A Network Framework for Noisy Label Aggregation in Social Media

Xueying Zhan[1], **Yaowei Wang**[1], **Yanghui Rao**[1,*],
Haoran Xie[2], **Qing Li**[3], **Fu Lee Wang**[4], **Tak-Lam Wong**[2]
[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2] Department of Mathematics and Information Technology,
The Education University of Hong Kong, Hong Kong SAR
[3] Department of Computer Science, City University of Hong Kong, Hong Kong SAR
[4] Caritas Institute of Higher Education, Hong Kong SAR
{zhanxy5, wangyw7}@mail2.sysu.edu.cn, raoyangh@mail.sysu.edu.cn,
hrxie2@gmail.com, qing.li@cityu.edu.hk, pwang@cihe.edu.hk, tlwong@eduhk.hk

Abstract

This paper focuses on the task of noisy label aggregation in social media, where users with different social or culture backgrounds may annotate invalid or malicious tags for documents. To aggregate noisy labels at a small cost, a network framework is proposed by calculating the matching degree of a document's topics and the annotators' meta-data. Unlike using the back-propagation algorithm, a probabilistic inference approach is adopted to estimate network parameters. Finally, a new simulation method is designed for validating the effectiveness of the proposed framework in aggregating noisy labels.

1 Introduction

Social media allows users to share their views, opinions, emotion tendencies, and other personal information online. It is quite valuable to analyze and predict user opinions from these materials (Wang and Pal, 2015), in which supervised learning is one of the effective paradigms (Xu et al., 2015). However, the performance of a supervised learning algorithm relies heavily on the quality of training labels (Song et al., 2015). In social media, many training data are collected via simple heuristic rules or online crowdsourcing systems, such as Amazon's Mechanical Turk (www.mturk.com) which allows multiple labelers to annotate the same object (Zhang et al., 2013). Due to the lack of quality control, it can be hard for a model to reconcile such noise in training labels.

This study aims to aggregate noisy labels by matching annotators and documents. Unlike other noisy label aggregation and integration tasks (or algorithms), such as Learning to Rank (LtR) and integrating crowdsourced labels which rely on accurate instance sources (Ustinovskiy et al., 2016) or confidence scores (Oyama et al., 2013), we only need features that can be obtained with a small cost ($i.e.$, topics). Compared with acquiring accurate instance sources or confidence scores, which is very hard, extracting topics can be done conveniently by many existing topic models. Note that label noise is not always random, as *adversarial noise* may occur in real-world environments when a malicious agent is permitted to select labels for certain instances (Auer and Cesa-Bianchi, 1998). For example, a fake annotator is purchased to promote defective goods by giving high ratings. Noisy labels in such a manner are extremely difficult to be handled (Nicholson et al., 2015). To validate the effectiveness of aggregating the aforementioned noisy labels, we propose to design a new simulation method in Section 4.

2 Related Work

To aggregate or refine noisy labels, several approaches have been proposed recently. Whitehill et al. (Whitehill et al., 2009) explored a probabilistic model to combine labels from both human labelers and automatic classifiers in image classification. Raykar et al. (Raykar et al., 2010) used a Bayesian approach for supervised learning over

*The corresponding author.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 484–490
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-2077

noisy labels from multiple annotators. Oyama et al. (Oyama et al., 2013) proposed to integrate labels of crowdsourcing workers using their confidence scores. Song et al. (Song et al., 2015) developed a single-label refinement algorithm to adjust noisy and missing labels. Ustinovskiy et al. (Ustinovskiy et al., 2016) proposed an optimization framework via remapping and reweighting methods to solve the problem of LtR with the existence of noisy labels.

Different from the previous study that modeled the difficulties of instances and the user's authority (Whitehill et al., 2009), we target at integrating multiple labels for each instance by estimating the *matching degree* of documents and annotators. Consequently, our work is applicable to aggregating individual sentiment labels in social media, where users under various scenarios ($e.g.$, character and preference) may express invalid or noisy sentiments to different topics.

