

Distilling Trustworthy Knowledge from Crowdsourced Data

BY

Sihong Xie

B.E., Software Engineering, Sun Yat-Sen University, 2008

M.E., Software Engineering, Sun Yat-Sen University, 2010

DOCTORAL THESIS

submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2016

Chicago, Illinois

Defense Committee:

Philip S. Yu, Chair and Advisor

Bing Liu

Brian Ziebart

Wei Fan, Big Data Lab, Baidu Research

Yuheng Hu, Information and Decision Sciences, UIC

This thesis is dedicated to my family.

ACKNOWLEDGMENTS

First I would like to thank my PhD advisor Professor Philip Yu, who is extremely supportive in all aspects. I learned from him about how to conduct cutting-edge research, and his experience and insights that are priceless in my future career for many years to come. I am grateful to Dr. Wei Fan, who led me into the field of data mining and has been a wonderful mentor and collaborator for many years. Professor Bing Liu is always full of novel ideas, which inspired many of my research works. I would like to thank Professor Brian Ziebart for introducing me to the fascinating area of machine learning. Professor Yuheng Hu has been very encouraging, and provided valuable advice for my academic job search.

My life at UIC has become very memorable through my interactions with the following co-workers, faculties, staffs and friends: Yan Xie, Yuchen Zhao, Xiangnan Kong, Xiaoxiao Shi, Fengjiao Wang, Shuyang Lin, Guan Wang, Xiaokai Wei, Weixiang Shao, Qingbo Hu, Chun-Ta Lu, Jingyuan Zhang, Jiawei Zhang, Bowen Dong, Bokai Cao, Chenwei Zhang, Vahid Noroozi, Hu Xu, Lei Zheng, Lifang He, Guoqiong Liao, Chuan Shi, Xi Zhang, Senzhang Wang, Guiling Li, Xiangli Chen, Zhiyuan Chen and Huayi Li.

Last but not least, my family has been supportive in my pursue of the degree. I will not be able to complete my degree without their supports and love. My deepest love goes to them.

SX

CONTRIBUTION OF AUTHORS

Chapter 1 introduces various research problems in crowdsourcing, along with a short history and examples. The organization of the thesis is also presented. Chapter 2 is a published manuscript (59) for which I am the primary author. Dr. Wei Fan and Prof. Philip S. Yu helped with the revision of the paper.

Chapter 3 is a published manuscript (62) for which I am the primary author. Xiangnan and other co-authors shared with me their opinions and helped me proof-read the paper.

Chapter 4 presents a published paper (60) for which I formulated the problem, proved the theorems and conducted the experiments. I started the research of this problem when I was an intern at IBM T.J Watson research under the supervision of Dr. Wei Fan. Prof. Jing Gao from University of Buffalo pointed me to the issue of overfitting in consensus maximization. All co-authors helped me proof-read the paper.

Chapter 5 presents a paper (61) for which I was the primary author. The other authors helped proof-reading the paper.

Chapter 6 presents a published paper (63) for which I was the primary author. The other authors helped proof-reading the paper.

Chapter 7 presents a published paper (64). for which I was the primary author. The other authors helped proof-reading the paper.

TABLE OF CONTENTS

<u>CHAPTER</u>		<u>PAGE</u>
1	INTRODUCTION	1
1.1	What is crowdsourcing	1
1.2	A short history of crowdsourcing research	3
1.3	Organization of the thesis	4
2	CROWD COMPETENCE-BASED KNOWLEDGE DISCOVERY	6
2.1	Problem statement	6
2.2	Consensus maximization	8
2.3	Iterative Re-weighted Consensus Maximization	11
2.3.1	A probabilistic View of IRCM	16
2.4	Experiments	19
2.4.1	Baseline Methods	20
2.4.2	Overall Performance Study	21
2.4.3	The effectiveness of filling up missing labels	23
2.4.4	Sensitivity Study	25
3	KNOWLEDGE DISCOVERY FROM MULTI-LABELED CROWD-SOURCED DATA	27
3.1	Problem statement	27
3.2	Preliminary	28
3.3	Multilabeled Consensus Maximization for Ranking Loss	30
3.3.1	Analysis of MLCM-r	33
3.4	Multilabel Consensus Maximization for microAUC	37
3.4.1	microAUC and its properties	37
3.4.2	MLCM-a	38
3.5	Experiments	42
3.5.1	Datasets	42
3.5.2	Evaluation Metrics	43
3.5.3	Baselines	44
3.5.4	Experiment settings	44
3.5.5	Results	45
4	THEORETICAL ANALYSIS OF MULTI-CLASS CROWDSOURCED DATA FUSION	48
4.1	Introduction	48
4.2	Overfitted Consensus Maximization	52
4.2.1	CM overfits	52

TABLE OF CONTENTS (Continued)

<u>CHAPTER</u>		<u>PAGE</u>
4.3	Class-Distribution Regularized Consensus Maximization . . .	54
4.3.1	Regularization over class distributions	54
4.3.2	Optimization of the Class-distribution Regularized Model . .	56
4.3.3	Projection to the Probabilistic Simplex	59
4.4	Generalization Error of RCM	60
4.5	Experimental Results	65
4.5.1	Experimental Settings	65
4.5.2	Overfitting in Consensus Maximization	67
4.5.3	Accuracy	70
4.5.4	Convergence Study	71
5	DISTILLING TRUSTWORTHY CROWDSOURCED RATINGS VIA SINGLETON SPAMMING ATTACK DETECTION	72
5.1	Background and Motivation	72
5.2	Singleton Review/Rating Spam Detection Model	75
5.2.1	The Model of Reviewer Behavior	75
5.2.2	A Correlated Temporal Anomalies Discovery based Approach	77
5.2.2.1	Time Series Construction	77
5.2.2.2	Correlated Abnormal Patterns Detection in Multidimensional Time Series	79
5.2.2.3	A Hierarchical Framework for Robust Singleton Review Spam Detection	84
5.3	Experiments	86
5.3.1	Review Data Description	86
5.3.2	Human Evaluation	87
5.3.2.1	Suspicious Store Detection	87
5.3.2.2	Singleton Reviews on a Detected Store	88
5.3.3	Spam Detection Case Study	89
5.3.3.1	First Case Study	89
5.3.3.2	Second Case Study	93
5.3.3.3	Third Case Study	94
6	DEBIASING CROWDSOURCED RATINGS VIA CONSENSUS RANKING DUAL TRANSFER	96
6.1	Introduction	96
6.2	Preliminary	98
6.2.1	Unsupervised bias correction	99
6.2.2	Semi-supervised bias correction	101
6.3	Correcting crowd bias via transfer learning	101
6.3.1	Two product-centric single transfer strategies	102
6.3.1.1	Product rating single transfer	102
6.3.1.2	Product ranking single transfer	102

TABLE OF CONTENTS (Continued)

<u>CHAPTER</u>		<u>PAGE</u>
6.3.2	A robust dual transfer approach	104
6.3.2.1	Anchor reviewer reliability estimation and confident anchor reviewer identification	104
6.3.2.2	Incorporating reviewer reliability in the single transfer strategy	105
6.3.3	Computational complexity analysis and incremental model up- date	106
6.4	Experiments	107
6.4.1	Datasets and Performance Metrics	107
6.4.2	Baselines and experimental protocol	108
6.4.3	Results	109
6.4.3.1	Sensitivity study	110
7	A CONTEXT-AWARE APPROACH TO DETECTION OF SHORT IRRELEVANT TEXTS	112
7.1	Introduction	112
7.2	Irrelevant content detection	115
7.3	Context-Agnostic Detection Models	119
7.3.1	Simple Language Model	119
7.3.2	Probabilistic Topic Models	120
7.3.3	Matrix Factorization based Models	121
7.3.4	Detection Signals based on Context-Agnostic Models	122
7.4	Context-Aware Detection Signals	123
7.4.1	Native Contexts	123
7.4.2	Early Detection of Irrelevant Comments	126
7.5	Experiments	129
7.5.1	Preparation of Datasets	129
7.5.2	Experimental Settings and Results	130
	COPYRIGHTS	139
	CITED LITERATURE	147
	VITA	153

LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
I	UNCERTAIN AND INCOMPLETE ANSWERS IN CROWDSOURC- ING	7
II	NOTATIONS FOR CONSENSUS MAXIMIZATION	9
III	NOTATIONS FOR DATA	12
IV	SUMMARY OF DATASETS	19
V	Overall Performance	21
VI	NOTATIONS	29
VII	NOTATIONS FOR MLCM-R	32
VIII	DATASETS	42
IX	RESULTS ON ENRON DATASET	46
X	RESULTS ON MEDICAL DATASET	46
XI	RESULTS ON RCV1 SUBSET 1 DATASET	46
XII	RESULTS ON RCV1 SUBSET 2 DATASET	47
XIII	RESULTS ON SLASHDOT DATASET	47
XIV	RESULTS ON BIBTEX DATASET	47
XV	RUNNING CM EXAMPLE	51
XVI	NOTATIONS	51
XVII	DATASETS AND BASE MODELS	66
XVIII	Overall Performance on Text Classification Tasks	68

LIST OF TABLES (Continued)

<u>TABLE</u>		<u>PAGE</u>
XIX	HUMAN EVALUATION RESULTS ON STORES	88
XX	HUMAN EVALUATION RESULTS ON REVIEWS	89
XXI	NOTATIONS	99
XXII	CHARACTERISTICS OF RATING DATASETS	108
XXIII	NOTATIONS	120
XXIV	CONTEXT-AGNOSTIC IRRELEVANT COMMENT DETECTION SIGNALS	123
XXV	DATASET CHARACTERISTICS	130

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
1	Bipartite graph used in CM	10
2	Overall accuracy comparison	21
3	Accuracies of EM-PMLA and EM-SLME with different percentage of missing labels completed	24
4	Accuracies of IRCM with varying number of iterations	26
5	Applying BGCM to multilabel prediction combination	30
6	Bipartite graph for MLCM-r and its collapse to group nodes	31
7	Comparison of ranking loss and microAUC	39
8	Hypothesis space of various methods: the tips of the triangles represent the bases \mathbf{e}_ℓ	54
9	Consensus Loss	68
10	Consensus loss and entropy of CM and RCM	69
11	Convergence of RCM	70
12	Bursty Patterns Detected in Store 24779	82
13	Contributions of reviewers	86
14	Anomaly detection on multi-scale multidimensional time series	90
15	Topic Hotness Trend	90
16	Bursty Patterns Detected in Store 24938	94
17	Overall comparisons of the proposed method and the baselines	110

LIST OF FIGURES (Continued)

<u>FIGURE</u>		<u>PAGE</u>
18	Sensitivity of CRDT (“best” indicates the baseline with the best performance)	110
19	Early detection as a real world problem	113
20	Distribution of length of comments	119
21	Context-Agnostic vs. Context-Aware methods	125
22	Transferred Contexts	128
23	Effectiveness of Native Context on the News dataset	131
24	Effectiveness of Native Context on the Blog dataset	132
25	Precision-Recall Curves for the context-agnostic and context-aware detections	137
26	Effectiveness of Transferred Contexts on the News dataset	138

SUMMARY

Crowdsourcing, a technique referring to sourcing data from a large crowd of human workers, has become an effective, efficient and scalable data collecting paradigm in domains like text and image tagging, spam detection, product rating and ranking, etc., that are easier for human beings than for computers. However, the crowdsourced data are usually noisy, incomplete, errorous due to incompetence of crowdsourcing workers, malicious injection of false information, etc., leading to trustworthiness issues in the crowdsourced data. In this thesis, I explore the issues in two settings: 1) crowdsourcing with a panel and 2) crowdsourcing in the wild.

Under the first setting, in Chapter 2, I model, infer and exploit the crowdsourcing workers' competences to better sift out the less accurate workers and emphasize the input from more reliable ones. The problem is modeled in a probabilistic way, with an EM (Expectation Maximization) algorithm inferring the latent variables. In Chapter 3, I handle crowdsourced multi-labeled annotations for images or texts. I propose two frameworks, one based on propagation over graph and the other based on a probabilistic model, to jointly infer the label correlations and the more trustworthy multi-labeled annotations. In Chapter 4, I explore the theoretical aspect of distilling trustworthy information from crowdsourced data with a panel. In particular, using the (SRM) Structural Risk Minimization framework, I show that previous methods either under-fit or overfit the crowdsourced data in seeking for trustworthy information. I then propose a large margin based framework to find the best parameter space for distillation of trustworthy information from crowdsourced data.

SUMMARY (Continued)

The situations are quite different when crowdsourcing information from a crowd in the wild, such as rating and ranking systems where a large number of unknown workers contribute their opinions. The challenges mainly come from malicious workers in the crowd and the goal is to detect and remove such workers. In Chapter 5, I study singleton rating attacks to such systems. Such attacks are widely adopted by attackers due to significant financial incentive and the well-covered trails of attacks. I propose a time series pattern mining based approach to collectively detect such attacks. Case studies show that the detected time windows do contain convincing evidences of singleton rating attacks, which can be reviewed and removed by domain experts to restore a more trustworthy ratings and ranking. In Chapter 6, I study various biases in the crowdsourced ratings due to sample selection bias and subjectivity. Then I propose a transfer learning based iterative bias correction method that is efficient in terms of human supervision. The method is shown to be able to restore the relative rankings of more pairs of products, and can converge in a few iterations. Lastly, in Chapter 7, I propose a framework based on dimension reduction to detect the irrelevant text comments crowdsourced on social medias. Such irrelevant comments usually carry untrustworthy information and can be harmful to the visitors of the social medias. I propose a novel concept called “local language model” that is effective in capturing the context-aware semantics of the terms, such that irrelevant comments can be more effectively detected.

CHAPTER 1

INTRODUCTION

1.1 What is crowdsourcing

Traditionally, extracting knowledge from data is done by domain experts. For example, botanists examine and recognize the pedals of iris (data) to determine the species of the specific plant, while editors read news articles (data) and put them under the suitable categories (knowledge). With huge amounts of data being generated every second, it is becoming more and more expensive to extract accurate knowledge in the speed and scale that the data are being generated, simply because that the labors of the human experts are not scalable. Crowdsourcing is a technique that uses a crowd of inexpensive non-experts to help alleviate such limitation in knowledge extraction. Specifically, human beings are used as “sensors” to measure the unknown information, a process called “crowdsourcing”, and the sensed (or crowdsourced) data are then processed to distill the desirable information. Formally, a human being involved in crowdsourcing is defined as a crowdsourcing worker, and the task of measuring a piece of information is defined as a crowdsourcing tasks. Examples includes:

- in news categorization or image digitalization, multiple non-experts can provide tags to the contents as they consume the contents. These tags become useful knowledge that describes the contents that would be otherwise very time consuming to be labeled by domain experts.

- in e-commerce and recommender systems, rating and textual comments are two types of human knowledge about the entities such as products and online contents. Ratings reflect the quality of the entities, while comments are free and unstructured texts that can provide many properties of the entities.
- in online question answering websites like Yahoo Answer, Quora and Stackoverflow, the crowd of users is driven by a reward system and answer a large number of questions in a timely and accurate manner. The knowledge in such systems is embedded in the answers.
- machine translation systems are trained over a huge amount of labeled data, which can take linguists many hours to label. With crowdsourcing techniques like Duolingo, large number of language learners are enrolled to translate many documents as exercises and later as training data for machine translation systems.
- in medicine, bioinformatics and molecular biology, it is important to understand how proteins fold in the three-dimensional space. The task is used to be solved by computational intensive bioinformatic algorithms, with less satisfactory results due to the complexity of the problem. Foldit is a crowdsourcing project that involves human workers in the process of protein folding, with gamification increasing the engagements of the workers in solving the folding problems. The crowdsourcing technique has led to superior performance in enzyme design, among other tasks.
- optical character recognition is important to many applications like book digitalization. To train a recognizer, again large amount of labeled training data is required and can be

obtained using crowdsourcing. Examples include reCaptcha that integrates verification and OCR.

Notice that since the human sensors are noisy and error-prone, it is desirable to have multiple human beings to measure the same piece of information to cancel out the noises. Thus, after the crowdsourcing process, an aggregation step is necessary to distill the final or conclusive measurement of the piece of information that is of genuine interest. It is this aggregation step that is the focus of this thesis.

1.2 A short history of crowdsourcing research

The research of crowdsourcing has been conducted over the past few decades and we can view the technique from multiple aspects.

- Worker quality estimation: one basic question in crowdsourcing is how good the workers in providing their input to the task. Depending on the formats of input data and the crowdsourced data, there have been various metrics to quantify the worker quality.
- Budget allocation: the resources such as money and time used to crowdsource the desired information are limited, therefore it is necessary to optimally allocate the resources to obtain the most desirable piece of information from the crowd.
- Crowdsourced data aggregation: once data are crowdsourced, one needs to extract conclusive information. The process of distilling useful information from the crowdsourced data is the focus of this thesis. Depending on the structures of the crowdsourced data, different aggregation methods are required.

- Incentive and reward system: given the same amount of resources, the mechanisms that can encourage more capable workers to provide high quality input in a shorter time is more desirable. It is an important question how to design efficient and effective crowdsourcing mechanisms.

1.3 Organization of the thesis

In this thesis, we study several challenges of the aggregation of crowdsourced data in the pursue of trustworthy information. In Chapter 2, we address the issue of the varying worker competence, and show that it is important and possible to extract and exploit worker competences to distill more reliable information from crowdsourced data. In Chapter 3, we investigate the aggregation of multi-labeled crowdsourced data, which is a more challenging task than the aggregation of binary or multi-class crowdsourced data. Chapter 4 is a short chapter that touches the theoretical aspect of multi-class crowdsourced data aggregation problems from the machine learning algorithmic perspective. These above three chapters assume a fixed panel of workers for the crowdsourcing tasks, and in the following three chapters, I relax the assumption and allow arbitrary number of workers to join in a task, and study the problem of trustworthy information distillation from data such crowdsourced. In Chapter 5, I tackle ratings of products crowdsourced from online reviewers on e-commerce websites, and propose a method that can jointly infer the reliability of the reviewers and calibrate the crowdsourced ratings in order to recover the underlying true ratings. In Chapter 6, I look at the extreme case, where a reviewer can contribute very few (one or two) ratings. The challenge is the sparsity of information of such reviewers, making it difficult to quantify the trustworthiness of the reviewers and thus

their ratings. I propose to address the challenge via a time series pattern mining approach to identify groups of suspicious reviewers who might have been hired to post extreme ratings to manipulate the rating systems. In Chapter 7, different from all previous problem settings where structured data are assumed, I handle unstructured textual data crowdsourced arbitrary number of workers. The goal is to quantify the relevance of each piece of text and leave only the more relevant ones.

CHAPTER 2

CROWD COMPETENCE-BASED KNOWLEDGE DISCOVERY

(This chapter includes the paper published in *Sihong Xie, Wei Fan and Philip S. Yu. “An Iterative and Re-weighting Framework for Rejection and Uncertainty Resolution in Crowdsourcing”*. In **SDM 2012**. (DOI: <http://epubs.siam.org/doi/abs/10.1137/1.9781611972825.95>).

2.1 Problem statement

Crowdsourcing has been a popular and inexpensive way to collect labels for such tasks. However, these low price tag labels usually come with a couple of drawbacks. First, the labels obtained could be noisy and uncertain due to the difficulty of the task, each labeler’s perception, lack of interest and insufficient background knowledge. Second, it is not uncommon for labelers to have different opinions. For example, given an article talking about Google’s acquisition of Motorola Mobility, it could be labeled as an information tech or a business article. The collected labels are usually inconsistent to some extent, such that one cannot simply treat these labels as ground truths. Third, missing and uncertain labels are ubiquitous in these tasks, especially for large scale data collected from real applications. In the example given above, a labeler might not be confident enough to label that article, so he/she simply omits the example. Collecting labels for all data is unrealistic for large datasets, or labelers are simply unable label all the data. For example, most of the Amazon reviewers review only a couple of products. Therefore,

TABLE I
UNCERTAIN AND INCOMPLETE ANSWERS IN CROWDSOURCING

Data	Ground_truth	label 1	label 2	label 3
\mathbf{x}_1	-1	1	1	-1
\mathbf{x}_2	1	na	na	1
\mathbf{x}_3	1	1	-1	-1

it is highly unlikely that all available labelers are able to provide labels for all queries. Lastly, labelers tend to have different but hidden competence for a task, depending on many factors such as background knowledge. Giving the same weights to all labelers is unreasonable, or at least suboptimal. As the labels provided by less competent labelers are more noisy and could therefore contaminate the labels from more competent labelers. Weighting labels from different labelers should perform better, yet how to learn the weights and exploit them effectively are nontrivial. Interestingly, it is hard to pre-assign a weight vector for labelers as their performance varies from task to task.

To summarize the challenges, we are given a set of uncertain, incomplete and inconsistent labels from which we wish to draw the ground truth while taking labelers' competencies into account. We use a toy example to demonstrate the challenges. In Table I, the third label of \mathbf{x}_1 contradicts the other two labels, which are incorrect, causing majority voting to fail. For \mathbf{x}_2 , the first two labels are missing, so there is not enough information to infer the true label. Note that given an instance, not all labelers necessarily provide a label. This is different from the semi-supervised setting (69)

2.2 Consensus maximization

We briefly introduce the Consensus Maximization (CM) framework (24). Quite different from traditional Bayesian model averaging approaches, CM combines the outputs of multiple models by considering their cross-example correlation and consistencies. Essentially, CM can resolve conflicts among labelers' opinions and infer the best possible ground truth in crowd-sourcing. In the proposed, we improve CM in order to infer each labeler's competence.

