060	34,624
# of negative posts	11,190	13,939
Size of vocabulary	30,171	23,181
# of sentiment words	4,495	2,457
# of topic words	22,758	17,899
# of syntactic units	314,712	213,590
# of sentiment units	40,775	42,650
# of topic units	164,529	98,987

Table 1: The statistics of the microblog datasets we used

(a) Weibo dataset   (b) Twitter dataset

Figure 3: Decomposition alleviated sparsity of user data

(a) Weibo dataset   (b) Twitter dataset

Figure 4: Influence of the depth of followers considered

they are assumed suitable personalized sentiment labels [Go *et al.*, 2009; Pak and Paroubek, 2010]. We selected 33 frequently used positive emoticons and 57 negative ones to derive ground-truth polarities. We crawled the microblog posts of 281 Sina Weibo users and 674 Twitter users using Weibo API[2] and Twitter API[3], respectively. We obtained 43,250 Weibo posts and 48,563 tweets, each containing one positive or negative emoticon. For Weibo corpus, we used an effective Chinese tokenizer[4] for word segmentation. For both datasets, Stanford POS tagger and neural network dependency parser were employed for POS tagging and dependency parsing, respectively (see Section 4.1). The Chinese sentimental words ontology bank[5] and NRC's EmoLex and MaxDiff Twitter Sentiment Lexicon[6] were used as sentiment lexicons for Weibo posts and tweets, respectively. The statistics about the two datasets are shown in Table 1.

We used 10-fold cross validation for evaluation, where 8 folds were for training, 1 for development and 1 for test. We implemented our models based on the generic factorization tool SVDFeature[7]. A common issue with microblog data is the imbalanced sentiment class distribution [Li *et al.*, 2011b; Liu *et al.*, 2013]. We re-sampled the training instances *for each user* to balance the proportion of positive and negative posts while keeping the development and test data are intact. We used geometric mean [Kubat and Matwin, 1997] as the metric for evaluation considering imbalanced nature of our dataset, which was defined as $G\text{-}mean = \sqrt{Sensitivity \times Specificity}$, where *Sensitivity* is the true positive rate and *Specificity* is the true negative rate.

5.2 Experiments and Results

Fixed parameters setting

We optimized the f parameter via validation on the development set by performing a grid search on all values of $10 * x$ with $x \in \{1, 2, ..., 10\}$. Basically the performance was not sensitive with respect to f, and we fixed $f = 60$ which is slightly better than other choices. We tuned λ using the development data and fixed it as $1.0e\text{-}4$. Since λ_U and λ_T are calculated by $\frac{\delta^2}{\delta_U^2}$ and $\frac{\delta^2}{\delta_T^2}$ (see formula 4), we set the two ratios to fixed values as $1.0e\text{-}3$ as we found that varying them just influenced the results slightly.
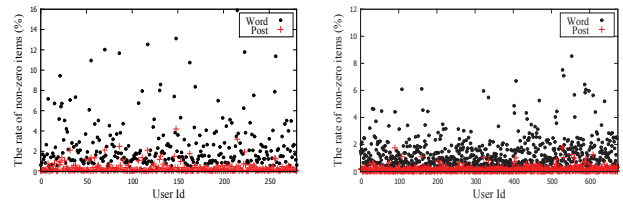
---

[2] open.weibo.com/

[3] https://dev.twitter.com/overview/api

[4] ictclas.nlpir.org/

[5] ir.dlut.edu.cn/EmotionOntologyDownload.aspx
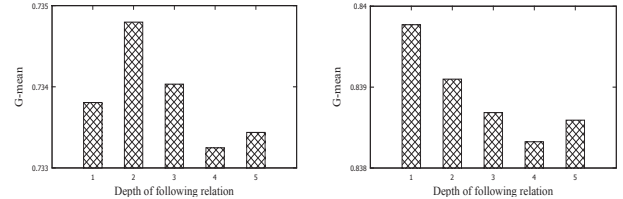
[6] www.saifmohammad.com/Lexicons/

[7] svdfeature.apexlab.org/wiki/Main_Page

Effects of decomposition and following relation

We examined the effect of word-level decomposition for dealing with feature sparsity. We compare the ratio of non-zero features of *each user* before decomposition, i.e., $\frac{\text{observed posts \#}}{\text{total posts \#}}$, and that after it, i.e., $\frac{\text{observed words \#}}{\text{total words \#}}$. The ratios are displayed in Figure 3. Overall, the non-zero rate of the personal data raised considerably from 0.36% to 2.9% for Weibo data and from 0.15% to 1.26% for Twitter data. This implies that the decomposition is helpful to alleviate the sparsity.