3 Noisy Label Aggregation Framework

3.1 Problem Definition

The problem of noisy label aggregation is defined as follows: Given N documents (instances) annotated by M users (annotators) over C kinds of labels, we generate D topics by existing unsupervised topic models. Let $\mathbf{T} \in \mathbb{R}^{N\times D}$ be topics of all instances, where the i-th row of $\mathbf{T}$ ($i.e.$, $\mathbf{T}_i$) is the topic distribution of document i, and the size of $\mathbf{T}_i$ ($i.e.$, $|\mathbf{T}_i|$) is D. Let $\mathbf{F} \in \mathbb{R}^{M\times U}$ be features ($e.g.$, age and gender) of all annotators, where $\mathbf{F}_j$ is the feature distribution of user j and $|\mathbf{F}_j| = U$. To model different dimensions of document topics (D) and annotator features (U) jointly, we map $\mathbf{T}_i$ and $\mathbf{F}_j$ to K latent factors denoted as $\mathbf{S}_i$ and $\mathbf{A}_j$, $i.e.$, $|\mathbf{S}_i| = |\mathbf{A}_j| = K$.

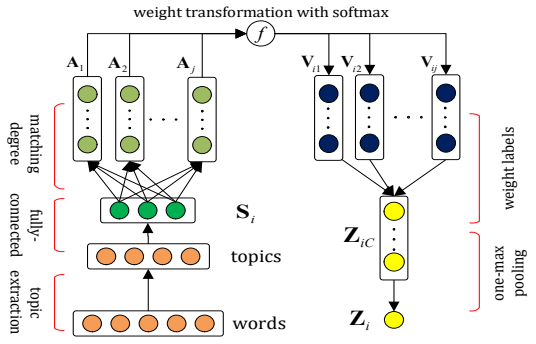


Figure 1: Our proposed network framework.

To estimate the ground truth label Z_i, we propose a novel network framework via aggregating the observable labels $\mathbf{V}_i$, as shown in Fig. 1. In our framework, the correctness of V_{ij} depends on whether annotator j *matches* document i.

3.2 Detailed Steps

Topic Extraction (TE): For document features, it is rough to use *tf* or *tf-idf* since they ignore the versatility of semantics among various contexts. Without considering the semantic units called topics, the accurate category of each document may be hard to access (Song et al., 2016). Short messages ($e.g.$, tweets) are prevalent in social media, which differ from normal documents insofar as the number of words is fewer and most words only occur once in each instance. To extract topics from such a sparse word space, we employ the Biterm Topic Model (BTM) by breaking each document into biterms and leveraging the information of the whole corpus (Yan et al., 2013).

Fully-connected Operation (FcO): There can be a large difference between dimensions of document topics and annotator features, so we need convert $\mathbf{T}$ and $\mathbf{F}$ to the same latent space. This step conducts linear transformation by introducing fully-connected weights $\mathbf{W_T} \in \mathbb{R}^{D\times K}$ and $\mathbf{W_F} \in \mathbb{R}^{U\times K}$, as follows: $\mathbf{S} = \mathbf{TW_T}$ and $\mathbf{A} = \mathbf{FW_F}$. The values of $\mathbf{S}$ and $\mathbf{A}$ are proportional to the label correctness probability.

Since more cohesive topics may indicate that the document's category is more concentrated and can be correctly annotated by more users, the topic distribution embeds key information on the document factors $\mathbf{S}$. To map $\mathbf{T}$ to $\mathbf{S}$ well, we propose the concept of *topic entropy* that acts as the constraint factor, by calculating the centralization of each document's topics: $H(d_i) = -\sum_{z=1}^{D} p(t_z|d_i)\ log_D\big(p(t_z|d_i)\big)$, where $p(t_z|d_i)$ is the probability of the z-th topic conditioned to document i, and D constrains the values ranging from 0 to 1. The lower $H(d_i)$, the higher the concentration of topics and the label correctness for document i. We thus infer the relationship between $\mathbf{S}_i$ and $H(d_i)$ as $||\mathbf{S}_i||_2 \propto 1/H(d_i)$, where $||\mathbf{S}_i||_2$ is the Euclidean norm of $\mathbf{S}_i$.

Matching Degree Calculation (MDC): This step calculates the *matching degree* of document i and annotator j, which is denoted as g_{ij} by the similarity/distance between latent factors $\mathbf{S}_i$ and $\mathbf{A}_j$. Intuitively, a basketball enthusiast j matches

close to a document i that contains the "basketball" topic, which indicates that the "matching degree" of i and j is high with a large similarity. The inner product is used here, and it can be replaced by distance measures.