In CM, classification results of multiple labelers can be represented by a bipartite graph with two types of nodes: object and group nodes. \mathbf{x}_i is represented by an object node o_i . For model k , $k = 1, \dots, \ell$, there are 2 group nodes $h_{2 \times k-1}$ and $h_{2 \times k}$ associated with it, representing class -1 and 1, respectively. Object node o_i is connected to group node $h_{2 \times k-1}$ (or $h_{2 \times k}$) if instance \mathbf{x}_i is classified to -1 (or 1) by the k -th model. Object node o_i is associated with a random vector $\mathbf{u}_i = [u_{i1}, u_{i2}]$ for $i = 1, \dots, n$, $u_{i1} + u_{i2} = 1$, where u_{i1} and u_{i2} represent the probability that node o_i belongs to class -1 and 1, respectively. Similarly, group node h_j is associated with a random vector $\mathbf{q}_j = [q_{j1}, q_{j2}]$, $q_{j1} + q_{j2} = 1$ for $j = 1, \dots, v$, $v = 2\ell$ where q_{j1} and q_{j2} are the probabilities that h_j belongs to class -1 and 1, respectively. These random vectors can be organized into two matrices $U_{n \times 2} = [\mathbf{u}_1, \dots, \mathbf{u}_n]^\top$ and $Q_{v \times 2} = [\mathbf{q}_1, \dots, \mathbf{q}_v]^\top$. The meanings of U and Q will be clear in Equation 2.3 and Equation 2.4. The connections between object and group nodes are given by matrix $A_{n \times v} = [\mathbf{a}_1, \dots, \mathbf{a}_v]$, where $a_{ij} = 1$ if o_i is connected to h_j and 0 otherwise. Each group node has an initial class prediction $\bar{\mathbf{y}}_j = [1, 0]$ if node h_j represents class -1 and $\bar{\mathbf{y}}_j = [0, 1]$ otherwise. Note that $\bar{\mathbf{y}}_j$ are fixed and should

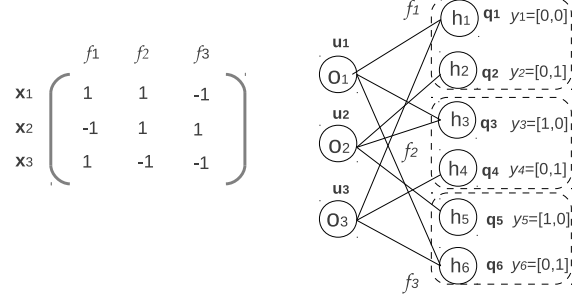
TABLE II

NOTATIONS FOR CONSENSUS MAXIMIZATION

o_i	Object node for \mathbf{x}_i
h_j	Group node for the $\lceil j/2 \rceil$ -th model
\mathbf{u}_i	Probability distribution over o_i
\mathbf{q}_j	Probability distribution over h_j
$\bar{\mathbf{y}}_j$	Initial probability distribution over h_j
\mathbf{a}_j	Indicates the o_i that h_j connects to

be distinguished from the label vector $\mathbf{y}_i = [y_i^1, \dots, y_i^\ell]$ of instance \mathbf{x}_i given by ℓ labelers. Let $\bar{Y}_{v \times 2} = [\bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_v]^\top$. Table II summarizes these notations.

Figure 1 demonstrates how to construct a bipartite graph in CM. Assume the missing labels in Table I are filled up and the resulting labels are shown in 1(a). The corresponding bipartite graph is shown in 1(b). Here we have 3 models (f_1 , f_2 and f_3) and two classes, so we need 6 group nodes sitting on the right of the bipartite graph. On the left hand side there are 3 object nodes $\{o_1, o_2, o_3\}$ representing \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 . \mathbf{x}_1 is classified to 1, 1 and -1 by models f_1 , f_2 and f_3 (row 1 in 1(a)). Therefore, o_1 is connected to $h_{2 \times 1 - 1}$, $h_{2 \times 2 - 1}$ and $h_{2 \times 3}$. Similarly, o_2 is connected to $h_{2 \times 1}$, $h_{2 \times 2 - 1}$ and $h_{2 \times 3 - 1}$, since it is classified to $-1, 1, 1$ by three models, respectively (row 2). Lastly, row 3 gives connections between o_3 and h_1 , h_4 and h_6 .



(a) Three instances with their labels given by three resending the labeling models

Figure 1. Bipartite graph used in CM

The objective of CM can be expressed as follows:

$$\min_{Q, U} \sum_{i=1}^n \sum_{j=1}^v a_{ij} \|\mathbf{u}_i - \mathbf{q}_j\|^2 + \alpha \sum_{j=1}^v \|\mathbf{q}_j - \bar{\mathbf{y}}_j\|^2 \quad (2.1)$$

$$\text{s.t.} \quad u_{i1}, u_{i2} \geq 0, u_{i1} + u_{i2} = 1, i = 1, \dots, n \quad (2.2)$$

$$q_{j1}, q_{j2} \geq 0, q_{j1} + q_{j2} = 1, j = 1, \dots, v$$

The term $\sum_{j=1}^v a_{ij} \|\mathbf{u}_i - \mathbf{q}_j\|^2$ enforces the probability distribution of the object node \mathbf{o}_i to be close to those of the group nodes it connects to and respect the original class predictions to some extent. Parameter α controls how much the consensus results \mathbf{q}_j to be consistent with the initial classification models' outputs $\bar{\mathbf{y}}_j$: a larger α encourages each \mathbf{q}_j to be closer to $\bar{\mathbf{y}}_j$. The optimization problem is solved via block-wise gradient descent, where Q and U are updated using:

$$Q^t = (D_v + \alpha I)^{-1}(A^\top U^{t-1} + \alpha \bar{Y}) \quad (2.3)$$

$$U^t = D_n^{-1} A Q^t \quad (2.4)$$

where $D_n = \text{diag}\{\sum_{j=1}^v a_{ij}, i = 1, \dots, n\}$, and $D_v = \text{diag}\{\sum_{i=1}^n a_{ij}, j = 1, \dots, v\}$, $\bar{Y}_{v \times 2} = [\bar{\mathbf{y}}_1, \dots, \bar{\mathbf{y}}_v]^\top$ and the superscript t on U and Q denotes the number of iterations. It is easy to see that the probability distribution \mathbf{u}_i is the average of the \mathbf{q}_j 's that are \mathbf{u}_i 's neighbors defined by A . Similarly, \mathbf{q}_j is defined by the average of probability distributions of the object nodes \mathbf{u}_i it connects to, plus $\alpha \bar{\mathbf{y}}_j$.

2.3 Iterative Re-weighted Consensus Maximization

We propose an IRCM framework (Iterative Re-weighted Consensus Maximization) to address the above challenges. Unlike previous methods (70; 45; 69), which ignore missing labels, IRCM first fills up the missing labels by building classifiers (f_1, \dots, f_ℓ in the chart) from the available labels (Y in the chart) and data (not shown), and then predicting missing labels. By doing so, one obtain a completed label matrix (" Y_c " in the chart). This step is shown in red dotted lines. Next, as shown in green solid lines in the chart, we infer ground truth (" \mathbf{y} " in the chart), by feeding weights of labelers (initially uniform distributed) and the completed label matrix to the proposed reweighted consensus maximization algorithm (RCM for short in the sequel). As the third step (blue dashed lines), the estimated ground truth and the completed label matrix are further used to update each labeler's competence (\mathbf{w} in the chart). IRCM goes back to the second step, and the iteration continues until it converges.

TABLE III

NOTATIONS FOR DATA	
$\mathbf{x}_i \in \mathbb{R}^d$	An instance of data
$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$	Collection of instances
$Z = [z_1, \dots, z_n]^\top$	Ground truth labels of \mathcal{D}
$\mathbf{y}_i = [y_i^1, \dots, y_i^\ell]$	Labels for \mathbf{x}_i given by labelers
$Y = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\top$	Label matrix
$Y_c \in \{-1, 1\}^{n \times \ell}$	Completed label matrix
$\mathbf{w} = [w_1, \dots, w_\ell]$	Competencies of labelers
$f : \mathbb{R}^d \rightarrow \{-1, 1\}$	Classification model

Assume that we are given a sample $\mathcal{D} = \{\mathbf{x}_i, i = 1, \dots, n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, with labels provided by ℓ labelers. These labels are denoted by $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\top$ with $\mathbf{y}_i = [y_i^1, \dots, y_i^\ell]$, which are the labels (probably missing) provided by ℓ labelers for \mathbf{x}_i . If an entry in \mathbf{y}_i is missing, it has value 0, otherwise, it is either -1 or 1. The completed label matrix is denoted by Y_c with entries taking value -1 or 1. The objective is to model labelers' competencies $\mathbf{w} = [w_1, \dots, w_\ell]$ and infer the ground truth labels $Z = [z_1, \dots, z_n]^\top$ of \mathcal{D} . Each labeler can be seen as a mapping from \mathbb{R}^d to $\{-1, 1\}$, denoted by $f_k(\mathbf{x}), k = 1, \dots, \ell$. The k -th column in Y is the output when f_k evaluated on a subset of \mathcal{D} , while the k -th column in Y_c is f_k evaluated on all data in \mathcal{D} . These notations are summarized in Table Table III.

It is natural that different labelers have different competencies in labeling data. Treating all labelers equally when aggregating their decisions is not optimal. Instead, one should pick up the outputs of the best ones to make final decisions, or weigh their outputs using competencies. Nonetheless, the original CM does not take competence or accuracy of base classifiers into

account and therefore, is not optimal. To incorporate weights of base classifiers in CM, there are two questions to be addressed. First, without any ground truth or supervision, how can one estimate the competence of each labeler? Second, even if the weights are available, how to use this information effectively in CM?

For the first question, instead of requiring accurate computation of competencies, we ask only for a rough estimation, such that the relative order is preserved. For example, if labeler Bob is doing better than labeler Alice, we only need an estimation of competencies which weighs Bob higher than Alice, instead of an estimation close to the real competence of the labelers. Assume that CM outputs a reasonable estimation of ground truth, we use this information to estimate the competencies of labelers, denoted by $w_j, j = 1, \dots, \ell$.

The second problem is more difficult to solve. A straightforward solution is to put the weights in the original CM algorithm by weighting $\|\mathbf{q}_j - \bar{\mathbf{y}}_j\|^2$ using $w_j, j = 1, \dots, v$. The higher the w_j , the more penalty will be incurred when \mathbf{q}_j deviates more from $\bar{\mathbf{y}}_j$, while a lower weight allows the final model \mathbf{q}_j to go relatively far away from its original distribution $\bar{\mathbf{y}}_j$. However, we experimentally find out that this seemingly reasonable and simple method does not work as expected.

We propose to weight the base models in CM via functional space sampling. Specifically, we emphasize “good” functions by sampling them more frequently. We generate a sample of functions by adding new columns to the completed label matrix Y_c . The resulting matrix is called the extended label matrix, also denoted by Y_c , but with new columns added. The new columns are generated entry-wise. That is, for each entry of a new column in Y_c , we randomly

Algorithm 1 IRCM with Missing Labels

```

1: Input:Data  $\mathcal{D}$ , incomplete label matrix  $Y$ 
2: Output:Inferred ground truths  $Z$ 
3: for all labeler  $j \in \{1, \dots, \ell\}$  do
4:   Build a classifier using  $Y(:, j)$  and  $\mathcal{D}$ .
5:   Complete  $Y(:, j)$  using this classifier to get  $Y_c(:, j)$ .
6: end for
7: Infer  $Z$  using CM from  $Y_c$ .
8: Estimate competencies of labelers  $\mathbf{w}$ .
9: while Not converge do
10:  Infer  $Z$  from  $Y_c$  and  $\mathbf{w}$  using Algorithm 2.
11:  Re-estimate  $\mathbf{w}$ .
12: end while

```

pick the entries in the same row in the first ℓ columns of Y_c , where the probability that an entry being picked is proportional to the weight on that column. More formally, suppose we need to add C new columns,

$$Y_c(i, j') = Y_c(i, j) \text{ with probability } w_j/K \quad (2.5)$$

where $j' = \ell + 1, \dots, \ell + C$, $j = 1, \dots, \ell$ and $K = \sum_{j=1}^{\ell} w_j$. In this way, the new columns can be seen as a weighted sample of the original ones. The IRCM algorithm is given in Algorithms 1 and 2. Note that Y_c in line 10 of Algorithm 1 refers to the completed label matrix without extension. The extension of Y_c is done in Algorithm 2 with a new estimation of \mathbf{w} .

The idea of column generation is similar to importance sampling (52), which emphasizes on certain instances by sampling them more frequently. Instead of sampling instances in the usual sample space, we are sampling functions in functional space. Suppose we have a space of functions $\mathcal{F} = \{f : \mathbb{R}^d \rightarrow \{-1, 1\}\}$. Each column in the label matrix can be seen as a function

Algorithm 2 RCM: Reweighted CM

Input: Y_c (the completed label matrix), \mathbf{w} (weights on each column), C (number of columns to generate)
Output: Inferred ground truths Z .
for $j = \ell + 1 \rightarrow \ell + C$ and $i = 1 \rightarrow n$ **do**
 Assign value to $Y_c(i, j)$ according to Eq.(Equation 2.5).
end for
Estimate Z by applying to Y_c .

$f \in \mathcal{F}$ evaluated on data points in \mathcal{D} . Though we have no access to the entire functional space \mathcal{F} , existing columns can be seen as a finite sample of functions from \mathcal{F} . We denote this sample of functions by \mathcal{F}_0 . Assume that the other good functions outside \mathcal{F}_0 are similar to the good ones in \mathcal{F}_0 , we would like to uncover more good functions in \mathcal{F} . Since the “goodness” and “badness” of a function in \mathcal{F}_0 are only estimated on a finite number of data, there are some uncertainties associated with them. There is no reason to assert that a good function in \mathcal{F}_0 can be generalized well to all instances. We also cannot simply reject “bad” functions from \mathcal{F}_0 , as they might be doing well on some unseen samples. A good function should be a mixture of functions in \mathcal{F}_0 with the constraint that they are closer to good functions than to bad functions in \mathcal{F}_0 . Therefore, we do not take one of the existing columns as a whole to form a new column, but sample at the entry-wise level (see Eq.(Equation 2.5)). We show in the next section that this functional space sampling method might also be viewed as finding a good similarity measure between instances. We denote the newly generated functions together with the original functions \mathcal{F}_0 as $\sigma(\mathcal{F}_0)$, which can be seen as equivalent to Y_c .

2.3.1 A probabilistic View of IRCM

We formulate IRCM using EM framework to interpret the functional space sampling method. Using this formulation we compare IRCM to the state-of-the-art crowdsourcing methods (70; 45) and bring out the distinction of IRCM. Given sample \mathcal{D} and the extended and completed label matrix Y_c , we need to infer two groups of parameters: competencies \mathbf{w} and ground truths Z . Let Z be the latent random variables, the goal is to maximize the log-likelihood

$$\Pr[Y|\mathcal{D}, \mathbf{w}] = \sum_{i=1}^n \ln \Pr[\mathbf{y}_i|\mathbf{w}, \mathbf{x}_i] \quad (2.6)$$

In the following, all probabilities depend on \mathcal{D} , but to keep the formula uncluttered, we only write down this dependency explicitly when necessary. Take the latent variables into account and use the total probability formula, $\Pr[\mathbf{y}_i|\mathbf{w}] = \mathbb{E}\{\Pr[\mathbf{y}_i|\mathbf{w}, z_i]\}$ where the expectation is taken over $\Pr[Z]$. It is difficult to maximize Eq.(Equation 2.6) directly, we instead maximize its lower bound $\sum_{i=1}^n \mathbb{E}\{\ln \Pr[\mathbf{y}_i|\mathbf{w}, z_i]\}$ by the concavity of the logarithmic function \ln and Jensen inequality.

Assume either $\Pr[z = 1] = 1$ or $\Pr[z = -1] = 1$ and consider the competence as a labeler's accuracy, namely, $w_j = \Pr[y_i^j = z_i]$ for any $i = 1, \dots, n$. Then we model $\Pr[\mathbf{y}_i|\mathbf{w}, z_i]$ using Bernoulli model as in (70)

$$\Pr[\mathbf{y}_i|\mathbf{w}, z_i] = \prod_{j=1}^{\ell} w_j^{1-|y_i^j-z_i|/2} (1-w_j)^{|y_i^j-z_i|/2}$$

and the lower bound of Eq.(Equation 2.6) can be written as

$$\sum_{i=1}^n \ln \left(\prod_{j=1}^{\ell} w_j^{1-|y_i^j - z_i|/2} (1 - w_j)^{|y_i^j - z_i|/2} \right) \quad (2.7)$$

M-step Maximize Take the derivative of Eq.(Equation 2.7) with respect to w_j and let it equal to 0 we get

$$w_j = \frac{\sum_{i=1}^n (1 - |y_i^j - z_i|/2)}{n} \quad (2.8)$$

E-step Compute Eq.(2.3.1) To compute Eq.(Equation 2.7), we need to know a particular assignment of labels to Z , which can be obtained via $\Pr[Z|\mathcal{D}]$. We propose a unique way to derive this probability. Similar to the work in (69) (refer to related work for more details), we impose a graph prior over the labels Z , namely, we construct a similarity graph of instances where nearby instances are assigned similar labels. Unlike their work, we incorporate labelers' competencies in the graph construction such that the inferred labels directly depend on the competencies.

In particular, given the competencies \mathbf{w} and original functions \mathcal{F}_0 , we can generate a sample of functions $\sigma(\mathcal{F}_0)$ (represented by the extended complete label matrix Y_c , see Eq.(Equation 2.5)). These generated functions can be seen as “virtual” labelers whose classification decisions are weighted combinations of real labelers. Next, based on $\sigma(\mathcal{F}_0)$, we derive a bipartite graph where functions (instances) are represented by group (object respectively) nodes and classification re-

sults are represented by the connections between group nodes and object nodes. This graph can be represented by a connection matrix A with $\ell + C$ columns. Re-write Eq.(Equation 2.3) and Eq.(Equation 2.4) in the form of random walk iterations,

$$Z^t = \tilde{P}Z^{t-1} + \alpha D_n^{-1} A (D_v + \alpha I)^{-1} \bar{Y}$$

where we restrict the soft cluster membership U to hard partition indicators Z . This formula is the same as the random walk with probability transition matrix $\tilde{P} = D_n^{-1} A (D_v + \alpha I)^{-1} A^\top$, encoding similarity between instances by the number of group nodes two instances share, normalized by some factors. The iteration converges to the solution maximizing the posterior:

$$\Pr[Z|\mathcal{D}] \propto \exp\{-Z^\top L Z\} \quad (2.9)$$

where $L = I - \tilde{P}$ is the graph Laplacian matrix. We can see that Z is inferred using the function sample $\sigma(\mathcal{F}_0)$.

The conjecture is that, with a better estimation of the competencies and sufficient sampling of the functional space \mathcal{F} , we would be able to recover the underlying cluster structure of the data. In this way, IRCM assigns a label to an instance by considering its neighbors' labels and *consensus* among labelers. Note that the methods in (45; 69; 70) assume that the labelers are independent given the sample \mathcal{D} . They also simply use labelers' competencies for weighted majority voting. In contrast, we are the first to introduce CM to model consensus among labelers in crowdsourcing. We are also the first to exploit labelers' competencies in the

TABLE IV

SUMMARY OF DATASETS		
Tasks	# Features	# Instances
pltcs_vs_bsns	1389	596
pltcs_vs_tech	1409	597
bsns_vs_tech	1326	597

consensus maximization framework. As we shall in Sections 2.4, CM outperforms weighted majority voting.

2.4 Experiments

To create datasets with instances labeled by multiple labelers, we ask 5 human annotators to label articles crawled from the Yahoo! news website¹. We choose 3 categories of news for experiments (politics (pltcs), business (bsns) and technologies (tech)). The gold standard labels are provided according to the classification of Yahoo! news. The reason of using these 3 classes is that the selected articles can usually be classified into more than one categories, therefore, it is more confusing for human to label, introducing more noise in the labels. For each category, we fetched roughly 300 articles from the website (900 articles in total). Then we mix them together and select 5 subsets of 90 articles randomly. Each labeler labels a subset of articles. Different labelers could label the same article and their opinions might contradict those of others. The resulting label matrix has 90% missing labels. We then create 3 binary classification problems by combining articles from any 2 out of all 3 categories. After text

¹<http://news.yahoo.com/>

preprocessing such as stop words elimination, TF-IDF transformation, we vectorize the articles in each problem, the properties of the data of 3 problems are summarized in Table Table IV.

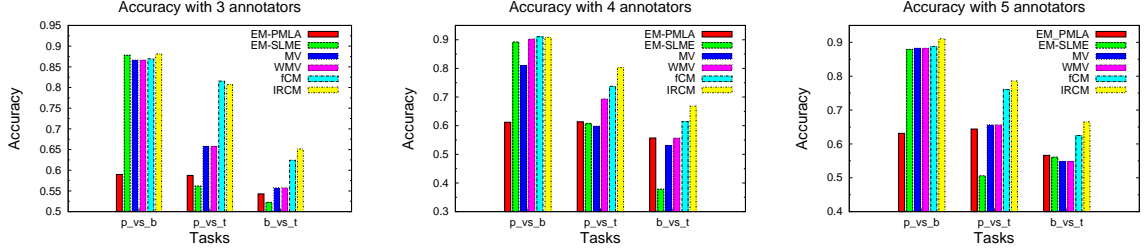
2.4.1 Baseline Methods

In the experiments, we compare the proposed algorithm with two state-of-the-art methods dealing with crowdsourcing data in (70) (EM-PMLA) and (45) (EM-SLME). We refer the readers to the related work for more details. In EM-PMLA, we use LBFGS¹ in the M-step. In EM-SLME, we use the L1-regularized logistic regression provided by Liblinear package (20). To compare these methods with the proposed method, we feed label matrices with different levels of completeness (see Section 2.4.2 and 2.4.3). Besides these sophisticated methods, we consider simplified versions of IRCM. The simplest version is just to use the first step of IRCM to fill up the missing labels and then apply majority voting (MV). Next, after the first round of IRCM, an initial set of competence estimates is obtained. Even without the iteration, we can apply these rough estimates to perform weighted majority voting (WMV). We also consider a degenerated case of IRCM which performs CM alone after filling up the missing labels without taking competence into consideration, we called this fCM. As we shall see, while filling up missing labels can help, introducing rough competence estimate has only marginal effect. Although fCM is more effective than majority voting, the iteration step to refine the competence estimates is

¹<http://users.eecs.northwestern.edu/~nocedal/lbfgs.html>

TABLE V. Overall Performance

	3 Labelers				4 Labelers				5 Labelers			
Tasks	p vs b	p vs t	b vs t	avg	p vs b	p vs t	b vs t	avg	p vs b	p vs t	b vs t	avg
EM_PMLA	<u>0.5899</u>	<u>0.5876</u>	<u>0.5429</u>	0.5735	<u>0.6120</u>	<u>0.6138</u>	<i>0.5572</i>	0.5943	<i>0.6312</i>	<u>0.6439</u>	<i>0.5662</i>	0.6138
EM_SLME	0.8780	0.5618	0.5224	0.6541	0.8923	0.6070	0.3786	0.6260	0.8792	0.5059	0.5611	0.6487
MV	0.8664	0.6576	0.5578	0.6939	0.8101	0.5983	0.5307	0.6464	0.8826	0.6566	0.5477	0.6956
WMV	0.8664	0.6576	0.5578	0.6939	0.9027	0.6941	0.5568	0.7179	0.8826	0.6566	0.5477	0.6956
fCM	0.8693	0.8154	0.6240	0.7696	0.9107	0.7374	0.6144	0.7542	0.8876	0.7605	0.6248	0.7576
IRCM	0.8809	0.8068	0.6513	0.7797	0.9082	0.8028	0.6683	0.7931	0.9107	0.7859	0.6654	0.7873



(a) ptcs vs bsns with 3 labelers (b) ptcs vs bsns with 4 labelers (c) ptcs vs bsns with 5 labelers

Figure 2. Overall accuracy comparison

indeed critical to boost up the performance in IRCM. We adopt libsvm¹ to fill up the missing labels.