We also studied how the depth of following relation considered can influence the performance. Here we use the formula $\mathbf{C} = \sum_{i=1}^{n} \frac{1}{i} \mathbf{C}^i$ to calculate the entry values of connection matrix up to n level of depth in the following relation. The strength of connection is decayed by a factor of $\frac{1}{i}$ for the i-th level relation. Figure 4 shows that our model achieved best result when $n=2$ on Weibo and $n=1$ on Twitter. This is because the user relation on our Twitter data is much denser. The result seems to indicate that the first-level connection is sufficient for Twitter, but followers one more step deeper is helpful for Weibo, and more depth may bring too much noise. But overall the impact of depth does not appear very strong.

Comparison of different configurations

We compared the performance of five different settings: (1) **Basic:** direct application of LFM using the original user-post matrix; (2) **BOW:** our bag-of-words LFM without considering dependency relation and user following relation, which is equivalent to modifying the reviewer-product-review model [Li *et al.*, 2011a] to remove the product dimension not needed; (3) **Follow**: our model that considers following connection and uses only bag of words; (3) **Depend**: our model that considers dependency-based syntactic units but not using following relation; (5) **Full:** our fully configured model.

As shown in Table 2, **Basic** performs the worst on G-mean due to sparsity of the user-post matrix. Other models with the decomposition appear much better. This indicates that allevi-

Dataset	Metric	Basic	BOW	Follow	Depend	Full
Weibo	SEN	.725	.699	.707	**.748**	.745
	SPE	.594	.713	.721	.709	**.725**
	GM	.656	.705§	.714§	.727‡§	**.735**‡§
Twitter	SEN	.747	.799	.800	.823	**.835**
	SPE	.832	.839	.843	.846	**.847**
	GM	.779	.819§	.825§	.835‡§	**.840**‡§

Table 2: Comparison of different model configurations. ‡, §: significantly better than BOW and Basic, respectively ($p <$0.01). SEN: Sensitivity; SPE: Specificity; GM: G-mean

Dataset	Metric	SVM	MFP	PSVM	SNM	Co-train	Ours
Weibo	SEN	.719	.642	.691	.718	.715	**.745**
	SPE	.501	.605	.652	.654	.695	**.725**
	GM	.587*	.623*	.667*	.675*	.705*	**.735**
Twitter	SEN	.514	.654	.704	.790	.743	**.835**
	SPE	.746	.621	.624	.764	.735	**.847**
	GM	.619*	.633*	.657*	.776*	.739*	**.840**

Table 3: Comparison of different approaches. * denotes significantly different G-mean from ours ($p <$0.01). SEN: Sensitivity; SPE: Specificity; GM: G-mean

ating matrix sparsity is effective to the personalized sentiment classification because many word-level features can take effect after the decomposition. Since **BOW** only adopted linear combination of words and neglected the importance of word dependencies and social effects, it performs the worst among other contrastive models. **Follow** improves **BOW** by enhancing the individuality of target user by interests of connected users, but it still can not compete with **Depend** and **Full** on G-mean due to its coarse sentiment-topic representation. Higher G-mean of **Full** implies that both dependency and following relations are complementary to improving sentiment model.

Comparison of different approaches

We compared our fully configured model with some traditional and advanced baselines *all based on the same setup of our datasets*. We first compared to standalone models which can be categorized as non-personalized and personalized:

– The non-personalized models did not distinguish user data. We used (1) bag-of-words **SVM** [Pang and Lee, 2005] with word frequency features; (2) the non-negative matrix tri-factorization based approach with lexicon prior knowledge [Li *et al.*, 2009] named **MFP** as conventional and strong non-personalized baselines, respectively.

– We used the classifier-fusion framework for multi-domain sentiment classification [Li and Zong, 2008] to simulate personalized sentiment model based on SVM, named as **PSVM**. The model was obtained by training a meta-classifier with input attributes that are the output of base classifiers. We manually grouped users into eight domains based on the occupation in their profiles (e.g., finance, education, etc.) or their following relation when the occupation is not given, and then trained a base classifier for each group of users.

We also compared to two ensemble-based models: (1) **Co-train**: Li et al. [2011b] proposed a feature-level ensemble method using co-training. The most confident positive and negative results are selected from the outputs of the two classifiers each learned from a subspace of features; (2) **SNM** (SVM+Naive Bayes+Maximum Entropy): Kittler et al. [1998] proposed an ensemble method with majority vote. So we combined SVM, Naive Bayes and Maximum Entropy with this voting method where the ensemble is of model level.

We tuned all these models on the development set to obtain their optimal parameters. The results of comparison on test set are displayed in Table 3. Compared with standalone methods, ours is obviously higher than both non-personalized and personalized baselines in terms of G-mean. It improves 25.2% over **SVM**, 18.0% over **MFP** and 10.2% over **PSVM** on Weibo dataset, and more strikingly improves 35.7% over

SVM, 32.7% over **MFP** and 27.9% over **PSVM** on Twitter dataset. This indicates the effectiveness of our personalized sentiment model. **PSVM** outperforms **SVM** by 13.6% and 6.1% on Weibo and Twitter data, respectively, which manifests that the personalized data based on different user groups can better distinguish users' personal sentiment. Also, we find ensemble models **Co-train** and **SNM** outperform standalone methods, but still clearly worse than ours especially on Twitter dataset. These again verify the advantages of our personalized sentiment classification model.