Weight Transformation (WT): We employ transformation to distinguish different scores effectively. The activation function is *sigmoid* (softmax) or *tanh*. Since most document labels are assumed to be discrete independent variables, we encode V_{ij} as a binary vector. The higher g_{ij} of a label, the closer it is to the ground truth. Namely, we should weight these labels in such a way that if a label has high g_{ij}, its weight will be increased; meanwhile, other labels should be punished. For *sigmoid* and *tanh*, the punishment is $1 - w_{ij}$ and $-w_{ij}$, respectively. Take four labels, the transformation weight w_{ij} and $V_{ij} = (1, 0, 0, 0)$ as an example, the label weight via *sigmoid* is $V_{ij}^{new} = (w_{ij}, 1 - w_{ij}, 1 - w_{ij}, 1 - w_{ij})$.

Label Weighting (LW) and One-max Pooling: The final step is to output by integrating weighted labels, where the multiplicative combination is used in aggregation, and the output is the maximum one of aggregated labels $\mathbf{Z}_{iC}$.

3.3 Parameter Estimation

Since training labels may contain noise, it is inaccurate to employ the back-propagation method which uses the error between predicted and training labels as feedback for parameter estimation. Thus, we turn the estimation of model parameters $\mathbf{W_T}$ and $\mathbf{W_F}$ into a probabilistic problem. The graphical representation is illustrated in Fig. 2.

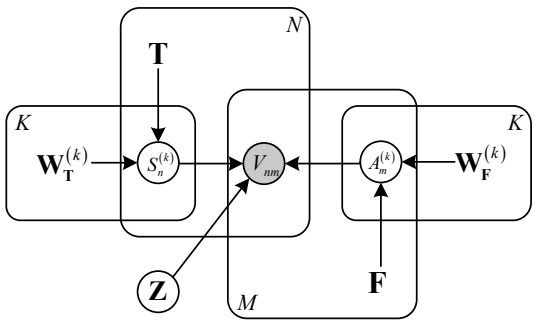


Figure 2: Probabilistic graphical representation.

Firstly, we define $\mathbf{W} = \{\mathbf{W_T}, \mathbf{W_F}\}$ for simplicity. Secondly, the parameter distribution is determined by the Maximum A Posteriori (MAP) principal: $\mathbf{W}^* = \arg\max_{\mathbf{W}} Pr(\mathbf{W}|\mathbf{V}, \mathbf{T}, \mathbf{F}) = \arg\max_{\mathbf{W}} \sum_{\mathbf{Z}} Pr(\mathbf{Z}) Pr(\mathbf{W}|\mathbf{V}, \mathbf{T}, \mathbf{F}, \mathbf{Z})$.

Finally, the following Expectation Maximization (EM) algorithm is used to estimate $\mathbf{W}^*$.

Initialization: We first initialize $\mathbf{W}$ randomly. The prior of ground truth $\mathbf{Z}$ can be set to $1/C$ or the frequency of each observable label.

Expectation (E): We then compute the expectation of the joint log-likelihood of observable and hidden variables given $\mathbf{W}$ ($i.e.$, the Q function), as follows: $Q(\mathbf{W}) = E[\ln Pr(\mathbf{V}, \mathbf{Z}, \mathbf{T}, \mathbf{F}|\mathbf{W})] = E[\ln Pr(\mathbf{V}|\mathbf{Z}, \mathbf{T}, \mathbf{F}, \mathbf{W})] + E[\ln Pr(\mathbf{Z}, \mathbf{T}, \mathbf{F}|\mathbf{W})]$.

Maximization (M): According to the Q function, the maximum likelihood of hidden variables is estimated by the gradient ascent method.

Alternation: The above E and M steps are alternately performed until the likelihood converges.

4 Experiments

4.1 Datasets and Baselines

As sentiment and emotion detection are widely studied in social media analysis (Wang and Pal, 2015), we test model performance based on the Stanford Twitter Sentiment (STS) and the International Survey on Emotion Antecedents and Reactions (ISEAR) corpus. The original STS dataset (Go et al., 2009) contains 1.6 million tweets that were automatically labeled as positive or negative using emoticons as labels, in which $80K$ (5%) randomly selected tweets were used to speed up the training process, $16K$ (1%) randomly selected tweets were used as the validation set, and 359 tweets were manually annotated as the testing set (dos Santos and Gatti, 2014). ISEAR is composed of $7,666$ sentences annotated by $1,096$ participants with different culture backgrounds (Scherer and Wallbott, 1994). These participants completed questionnaires about their 34 kinds of personal information ($e.g.$, age, gender, city, country, and religion), as well as their experiences and reactions over seven emotions. For the ISEAR corpus, we randomly selected 60% of sentences as the training set, 20% as the validation set, and the remaining 20% as the testing set.