2.4.2 Overall Performance Study

We show the overall accuracy of the proposed method and the baseline methods in Table Table V and Figure Figure 2. The number of columns generated in IRCM is fixed to 60 and the number of iterations is set to 15 (see next section for sensitivity study). For a given binary

¹<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

classification problem and a number of labelers (corresponding to one column in the table), we run IRCM 100 times and EM-PMLA 20 times using different combinations of labelers on all three classification problems. The average accuracy and standard variance are recorded for each of such combination. Then these statistics are averaged over all combinations of labelers. For example, when there are 3 labelers, then we have $\binom{5}{3} = 10$ combinations of labelers. All algorithms run on these 10 combinations and each gets 10 copies of accuracies and standard deviations. The accuracies under the header “3 Labelers” in Table Table V are the averages of these 10 copies. For EM-SLME, there is no randomness associated with it, so we do not need to repeatedly run it as other algorithms. If the standard variance is greater than 0.02 (0.01, respectively), then the corresponding accuracy is underlined (in italics font, respectively). For a given number of labelers, we also average performance of each algorithm over 3 classification problems, as shown in the columns with header “avg”.

From the table and the bar charts, we have the following observations. First, IRCM has the best performance 7 out of 9 tasks, with the exceptions that fCM slightly better than IRCM. Second, EM-PMLA and EM-SLME can sometimes perform even worse than weighted majority voting. Third, EM-PMLA is unstable, as it has the highest standard deviation in all methods across tasks. In contrast, the performance of IRCM are stable as none of the standard variance is higher than 0.01. Lastly, though sometimes EM-PMLA and EM-SLME have performance close to the proposed method, their accuracy can go down to a very low level. For example, in task *b_vs_t* with 4 labelers, EM-SLME is worse than random guess with only 37.86% accuracy.

The following conclusions can be drawn out of these observations, First, even using the first step of IRCM to fill up missing labels can be helpful for simple strategy like MV. In contrast, the weights learned by these EM-based methods are not effective when the labels are sparse. As we have seen, even the simple MV and WMV (using our weights) work better than weighted voting in EM-PMLA and EM-SLME. Second, The weights learned from IRCM are good indicators to pick up labelers. When there is a tie in majority voting with an even number of labelers, the weights can be helpful to decide which labelers to trust more. Third, As we show in formal analysis, CM is able to find a consensus results among labelers such that the generalization error bound is minimized. This is confirmed by the improved performance of fCM compared to MV and WMV. Lastly, IRCM iteratively refines and exploits competencies of labelers. The iterative re-weighted method achieves even better results than fCM, which does not consider labelers' competencies. This demonstrates that, in situations with multiple labelers such as crowdsourcing, competence is a critical factor to improving classification performance. This also shows that the functional space sampling method is an effective way to incorporate weights in the original CM.

2.4.3 The effectiveness of filling up missing labels

The proposed framework first predicts the missing labels before estimating competencies of labelers. Here, we demonstrate the effectiveness of this missing label filling up step. Different percentage of *missing labels* are filled up, and accuracies are obtained in the same way as we do in the last section. The number of EM iterations in EM-PMLA and EM-SLME is set to 15. From Figure Figure 3, we can see that in most of the cases, the performance of EM-PLMA (7

out of 9 cases) and EM-SLME (6 out of 9 cases) go up as missing label are filled up, therefore, the proposed framework does help existing state-of-the-art methods gain performance.

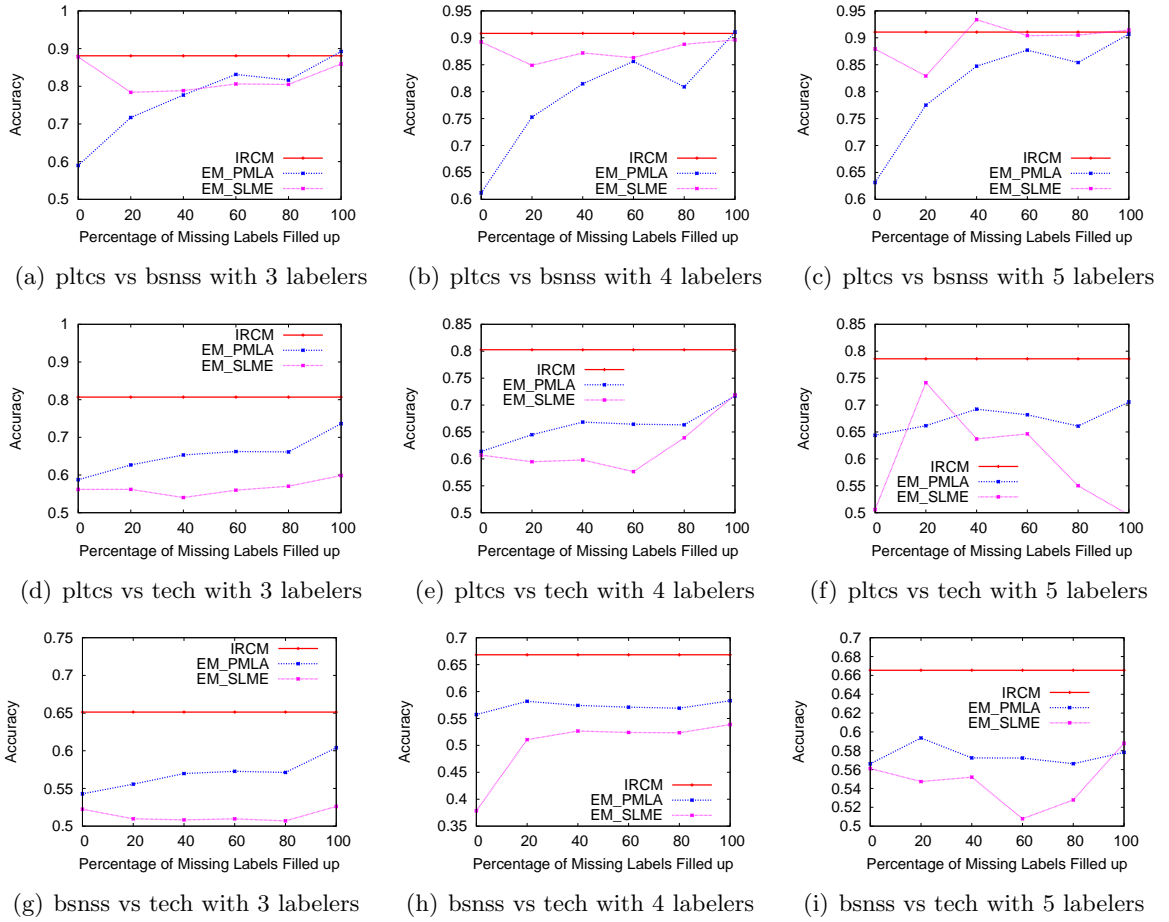
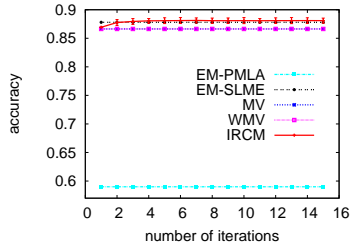


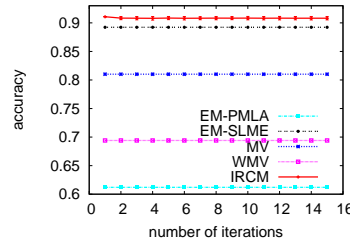
Figure 3. Accuracies of EM-PMLA and EM-SLME with different percentage of missing labels completed

2.4.4 Sensitivity Study

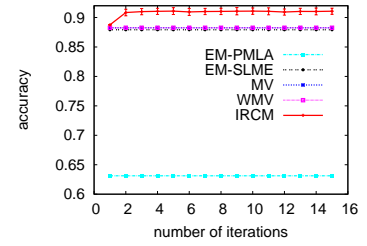
We study how the performance of IRCM varies with the number of iterations with different number of labelers. IRCM in the first iteration is equivalent to fCM with uniform weights on labelers. In Figure Figure 4, we plot the average accuracy with standard variance over 100 trials as error bars. From these figures, we can see that the accuracy of IRCM becomes better and better in 7 out of 9 cases, with two exceptions in Figure 4(b) and Figure 4(d), where the accuracies of IRCM go down slightly, but still better than those of the baseline methods. Therefore, we demonstrated the effectiveness of incorporating weights in CM in an iterative manner. The number of iterations required to converge is generally quite small (< 5).



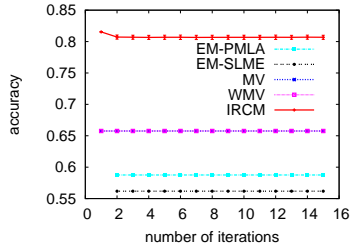
(a) pltcs vs bsns with 3 labels



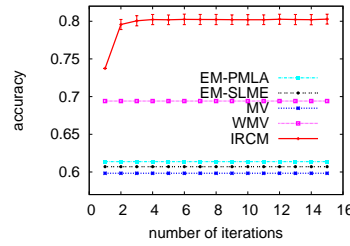
(b) pltcs vs bsns with 4 labels



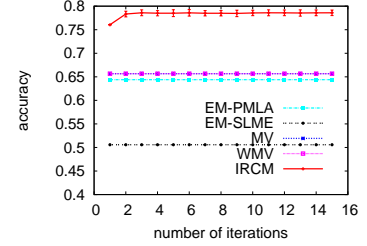
(c) pltcs vs bsns with 5 labels



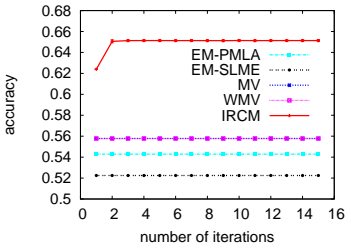
(d) pltcs vs tech with 3 labels



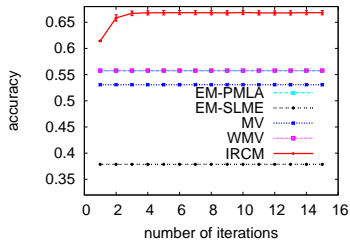
(e) pltcs vs tech with 4 labels



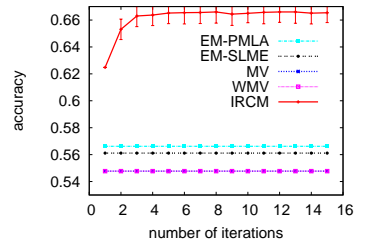
(f) pltcs vs tech with 5 labels



(g) bsns vs tech with 3 labels



(h) bsns vs tech with 4 labels



(i) bsns vs tech with 5 labels

Figure 4. Accuracies of IRCM with varying number of iterations

CHAPTER 3

KNOWLEDGE DISCOVERY FROM MULTI-LABELED CROWDSOURCED DATA

(This chapter includes the paper published in *Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, Philip Yu. “Multilabel Consensus Classification”. In **ICDM 2013**. ©2013 IEEE. Reprinted, with permission. (DOI: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6729628>)).*

3.1 Problem statement

Given the practical needs to combine multilabel predictions from multiple workers, we identify the following challenges. First, although state-of-the-art multilabeled classification methods show that label correlations can help improving classification performances, how to exploit label correlations solely using crowdsourced data has not been addressed before. Second, there are various evaluation metrics for multilabeled classification, such as microAUC, ranking loss, one error, etc. (14; 19), it is more desirable to design algorithms that can be proved to be optimal for a specific metric, as different applications require different quality measures. Although, in (14), they pointed out that optimizing different metrics translates into the modeling of different label correlations, it is non-trivial to align prediction combination methods with the modeling of label correlation in order to optimize a specific metric. There is no existing work that addresses the above issues.

In this chapter, we propose two algorithms that can model label correlations given only the crowdsourced multilabeled annotations. The algorithms are designed and proved to optimize two widely used but fundamentally different evaluation metrics, respectively. The first algorithm MLCM-r consolidates the annotations via maximizing model consensus and exploiting label correlations using random walk in the label space. The algorithm is proved to optimize ranking loss, which measures the quality of the predictions on a per instance basis (e.g. find relevant labels for a query in image search engine). Another important multilabel performance metric is microAUC, which treats all instances combined as a single prediction task (e.g. find tags to describe a set of images). Since a model that optimizes ranking loss might not be optimal for microAUC, it is necessary to develop an alternative model to distill knowledge from annotations to optimize microAUC. We propose a second algorithm called MLCM-a (MultiLabel Consensus Maximization for microAUC) for this purpose. MLCM-a is formulated as an optimization problem that regularizes annotation aggregation using partial correlations between labels, and we show that the objective optimizes microAUC.

3.2 Preliminary

In multilabel-label classification problems, the data are in the form of (\mathbf{x}, \mathbf{z}) , where \mathbf{x} is the feature vector of an instance and \mathbf{z} is the label vector. Suppose L is the set of all l possible labels, then \mathbf{z} is a vector of length $|L| = l$ and $z_\ell \in \{0, 1\}$ denotes the value of the ℓ -th label. Multilabel classification is different from multiclass classification. In multiclass classification, an instance have one and only one label, which can take more than two values (or classes). However, in multilabel classification, an instance can have more than one label, each of which

TABLE VI

NOTATIONS	
Symbol	Meaning
m	Number of multilabel classifiers
n	Number of instances
l	Number of labels
\mathbf{x}	An instance
\mathbf{z}	Ground truth labels of \mathbf{x}
Y^k	output of the k th model
\bar{Y}	Average of $Y^k, k = 1, \dots, m$
\mathbf{y}_i^k	prediction of the k th model for the i th instance
Y	
	Consolidated prediction of $Y^k, k = 1, \dots, m$
$\langle \cdot, \cdot \rangle$	inner product of two vectors
$ \cdot $	Determinant of a matrix
$\ \cdot\ $	Frobenius norm of a matrix
$\mathbb{1}[\cdot]$	indicator of a predicate
$\text{card}(A)$	cardinality of the set A

can take one and only one of the multiple values (classes). For example, an account on a social network (LinkedIn, Facebook, etc.) can have multiple labels such as “sex” and “is employed”, while there can be only one specific value for the label “is employed”. Multilabeled classification introduces various unique challenges, such as sparsity and imbalance of labels. Among these challenges, how to model and exploit label relationships to improve accuracy has been studied intensively in (13; 43; 46; 53; 74). There are various types of label relationships, and the simplest one is pair-wise correlation, which specifies how often two labels co-occur. There are also some more complicated label relationships, such as hierarchical organizations of labels or high order relationships. Recently, certain types of label relationship are shown to be connected to certain corresponding evaluation metrics. For example, it is shown in (14) that if one can compute the

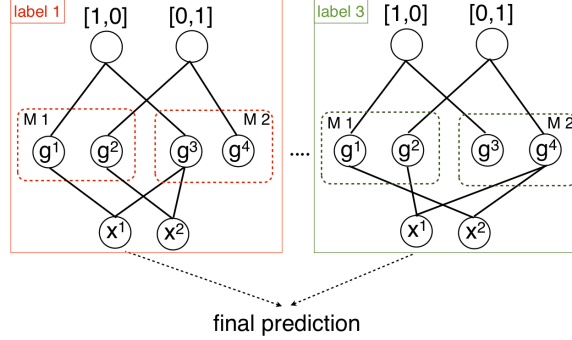


Figure 5. Applying BGCM to multilabel prediction combination

relevance score of each individual label given an instance, the ranking according to the scores would yield the minimum ranking loss. Conventional multilabeled classification algorithms mainly focus on how to exploit label correlations from training data. These methods cannot directly address the challenge of aggregating crowdsourced multilabeled annotations without access to training or test data.

3.3 Multilabeled Consensus Maximization for Ranking Loss

We propose MLCM-r that adopts BGCM (see the appendix) to aggregate crowdsourced multi-labeled annotations to minimize ranking loss. For simplicity, we assume that each label consists of two classes. In particular, we let the n by v ($v = m \times l$) connection matrix A encode the multilabeled predictions, where the $(i, (k-1) \times l + j)$ -th entry is 1 if the k -th worker predicts that the i -th instance takes class 1 on the j -th label, otherwise the entry is 0. Viewing A as a connection matrix between instances and labels, a bipartite graph can be constructed for MLCM-r. An example of the bipartite graph of MLCM-r for 2 instances, 3 classes and two

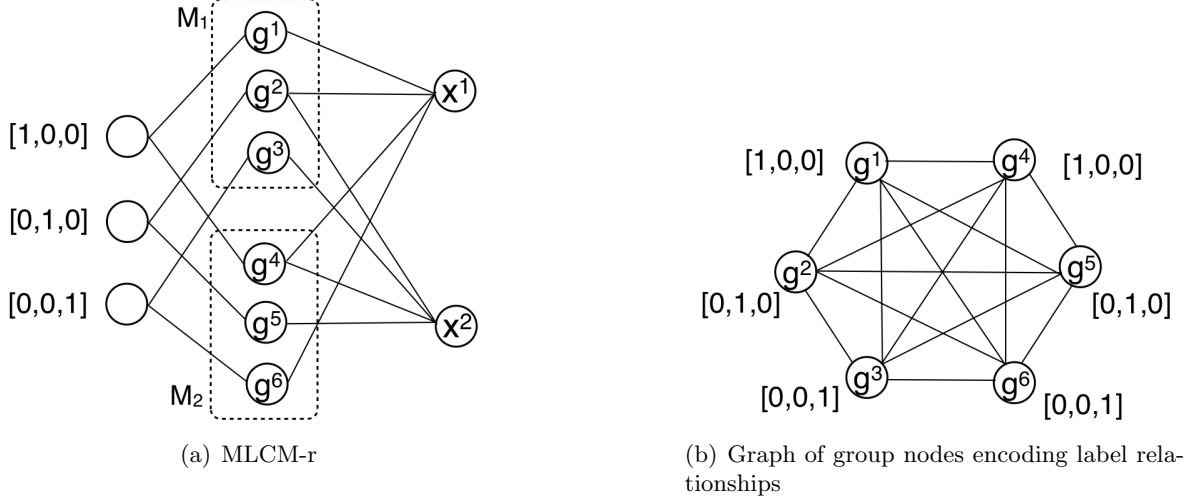


Figure 6. Bipartite graph for MLCM-r and its collapse to group nodes

workers is shown in Figure 6(a). Similar to the bipartite graph used by BGCM, the bipartite graph for MLCM-r has both group nodes and instance nodes, annotated by the letters g and x . However, there are some differences between two bipartite graphs. Surrounded by a rectangle with dashed line are the group nodes from a worker (e.g. the rectangle M_1 includes the group nodes from the first worker). A group node in Figure 6(a) represents a label instead of a class in Figure 5. An instance node in Figure 6(a) can be connected to more than one group nodes from a worker, naturally representing a multilabeled prediction. These differences between Figure 6(a) and Figure 5 bring more expressive power, as summarized below:

- The connections between an instance and *all* labels are fully given by a single graph in MLCM-r, instead of being broken down into multiple bipartite graphs in BGCM.

TABLE VII

NOTATIONS FOR MLCM-R	
Symbol	Meaning
A	$a_{i,j}$ is the prediction of label $(j \bmod l)$ on \mathbf{x}_i by the $\lfloor j/l \rfloor$
B	Label node class distribution
U	$u_{i\ell}$ is the probability that label ℓ is relevant to \mathbf{x}_i
Q	$q_{j\ell}$ is the probability of seeing label ℓ given label j
I_k	k dimensional identity matrix

- most importantly, the relationship between labels can be derived in MLCM-r using Figure 6(a), as shown in the graph of group nodes in Figure 6(b). We give more details of this property of MLCM-r in Section 3.3.1.

According to the newly defined A , we re-define the distributions associated with the nodes. $u_{i\ell}$ (the ℓ -th entry of \mathbf{u}_i) is now defined to be the probability of the i -th instance taking class 1 on the ℓ -th label. Similarly $q_{j\ell}$ is defined as the probability of seeing the ℓ -th label given the j -th label (the reason of this definition is explained in the next section). If the j -th group node represents the ℓ -th label, it is connected to a label node with distribution \mathbf{b}_j , which has 1 on its ℓ -th entry and 0 for the other entries, let $B = [\mathbf{b}'_1, \dots, \mathbf{b}'_v]'$ similarly as in BGCM. With the re-defined variables and constants (see Table VII), MLCM-r maximizes model consensus by solving a similar optimization problem in Equation 2.1. The closed form optimal solution is given in Equation 3.1 and Equation 3.2, which infer and exploit label correlations to minimize ranking loss, as analyzed in the next section.

3.3.1 Analysis of MLCM-r

In this section, we first analyze the property of MLCM-r, which is shown to perform a random walk in label space and thus infer label correlations. Then we introduce ranking loss, which is connected to MLCM-r to show that MLCM-r indeed optimizes ranking loss.