Significance test

We conducted two-tailed pairwised *t*-test on G-mean between our model and approaches in Table 3 and among the different configurations in Table 2 by running 10 times 10-fold cross-validation based on the re-partitioned folds each time [Han, 2005]. Therefore, we obtained 100 sample scores per model. As shown in both tables, most pairs of results examined are significantly different from each other. This confirms the effectiveness of our approach. It seems that our model using follow relation could be further improved as it cannot significantly outperform BOW. It might be because the user connectivity on our Weibo dataset is too sparse, but that on our Twitter dataset is much dense. We leave this for future study.

6 Conclusion

In this work, we focus on the personalized sentiment classification for microblog users. We proposed novel extensible latent factor models to automatically capture the latent individuality from the underlying context of sentimental microblog posts. In particular, our approach decomposed microblog posts into word level for the sparsity of personal data, and considered dependency relations between words for better estimating the word features; it also integrated user's following relations to strengthen the catch of individuality for target users. Experimental results on two real-world microblog datasets show that the performance of our method is very promising and significantly outperforms state-of-the-art baseline algorithms.

Acknowledgments

References

[Agarwal *et al.*, 2011] Deepak Agarwal, Bee-Chung Chen, and Bo Pang. Personalized recommendation of user comments via factor models. In *Proceedings of EMNLP*, pages 571–582, 2011.

[Bottou, 2004] Léon Bottou. Stochastic learning. In *Advanced Lectures on Machine Learning*, pages 146–168. Springer Verlag, 2004.

[Calais Guerra *et al.*, 2011] Pedro Henrique Calais Guerra, Adriano Veloso, Wagner Meira, Jr., and Virgílio Almeida. From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In *Proceedings of SIGKDD*, pages 150–158, 2011.

[Chen *et al.*, 2012] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. Collaborative personalized tweet recommendation. In *Proceedings of SIGIR*, pages 661–670, 2012.

[Davidov *et al.*, 2010] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of COLING*, pages 241–249, 2010.

[Feng *et al.*, 2011] Shi Feng, Daling Wang, Ge Yu, Wei Gao, and Kam-Fai Wong. Extracting common emotions from blogs based on fine-grained sentiment clustering. *Knowledge and Information Systems*, 27(2):281–302, 2011.

[Go *et al.*, 2009] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. In *Technical Report, Stanford University*, pages 1–6, 2009.

[Han *et al.*, 2013] Bo Han, Paul Cook, and Timothy Baldwin. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology*, 4(1):5:1–5:27, 2013.

[Han, 2005] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[Hassan *et al.*, 2010] Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. What's with the attitude? identifying sentences with attitude in online discussions. In *Proceedings of EMNLP*, pages 1245–1255, 2010.

[Hu *et al.*, 2013] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of WSDM*, pages 537–546, 2013.

[Hutto and Gilbert, 2014] C. J. Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of ICWSM*, pages 216–225, 2014.

[Kittler *et al.*, 1998] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[Koren, 2009] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of SIGKDD*, pages 447–456, 2009.

[Kubat and Matwin, 1997] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of ICML*, pages 179–186, 1997.

[Li and Zong, 2008] Shoushan Li and Chengqing Zong. Multi-domain sentiment classification. In *Proceedings of ACL-HLT*, pages 257–260, 2008.

[Li *et al.*, 2009] Tao Li, Yi Zhang, and Vikas Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of ACL-IJCNLP*, pages 244–252, 2009.

[Li *et al.*, 2011a] Fangtao Li, Nathan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. Incorporating reviewer and product information for review rating prediction. In *Proceedings of IJCAI*, pages 1820–1825, 2011.

[Li *et al.*, 2011b] Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. Semi-supervised learning for imbalanced sentiment classification. In *Proceedings of IJCAI*, pages 1826–1831, 2011.

[Liu *et al.*, 2013] Zhiguang Liu, Xishuang Dong, Yi Guan, and Jinfeng Yang. Reserved self-training: A semi-supervised sentiment classification method for chinese microblogs. In *Proceedings of IJCNLP*, pages 455–462, 2013.

[Pak and Paroubek, 2010] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC*, pages 1320–1326, 2010.

[Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124, 2005.

[Salakhutdinov *et al.*, 2007] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of ICML*, pages 791–798, 2007.

[Speriosu *et al.*, 2011] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 53–63, 2011.

[Tan *et al.*, 2011] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *Proceedings of SIGKDD*, pages 1397–1405, 2011.

[Wang *et al.*, 2011] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach. In *Proceedings of CIKM*, pages 1031–1040, 2011.