We use the following models for comparison: Majority Voting (MV) (Sheng et al., 2008), Maximum Likelihood Estimator (MLE) (Raykar et al., 2010), and Generative model of Labels, Abilities and Difficulties (GLAD) (Whitehill et al., 2009). The baselines of MV and MLE are implemented by following (Sheng et al., 2008; Raykar et al., 2010), and GLAD is run by the software that is available in public at (Whitehill et al., 2009). We

also implement the multivariate version of GLAD, called MGLAD as the baseline for the ISEAR corpus with seven emotions. Although there are some more recent models on label aggregation (Oyama et al., 2013) or refinement (Song et al., 2015; Ustinovskiy et al., 2016), they either require additional features like users' reported confidence scores, or are only suitable to a corpus with one label for each document. To compare sentiment and emotion classification performance using the aggregated labels for training, we further apply the above noisy label aggregation models to a linear Support Vector Machine (SVM) with squared hinge loss (Chang and Lin, 2011). As shown in the existing studies with refined labels, the linear SVM performed well on sentiment classification of reviews (Pang et al., 2002) and tweets (Vo and Zhang, 2015).

4.2 Experimental Design

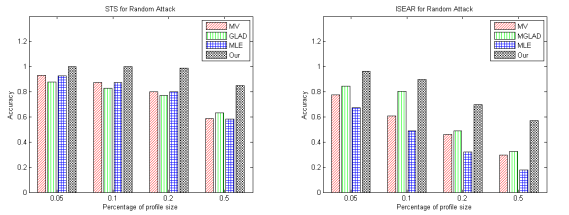
To evaluate the performance of noisy label aggregation models, each instance should be annotated by multiple users. Unlike previous studies which introduced a parameter to disturb ground truth labels (Sheng et al., 2008) or employed online crowdsourcing systems (Whitehill et al., 2009; Raykar et al., 2010) to generate noisy annotations, we design a new simulation approach by following the process of Profile Injection Attack in Collaborative Recommender Systems (Williams and Mobasher, 2006). This is because the existing methods can not assign multiple labels to each instance, or are difficult to generate virtual users and access their information ($e.g.$, age and gender). In particular, the following steps have been performed. First, we generate virtual users with different features, making them the neighbors of existing (actual) annotators. For each dimension of the actual annotators' features, we take the mean value if the attribute is continuous. For discrete attributes, we randomly select one type from the existing attribute values. If the dataset has no user features, we set it as a unit vector. Second, we generate document annotating vectors for virtual users. Each annotating vector is composed of three parts: annotating for filler instances (I_F), which is a set of randomly chosen filler instances drawn from the whole dataset, untagged instances ($I_\varnothing$), and the target instance (i_t). The purpose of setting I_F and $I_\varnothing$ is to make the virtual user looks like an ordinary annotator. We select three simulation types from Profile Injection Attack (Williams and Mobasher, 2006), $i.e.$, random, average, and love/hate. In the random method, the label for each instance $i \in I_F$ is drawn from a normal distribution around the annotations across the whole dataset, and the probability of labeling correctly to i is $1/C$. The corresponding probabilities are 0.5 and 1 for the average and love/hate methods, respectively. In all these methods, the annotation for i_t is randomly selected from wrong labels.

We tune the number of topics D and annotator features U by performing a grid search over all D and U values, with $D \in \{2, 3, 4, ..., 10\}$ on both datasets, $U = 34$ on ISEAR, and $U \in \{1, 10, 100, 500, 1000\}$ on STS that contains user ID only. The value of K is set to the maximum of D and U. Based on the performance on the validation set, we set $D = 6, U = 1000, K = 1000$ for STS, and $D = 2, U = 34, K = 34$ for ISEAR. For the sum of $|I_F|$ and $|i_t|$ ($i.e.$, attack size) for each virtual user, we set it as the mean number of annotations in actual users. The sum of selecting i_t in each simulation is called the profile size, and the percentage of the profile size is denoted as o. Following the previous criterion of choosing the noise rate (Auer and Cesa-Bianchi, 1998), we set $o \in \{0.05, 0.1, 0.2, 0.5\}$. According to (Ustinovskiy et al., 2016), each target instance except for those in I_F is annotated by three users. Thus, the number of virtual users is set to $2oN$. We set the parameter values of MV, MLE, and M/GLAD according to (Sheng et al., 2008; Raykar et al., 2010; Whitehill et al., 2009), and apply the grid search method to obtain the optimal parameters for SVM.