According to (24), a closed form solution for the optimization problem is

$$Q^* = (I_v - D_\lambda D_v^{-1} A' D_n^{-1} A)^{-1} D_{1-\lambda} B \quad (3.1)$$

where $D_v = \text{diag}(\mathbb{1}' A)$, $D_n = \text{diag}(\mathbb{1}' A')$, $\mathbb{1}$ is a column vector with all entries being 1. $D_\lambda = D_v(\alpha I + D_v)^{-1}$ and $D_{1-\lambda} = \alpha(\alpha I + D_v)^{-1}$. After Q^* is obtained, U^* is obtained using

$$U^* = D_n^{-1} A Q^* \quad (3.2)$$

Equation 3.1 actually solves a problem similar to personalized pagerank over the graph in Figure 6(b). The graph consists of group nodes from Figure 6(a), with edges indicating strength of connection between group nodes. In particular, the graph expresses the chances of co-occurrence of two labels in terms of the proportion of instances that have both labels simultaneously. The results of the random walk is simply the probabilities that one node hits another node during a specific random walk. Since the nodes represent labels, the solution can be seen as the probabilities of seeing one label when starting from another label. We analyze this intuition more formally. We wish to establish the solution of the random walk $Q_{\ell j}^*$ as the probability of

seeing the j -th label given the ℓ -th label. Fixing j and looking at Equation 3.1 in a column-wise perspective, for $j = 1, \dots, v$, we obtain

$$Q_{\cdot,j}^* = (I_v - D_\lambda D_v^{-1} A' D_n^{-1} A)^{-1} D_{1-\lambda} B_{\cdot,j} \quad (3.3)$$

where $Q_{\cdot,j}$ and $B_{\cdot,j}$ are the j -th column of Q and B , λ_j is the j -th diagonal entry of D_λ . Let $S = D_v^{-1} A' D_n^{-1} A$, which is the transition matrix. Each row of S is a probability distribution where S_{ij} is the transition probability from group node i to group node j . By the identity $(I - S)^{-1} = \sum_{t=0}^{\infty} S^t$, we can re-write Equation 3.3 as

$$Q_{\cdot,j}^* = \left(\sum_{t=0}^{\infty} (D_\lambda S)^t \right) D_{1-\lambda} B_{\cdot,j} \quad (3.4)$$

Out of Equation 3.4, we can construct a random walk where a walker takes from 0 to infinitely many steps to eventually settle down at any one of the group nodes for label j (note that there can be multiple group nodes for label j given multiple workers, e.g. both group nodes g^1 and g^4 represent label j). At each step, the walker can choose to settle down with probability $1 - \lambda_i$ at group node i , or to take one more transition with probability λ_i , given the current position being the i -th group node. $(D_\lambda S)^t$ can be interpreted similar to traditional random walk. For the base case, $(D_\lambda S)^0 = I$ gives the probability that one starts from any one of the group nodes and reaches *any* nodes in zero step. Assume $((D_\lambda S)^{t-1})_{ij}$ is the probability that the person reaches node j starting from node i in $t - 1$ steps. Then $((D_\lambda S)^t)_{ij} = \lambda_i \sum_{k=1}^v S_{ik} ((D_\lambda S)^{t-1})_{kj}$, which can be interpreted as the walker chooses to con-

tinue walking with probability λ_i , and ends up at node j with probability $\sum_{k=1}^v S_{ik}((D_\lambda S)^{t-1})_{kj}$. By induction, $(D_\lambda S)^t$ gives the probabilities of moving from one node to another in t steps without settling down.

Given the above interpretation and fixing $j = 1$, we obtain

$$\begin{aligned} ((D_\lambda S)^t D_{1-\lambda} B)_{i1} &= \sum_k ((D_\lambda S)^t)_{ik} (1 - \lambda_k) B_{k1} \\ &= \sum_k ((D_\lambda S)^t)_{ik} (1 - \lambda_k) \mathbb{1}(B_{k1} = 1) \end{aligned}$$

Note that B is a matrix with 0 or 1 entries and $B_{k1} = 1$ iff the k -th group node for label 1. Also note that $(1 - \lambda_k)$ is the probability of settling down at the k -th group node. Then a summand in the above summation is the probability of starting from group node i and settling down after t steps of transition at the k -th group node belonging to label 1. The sum of these probabilities is the probability that settling down at *any* of the group nodes for label 1. $Q_{\cdot 1}^* = (\sum_{t=0}^\infty (D_\lambda S)^t) D_{1-\lambda} B_{\cdot 1}$, and $Q_{\ell 1}^*$ gives the probability that, starting from the ℓ -th group node, one reaches any group nodes of class 1.

According to Equation 3.2, the (i, ℓ) -th entry of U is

$$\begin{aligned} U_{i\ell} &= \frac{1}{d_i} \sum_{j=1}^v a_{ij} Q_{j\ell}^* \\ &= \sum_{k=1}^c \frac{n_k}{d_i} \left(\sum_{j=1}^v \mathbb{1}[B_{jk} = 1] \frac{a_{ij}}{n_k} Q_{j\ell}^* \right) \\ &= \sum_{k=1}^c p(k|\mathbf{x}_i) p(\ell|k, \mathbf{x}_i) \end{aligned}$$

where $n_k = \sum_j a_{ij} \mathbb{1}[B_{jk} = 1]$, which is the total number of group nodes of label k that \mathbf{x}_i connects to. $p(k|\mathbf{x}_i) = n_k/d_i$ is the probability that \mathbf{x}_i has label k according to m base models. $p(\ell|k, \mathbf{x}_i) = \sum_{j=1}^v \mathbb{1}[B_{jk} = 1](a_{ij}/n_k)Q_{j\ell}^*$ is simply the average of $Q_{j\ell}^* \mathbb{1}[B_{jk} = 1]$, which is probability of going from label k to label ℓ . These two probabilities depend on \mathbf{x}_i due to the term d_i and n_k , which depend on the connectivity between \mathbf{x}_i and the group nodes. Therefore, MLCM-r computes the probabilities $p(y_\ell = 1|\mathbf{x}_i)$.

The above results connects MLCM-r to ranking loss, which is defined below. between pairs of labels. Let P_i be the set of relevant labels for \mathbf{x}_i , and N_i the set of irrelevant labels. $P_i \times N_i$ is the set of all pairs of relevant and irrelevant labels. Given the relevance scores $f(\ell, \mathbf{x}_i)$ of label ℓ of \mathbf{x}_i , $\ell = 1, \dots, l, i = 1, \dots, n$, ranking loss is defined as

$$\text{ranking loss} = \sum_{i=1}^n \sum_{\ell \in P_i, \ell' \in N_i} \frac{\mathbb{1}[f(\ell, \mathbf{x}_i) \leq f(\ell', \mathbf{x}_i)]}{\text{card}(P_i \times N_i)} \quad (3.5)$$

In (14), it was proved that the expected ranking loss is minimized by the ranks of the relevance scores, which is defined as the posterior probability $p(y_{i\ell} = 1|\mathbf{x}_i)$. In other words, so long as the probability $p(y_{i\ell} = 1|\mathbf{x}_i)$ can be estimated accurately, one should be able to achieve a low ranking loss. But this is what exactly MLCM-r does, as we show above. Therefore we conclude that MLCM-r minimizes ranking loss.

3.4 Multilabel Consensus Maximization for microAUC

In this section, we propose another algorithm for multilabeled annotation aggregation. This algorithm differs from the first one in that it optimizes microAUC, which is both theoretically and practically different from ranking loss.

3.4.1 microAUC and its properties

AUC (Area Under the Curve) is a binary classification metric for situations where one class greatly out-numbers the other class. In multilabeled classification, an instance usually has only a small number of all labels. For example, in text tagging, there can be thousands of tags, yet an article usually has only a couple of tags. Since there can be much more irrelevant label than relevant labels, AUC can be adopted in the multilabeled setting, where the metric is called microAUC. Formally, the label matrix $Z = [\mathbf{z}'_1, \dots, \mathbf{z}'_n]'$ for n instances has a total of $n \times l$ entries. Let P be the set of positive (relevant) entries and N the set of negative (irrelevant) entries, $\text{card}(P) \ll \text{card}(N)$. Given a list of relevance scores of all entries, microAUC (11; 26) is defined as

$$\text{microAUC} = \sum_{i \in P} \sum_{j \in N} \frac{\mathbb{1}[f(i) > f(j)]}{\text{card}(P) \times \text{card}(N)} \quad (3.6)$$

where $f(i)$ is the relevance score of entry i . Observe that microAUC is the ratio between the number of correctly ordered pairs and that of the total pairs. A fundamental difference between two metrics is that, ranking loss does not compare the ranks between labels of two different instances, while microAUC compares the ranks of all possible pairs of labels, no matter they

are from the same instance or not. In this sense, approaches that optimize ranking loss does not necessarily optimize microAUC.

3.4.2 MLCM-a

We examine microAUC more closely to motivate the method to be proposed. In Figure 7, we graphically demonstrate the differences between ranking loss and microAUC. Assume we have 3 labels and 3 instances $\{\mathbf{x}_1, \dots, \mathbf{x}_3\}$. The ground truth labels of the 3 instances are layed out as in the 3×3 label matrix $Z = [\mathbf{z}'_1, \dots, \mathbf{z}'_3]'$ where \mathbf{z}_i is a row vector of the values of all labels for \mathbf{x}_i . The values of the entries for a label are grouped in a rectangle, while each row represents the labels of an instance. Ranking loss accounts the pairwise relationship between the labels *within* an instance. Therefore, in Figure 7(a), 3 pairs of relative ranks of entries will contribute to the ranking loss, as indicated by arrows pointing from relevant labels to irrelevant ones within each instance. However, there are more pairs of entries that microAUC accounts for. Given a relevant label for an instance, microAUC pairs it with *all* other irrelevant labels of all instances, including itself. In Figure 7(b), example pairs of relevant and irrelevant entries are indicated by arrows labeled by letters. We do not draw all pairs in Figure 7(b) to avoid untidiness. Note that arrow **a** indicates the sort of pairs of entries considered by ranking loss. Arrow **b** indicates pairs of entries within a label for different instances, and arrow **c** points from a label of an instance to a different label of a different instance.

Pairs of entries indicated by arrow **b** or **d** must have been handled by reasonable workers, who can assign a label to relevant instances. Pairs of entries indicated by arrow **a** consist of only a small portion of all pairs if n is large, due to the sparsity of relevant labels. Therefore,

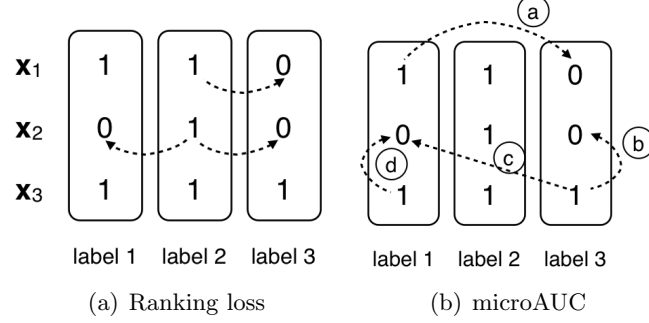


Figure 7. Comparison of ranking loss and microAUC

the major challenge in optimizing microAUC is how to enforce preference of one label over other labels across different instances, such as the pairs indicated by arrow **c**. Without loss of generality, take Figure 7(b) as an example, given two instances \mathbf{x}_2 and \mathbf{x}_3 , we need to estimate the posteriors $p(y_j = 1|\mathbf{x}_i)$ for $i \in \{2, 3\}, j \in \{1, 3\}$, in order to derive preferences between relevant and irrelevant labels. Suppose with high probability that $p(y_1 = 1|\mathbf{x}_3) > p(y_1 = 1|\mathbf{x}_2)$ (arrow **b**). If label 1 and 3 are correlated, we would like the estimations of $p(y_3 = 1|\mathbf{x}_2)$ and $p(y_3 = 1|\mathbf{x}_3)$ (arrow **d**) to reflect such correlations to certain extent according to how much these two labels are correlated. This can be achieved by enforcing $p(y_3 = 1|\mathbf{x}_2)$ and $p(y_3 = 1|\mathbf{x}_3)$ to satisfy similar label preference, namely, $p(y_3 = 1|\mathbf{x}_3) > p(y_3 = 1|\mathbf{x}_2)$ with certain high probability according to the correlation between two labels. As a by-product, we have $p(y_3 = 1|\mathbf{x}_3) > p(y_1 = 1|\mathbf{x}_2)$ (arrow **c**) and therefore enforce label preference across labels and instances to follow the correlation between labels. In summary, we can tackle the challenge in two steps:

- estimate the correlations between labels accurately
- optimize microAUC by estimating label relevance according to the label correlations estimated above.

We describe the second step first. A representation of label correlation is needed. Here we model all pairs of label correlation using the partial correlation matrix of labels.

Definition 1 (Partial Correlations). *Partial correlation between labels ℓ and ℓ' is the correlation between two labels given the other labels.*

The partial correlations can be captured by an $l \times l$ symmetric matrix Ω^{-1} , which is called precision matrix in multivariate statistics. To estimate the relevance scores of the labels Y following the estimated Ω^{-1} , we set up an optimization objective that combines two goals. The first goal is to minimize certain loss function employed in model combination algorithms. The second goal is to maximize the correlation between the label partial correlation (matrix Ω^{-1}) and the empirical label correlation (given by $Y'Y$). The latter goal can be formulated by the inner product of two matrices: $\text{tr}(Y'Y\Omega^{-1}) = \text{tr}(Y\Omega^{-1}Y')$. The optimization problem is

$$\min_Y J = \|\bar{Y} - Y\|^2 + \text{tr}(Y\Omega^{-1}Y') \quad (3.7)$$

where Y is the final aggregated labels, and \bar{Y} is the simple average of the crowdsourced multi-labeled annotations. Taking the derivative of J with respect to the i th row of Y , \mathbf{y}_i , we obtain

$$\frac{\partial J}{\partial \mathbf{y}_i} = -2(\bar{\mathbf{y}}_i - \mathbf{y}_i)' + 2\Omega^{-1}\mathbf{y}_i' \quad (3.8)$$

Equating the above derivative to 0, we get

$$\mathbf{y}_i = \sum_{k=1}^m \mathbf{y}_i^k (\Omega^{-1} + mI_l)^{-1} = m\bar{\mathbf{y}}_i (\Omega^{-1} + mI_l)^{-1} \quad (3.9)$$

The label correlation Ω is now taken into account when producing the consolidated predictions.

Note that we assume Ω is given in the above optimization problem. In reality, Ω is usually unknown and has to be estimated from data. Below we show how to estimate Ω using MLE. In order to set up an MLE problem, one needs to assume density functions for the observed data given the parameter. Here we treat $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ as the data independently generated from the normal density

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \Omega) = \frac{1}{C} \exp\left\{-\frac{1}{2}\mathbf{y}_i' \Omega^{-1} \mathbf{y}_i\right\} \quad (3.10)$$

where $C = (2\pi)^{l/2} |\Omega|^{1/2}$ is the normalization constant. The likelihood of Y given Ω is

$$p(Y|\Omega) = \prod_{i=1}^n p(\mathbf{y}_i|\Omega) = \frac{1}{C^n} \exp\left\{-\frac{1}{2} \sum_{i=1}^n \mathbf{y}_i' \Omega^{-1} \mathbf{y}_i\right\} \quad (3.11)$$

According to the MLE of the covariance matrix of multivariate Gaussian distributions, Ω is estimated as

$$\hat{\Omega}_{MLE} = \frac{1}{n} Y' Y$$

Now we can put the above two steps together to build the MLCM-a algorithm (Algorithm 3).

Algorithm 3 MLCM-a

-
- 1: **Input:** Predictions from base models $\{Y^1, \dots, Y^m\}$
 - 2: **Output:** Consolidated predictions Y .
 - 3: Estimate $Y = \bar{Y}$
 - 4: **for** $t = 1 \rightarrow T$ **do**
 - 5: Estimate covariance $\Omega = \frac{1}{n}Y'Y$
 - 6: Estimate Y using Eq.(Equation 3.9)
 - 7: **end for**
-

TABLE VIII

DATASETS			
datasets	# of instances	# of features	# of labels
enron	1702	1054	53
medical	978	1449	45
rcv1 subset 1	2997	47337	101
rcv1 subset 2	2951	47337	101
slashdot	3782	1101	22
bibtex	3701	1995	159

3.5 Experiments**3.5.1 Datasets**

With 6 datasets widely used in multilabel classification community, we demonstrate the effectiveness of the proposed methods. Their properties are summarized in Table Table VIII. Note that these datasets have a relatively large number of labels, it can be very time-consuming for multilabel classification models to account for complex label correlations during training.

3.5.2 Evaluation Metrics

We further include certain popular metrics to give some empirical observations as guidance for the use of the proposed methods in practice. For a multilabel classifier f , the ranking of the labels of an instance \mathbf{x} is given by $\{\ell'_1, \dots, \ell'_c\}$ where $f(\ell'_1, \mathbf{x}) \geq f(\ell'_2, \mathbf{x}) \geq \dots \geq f(\ell'_c, \mathbf{x})$ and $f(\ell, \mathbf{x})$ is the relevance score of the label ℓ to \mathbf{x} according to f .

- *one error*: an error occurs when the top-ranked label is not a relevant one, otherwise there is no error, regardless of how the other labels are ranked.

$$one\ error = \frac{1}{n} \sum_{i=1}^n \mathbb{1}[\ell'_1 \notin \mathbf{z}_i] \quad (3.12)$$

where ℓ'_1 is the most relevant label to \mathbf{x}_i according to f and \mathbf{z}_i is the set of relevant labels of \mathbf{x}_i . $\mathbb{1}[\cdot]$ is 1 if and only if the statement in the brackets is true. The lower the one error, the better an algorithm performs.

- *average precision*: evaluates the precision averaged over all instances and all possible numbers of retrieved labels.

$$average\ precision = \frac{1}{n} \sum_{i=1}^n \frac{1}{c} \sum_{s=1}^c \frac{|\{\ell'_1, \dots, \ell'_s\} \cap \mathbf{z}_i|}{s} \quad (3.13)$$

where $\{\ell'_1, \dots, \ell'_s\}$ is the top s labels retrieved for instance \mathbf{x}_i (the subscript i is ignored in the retrieved labels). The higher the average precision, the better an algorithm performs.

3.5.3 Baselines

We compare the proposed methods to two baselines. First, evaluation metrics are computed for each base model, the averaged performance of base models (denoted by BM in the sequel) are obtained as one of the baselines. Second, we also report the performance of majority voting method (MV in the sequel). The predictions of all base models are averaged and evaluation metrics are computed using the averaged predictions. By comparison of these two methods, we would be able to see how model averaging improves the performance in the multilabel setting. This confirm the effectiveness of ensemble method used in multilabel classification (46; 53). Since we do not assume the base models have considered label correlation in training or testing phase, while majority voting cannot discover and exploit label correlations, the proposed methods should be able to outperform the base models and majority voting.

3.5.4 Experiment settings

A base model is obtained by first randomly shuffling the dataset, followed by 10-fold CV. For each dataset, we training 10 such base models. For each base model, one can calculate its performance using the metrics mentioned above. The predictions of these base models are used as input to MV, MLCM-r and MLCM-a, each of which produces consolidated predictions. Based on the consolidated predictions, we can evaluate the performance of MV, MLCM-r and MLCM-a. This experiment is repeated for 10 times for each dataset and the averaged performance is reported next.

3.5.5 Results

We show the performance of the proposed algorithms and baselines in Table IX-Table XIV. We have a couple of observations. First, by comparing results in the rows for BM and MV, one can see that combining model can boost the performance of multilabel classification, even only using the simplest way of combination (simple averaging here). The maximum improvements of MV over BM are 41% and 12.8% for ranking loss and microAUC, respectively. This is not surprising, as this method is widely used in ensemble multilabel classification methods like (46; 53; 66; 48; 73; 49; 75). Second, by comparing the results of the proposed methods and simple averaging, we observe that simple averaging is not sufficient to fully exploit label correlations, especially when the base models do not take the correlations into account. The maximum improvement of either the proposed algorithms over MV is 45% in ranking loss and 20% in microAUC. Third, out of 6 tasks, MLCM-r wins MLCM-a 5 times in ranking loss, with a maximum of 12% improvement, and MLCM-a wins MLCM-r 4 times in microAUC, with a maximum of 5.8% improvement. The above comparisons show the superiority of the proposed methods over the baselines for multilabel predictions combination tasks, and also how to choose from the proposed methods when different metrics are considered. Lastly, besides ranking loss and microAUC, the proposed methods also outperform the baselines with the other two metrics, and this shows the wide applicability of the proposed methods.

TABLE IX

RESULTS ON ENRON DATASET				
Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.7342	0.5024	0.2967	0.4592
MV	0.8289	0.3398	0.1848	0.6020
MLCM-r	0.8759	0.6233	0.1003	0.5252
MLCM-a	0.8931	0.2675	0.1070	0.6556

TABLE X

RESULTS ON MEDICAL DATASET				
Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.8887	0.2041	0.0989	0.7953
MV	0.9321	0.1410	0.0582	0.8639
MLCM-r	0.9536	0.1327	0.0494	0.8750
MLCM-a	0.9556	0.1322	0.0530	0.8649

TABLE XI

RESULTS ON RCV1 SUBSET 1 DATASET				
Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6194	0.6036	0.3373	0.3218
MV	0.6787	0.4792	0.2838	0.4164
MLCM-r	0.7867	0.3554	0.2316	0.5017
MLCM-a	0.8069	0.3120	0.2605	0.4967

TABLE XII

RESULTS ON RCV1 SUBSET 2 DATASET

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6220	0.5652	0.5652	0.3659
MV	0.6678	0.4730	0.4730	0.4389
MLCM-r	0.7581	0.2955	0.2955	0.5146
MLCM-a	0.8020	0.2830	0.2830	0.5073

TABLE XIII

RESULTS ON SLASHDOT DATASET

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.7377	0.4875	0.2062	0.5856
MV	0.8210	0.4085	0.1482	0.6689
MLCM-r	0.8782	0.4123	0.1203	0.6736
MLCM-a	0.8702	0.3887	0.1289	0.6800

TABLE XIV

RESULTS ON BIBTEX DATASET

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6620	0.5469	0.3095	0.3575
MV	0.7266	0.4329	0.2508	0.4567
MLCM-r	0.8668	0.4713	0.1599	0.4828
MLCM-a	0.8645	0.3790	0.1755	0.4937

CHAPTER 4

THEORETICAL ANALYSIS OF MULTI-CLASS CROWDSOURCED DATA FUSION

(This chapter includes the paper published in *Sihong Xie, Jing Gao, Deepak Turaga, Wei Fan, Philip Yu. “Class-Distribution Regularized Consensus Maximization for Alleviating Overfitting in Model Combination”. In **KDD 2014**. (DOI: <http://dl.acm.org/citation.cfm?doid=2623330.2623676>)).*

4.1 Introduction

Combining multiple supervised and unsupervised models can be desirable and beneficial, or sometimes even a must. For example, in crowdsourcing, privacy-preserving data mining or big data applications, there could be only predictions from multiple models available, with raw features of the data being withheld or discarded. One has to merge the output of these models to obtain the final classification or clustering results. On the one hand, there are various new consensus-based solutions, such as those proposed in (24; 59; 57; 21; 34; 50; 71). One common idea that these algorithms share is to learn a model that has highest prediction consensus among base models. On the other hand, simple model combination algorithms, such as majority voting (23), that do not pursue model consensus are portrayed as baselines inferior to the algorithms seeking consensus. These comparisons give the illusion that the more consensus one can achieve, the more likely the consolidated predictions will be accurate. One might

ask: are the consolidated predictions that achieve maximal consensus the best choice? Could these consensus-based methods overfit the noisy and limited *observed* data, leading to results inconsistent with the *true* data distribution? After all, the goal of classification/clustering is to produce discriminative predictions (4; 55).