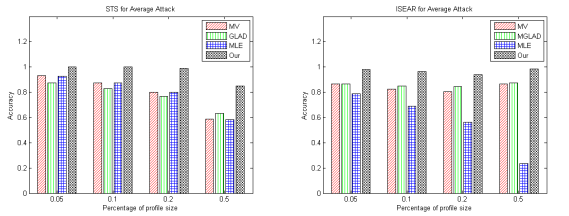
4.3 Results and Analysis

Firstly, we evaluate the noisy label aggregation performance of different models by comparing the proportion of estimated labels which match the actual categories ($i.e.$, accuracy). The results are shown in Fig. 3, which indicates that our model performs the best under various conditions. From the aspect of simulation methods, the accuracy of the random one is the lowest and the Love/Hate one is the highest, which is consistent to the correctly labeling probability for each method. The results of the random and average ones over STS are similar, because $C = 2$ on STS.

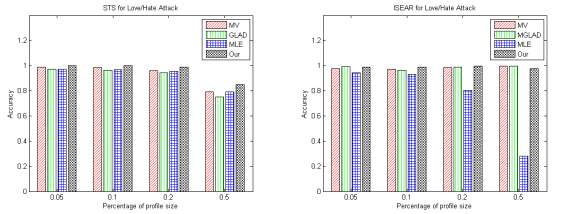
Particularly, our model performs better than



(a) Random over STS (b) Random over ISEAR

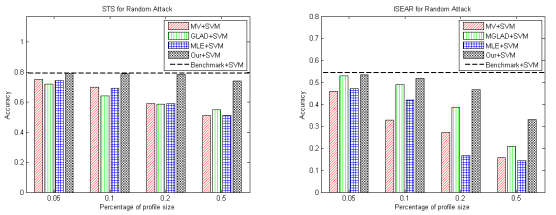


(c) Average over STS (d) Average over ISEAR

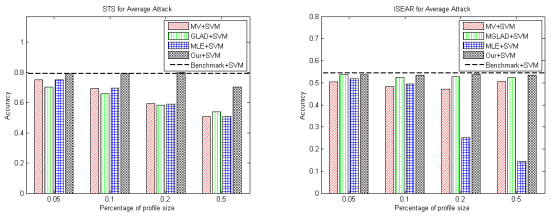


(e) Love/Hate over STS (f) Love/Hate over ISEAR

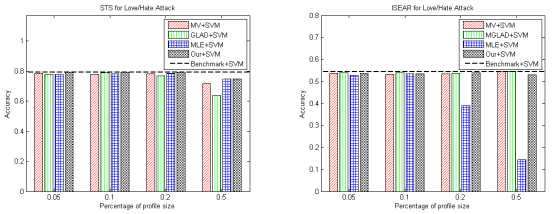
Figure 3: Label aggregation performance.



(a) Random over STS (b) Random over ISEAR



(c) Average over STS (d) Average over ISEAR



(e) Love/Hate over STS (f) Love/Hate over ISEAR

Figure 4: Classification performance.

baselines in aggregating noisy labels, especially when the noise scale becomes large. For instance, our model achieves 85% and 57% accuracies on STS and ISEAR when using the random method and $o = 0.5$, which indicates that our model has higher capability of recognizing adversarial noise (i_t). In the random method, we can also observe that the performance differences are more significant on ISEAR than STS. This is because ISEAR has more elaborate, $i.e.$, 34 kinds of observable user information, which validates the joint influence of users and documents on noisy label aggregation. To evaluate the performance differences statistically, we use the 12 groups of results over all methods and o values based on the conventional significance level ($i.e.$, p value) of 0.05. The p values of t-tests between our model and MV, M/GLAD, MLE are 0.0087, 0.0009, 0.0067 over STS, and 0.0535, 0.1037, 0.0007 over ISEAR, which indicates that the performance differences between our model and baselines are statistically significant on both datasets, except for MV and MGLAD in the love/hate method over ISEAR. The reason may be that each virtual user annotates around seven instances on ISEAR, and only one label is incorrect for the love/hate method, which makes the simple MV perform competitively.

Secondly, we compare the classification performance of SVM using labels from different noisy label aggregation models for training. The accuracies are shown in Fig. 4, in which dotted lines represent results on benchmark datasets without conducting the Profile Injection Attack process. Compared to other methods, the performance of SVM based on the aggregated labels from our model is almost closer to that of SVM using benchmark datasets. For the average method and $o = 0.2$ over STS, we can observe that SVM in conjunction with our model performs even better than that on the benchmark dataset. This is because emoticons are used as annotations for STS, which may introduce errors to the original labels.

5 Conclusions

In this paper, we proposed a network framework for noisy label aggregation by calculating the *matching degree* of documents and annotators. Experiments using a new simulation method of generating noisy labels validated the effectiveness of the proposed framework. As our model is linear in feature transformation, it is flexible to handle large-scale datasets. In the future, we plan to compare the model performance using different topic models, improve our model by exploiting the feedback of a small proportion of refined labels, and recruit actual participants to provide noisy labels.

Acknowledgments

The authors are thankful to the reviewers for their constructive comments and suggestions on this paper. The work described in this paper was supported by the National Natural Science Foundation of China (61502545), a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS11/E03/16), the Start-Up Research Grant (RG 37/2016-2017R), and the Internal Research Grant (RG 66/2016-2017) of The Education University of Hong Kong.

References

P. Auer and N. Cesa-Bianchi. 1998. On-line learning with malicious noise and the closure algorithm. *Annals of Mathematics and Artificial Intelligence* 23(1-2):83–99.

C.-C. Chang and C.-J. Lin. 2011. LIBSVM: A library for support vector machines. *Journal of ACM Transactions on Intelligent Systems and Technology* 2(3):27:1–27:27.

C.N. dos Santos and M. Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*. pages 69–78.

A. Go, R. Bhayani, and L. Huang. 2009. Twitter sentiment classification using distant supervision. *Cs224n Project Report* .

B. Nicholson, J. Zhang, V.S. Sheng, and Z. Wang. 2015. Label noise correction methods. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. pages 1–9.

S. Oyama, Y. Baba, Y. Sakurai, and H. Kashima. 2013. Accurate integration of crowdsourced labels using workers' self-reported confidence scores. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. pages 2554–2560.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 79–86.

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

K.R. Scherer and H.G. Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of Personality & Social Psychology* 66(2):310–328.

V.S. Sheng, F. Provost, and P.G. Ipeirotis. 2008. Get another lable? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. pages 614–622.

K. Song, W. Gao, L. Chen, S. Feng, D. Wang, and C. Zhang. 2016. Build emotion lexicon from the mood of crowd via topic-assisted joint non-negative matrix factorization. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*. pages 773–776.

Y. Song, C. Wang, M. Zhang, H. Sun, and Q. Yang. 2015. Spectral label refinement for noisy and missing text labels. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*. pages 2972–2978.

Y. Ustinovskiy, V. Fedorova, G. Gusev, and P. Serdyukov. 2016. An optimization framework for remapping and reweighting noisy relevance labels. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. pages 105–114.

D. Vo and Y. Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. pages 1347–1353.

Y. Wang and A. Pal. 2015. Detecting emotions in social media: A constrained optimization approach. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. pages 996–1002.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J.R. Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*. pages 2035–2043.

C.A. Williams and B. Mobasher. 2006. Thesis: Profile injection attack detection for securing collaborative recommender systems. *Service Oriented Computing & Applications* 1(3):157–170.

R. Xu, T. Chen, Y. Xia, Q. Lu, B. Liu, and X. Wang. 2015. Word embedding composition for data imbalances in sentiment and emotion classification. *Cognitive Computation* 7(2):226–240.

X. Yan, J. Guo, Y. Lan, and X. Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. pages 1445–1456.

J. Zhang, X. Wu, and V.S. Sheng. 2013. Imbalanced multiple noisy labeling for supervised learning. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*. pages 1080–1085.

A ISEAR's Annotator Features

The ISEAR corpus contains 34 kinds of personal information of participants. For clarity, the total set of annotator features is given below.