In this paper, we study the above questions based on the Consensus Maximization framework (24) (CM for short in the sequel), due to its generality and effectiveness. We first present a running example of CM in Table Table XV to demonstrate that solely maximizing the consensus can lead to undesirable results. Suppose we have 5 instances in 2 classes, whose ground truth labels are shown in the first column of the table. There are 2 supervised models (M_1 and M_2) and 2 unsupervised model (M_3 and M_4). A supervised (resp. unsupervised) model predicts the class (resp. cluster) labels of all instances. The predictions from a model are shown under the header with the model’s name. Note that neither the correspondence between class labels and cluster labels, nor the correspondence between cluster labels from different clustering models is known. We describe the details of CM later and for the moment, one can think of CM as a black box that consolidates the predictions of base models and outputs predictive posteriors $p(y = 1|\mathbf{x})$ that achieve maximal consensus among base models. For majority voting (MV for short in the sequel), it simply averages the predictions from supervised models (predictions of unsupervised models cannot be used by MV because the correspondence between classes and cluster labels is unknown). The consolidated predictions produced by CM and MV are shown in the last two columns of the table.

From this running example, one can see that CM makes more correct predictions than MV does. However, the posteriors $p(y = 1|\mathbf{x})$ produced by CM tend to be closer to the decision boundary and the margins between $p(y = 1|\mathbf{x})$ and $p(y = 0|\mathbf{x})$ are quite small. We have two observations. First, according to the margin-based generalization error analysis (42), a smaller margin of posterior class distributions between different classes leads to a higher *empirical margin risk*, which contributes to the overall generalization error. If one can produce consolidated predictions with a large posterior margin, a tighter upper bound on the generalization error can be obtained. Second, if the hypothesis space for a model combination algorithm has large capacity (measured by VC-dimension, growth function or covering number, etc.), then the upper bound of generalization error is also higher. One may incorporate certain relevant prior knowledge of the data to shrink the size of the hypothesis space. For example, for multi-class single-label classification, desirable consensus predictions should be discriminative in the sense that an instance belongs to one and *only* one class. Our goal is to reduce empirical margin risk and the capacity of the hypothesis space of model combination methods such as CM, and obtain a smaller upper bound on the generalization error.

We propose a family of regularization objectives over class distribution to reduce generalization errors. As a solid instance, we add regularization objectives to CM to obtain Regularized Consensus Maximization (RCM). In terms of algorithmic effectiveness, though the regularization introduces many tuning parameters and makes the optimization problem not jointly convex, we develop a simple yet efficient approximation to the regularization term without introducing additional parameters. An alternative optimization procedure is developed to find

TABLE XV
RUNNING CM EXAMPLE

y	Predictions				MV Results	CM Results
	M_1	M_2	M_3	M_4	$P(y = + \mathbf{x})$	$P(y = + \mathbf{x})$
+	+	+	C1	R0	1	0.5073
+	+	+	C0	R0	1	0.5097
+	−	+	C0	R0	.5	0.5024
−	+	−	C1	R1	.5	0.4946
−	−	−	C1	R1	0	0.4873

TABLE XVI

NOTATIONS

$\mathbf{x}^i \in \mathcal{X} = \mathbb{R}^d$	An instance of data
$\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$	Collection of instances
$Y = [y_1, \dots, y_n]^\top$	Ground truth labels of \mathcal{D}
$U_{n \times c}$	Membership indicators of data
$Q_{v \times c}$	Membership indicators of groups
$A_{n \times v}$	Affinity matrix
c	Number of classes
\mathbf{u}^i	The i -th row of matrix U
\mathbf{u}_j	The j -th column of matrix U

a local minimum with reasonably good empirical results. In terms of theoretical effectiveness of learning, we give a detailed analysis of the algorithm and formally prove that, comparing to the original version, RCM achieves a smaller upper bound on generalization error.

4.2 Overfitted Consensus Maximization

In this section, we recapitulate some basic concepts used in the CM framework, which is followed by an analysis of why it tends to overfit the data.

4.2.1 CM overfits

The hypothesis space of a learning algorithm is the set of all feasible solutions of the algorithm. A larger hypothesis space has more expressive power comparing to a smaller one, leading to less training errors. However, models with a larger hypothesis space is more complicated and can lead to less generalization ability and more predicting errors (55). Therefore, one needs to trade-off between minimizing training error and model complexity. We compare the hypothesis spaces of two model combination methods, MV and CM, leading to some insights into the overfitting issue of CM.

Suppose there are m base models (for ease of presentation, assume all models are supervised models). For each instance and each class, a model outputs 1 (a vote) or 0 (no vote). A model combination method is a function f that maps predictions of base models to posterior distributions on the probabilistic simplex:

$$f : X \rightarrow S \tag{4.1}$$

$$S = \{p \in \mathbb{R}^\ell | p = \sum_{\ell=1}^c \theta_\ell \mathbf{e}_\ell = [\theta_1, \dots, \theta_\ell], \sum_{\ell=1}^c \theta_\ell = 1, \theta_\ell \geq 0\}$$

where we abuse the notation X , such that X is the collection of base model predictions. \mathbf{e}_ℓ is the standard basis having 1 in its ℓ -th position and 0 anywhere else, representing the distribution

of class ℓ , θ_ℓ is the probability that an instance belongs to class ℓ . When $c = 3$, an example of 2-simplex is shown in Figure 8(a). Various model combination methods can be seen as ways of searching a suitable mapping f in the hypothesis space \mathcal{F} of all such mappings, to optimize certain objectives. Existing methods differ in their hypothesis spaces \mathcal{F} and the way they searches, but the capacity of the hypothesis space is directly related to the generalization ability of a method. Note that the domain of all model combination methods are the same, so the capacities of their hypothesis spaces are completely determined by the images of the maps $f(X) \subset S$.

Majority voting simply sums up the number of votes for each class and assigns an instance to the class having the most votes. Formally, given the output of r models for an instance, say, $[\hat{y}_1, \dots, \hat{y}_r]$, $\hat{y}_k \in \{1, \dots, c\}$, the decision of majority voting is made based on the the vector:

$$\frac{1}{r} \left[\sum_{k=1}^r \mathbb{1}[\hat{y}_k = 1], \dots, \sum_{k=1}^r \mathbb{1}[\hat{y}_k = c] \right] \in S \quad (4.2)$$

Note that majority voting maps the predictions of base models to rational vectors on the simplex, with denominators equal to the number of models. For example, if an instance receives two votes for class 1, one votes for class 2 and 0 vote for class 3, from a total of three classifiers, then the output of f is $[2/3, 1/3, 0]$, shown as the square with an arrow in Figure 8(b).

CM maps predictions of base models of an instance to a posterior distribution in S , and the image of the map is the whole simplex S . The relaxation from rational vectors to real vectors allows a larger hypothesis space such that CM can find an f to attain low consensus

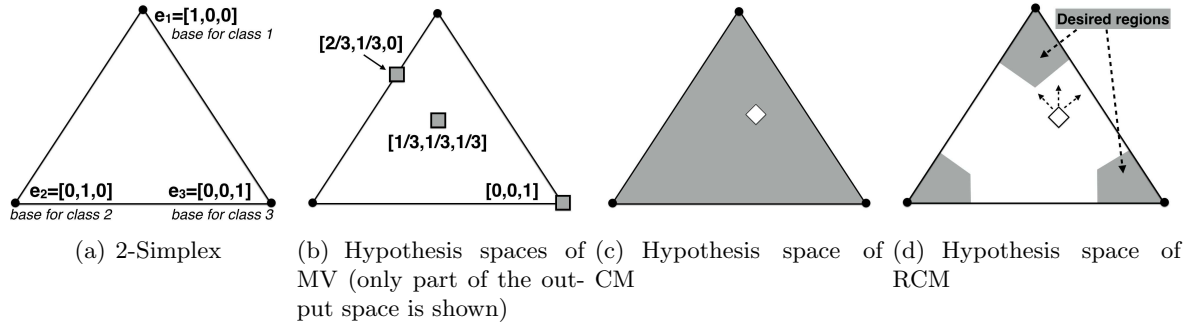


Figure 8. Hypothesis space of various methods: the tips of the triangles represent the bases \mathbf{e}_ℓ

loss (see Section 4.5.2). However, it also allows CM to pick an f that outputs predictions close to uniform distribution with small margin (like the diamond in Figure 8(c)), leading to higher empirical margin risk (see Section 4.4). It is verified in Section 4.5 that CM does tend to output predictions that have small consensus losses and small margins. Here we define “overfitting” in model combination in a vague sense, and defer the formal analysis to Section 4.4.

Definition 2 (Overfitting in model combination). *A model combination method consolidates predictions of base models to achieve a high degree of model consensus but with higher generalization error upper bound.*

4.3 Class-Distribution Regularized Consensus Maximization

4.3.1 Regularization over class distributions

According to the above analysis, if we adopt a reasonably small but rich enough hypothesis space for CM, then we could avoid over-fitting and achieve better performance. How can we specify a suitable hypothesis space for CM? Note that the predictions lying near the corners

of the simplex (shadows in Figure 8(d)) have a more dominating component θ_{ℓ_0} for some class ℓ_0 . On the one hand, when the difference between $p(y = \ell_0|\mathbf{x})$ and any other $p(y = \ell|\mathbf{x})$ is larger, the prediction is more discriminative to reflect the true class distribution. On the other hand, if the number of dominating entries in $p(y|\mathbf{x})$ is greater than 1, then those dominating classes are correlated since they co-occur, conflicting the multi-class distribution assumption. This observation indicates that when searching for solutions in the hypothesis space, CM should penalize solutions that lie too far away from any corner of the simplex and encourage solutions that lie close to the corners. For CM to reduce the penalized consensus loss $\mathcal{L}(U, Q)$, it must move its predictions towards one of the corners on the simplex, as shown by the arrows in Figure 8(d). The above intuition suggests that the consolidated predictions should exhibit some sort of independence between classes, given the problem is a multi-class single label problem.

Specifically, recall that U is the consolidated prediction, with the ℓ -th column being the posterior probabilities $p(y = \ell|\mathbf{x})$, we can compute the empirical class correlation matrix $\Sigma = U^\top U$. We want the matrix Σ to be close to a $c \times c$ matrix D , which represents the ideal class correlations. For example, to enforce independence between classes in multi-class classification problems, we can set the diagonal elements of D to a positive number whose scale is comparable to the empirical correlations, and set the off-diagonal elements to a positive number much smaller than the diagonal elements.

By adopting the Frobenius norm, we obtain the following regularization term

$$\Delta_F = \frac{1}{2} \|\Sigma - D\|_F \quad (4.3)$$

or by adopting the relative entropy (8)

$$\Delta_E = \frac{1}{2} \sum_{i,j=1}^c \Sigma_{ij} \log \frac{\Sigma_{ij}}{D_{ij}} \quad (4.4)$$

Adding any of the above regularization terms to the objective of CM, we obtain the following optimization problem:

$$\begin{aligned} \min_{U, Q} \quad & \sum_{i=1}^n \sum_{j=1}^v a_{ij} \|\mathbf{u}^i - \mathbf{q}^j\|^2 + \alpha \sum_{j=1}^v b_j \|\mathbf{q}^j - \bar{\mathbf{y}}^j\|^2 + \lambda \Delta \\ \text{s.t.} \quad & u_\ell^i \geq 0, \|\mathbf{u}^i\|_1 = 1, i = 1, \dots, n \\ & q_\ell^j \geq 0, \|\mathbf{q}^j\|_1 = 1, j = 1, \dots, v \end{aligned}$$

where $\Delta = \Delta_F$ or Δ_E . The parameter λ controls the trade-off between model consensus and class independence. We will see that the regularization helps reduce the capacity of hypothesis space and also the empirical margin risk.

4.3.2 Optimization of the Class-distribution Regularized Model

Our plan for solving the optimization problem Eq.(Equation 4.5) is to first ignore the constraints that \mathbf{u}^i and \mathbf{q}^j are probability distributions and solve the unconstrained optimization problem using gradient descent, then we address the probabilistic constraints on \mathbf{u}^i and \mathbf{q}^j in the next section. The gradient descent steps for the first two terms in the above objective func-

tion are given in Eq.(Equation 2.3) and Eq.(Equation 2.4), the gradients of the regularization term Δ with respect to column \mathbf{u}_j are as follows:

$$\begin{aligned}\frac{\partial \Delta_F}{\partial \mathbf{u}_j} &= \sum_{i=1}^c (\Sigma_{ij} - D_{ij}) \mathbf{u}_i \\ \frac{\partial \Delta_E}{\partial \mathbf{u}_j} &= \sum_{i=1}^c (1 + \log \frac{\Sigma_{ij}}{D_{ij}}) \mathbf{u}_i\end{aligned}$$

Thus a gradient descent step for the regularization term with respect to column \mathbf{u}_j are:

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t \sum_{i=1}^c (\Sigma_{ij} - D_{ij}) \mathbf{u}_i \quad (4.5)$$

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t \sum_{i=1}^c (1 + \log \frac{\Sigma_{ij}}{D_{ij}}) \mathbf{u}_i \quad (4.6)$$

where \leftarrow indicates the assignment of an updated \mathbf{u}_j to itself. η_t is the learning rate in the t -th iteration with $\eta_t = \eta_0/\sqrt{t}$ and η_0 is the initial learning rate. We let the trade-off parameter λ in the RCM objective be absorbed in η_0 . Eq.(Equation 4.5) and Eq.(Equation 4.6) have a quite intuitive meaning: for each column \mathbf{u}_i representing the i -th class, depending on whether the empirical class correlation Σ_{ij} exceeds the ideal class correlation D_{ij} , \mathbf{u}_j is moved away from ($\Sigma_{ij} > D_{ij}$) or towards ($\Sigma_{ij} < D_{ij}$) \mathbf{u}_i , and the amount of displacement is proportional to the distance between the empirical and ideal class correlation. In practice, it is not easy to specify the ideal class correlation matrix D , and the scaling parameters $\beta_{ij} = \Sigma_{ij} - D_{ij}$ (or $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$) may be sensitive to the choice of D . Simply setting all the β_{ij} to be 1 will actually hurt the performance, as we ignore the information about the class correlations.

We propose an approximation of Eq.(Equation 4.5) and Eq.(Equation 4.6) to avoid specifying the parameters D and to maintain the effect of the regularization, namely, a large margin between class distributions. Note that in Eq.(Equation 4.6), for $i \neq j$, D_{ij} should be some small number and if $\Sigma_{ij} \gg D_{ij}$, the scaling parameter $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$ will be large; on the other hand, if Σ_{ij} is about the same as D_{ij} , $1 + \log \frac{\Sigma_{ij}}{D_{ij}}$ will be close to 1. According to this observation, when computing the gradient for the column \mathbf{u}_j , we can set β_{ij} as follows:

$$\beta_{ij} = \begin{cases} 1 & \text{if } i = \arg \min_{k \neq j} \|\mathbf{u}_k - \mathbf{u}_j\|_2 \\ -1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

The resulting regularization term is

$$\Delta_A = \frac{1}{2} \sum_{j=1}^c \|\mathbf{u}_j - \mathbf{u}_{d(j)}\|_2^2 \quad (4.8)$$

where

$$d(j) = \arg \min_{k \neq j} \|\mathbf{u}_j - \mathbf{u}_k\|_2 \quad (4.9)$$

Eq.(Equation 4.5) and Eq.(Equation 4.6) become

$$\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t (\mathbf{u}_{d(j)} - \mathbf{u}_j) \quad (4.10)$$

So far we have specified all necessary gradient descent steps for RCM. Nonetheless, the original CM gradient descent steps involve the rows of the matrices U and Q , while to minimize the regularization term Δ , one has to work with the columns of U . It is non-trivial to derive gradient descent steps involving both rows and columns of a matrix. We adopt an alternative optimization procedure that first minimizes the consensus loss $\mathcal{L}(U, Q)$ through Eq.(Equation 2.3) and Eq.(Equation 2.4), then minimizes Δ_A through Eq.(Equation 4.10). These two steps are alternatively repeated until it converges.

4.3.3 Projection to the Probabilistic Simplex

The converted unconstrained optimization problem ignores the constraints:

$$\begin{aligned} u_\ell^i &\geq 0, \|\mathbf{u}^i\|_1 = 1, i = 1, \dots, n \\ q_\ell^j &\geq 0, \|\mathbf{q}^j\|_1 = 1, j = 1, \dots, v \end{aligned} \tag{4.11}$$

Although Eq.(Equation 2.3) and (Equation 2.4) maintain rows of U and Q as probability distributions, Eq.(Equation 4.10) might bring any entry of U to be greater than 1 or less than 0, and a row in U or Q might not sum up to 1. We propose to perform *probabilistic projection* for all \mathbf{u}^i after all gradient descent steps in each iteration. More formally, the following optimization problem finds \mathbf{v} , the projection of \mathbf{u}^i onto the probabilistic simplex

$$\begin{aligned} \min_{\mathbf{v}} \quad & \|\mathbf{v} - \mathbf{u}^i\|_2 \\ \text{s.t.} \quad & \|\mathbf{v}\|_1 = 1, v_\ell \geq 0, \ell = 1, \dots, c \end{aligned}$$

The optimal solution \mathbf{v}^* serves as the new \mathbf{u}^i for the next iteration, with the probabilistic constraints satisfied. An efficient algorithm (in $O(cn)$) with implementation to solve the above problem can be found in (17). The complete algorithm is described in Algorithm 4.

Algorithm 4 Regularized Consensus Maximization (RCM)

```

1: Input: Affinity matrix  $A$ , initial learning rate  $\eta_0$ 
2: Set  $\mathbf{u}^j$  to uniform distribution.
3: for  $t = 1 \rightarrow \text{MaxIterNum}$  do
4:    $Q = (D_v + \alpha K_v)^{-1}(A^\top U + \alpha K_v Y)$ 
5:    $U = D_n^{-1} A Q$ 
6:    $\eta_t = \eta_0 / \sqrt{t}$ 
7:   for  $j = 1 \rightarrow c$  do
8:      $d(j) = \arg \min_{k \neq j} \|\mathbf{u}_k - \mathbf{u}_j\|$ 
9:      $\mathbf{u}_j \leftarrow \mathbf{u}_j - \eta_t(\mathbf{u}_{d(j)} - \mathbf{u}_j)$ 
10:  end for
11:  Project  $\mathbf{u}^i$  to the probabilistic simplex.
12: end for
```

4.4 Generalization Error of RCM

In this section, we prove that, compared to CM, the proposed regularization leads to a smaller upper bound on generalization error. The generalization error bound consists of two terms: the empirical margin risk on training data and a term measuring the capacity of the hypothesis space explored by a learning algorithm. Regarding the empirical margin risk, we first define the multi-class margin (42).

Definition 3 (Canonical Function). *Given a function $f \in \mathcal{F}$ that maps predictions of base models to posterior distribution (see Section 4.2.1). For the instance \mathbf{x} , $f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_c(\mathbf{x})] \in$*

S where $f_\ell(\mathbf{x})$ is the probability that \mathbf{x} belongs to class ℓ , according to f . Let M_1 be the smallest index ℓ such that $f_\ell(\mathbf{x}) = \max_k f_k(\mathbf{x})$, and M_2 be the smallest index ℓ such that $f_\ell(\mathbf{x}) = \max_{k \neq M_1} f_k(\mathbf{x})$. The canonical function $\Delta f : X \rightarrow [-1, 1]^c$, with the ℓ -th component being:

$$\Delta f_\ell(\mathbf{x}) = \begin{cases} f_\ell(\mathbf{x}) - f_{M_2}(\mathbf{x}) & \text{if } \ell = M_1 \\ f_\ell(\mathbf{x}) - f_{M_1}(\mathbf{x}) & \text{otherwise} \end{cases} \quad (4.12)$$

M_1 is the label selected by Bayes decision rule and M_2 is the closest runner-up. Δf_ℓ measures how far away the selected label is from the other competitors. Based on the canonical function, we define the multi-class empirical margin risk

Definition 4 (Empirical Margin Risk). For $\gamma > 0$ and training set $s = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$, the empirical margin risk $R_s^\gamma(f)$ of the function f is

$$R_s^\gamma(f) = \frac{1}{m} |\{\mathbf{x}_i | \exists \ell \in \{1, \dots, c\}, y_{i\ell} \cdot \Delta f_\ell(\mathbf{x}_i) < \gamma\}| \quad (4.13)$$

where $y_{i\ell}$ is the ℓ -th component of the true label vector \mathbf{y}_i .

Next we define necessary concepts to measure the capacity of hypothesis spaces.

Definition 5 (Supremum Metric for functions). (42; 3) Suppose \mathcal{F} is the collection of functions mapping from X to S , and $s = \{\mathbf{x}_i\}_{i=1}^m \subset X$ is a given set of instances. Define the metric (distance measure) for functions $d(\cdot, \cdot) : \mathcal{F} \times \mathcal{F} \rightarrow [0, +\infty)$ on s by

$$d_s(f, \tilde{f}) = \max_{\mathbf{x}_i \in s} \sum_{\ell=1}^c |f_\ell(\mathbf{x}_i) - \tilde{f}_\ell(\mathbf{x}_i)| \quad (4.14)$$

Note that the metric such defined depends on the set of instances s .

Definition 6 (Covering number). *Let (\mathcal{F}, d_s) be the space of functions equipped with the supremum metric, where $s \subset X$ a finite set of instances. Define $B_s(f, r)$ the closed ball centered at f with radius r :*

$$B_s(f, r) = \{g \in \mathcal{F} | d_s(f, g) \leq r\} \quad (4.15)$$

The covering number $\mathcal{N}(\epsilon, \mathcal{H}, d_s)$ of a set $\mathcal{H} \subset \mathcal{F}$ is defined as

$$\mathcal{N}(\epsilon, \mathcal{H}, d_s) = \inf_T \{|T|\} \text{ s.t. } \mathcal{H} \subset \cup_{f \in T} B_s(f, \epsilon) \quad (4.16)$$

The set T is called an ϵ -cover of the subset \mathcal{H} .

The following bound on generalization error for multi-class classification is given in (42):

Theorem 1. *Let \mathcal{F} be a set of functions from X to S and $\Delta\mathcal{F}$ be the set of canonical functions Δf . Let s be a learning set of size m drawn iid. from a probability distribution P . Let $0 < \gamma < 1$. With probability $1 - \delta$, $\forall f \in \mathcal{F}$,*

$$R(f) \leq R_s^\gamma(f) + \sqrt{\frac{1}{2m} \ln \left(\frac{2\mathcal{N}_\infty(\gamma/2, \Delta\mathcal{F}^\gamma)}{\delta} \right)} \quad (4.17)$$

where

$$\mathcal{N}_\infty(\gamma, \mathcal{F}) = \sup_{s: |s|=2m} \mathcal{N}(\gamma, \mathcal{F}, d_s) \quad (4.18)$$

$\Delta\mathcal{F}^\gamma = \{\pi_\gamma \circ \Delta f : \Delta f \in \Delta\mathcal{F}\}$ where π_γ is the truncation function applied to each of the c components of Δf

$$\pi_\gamma(f_\ell(\mathbf{x})) = \begin{cases} \gamma \cdot \text{sign}(f_\ell(\mathbf{x})) & \text{if } |f_\ell(\mathbf{x})| \geq \gamma \\ f_\ell(\mathbf{x}) & \text{otherwise} \end{cases} \quad (4.19)$$

Given the bound in Eq.(Equation 4.17), we want to prove that both terms in the bound for the regularized CM are smaller than those for the original CM, and obtain the following theorem:

Theorem 2. *RCM has a smaller upper bound on generalization error compared with that of CM.*

The above theorem is proved in two steps in the following two lemmas.

Lemma 1. *RCM achieves a lower empirical margin risk if we use Δ_E as our regularization term and the matrix D is such set that the scaling parameters $\beta_{ij} = \beta_{ji}$ and $\beta_{ii} = 1$.*

Proof. Given training data s , $0 < \gamma < 1$, $1 - R_s^\gamma(f)$ is the proportion of correctly classified instances with margin greater than γ . Suppose f is the prediction function found by CM and \tilde{f} is that found by RCM. In other words, \tilde{f} is obtained by applying Eq.(Equation 4.6) to f . Note that $R_s^\gamma(\tilde{f}) \leq R_s^\gamma(f) \iff 1 - R_s^\gamma(\tilde{f}) \geq 1 - R_s^\gamma(f)$, we need to prove, for any correctly classified instance with margin greater than γ , its margin under \tilde{f} is not smaller than that under f .

Let $\mathbf{u} = [f_1, \dots, f_c]$ and $\tilde{\mathbf{u}} = [\tilde{f}_1, \dots, \tilde{f}_c]$ be the evaluations of f and \tilde{f} at some point \mathbf{x} that is correctly classified with margin larger than γ (we ignore the arguments of f and \tilde{f}). Assume

$1 = \operatorname{argmax}_{\ell} f_{\ell}$ and $2 = \operatorname{argmax}_{\ell \neq 1} f_{\ell}$. Then $y_1 \cdot \Delta f_1 \geq \gamma$. But $y_1 = 1$, so $\Delta f_1 = f_1 - f_2 \geq \gamma$.

The gradients Eq.(Equation 4.6) at \mathbf{x} be

$$g_j = \eta_t \sum_{i=1}^c (1 + \log \frac{\Sigma_{ij}}{D_{ij}}) f_i > 0, j = 1, 2 \quad (4.20)$$

Assume that proper values are set to matrix D , such that $\Sigma_{ii} = D_{ii}$ but $\Sigma_{ij} \gg D_{ij}$ for $i \neq j$.

Then the gradients are

$$g_j = \eta_t \sum_{i=1}^c \beta_{ij} f_i, j = 1, 2 \quad (4.21)$$

where $\beta_{ii} \ll \beta_{ij}, i \neq j$. That is, for a given j , f_i has a much larger weight than f_j in g_j for $i \neq j$. If $\beta_{ij} = \beta_{ji}$, then by $f_1 > f_2$, we have $g_2 > g_1$,

$$\Delta \tilde{f}_1 = \tilde{f}_1 - \tilde{f}_2 = (f_1 - g_1) - (f_2 - g_2) = \Delta f_1 - (g_1 - g_2) > \Delta f_1 \quad (4.22)$$

□

Lemma 2. *The hypothesis space of RCM has smaller covering number than the hypothesis space of CM.*

Proof. Let $\Delta \mathcal{F}^{\gamma} = \{\pi_{\gamma} \circ \Delta f : \Delta f \in \Delta \mathcal{F}\}$ and $\Delta \tilde{\mathcal{F}}^{\gamma} = \{\pi_{\gamma} \circ \Delta \tilde{f} : \Delta \tilde{f} \in \Delta \tilde{\mathcal{F}}\}$ where \mathcal{F} is the collection of functions $f : X \rightarrow S$ and $\tilde{\mathcal{F}}$ are their large margin version as defined in Lemma 1, Δf is the canonical function and π_{γ} is the truncation function Eq.(Equation 4.19). Then $\Delta \tilde{\mathcal{F}}^{\gamma} \subset \Delta \mathcal{F}^{\gamma}$ since for any $f \in \mathcal{F}$, its large margin version $\tilde{f} \in \mathcal{F}$, thus we have $\Delta \tilde{\mathcal{F}} \subset \Delta \mathcal{F}$. After truncation, $\Delta \tilde{\mathcal{F}}^{\gamma} \subset \Delta \mathcal{F}^{\gamma}$.

Given any training data s of size $2m$, any $\gamma/2$ -cover of $\Delta\mathcal{F}^\gamma$ is also a $\gamma/2$ -cover of $\Delta\tilde{\mathcal{F}}^\gamma$.

Therefore by definition Eq.(Equation 4.16),

$$\mathcal{N}(\gamma/2, \Delta\mathcal{F}^\gamma, s) = \inf\{|T|\} \geq \inf\{|T'|\} = \mathcal{N}(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma, s) \quad (4.23)$$

where $T \in \{\gamma/2\text{-covers of } \Delta\mathcal{F}^\gamma\}$ and $T' \in \{\gamma/2\text{-covers of } \Delta\tilde{\mathcal{F}}^\gamma\}$. By the definition Eq.(Equation 4.18),

we conclude that

$$\begin{aligned} \mathcal{N}_\infty(\gamma/2, \Delta\mathcal{F}^\gamma) &= \sup_s \mathcal{N}(\gamma/2, \Delta\mathcal{F}^\gamma) \\ &\geq \sup_s \mathcal{N}(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma) = \mathcal{N}_\infty(\gamma/2, \Delta\tilde{\mathcal{F}}^\gamma) \end{aligned}$$

□

4.5 Experimental Results

In this section, we first summarize the experimental settings, including evaluation benchmarks and model combination baselines. Then we demonstrate how CM overfits the data and how the proposed RCM resolves the issue.

4.5.1 Experimental Settings

Benchmarks A model consolidation method consolidates the predictions of multiple supervised and/or unsupervised models to come up with improved predictive performance. Therefore, to evaluate the performance, we need the predictions from multiple base models for

TABLE XVII
DATASETS AND BASE MODELS

Datasets		# Instances	# Classes	Predictors
20NG	1	1568	4	Apply SVM, Logistic Regression, K-means and mini-cut to texts. 4 Predictors in total.
	2	1588	4	
	3	1573	4	
	4	1484	4	
	5	1584	4	
	6	1512	4	
Cora	1	663	3	Apply Logistic Regression K-means to citation/publication network and texts. 4 Predictors in total.
	2	977	4	
	3	1468	5	
	4	975	5	
DBLP		4236	4	

the datasets, whose information are summarized in Table Table XVII. The dataset¹ contains 11 text classification tasks. Each task contains the predictions given by the output of 2 classification and 2 clustering models. For details of how they processed the data, please refer to (24).

We compare RCM with CM in order to verify the effectiveness of the large margin constraint. CM and RCM share most of the parameters such as number of iterations, importance of supervised models, etc.. For the shared parameters, we adopt the parameter settings of CM (24). In addition, we set the initial learning rate η_0 to be 0.1. We also compare RCM with other state-of-the-art cluster ensemble methods: MCLA (50), HBGF (50), SNNMF (34), BCE (57) ECMC (71). MCLA and HBGF are graph partition based approaches, which use

¹available at <http://www.cse.buffalo.edu/~jing/>

spectral clustering (41; 16) to partition the bipartite or hyper graph constructed from the predictions of base models. There is no parameter to tune for these two methods. SNNMF is a matrix factorization based method, which derives clustering of instances using the similarity matrix constructed from base models' predictions. We run SNNMF to its convergence to obtain the final predictions. BCE is a Bayesian approach to consensus maximization. We set its parameters as follows: LDA parameters $\alpha = 0.5, \beta = 0.1$, number of iterations for Gibbs sampling is set to 50,000, the topic distributions of the words in documents are randomly initialized. We observe that performance the Gibbs sampling for BCE is sensitive to the initialization of the parameters and unstable, we run the BCE for 10 times and report its best performance. We also implemented BCE using variational inference, but the procedure did not converge after long runs, so we do not report the corresponding results. ECMC is a matrix factorization method with a de-noising step, we adopt the implementations of robust PCA and matrix completion packages¹, with $d_0 = 0.4, d_1 = 0.6$ and other parameters being the default values (see (71) for details).

4.5.2 Overfitting in Consensus Maximization

In Section 2.2 and 4.2.1 we theoretically showed that, CM produces predictions that minimize the consensus loss but overfit the data, and therefore might not generalize well, and in Section 4.3.1, we proposed RCM to solve the issues. By comparing CM and RCM in con-

¹http://perception.csl.illinois.edu/matrix-rank/sample_code.html

TABLE XVIII. Overall Performance on Text Classification Tasks

Methods	Newsgroups						Cora				DBLP
	1	2	3	4	5	6	1	2	3	4	1
MCLA	0.7574	0.8345	0.7816	0.8225	0.8039	0.8332	0.8522	0.8009	0.8442	0.8262	0.8604
HBGF	0.721	0.636	0.7677	0.6885	0.6421	0.7482	0.7966	0.6574	0.7655	0.7912	0.8146
SNNMF	0.5980	0.6904	0.6384	0.5733	0.6245	0.6753	0.7407	0.6492	0.7051	0.6989	0.6307
BCE	0.6639	0.2544	0.7082	0.7230	0.7247	0.7474	0.6546	0.8915	0.5565	0.2482	0.2887
ECMC	0.5599	0.6215	0.6294	0.6759	0.6338	0.4530	0.5973	0.6428	0.5252	0.8513	0.7771
CM	0.8131	0.9106	0.8608	0.9117	0.8857	0.9094	0.8688	0.9151	0.8951	0.9036	0.9412
RCM	0.8131	0.9030	0.8735	0.9232	0.8927	0.9134	0.8703	0.9222	0.9203	0.9128	0.9429

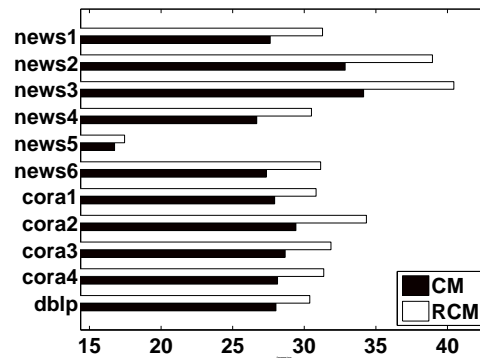


Figure 9. Consensus Loss

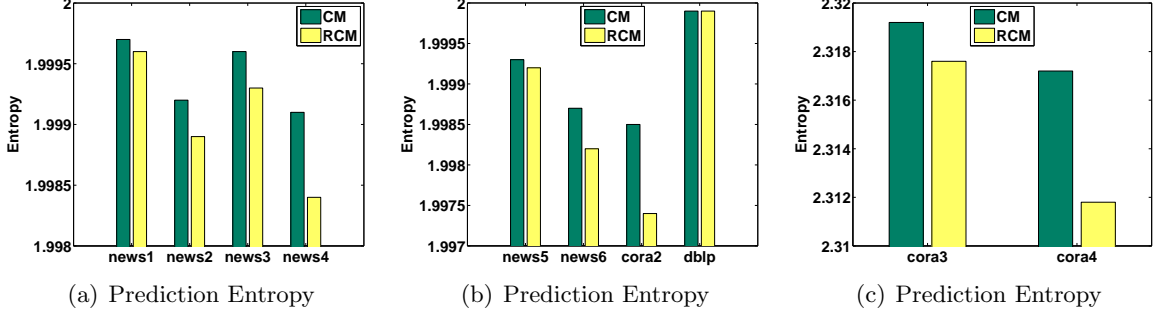


Figure 10. Consensus loss and entropy of CM and RCM

sensus loss, prediction margins and accuracy (next section), we verify that CM does have the overfitting issue and RCM can effectively mitigate overfitting.

On the one hand, one can see from Figure Figure 9 that CM has a lower consensus loss than RCM does across all datasets. This is because CM solely minimizes the consensus loss while RCM minimizes a regularized consensus loss and has a smaller hypothesis space. On the other hand, we use entropy of \mathbf{u}^i ($h^i = -\sum_{\ell=1}^c u_{\ell}^i \log u_{\ell}^i$) as a measure of prediction margin: the higher the entropy, the smaller the margin \mathbf{u}^i has and the less discriminative \mathbf{u}^i is. We show the averaged entropy $\frac{1}{n} \sum_{i=1}^n h^i$ for each dataset in Figure 10(a), 10(b), 10(c). From the figures, we can see that the entropy is higher in the predictions of CM across all datasets except on the *dblp* dataset. (the result on the *cora1* dataset is not shown due to the scale). Therefore on average, the predictions of CM have smaller margins than those of RCM. Since margin is used as an indicator of generalization performance of a learning algorithm (4), CM might overfit the data while RCM should improve the generalization ability and accuracy of CM.

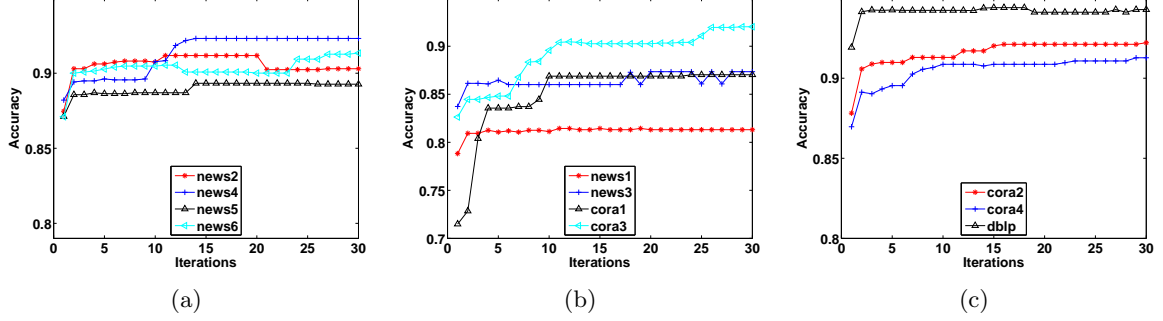


Figure 11. Convergence of RCM

4.5.3 Accuracy

In Table XVIII, we compare the accuracies of RCM and the baselines on 11 text classification tasks. From the table, we can see that BCE is very unstable and there are two main reasons for this. First, similar to LDA, BCE needs a lot of observed data to infer the consolidated labels, yet usually we have only a couple of base models. Second, Gibbs sampling is too sensitive to initial conditions while variational inference does not converge given only a handful of data. ECMC and SNNMF sometimes give reasonable performance, such as ECMC on the *cora4* task. However, their optimization are also sensitive to initialization, and their solutions are unstable. Both MCLA and HBGF in general have better performance than ECMC and SNNMF, though they are still outperformed by both CM and RCM.

The comparison between CM and RCM is more interesting. Using the proposed regularization over the class distributions, RCM controls the size of its hypothesis space and focuses on the more discriminative predictions. As we can see from the table, RCM outperforms CM on

10 out of 11 datasets. These evidences, together with the comparisons of consensus loss and entropy in Section 4.5.2, clearly demonstrate that CM overfits the data to produce highly consensus predictions, while RCM is able to trade-off between two objectives and achieves better accuracy.

Statistical significance of the results We verify that the improvements brought by the proposed method is statistically significant. According to (15), one can compare the effectiveness of different algorithms based on their performance on multiple datasets. Since among all baselines, CM has the closest performance to RCM, we compare these two methods using the Wilcoxon signed-ranks test. For the details of how to carry out the test, please refer to (15). The test shows that RCM is statistically significantly superior to CM with $\alpha = 0.05$, where α is the probability that RCM is *not* better than CM.

4.5.4 Convergence Study

For each of the text classification tasks, we record the accuracy at the end of each iteration of RCM. In Figure Figure 11, we plot the accuracies against the number of iterations. From the figure, we can see that, except for the *news3* and *dblp* tasks, RCM converges to some fixed accuracies. Even for those two exceptions where there are some zigzag’s at the tails of the curves, we notice that the lowest accuracies obtained after the 25th iteration are at least the same as the best baseline (CM in both cases). Therefore, we conclude that given a big enough number of iterations, the algorithm performs better than or comparable with the baselines.

CHAPTER 5

DISTILLING TRUSTWORTHY CROWDSOURCED RATINGS VIA SINGLETON SPAMMING ATTACK DETECTION

(This chapter includes the paper published in *Sihong Xie, Guan Wang, Shuyang Lin, Philip S. Yu*. “Review Spam Detection via Temporal Pattern Discovery”. In **KDD 2012**.. (DOI:<http://dl.acm.org/citation.cfm?doid=2339530.2339662>))

5.1 Background and Motivation

In this chapter, we will address the problem of distilling trustworthy ratings from crowdsourced ratings for products or stores on online commerce websites. Online ratings about products and stores are essential parts in today’s electronic commerce where they provide helpful information for potential customers. A product or store with a decent rating and a high proportion of positive reviews will attract more customers and larger amount of business, while a couple of negative reviews/ratings could substantially harm the reputation, leading to financial losses. Since there is no rule governing online reviews and ratings, some product providers or retailers are leveraging such public media to defame competitors and promote themselves unfairly, or even to cover the truth disclosed by genuine reviews. For example, suppose a customer finds the delivery service of a certain store unacceptably slow, she then writes a review about the fact and gives it a low rating on a review website. This review and rating present a unfavourable impression of the store to potential customers, who might choose other stores

after reading that review. In order to avoid the drainage of business caused by this negative yet truthful review, the store could employ or entice a group of people to post undeserving positive ratings about the delivery service. Similarly, the store could also ask these people to post unfavorable ratings about its competitors, from which the store would like to distract customers. These hired reviewers are called spammers and the fake ratings they post are called spamming ratings. In order to protect customers, honest online stores and the whole electronic commerce environment, it is desirable to detect the spamming ratings to distill a set of more trustworthy ratings of the products.

Previous works proposed to use features of posted review contents and reviewers' behaviors (18; 39; 40; 1) or graph connecting reviewers, stores and reviews (22), to detect spamming reviews or ratings. These methods work best in the situations where spammers post many reviews or ratings (see the related work). In reality, however, most reviewers post only one review or rating. For example, 68% of the reviewers post a single review or rating in the Amazon review dataset studied in (39), and this percentage is 90% in the dataset we study here. If a review/rating is the only review/rating a reviewer has post, we call it a *singleton review/rating* (SR for short). In fact, as we shall see later, the SR definition can be generalized to cover reviewers with a few reviews/ratings, not necessarily just one. One question is: are most of these SRs honest ones? The answer is probably not, due to the *nature* of spam attacks. For a store to rapidly raise its fame and rating (resp. defame others) it is desirable to have spammers post plenty of favorable (resp. unfavorable) reviews/ratings about it (resp. its competitors) in a short time. Usually a spammer would not post many reviews with similar ratings for a

store under the same name. Instead, he would rather post fake reviews/ratings under different names to avoid being caught. This spamming strategy brings a large number of SR. Most of the statistics adopted by previous works would not work on such singleton reviews/ratings. For example, the mean and standard deviation of ratings given by a reviewer (18) become meaningless if this reviewer has post only one rating; in the rule or frequent pattern based detection method (40), singleton reviews are also ignored due to their low significance.

Indeed, detecting SR spams can be challenging. If a reviewer has post only one review/rating, simply looking at this SR reveals little information about the true intention behind it, so it is hard for machines or even human beings to draw a conclusion. For example, in the experiment, we find that one reviewer said “For the few times that I’ve contacted customer service via phone, email, or chat, the person has always been helpful and gone out of his/her way.” At a first glance, this is a normal review talking about customer service. Since it is the only review the reviewer has written, existing spam detection algorithms will simply ignore this review. Even for human beings, it is extremely difficult to tell if this is a spam review or not. However, if we look at the aggregate reviewers’ behaviors in a temporal way, we can find that this review was written in a period when there was a burst of SRs and the rating of the store went up dramatically. It is abnormal for the number of reviews, the ratio of SR and the store rating to be temporally correlated, this review is pretty suspicious.