- Subject's backgrounds: (1) city, (2) Country, (3) ID suffix, (4) gender, (5) age, (6) religion, (7) practising religion, (8) father's job, (9) mother's job, and (10) field of study.
- Questionnaire: (11) when did the situation or event happen? (12) how long did you feel the emotion? (13) how intense was this feeling?
- Physiological symptoms of participants: (14) ergotropic arousal, (15) trophotropic arousal, and (16) felt temperature.
- Expressive behavior and other features of participants: (17) movement behavior, (18) laughing or smiling, (19) crying or sobbing, (20) nonverbal activity, (21) paralinguistic activity, (22) verbal activity, (23) moving against people or things, aggression, (24) did you expect the situation or event that caused your emotion to occur? (25) did you try to hide or to control your feelings so that nobody would know how you really felt? (26) did you find the event itself pleasant or unpleasant? (27) would you say that the situation or event that caused your emotion was unjust or unfair? (28) did the event help or hinder you to follow your plans or to achieve your aims? (29) who do you think was responsible for the event in the first place? (30) how did you evaluate your ability to act on or to cope with the event and its consequences when you were first confronted with this situation? (31) if the event was caused by your own or someone else's behavior, would this behavior itself be judged as improper or immoral by your acquaintances? (32) how did this event affect your feelings about yourself, such as your self-esteem or your self confidence? (33) how did this event change your relationships with the people involved? and (34) the "NEUTRO" attribute.

B Noisy Label Aggregation Algorithm

In our method of noisy label aggregation as shown in Algorithm 1, the cost of calculating $\mathbf{S}$ and $\mathbf{A}$ by *FcO* (line 6) is linear to the number of instances, *i.e.*, $O(NDK)$, and the total number of users, *i.e.*, $O(MUK)$, respectively. Before the EM iteration (lines 7 to 13), it takes $O(NM(K + C))$ to weigh all labels $\mathbf{V}$. For each iteration of EM (lines 14 to 15), the optimization with stochastic gradient descent takes $O(NMC+NK+MK)$ when each user annotates all documents. Assume that our algorithm converges after t iterations ($t < 10$ in our experiments), the overall time complexity is $O(NM(K + C)t)$, which is linear to the numbers of instances and users.

Algorithm 1 Noisy Label Aggregation

Input:
$\mathbf{V}$: Observable labels;
$\mathbf{F}$: Features of users;
ω: Words of documents;
δ: Threshold of convergence.
Output:
Aggregated labels.

1: $\mathbf{T} \leftarrow TE(\omega)$;
2: Initialize parameter $\mathbf{W}$ randomly;
3: $Q \leftarrow 0$;
4: **repeat**
5: $lastQ \leftarrow Q$;
6: $\{\mathbf{S}, \mathbf{A}\} \leftarrow FcO(\mathbf{W}, \mathbf{T}, \mathbf{F})$;
7: **for** each $i \in [1, N]$ **do**
8: **for** each $j \in [1, M]$ **do**
9: $g_{ij} = MDC(\mathbf{S}_i, \mathbf{A}_j)$;
10: $V_{ij}^{new} = WT(g_{ij}, V_{ij}, sigmoid)$;
11: **end for**
12: $\mathbf{Z}_{iC} = LW(\mathbf{V}_i^{new})$;
13: **end for**
14: $Q \leftarrow$ E-Step($\mathbf{Z}_{iC}$);
15: $\mathbf{W} \leftarrow$ M-Step(Q, $\mathbf{W}$);
16: **until** $|Q$ - $lastQ| < \delta$;
17: **return** Z_i, *i.e.*, the maximum one of $\mathbf{Z}_{iC}$.

C Gradient Derivation

Given the estimated value of $\mathbf{Z}_{iC}$, the Q function can be calculated by $Q(\mathbf{W}) = \sum_{ij} \mathbf{Z}_{iC} \ln V_{ij}^{new} + const$. Since the vector V_{ij}^{new} has two possible values when using *sigmoid* (*i.e.*, w_{ij} and $1 - w_{ij}$), the gradient of $\ln V_{ij}^{new}$ on parameter $W_T^{i,k}$ is $(V_{ij} - w_{ij})A_{jk}$, *i.e.*, $[w_{ij}(1 - w_{ij})]/w_{ij}A_{jk}$ and $[-w_{ij}(1 - w_{ij})]/(1 - w_{ij})A_{jk}$, respectively. Then, the gradient of Q on parameter $W_T^{i,k}$ can be derived as $\partial Q/\partial W_T^{i,k} = \sum_j \mathbf{Z}_{iC}(V_{ij} - w_{ij})A_{jk}$. Similarly, the gradient of Q on parameter $W_F^{j,k}$ is given by $\partial Q/\partial W_F^{j,k} = \sum_i \mathbf{Z}_{iC}(V_{ij} - w_{ij})S_{ik}$.