In particular, spammers have to post many positive (or negative) reviews/ratings in a short time, otherwise, the spammers are not effective in promoting or defaming a store or product. However, if a spammer posts his reviews/ratings quickly under the same name, he can be easily

detected by checking the duration between two consecutive reviews with similar rating from a single person. So posting spam reviews/ratings under different names is a safer way. Based on the above reasoning, we make the following conjectures on SR spam attacks. When such an attack occurs in a certain period, there tends to be a sharp increase in the number of reviews and the ratio of SRs, together with an increase (or decrease) in the average rating. Therefore, we can transform the SR spam detection problem to an abnormally correlated temporal pattern detection problem in multidimensional time series consisting of the above three indices. Note that spammers may want to evade the proposed method by writing more than one but not too many reviews. For these spammers, we can easily modify the algorithm to catch them (see Section 5.2.2.1) and we focus on the detecting SR spams. In the next section, we make several assumptions about reviewers' arrival patterns, which define a necessary condition of SR spam attacks.

5.2 Singleton Review/Rating Spam Detection Model

5.2.1 The Model of Reviewer Behavior

We make certain assumptions of reviewers' behaviors, divided into two phases: the arriving and posting phase. In the arriving phase, a customer buys something from a store or a spammer is hired or enticed by a store to post fake reviews/ratings. The posting phase is when a reviewer post a review/rating. There are mainly three patterns of arriving phase behaviors: normal arrival, promotion/sale event arrival, and spam attack pattern. First, the normal arrival pattern can be modeled by a homogeneous Poisson process with a fixed rate λ . A Poisson process is a set of random variables $\{N(t) : t \geq 0\}$ satisfying the following properties (30):

- $\Pr\{N(t+h) - N(t) = 1 | N(t) = n\} = \lambda h + o(h)$ as $h \rightarrow 0$, for $n = 0, 1, \dots$
- $\Pr\{N(t+h) - N(t) = 0 | N(t) = n\} = 1 - \lambda h + o(h)$ as $h \rightarrow 0$
- $N(0) = 0$

where $N(t)$ is the number of arrivals up to time t . λ is a constant controlling the intensity of arrivals, with a larger λ indicating more arrivals in a unit of time. Second, it is possible for a store to promote their products over a period and therefore increase the traffic of customers and reviews. We model this arrival pattern using a non-homogeneous Poisson process, with the rate parameter being a function of time $\lambda(t)$. Third, the spam attack arrival pattern is pretty much like that in the promotion mode, since a large number of spammers would be hired or enticed by the store.

In the writing phase, we model the writing behaviors of normal reviewers and spammers. First, in order to get the rewards offered by the store that tries to commit SR spam attacks, spammers tend to post spam reviews or ratings in a hurry, and there is seldom a delay. Therefore, we assume that the time when a spammer posts something is the same as the time she arrives, and the spamming reviews/ratings' arriving pattern is the same as spammers' arriving pattern, a bursty one. Second, for genuine reviewers, we claim that there are some random factors associated with the delays in posting their reviews or ratings after their shopping experiences, by the following reasons. A genuine reviewer seldom post a review/rating right after she shops with a store. Instead, most of them would do so after receiving and trying out the products for some time. Therefore, one random factor is the time spent on delivery, this factor depends on how a customer and a store choose the way of delivery, the traffic and logistic

conditions and so on. Another random factor is the time spent on tryouts, which depends on individual customer behaviors. The randomness associated with the delay in a genuine reviewer's posting activities smooth out the arrival intensity of reviews, even in a promotion event. In other words, their postings are less likely to concentrate in a short period and causing bursty peaks.

According to the above analysis, a spam attack tends to create a burst in the review arriving process, which is distinct from the normal and even promotion review arrivals. Nonetheless, as fluctuations in the volume of reviews do exist, bursty patterns in review arrival do not necessarily imply SR spam attacks. Observe that spammers are brought together to bring up or down the rating of a store, the spamming ratings are more likely to correlate with these reviews' arrivals. In contrast, because the opinions of genuine reviewers about a store can vary wildly, depending on their satisfaction with speed of delivery, quality of products and customer services, etc. If we average the ratings of genuine reviews in a certain period of reasonable length, the positive and negative ratings will cancel out each other, therefore, the average ratings should be stable over time and independent of genuine reviews' arrivals. In summary, we should look at the joint abnormal patterns in review arrival and averaged rating to detect such attacks more robustly.

5.2.2 A Correlated Temporal Anomalies Discovery based Approach

5.2.2.1 Time Series Construction

The detection approach is based on time series of the number of reviews, average ratings and the ratio of singleton reviews. The data we study here is a set of reviews with texts and ratings posted for different stores on a review website in a certain time period. To construct these time

series, we discard text information and keep the posting time and ratings of the reviews. This is reasonable as there exist other spam detection algorithms utilizing text information, so they are complementary methods to the proposed algorithm. The resulting data can be seen in this way: each store s has a series of ratings sorted in ascending order of posting time.

$$R(s) = \{r_1, \dots, r_{n_s}\}, \quad TS(s) = \{ts_1, \dots, ts_{n_s}\},$$

where n_s is the number of reviews for store s , and ts_i is the time stamp when r_i is written, $ts_i \leq ts_j$ for all $1 \leq i < j \leq n_s$. After choosing the time windows size (denoted by Δt), the time interval under investigation (denoted by $I = [t_0, t_0 + T]$) can be divided into $N = T/\Delta t$ consecutive time windows or sub-intervals. Each time window is of length Δt and contains reviews posted during that time window. Let I_n denote the n -th time window, so

$$I_n = [t_0 + (n-1)\Delta t, t_0 + n\Delta t], \quad I = \bigcup_{n=1}^N I_n$$

Given a time window I_n , we compute the average rating f_1 , the number of reviews f_2 , and the ratio of singleton reviews f_3 . Formally,

$$f_1(I_n) = \sum_{ts_j \in I_n} r_j / f_2(I_n)$$

$$f_2(I_n) = |\{r_j : ts_j \in I_n\}|$$

$$f_3(I_n) = |\{r_j : ts_j \in I_n, r_j \text{ is an SR}\}| / f_2(I_n)$$

where $|A|$ denotes the cardinality of the set A . Given a store s , time interval $I = [t_0, t_0 + T]$ and time window size Δt , these aggregate functions represent a three dimensional time series and can be collectively represented by

$$F_s(I, \Delta t) = \begin{bmatrix} f_1(1) & \dots & f_1(N) \\ f_2(1) & \dots & f_2(N) \\ f_3(1) & \dots & f_3(N) \end{bmatrix}_s$$

where $f_i(n)$ is a shorthand for $f_i(I_n)$, $i = 1, 2, 3$. In the following, we drop the index on stores and let $F(I, \Delta t)$ denote the time series constructed for a certain store. The way we construct these time series can be generalized to handle spammers who write just a few reviews with similar ratings. We can simply treat all the reviews as SR by ignoring reviewers' ids, then the way we construct these time series still makes sense and the proposed algorithm can detect SR attacks (see next section).

5.2.2.2 Correlated Abnormal Patterns Detection in Multidimensional Time Series

Given the three time series of a store, we would like to find out correlated abnormal blocks on all three series. In other words, these blocks should simultaneously present sudden increases in rating, ratio of singleton reviews and the number of reviews. Here we focus on the singleton review detection methodology based on burst detection algorithms. Instead of inventing a novel burst detection algorithm, which is not the focus in this paper, we use a three-step approach for the detection. First, on each dimension, we employ a Bayesian change point detection

algorithm (9) to fit curves using the time series (other curve fitting algorithms will do the job, too). As an example, we plot the time series along with the fitted curves in Figure 12. We then apply a simple template matching algorithm to the fitted curves to detect bursty patterns. Lastly, a sliding window finds out the blocks in time series corresponding to a joint burst in all dimensions of the time series. In the above example, a joint burst is highlighted by the red box in Figure 12.

Assuming that we have obtained the fitted curves, we describe in what follows the last two steps in details. For the curve fitting algorithm, please refer to (9). Let $C = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\}$ be the fitted curves of the three dimensions of a time series. All curves have the same length (number of samples), which is also defined as the length of C . First, we want to detect sudden increases in each of the three curves separately. Based on the description of the arrival process in Section 5.2.1, this can be transformed to the problem of template matching. We use a step function-like template to represent a sudden rise in values

$$\mathbf{v} = \{-0.5, -0.5, 0.5, 0.5, 0.5\}$$

Note that one can use other values for \mathbf{v} so long as it represents a sharp increase temporally. If a block on a fitted curve $\mathbf{c} = \{c_1, \dots, c_n\} \in C$ is found to “match” this template well, then we find an anomaly of interest on the curve. One can obtain all blocks of \mathbf{c} by sliding a window

through \mathbf{c} , and all consecutive points on \mathbf{c} falling into the window form a block, which is denoted by

$$\mathbf{b} = \{c_{i_1}, \dots, c_{i_5}\}$$

where $1 \leq i_k \leq n$ for $k = 1, \dots, 5$ and $i_k + 1 = i_{k+1}$ for $k = 1, \dots, 4$. Note that the length of a block is chosen to have the same length as the template. We use a modified longest common substring (LCS) for matching (54) between \mathbf{v} and \mathbf{b} . In general, suppose we want to find the degree of match between two sequences $\mathbf{z}^1 = \{z_1^1, \dots, z_n^1\}$ and $\mathbf{z}^2 = \{z_1^2, \dots, z_n^2\}$. Without loss of generality, one can think of \mathbf{z}^1 as \mathbf{v} and \mathbf{z}^2 as \mathbf{b} . In the modified LCS, how well two sequences match each other is measured by the number of points in one sequence matching those in the other sequence. By a “match” between two points, we mean the absolute difference between the values of two points is less than a given threshold ϵ . The modified LCS algorithm uses the following dynamic programming formula to find out how many matches occur between \mathbf{z}^1 and \mathbf{z}^2 , for $0 \leq i, j \leq n$ and $|i - j| \leq 1$:

$$M(i, j) = \begin{cases} 0, & \text{if } i \text{ or } j = 0 \\ 1 + M(i - 1, j - 1), & \text{if } |z_i^1 - z_j^2| < \epsilon \\ \max\{M(i - 1, j), M(i, j - 1)\}, & \text{otherwise} \end{cases}$$

where $M(i, j)$ records the number of matches between subsequences $\{z_1^1, \dots, z_i^1\}$ and $\{z_1^2, \dots, z_j^2\}$. The constraint $|i - j| \leq 1$ makes sure that, $z_i^1 \in \mathbf{z}^1$ is not matched to a point $z_j^2 \in \mathbf{z}^2$ far away from the position of z_i^1 .

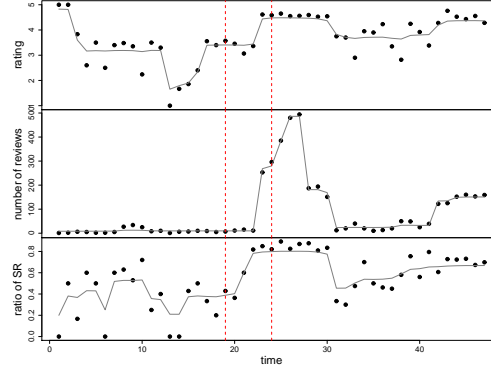


Figure 12. Bursty Patterns Detected in Store 24779

Algorithm 5 Bursts Detection in Single Time Series (**BD-STs**)

Input: fitted curve \mathbf{c} , template \mathbf{v}

Output: top k ranked periods with bursty pattern

m = length of \mathbf{c} .

n = length of \mathbf{v} .

for $i = 1 \rightarrow m - n + 1$ **do**

 Normalize $\mathbf{c}[i : i + n - 1]$.

$factor = range(\mathbf{c}[i : i + n - 1])$.

$s[i] = LCS(\mathbf{c}[i : i + n - 1], \mathbf{v}) \times factor$.

end for

return Periods corresponding to top k values in s .

Algorithm 5 describes bursty pattern detection in a single time series. We first normalize each block \mathbf{b} on a fitted curve \mathbf{c} . Then the modified LCS procedure described above finds out the number of matches between the template and the normalized block \mathbf{b} . By this step, we can find out, in the time series, the locations corresponding to bursty patterns. Taking the degree of burst into account, the number of matches in each block is multiplied by the range of values

in that block (the one before normalization), such that bursts that change more dramatically will be ranked higher.

Algorithm 6 Correlated Abnormal Patterns Detection in Multidimensional Time Series (CAPD-MDTS)

Input: Multidimensional curves C
Output: Periods when correlated anomalies appear
for each dimension \mathbf{c}_i **do**
 Time points of burst $L_i = \mathbf{BD-STS}(\mathbf{c}_i)$
end for
 $n = \text{length of } C, w = \text{time frame length (set to 5)}$
 $S = \emptyset$ // set of periods to return
for $b = 1 \rightarrow n - w + 1$ **do**
 $S = S \cup \{[b, b + w - 1]\}$ if $|\{x \in L_i : i = 1, 2, 3, x \in [b, b + w - 1]\}| == 3$
end for
return S

After we obtain a list of time points corresponding to the top k bursts in each of the dimensions, we need to find out the time windows corresponding to joint bursts in all three dimensions. By the first step, we know in each dimension the time when the bursty patterns appear, along with their intensities of burst. In the experiments, we take the top 5 time points. Then we slide a window of a certain size over the time axis. At each point, we find out how many top ranked locations in all dimensions are in the time frame specified by the current time window. A time window is reported if all three dimensions have bursty patterns falling into the window. These steps are formally described in Algorithm 6.

A running example based on the review data is shown in Figure 12. The length of the time window in time series construction is chosen to be 60 days. This example is also discussed in more detail in the experiment section. Each dimension of the time series is plotted in dark points (upper box - rating, middle box - number of reviews, lower box - ratio of singleton review). The solid lines are the fitted curves (to be discussed in the next subsection). We use red vertical dash lines to highlight one of the suspicious blocks detected in the time series by the proposed approach. The significant joint bursty pattern locates in $\{19 \rightarrow 24\}$ (from Oct 13, 2005 to Sep 12, 2006), as enclosed by the pair of vertical lines. The three curves all go up in this interval.

5.2.2.3 A Hierarchical Framework for Robust Singleton Review Spam Detection

Given the review records of a store, one can construct multiple time series using different time window sizes (resolutions). If the window size is set too small, the general trend of a time series would be buried in a large number of fluctuations, which might cause high false positive rate. Therefore, we propose a hierarchical framework, which incorporates Algorithm 6 to robustly detect SR spam attacks. We summarize this hierarchical SR spam detection algorithm in Algorithm 7. We first smooth out short-term fluctuations using a larger window (lower resolution). Then we fit curves using these time series and use Algorithm 6 (**CAPD-MDTS**) to detect any suspicious periods with correlated abnormal patterns, which indicate the high likelihood of SR spam attacks. A smaller window size (higher resolution) can be used to reveal more details (e.g. the exact time of the burst). This is accomplished by constructing new

time series with a higher resolution on the detected periods, and detecting any finer suspicious period. This process continues until one reaches the desired resolution such that the time of SR spam attacks can be easily pinpointed.

Algorithm 7 Multi-Scale Spam Detection Algorithm

- 1: **Input:** Reviews data of a store, initial window size Δt , time span I when all reviews are collected.
 - 2: **Output:** Detected time intervals of spam activities.
 - 3: Initialize time interval set $S_0 = \{I\}$. Scale $\ell = 0$.
 - 4: **while** Δt not small enough **do**
 - 5: $\ell = \ell + 1$, $S_\ell = \emptyset$.
 - 6: **for** Each time interval $I \in S_{\ell-1}$ **do**
 - 7: Construct time series $F(I, \Delta t)$.
 - 8: Fit a curve for each dimension of $F(I, \Delta t)$.
 - 9: Sample the curves to obtain clean time series C .
 - 10: $S_\ell = S_\ell \cup \text{CAPD-MDTS}(C)$.
 - 11: **end for**
 - 12: Decrease window size Δt ,
 - 13: **end while**
 - 14: **return** S_ℓ
-

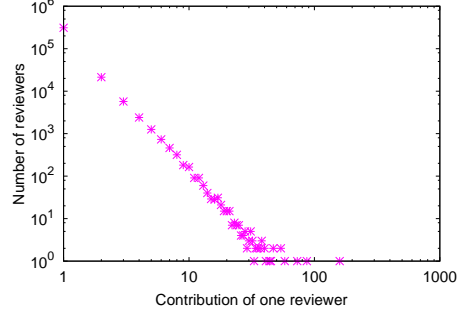


Figure 13. Contributions of reviewers

5.3 Experiments

In this section, we first describe the dataset we use, then we give a couple of case studies to provide evidences of spamming activities caught by the proposed method.

5.3.1 Review Data Description

The review data we use is a snapshot of a review website (www.resellerratings.com) on Oct 6th, 2010¹. It contains 408,469 reviews written by 343,629 reviewers (identified by their IDs on the website) for 25,034 stores. 310,499 reviewers ($> 90\%$) wrote only one reviews and about 76% ($310,499/408,469$) of the reviews are SRs. The distribution of the number of reviewers writing a certain number of reviews is plotted in logarithm scale in Figure 13. As we can see, the relation between these two quantities roughly follows the power distribution. This is also observed in (39). The main body of the data consists of reviews, along with information about

¹Thanks to Keith Nowicki

stores and reviewers. For each review we keep the following information: its rating (ranging from 1 to 5), the posting date and whether it is an SR.

5.3.2 Human Evaluation

In this section, we report the experimental results of human evaluation of the detected suspicious stores and reviews. We employ three human evaluators in this experiment.

5.3.2.1 Suspicious Store Detection

One way to use the algorithm is to run it against the reviews for a store to detect any singleton spam attack. We focus on stores with large number of SRs, so in the evaluation we select top 53 stores, each of which has more than 1,000 reviews. We ask human evaluators to read the reviews from all 53 stores and make decisions regarding the suspiciousness of these stores. If two or more evaluators vote a store as being likely to have committed an SR spam attack, we tag it to be a likely dishonest store. According to the human evaluation, there are a total of 29 stores having at least two votes. Out of the 53 stores, the proposed algorithm labels 36 ones as suspicious stores and the rest as normal ones. Out of the 36 detected ones, 22 stores have at least two votes for being suspicious. The proposed algorithm misses 7 suspicious ones. The recall is 75.86% (22/29), indicating that the proposed algorithm can catch most of the stores involved in SR spam attack. The precision is 61.11% (22/36). Though this precision looks a bit low, since our goal is to identify suspicious stores for human experts to investigate further, the proposed approach only enlarges the suspicious set moderately with a decent recall.

Table XIX shows the agreement between evaluators when evaluating the detected stores. The numbers on the diagonal show how many stores each evaluator considers as dishonest. For

TABLE XIX

HUMAN EVALUATION RESULTS ON STORES			
	Evaluator 1	Evaluator 2	Evaluator 3
Evaluator 1	17	14	16
Evaluator 2	-	20	19
Evaluator 3	-	-	24

example, evaluator 1 regards 17 out of 53 stores as suspicious ones. The off-diagonal numbers give how many stores that both evaluators in that row and column identify as dishonest stores. For example, the number on the intersection of Evaluator 1 and Evaluator 2 means that both evaluators 1 and 2 agree upon 14 stores that are suspicious stores. In any case, there are 26 stores at least one of the evaluators regarding it to be suspicious. Comparing the off-diagonal numbers with the diagonal numbers shows the limitation of the content-based approaches. Even human evaluators examining the contents cannot reach agreement a lot of the times as these cases are often very subtle.

5.3.2.2 Singleton Reviews on a Detected Store

We also ask three human evaluators to examine 147 reviews contained in the detected time window of burst given in the first case study (see next section). Each review is given a score (0-negative, 0.5-possibly, 1-positive) indicating the degree of being regarded as a spam review by each evaluator. Lastly, for each review, the scores from three evaluators are added up to get the final score. Among the 147 reviews, 43 reviews (38 are SR) have final score at least 2, and 12 reviews (11 are SR) have final score equal to 3. This indicates that many reviews identified

TABLE XX

HUMAN EVALUATION RESULTS ON REVIEWS			
	Evaluator 1	Evaluator 2	Evaluator 3
Evaluator 1	59	20	28
Evaluator 2	-	41	38
Evaluator 3	-	-	72

are indeed SR and the proposed algorithm can locate the period when SR are more likely to happen.

Table XX shows the results of human evaluation on spam reviews. Evaluator 1 tags 59 out of 147 reviews as spam reviews, while the other two regard 41 and 72 reviews as spam reviews, respectively. There are 98 reviews that at least one of the evaluators regard as spam. Similarly, the numbers off the diagonal show the agreement between evaluators. Again, this table shows that it is not easy for human beings to reach agreements on whether a review is an SR spam, and content-based methods will be less effective in the detection of this kind of spams.

5.3.3 Spam Detection Case Study

In this section, we closely study the evidences of SR attacks committed by several stores.

5.3.3.1 First Case Study

The results of running the proposed multidimensional multi-scale detection algorithm on the reviews of a store (id=24811) are shown in Figure 14. The multidimensional time series in the first subfigure (Figure 14(a)) is produced using a larger time window (30 days) with review data from Apr 2002 and to Aug 2010. The format of this figure is the same as that of Figure 12.

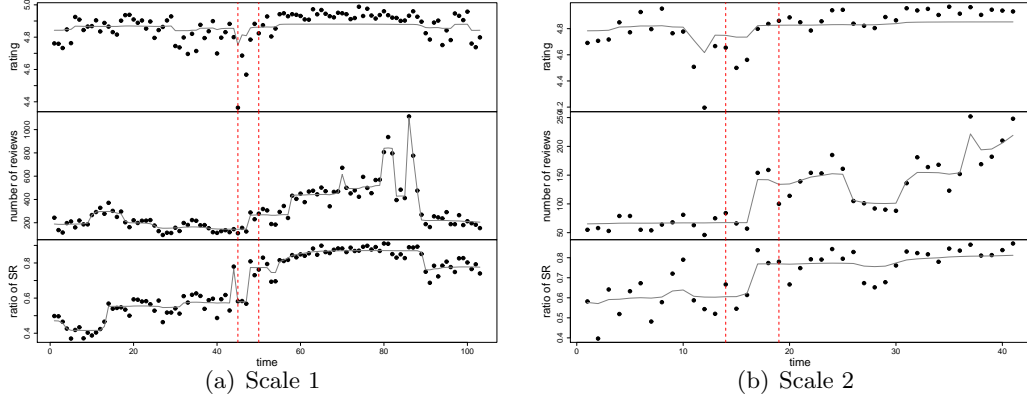


Figure 14. Anomaly detection on multi-scale multidimensional time series

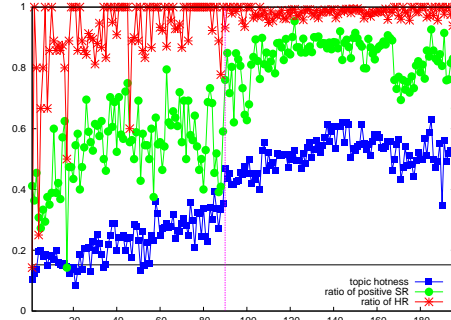


Figure 15. Topic Hotness Trend

In this higher level of detection, we find a sharp increase in $\{45 \rightarrow 50\}$, which corresponds to the time interval from Oct 30, 2005 to Mar 30, 2006. Notice the burst occurs in a two-month period. We construct another 3-dimensional time series from the review data in the detected time window to find more details of the burst. With the window size set to 15 days, we run

the detection algorithm again. The block $\{14 \rightarrow 19\}$ (from Dec 17, 2005 to Mar 3, 2006) with suspicious activities are found and highlighted in 14(b). Note that this detected time window is smaller than the previous one. In this time interval, the number of singleton reviews increases from 57 to 154, the rating goes up from 4.56 to 4.79, and the ratio of singleton reviews goes up from 61% to 83%. These all happen in a two-weeks period. It looks like the ratings were bolstered by the sudden increase of singleton reviews.

However, one might still not be convinced that there are probably spams activities in the detected time intervals. We provide further evidences by analyzing the review contents. Note that looking for these evidences is *not* part of the proposed algorithm, which only uses the reviewers' behaviors for detection. This is only for the purpose of validation. We find out that around the time when the bursts in all three dimensions are detected, the phrases “customer service” and “customer support” are unusually frequent among the SRs. Such correlation indicates that there could be spammers giving undeserving high ratings to “customer service” of that store. Next we study this correlation quantitatively.

When constructing multidimensional time series, we divide the time from Apr 2002 to Aug 2010 into intervals of two weeks. For each interval, we calculate the “hotness” of the topic “customer service”. The “hotness” of a topic is the ratio of reviews about that topic to all reviews in a certain period. If a review contains one of the phrases “customer service” and “customer support”, we consider it to be related to that topic. In Figure 15, we show the trend of the hotness of the topic in the blue curve with solid squares. One can see that there is a burst of topic hotness occurs at time 90 (Feb 06, 2006, indicated by the dashed line).

Note that this burst occurs in a two-week long period with the hotness goes up from 35% to 46%. Also, note that the time of this burst coincides with that of the burst detected in the multidimensional time series by the proposed algorithm. This makes the detected time interval look suspicious. The black horizontal solid line shows the topic hotness calculated from all reviews except those from the store being investigated. We can see that, on average, less than 16% of the reviews mentions the phrases. This number is calculated using 376,758 reviews out of the total 408,470 reviews, so it well represents the general interests of the reviewers about this topic. By comparison, we can see that the hotness within this store is twice as high as the average level. This is unlikely in normal business, since it is quite hard to gain the recognition of “customer service” from real customers in two weeks. After that time, the topic hotness keeps going up and is far higher than the average level. In particular, one out of two reviews is talking about “customer service” on average.

Besides “topic hotness”, we consider two other reviewer behaviors. The green curve with solid circles shows the ratio of singleton 5 star reviews to the topic-related reviews. We can see that from the time Feb 06, 2006 on, this ratio is rather high, namely, more than 80% of the singleton reviews are related to “customer service”. We can conclude that the burst and hotness of the topic is supported by the burst and high volume of singleton reviews. Lastly, the red curve with stars gives the ratio of reviews which are written by “hurry reviewers” (HR) to the singleton 5 star reviews. We define an HR to be a reviewer who writes a review on the same day she registers her id. From the figure, one can tell that, from Feb 06, 2006 on, a high percentage (over 90%) of 5 star singleton reviews about “customer service” are produced by

“hurry reviewers”. Since at least for those who registered in 2005 never write another review in the following 5 years, this is quite dubious. As a way of validation, we read reviews of the store in the period of topic hotness burst. We found a reviewer once disclosed the fact that the store emailed her for a favorable rating. The reviewer had an unpleasant experience with that store and got customer service only after she low-rated it on the review website.

5.3.3.2 Second Case Study

When we try to investigate a store *meritline* with high SR spams identified by the proposed algorithm, we find out it also operates under another name *cdrdvdrmedia*. Hence this case of spamming is quite interesting. The following facts support this observation: first, the addresses of the two companies are the same¹². Second, on the review website we are studying, *meritline* is an alien of *cdrdvdrmedia*. Third, according to a domain analysis website, these two stores have the same Google analytics account³. Lastly, one reviewer says the package and receipt she received were from *meritline* though she shopped with *cdrdvdrmedia*⁴. We perform the proposed multidimensional times series analysis on the reviews for *meritline*. Figure 12 (Section 5.2.2) shows the time interval when an SR spam attack is likely to have happened. *cdrdvdrmedia* sells the same set of products as *meritline* does, but with a much lower rating. There are only 48

¹www.cdrdvdrmedia.com/contact-us.html

²www.la.bbb.org/business-reviews/General-Merchandise-Retail-By-Internet/Meritline-in-City-of-Industry-CA-13135057

³domaintraker.com/meritline.com

⁴www.resellerratings.com/store/view/CDRDVDRMEDIA_17/page/1, see username “sableman”

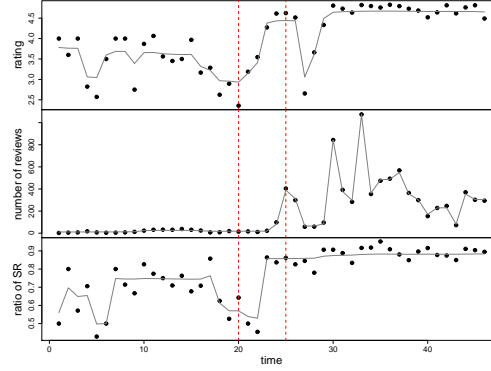


Figure 16. Bursty Patterns Detected in Store 24938

reviews in near 8 years (from Aug-2002 to Jan-2010). The average rating of the store is only 3.06 and people are talking about credit card problems, low-quality products and customer service. Therefore, the high volume of reviews and good rating for *meritline* are quite suspicious.

5.3.3.3 Third Case Study

We find another store (*supermediastore*) which is likely involved in singleton review spamming. The multidimensional time series for the store and the detected bursty patterns are shown in Figure 16. This store is probably owned by the same owner as *meritline* and *cdrdvdmedia*. This is supported by at least two forum posts¹. We also find an interesting review² telling that the reviewer was cheated by *supermediastore* when it tried to entice her into spamming. The

¹forum.doom9.org/archive/index.php/t-36023.html and forum.videohelp.com/threads/143262-Meritline-Very-Disappointed

²www.resellerratings.com/store/view/Supermediastore/page/895, see the review from the ID “defile”/

reviewer once received an email from the store about writing a review for it. In return, the reviewer would receive a “gift”, which she never receive. This is a direct evidence that this store is hiring/enticing people to write favorable reviews. This review is written during the time when there is a burst of singleton reviews.

CHAPTER 6

DEBIASING CROWDSOURCED RATINGS VIA CONSENSUS RANKING DUAL TRANSFER

(This chapter includes the paper published in *Sihong Xie, Qingbo Hu, Jingyuan Zhang, Jing Gao, Wei Fan, Philip S. Yu. “Robust Crowd Bias Correction via Dual Knowledge Transfer from Multiple Overlapping Sources”. In **BigData 2015**. ©2015 IEEE. Reprinted, with permission. (DOI: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7363827>)).*

6.1 Introduction

Crowdsourced and user-generated data are an important part of the big data being accumulated, such as the prevalent product rating and review data: hotels worldwide are reviewed and rated on Tripadvisor; several hundred million products are rated and reviewed by customers on Amazon. By hosting such comprehensive and opinionated data, these systems are not only vital to customers, but also to business owners. However, bias and noises are inevitable in these crowdsourced data, and to make sense of the data, it is important to infer objective and fair product quality measurements, such that customers can make informed decisions and business owners are not hurt by undeserved negative ratings.

The task is non-trivial since the crowdsourced data are biased for several reasons. First, although most of these rating systems employ state-of-the-art quality control mechanisms such as ReCaptcha (56), spammers can still infiltrate the systems and give arbitrary ratings to entice

customers into purchasing of low quality products. Second, affected by uncontrollable factors, regular users may rate products subjectively or spontaneously. For example, the perceptions of the quality and the ratings of a product can vary dramatically among reviewers.

There are existing works trying to address these issues in the pursue of objective product quality measurements. In (33; 32), they proposed an unsupervised method to jointly infer reviewer bias and product quality. These methods did not exploit supervision information and can possibly be misled by the inherent rating bias. In (38), they proposed to incorporate ground truth ratings to achieve better results. They further adopt active learning to further reduce the cost of expert supervisions. In (51), the authors proposed a supervised matrix factorization model to infer multiple latent factors, based on which expert ratings can be predicted using crowdsourced ratings. However, their method also require a significant amount of supervisions. In general, supervised methods are more effective in recovering unbiased information, but can incur expensive expert efforts.

Fortunately, big data provide a rich set of data sources, which we exploit via transfer learning to avoid the expensive expert efforts. The basic assumption is that multiple data sources contain information of the same subset of products. When expert ratings are too expensive to collect, weak supervisions transferred from auxiliary sources can be used to substitute expert input. For example, IMDB and Netflix can provide useful information about movies to infer a less biased movie ranking on Amazon.

We identify the following challenges in trading domain experts for multiple auxiliary sources. First, being user-generated, it is common for product ratings/rankings from multiple auxiliary

domains to be inconsistent or conflicting. For instance, IMDB might give 5 stars to a movie while Netflix gave the same product 3 stars. The challenge is to resolve such conflicts across domains. Second, due to the difference among domains, it is likely that the auxiliary domains only partially overlap with the target domain, and thus are not directly helpful to those non-overlapping products. Lastly, the auxiliary sources can be biased themselves, and are not immediately useful to the target domain.

We address the above challenges by a two-step pipeline. The first step is to resolve inter-source conflicts. It cancels out the bias in individual auxiliary sources, and extracts a single consensus product rating/ranking as transferable knowledge. The second step is to transfer knowledge from auxiliary source to the target domain. After exploring two product-centric transfer learning strategies, which are less effective in the presence of non-overlapping products, we propose a dual transfer approach, which applies the transferred knowledge to both products and reviewers (thus the name “dual”). Both the reliability of anchor reviewers and product ranking are estimated using auxiliary sources to regulate the bias correction procedure. Experiments on three real-world datasets show that the dual transfer approach outperforms previous approaches and the two single transfer approaches. Furthermore, we show that the inferred bias can be used as a signal for suspicious reviewer identification.

6.2 Preliminary

Suppose we have n products $\mathcal{V} = \{v_1, \dots, v_n\}$ which are rated by m reviewers $\mathcal{U} = \{u_1, \dots, u_m\}$. Let r_{ij} denote the rating given by user u_i to product v_j , and \mathcal{R} denote the

TABLE XXI

NOTATIONS	
Symbol	Meaning
\mathcal{U}	Set of users/reviewers, or the crowd
\mathcal{V}	Set of products to be rated
\mathcal{R}	Ratings of products from the users/reviewers
$n = \mathcal{V} $	Number of products
$m = \mathcal{U} $	Number of users
q_j	Quality of the j -th product
b_i	Bias of the i -th user
π_1, \dots, π_K	Partial orderings of \mathcal{V}
π_0	Ground truth ranking of \mathcal{V}
σ	Score function of \mathcal{V}
$\tau(\cdot, \cdot)$	Kendall- τ ranking correlation coefficient
$\rho(\cdot, \cdot)$	Spearman- ρ ranking correlation coefficient
L	Set of pairwise ranking constraints on products
S	Set of score constraints on products

collection of all ratings. \mathcal{R} can be seen as the union of \mathcal{R}^j (ratings dedicated to the j -th product): $\mathcal{R} = \cup_j \mathcal{R}^j$, or the union of \mathcal{R}_i (ratings given by the i -th reviewer): $\mathcal{R} = \cup_i \mathcal{R}_i$.

A ranking of the products is a function $\pi : \mathcal{V} \rightarrow \{1, \dots, n\}$, and $\pi(v_i) < \pi(v_j)$ (or $v_i \succeq_\pi v_j$) means that product v_i ranks higher (is better) than v_j . Let π_0 denote the unknown ground truth product ranking, the querying of which is expensive. Our goal is to correct bias in \mathcal{R} , such that the estimated ranking $\hat{\pi}$ is as close to π_0 as possible. The notations are summarized in Table XXI.

6.2.1 Unsupervised bias correction

A representative method is proposed in (38), which tries to infer unbiased product quality solely from crowdsourced ratings. Associate the product v_j with a quality score q_j and reviewer

u_i with bias b_i . We impose a reinforcement relationship between the product quality and reviewer bias:

$$q_j = \frac{1}{|\mathcal{R}^j|} \sum_{i \rightarrow j} r_{ij}(1 - b_i) \quad (6.1)$$

$$b_i = \frac{1}{|\mathcal{R}_i|} \sum_{i \rightarrow j} |r_{ij} - q_j| \quad (6.2)$$

The quality of a product is the averaged ratings dedicated to that product, adjusted by individual user bias, and the bias of a user is the averaged distance from his/her ratings to the estimated quality of the products he/she has rated.

Unfortunately, as pointed out in (38), without expert guidance, the above unsupervised algorithm is not very effective in inferring the true product quality. Suppose that the majority of the ratings for a product are fake ratings and biased toward the highest score, say 5-star, then during the first iteration, the estimated rating of the product is the average of the biased ratings and will be seriously biased towards 5-star. When a dishonest reviewer has only give a 5-star rating to that product, then the bias of this reviewer will be low, since his/her only rating agrees with the dominating fake 5-star ratings of the product.

6.2.2 Semi-supervised bias correction

In (38), the authors proposed to clip the scores of a subset of the products to expert evaluations, and infer the remaining product scores using the following equation:

$$q_j = \begin{cases} S(j) & \text{if } j \in S \\ \frac{1}{|\mathcal{R}^j|} \sum_{i \rightarrow j} r_{ij}(1 - b_i) & \text{otherwise} \end{cases} \quad (6.3)$$

where S is the set of products whose scores that are fixed at their ground truth scores (denoted by $S(j), j \in S$), and the reviewer bias is estimated as in Equation 6.2. The above equations propagate the supervision information in S to the remaining reviewers and products. By incorporating expert ratings, the method can better correct the rating bias, and the restored product scores are closer to editorial ratings. However, this semi-supervised algorithm requires large amount of input from experts ((38) labeled 50% of the products) to counteract the sensitivity of the graph-based propagation algorithm to the labeled set S .

6.3 Correcting crowd bias via transfer learning

We assume that there is no expert input to guide the bias correction, and seek for help from related external domains. We first explored two product-centric single transfer strategies, and then we point out their drawbacks and propose a dual transfer strategy.

6.3.1 Two product-centric single transfer strategies

6.3.1.1 Product rating single transfer

One can alter the semi-supervision bias correction algorithm in Section 6.2.2, and choose the overlapping products that are rated in both target and auxiliary domains as the set S . The ratings in S are fixed to the averaged scores computed from multiple auxiliary domains, while the ratings of the non-overlapping products are estimated as in Equation 6.1. Compared to the unsupervised bias correction, the transferred ratings are the average of several auxiliary domains. The underlying assumption is that the aggregated information from multiple data sources can be potentially less biased and be further propagated to the non-overlapping products, thereby overcoming the dominance of the biased ratings in the target domain. However, different domains typically have different rating scales. It is not clear how to optimally normalize the ratings from different domains to the same scale.

6.3.1.2 Product ranking single transfer

To handle the different rating scales, we can adopt product rankings from auxiliary domains as supervision, and enforce the relative rankings of the overlapping products to be the same as the transferred ranking. We first convert ratings (if any) from auxiliary domains to product rankings to eliminate difference in rating scales. After that, we let all auxiliary rankings be denoted by π_1, \dots, π_K , and define the indicator function on the pairs of products: $\mathbb{1}[v_1 \succeq_{\pi_k} v_2]$. To estimate a consensus product ranking out of the auxiliary rankings, define the following consensus score function: $\chi(v_1, v_2) = \frac{1}{K} \sum_{k=1}^K \mathbb{1}[v_1 \succeq_{\pi_k} v_2]$. χ serves as a ranking agreement measurement, and the higher χ is, the more the auxiliary rankings agree upon the ordering of

the two products. The weights of individual sources are chosen to be uniform since without expert input, it can be hard to determine which source is more reliable than the other. In the experiments, we only retrieve the product pairs with their orderings agreed upon by all auxiliary rankings.

$$v_1 \succeq_{\bar{\pi}} v_2 \iff \chi(v_1, v_2) = 1 \quad (6.4)$$

Essentially, we have extracted a partial ordering of the overlapping products, denoted by $\bar{\pi}$. We incorporate $\bar{\pi}$ in the unsupervised bias correction procedure as follows. After calculating the product quality using Equation 6.1, we solve the following optimization problem:

$$\begin{aligned} \mathbf{q}^* = \arg \min_{\mathbf{q}} \quad & \sum_{j=1}^n \|q_j - \bar{q}_j\| \\ \text{s.t.} \quad & q_j \geq q_\ell \text{ if } v_j \succeq_{\bar{\pi}} v_\ell \end{aligned} \quad (6.5)$$

where $\bar{\mathbf{q}} = [\bar{q}_1, \dots, \bar{q}_n]$ is the product quality scores found by Equation 6.1. This optimization problem models two objectives. First, the inferred scores of *all* products should be close to the averaged ratings obtained from the ratings, with reviewer bias taken into account. Second, the inferred scores of the overlapping products $q_i, i = 1, \dots, n$ are forced to follow the consensus ranking $\bar{\pi}$. Equation 6.2 then takes the solution of the above optimization problem as input to estimate reviewer bias, and the iterations go on until convergence. The algorithm is called **CRST** (**C**onsensus **R**anking **S**ingle **T**ransfer).

6.3.2 A robust dual transfer approach

In product-centric single transfer approaches, the transferred rating/ranking is only used to correct possible rating/ranking bias in the target domain. We want to utilize the transferred rating/ranking in a more effective way by adding a “reviewer-centric” perspective and propose a novel dual transfer approach, which robustly applies the transferred knowledge to both products and reviewers. In the following, we first introduce the concept of “anchor reviewer reliability” that can facilitate reviewer-centric transfer learning.

6.3.2.1 Anchor reviewer reliability estimation and confident anchor reviewer identification

Anchor reviewers have reviewed both overlapping and non-overlapping products *in the target domain*, and serve as a bridge between the overlapping and non-overlapping products. If we can robustly estimate their reliability using their ratings/rankings on the overlapping products, and incorporate these reliability in bias correction, then one can expect a better ranking of the non-overlapping products. Assume that the transferred ranking of the overlapping products is of reasonable quality (though may not be perfect), the similarity between the transferred ranking and the product ranking provided by an anchor reviewer can be indicative of the reliability of the reviewer. We adopt Kendall- τ ranking correlation coefficient (31) to measure the reliability of anchor reviewers. Formally, reliability of an anchor reviewer is defined as

$$\tau(\pi_1, \pi_2) = \frac{2(C - D)}{n(n - 1)} \quad (6.6)$$

where C (D , resp.) is the number of concordant (discordant, resp.) pairs of products in the two rankings π_1 (the transferred ranking) and π_2 (product ranking provided by an anchor reviewer). The function τ takes values in $[-1, 1]$, and a higher τ indicates the anchor reviewer is more reliable, and vice versa.

Note that if the denominator in Equation 6.6 is small, then the sample for computing the correlation is small and the estimated reliability of the anchor reviewer is less confident. We require there is a sufficiently large number of overlapping products between the transferred ranking and the product ranking provided by an anchor reviewer. However, if one requires a large number of overlapping products, too many anchor reviewers may be dropped off and bias correction may be affected. We will investigate this trade-off empirically later.

6.3.2.2 Incorporating reviewer reliability in the single transfer strategy

Now we need to use the anchor reviewers to help correct the bias. The bias of a reviewer is computed as in Eq. (Equation 6.2), but when computing the quality of a product (be it an overlapping one or not), we use the following equation:

$$q_j = \frac{1}{|\mathcal{R}_{\cdot j}|} \sum_{i \rightarrow j} r_{ij}(1 - b_i) \times (1 + rel(i)) \quad (6.7)$$

The meaning of this equation is that, for an anchor reviewer, his/her rating for the j -th product should be amplified or discounted by his/her reliability, while for a regular reviewer, there is no effect of reliability and Equation 6.7 is just the same as Equation 6.1. The algorithm is called **CRDT** (**C**onsensus **R**anking **D**ual **T**ransfer) and summarized in Algorithm 8. Compared with

CRST, **CRDT** has an additional step of anchor reviewer reliability estimation, and uses a different formula to estimate product scores, with anchor reviewer reliability taken into account. **CRDT** applies the transferred consensus ranking to both products (ranking constraint) and reviewers (reliability estimation) in the target domain.

Algorithm 8 Robust Bias Correction via Consensus Ranking Dual Transfer (**CRDT**)

Input: anchor reviewers $\{u_1, \dots, u_s\}$, product ratings \mathcal{R} in target domain, multiple external rankings $\pi_i, i = 1, \dots, k$

Output: q_j for the products.

Compute the consensus product ranking $\bar{\pi}$ from $\pi_i, i = 1, \dots, K$, using Equation 6.4.

for $i = 1 \rightarrow s$ **do**

 compute anchor reviewer reliability for u_i .

end for

while not convergent **do**

 Estimate reviewer bias using Equation 6.2.

 Estimate unbiased product rating using Equation 6.7.

 Enforce ranking of the overlapping products to agree with $\hat{\pi}$ by solving Equation 6.5.

end while

6.3.3 Computational complexity analysis and incremental model update

We first consider the time complexity of building **CRDT** from scratches. The time complexity to compute a consensus ranking is linear in the number of ratings from all auxiliary sources. These computations can be distributed to multiple machines as there is no information sharing among sources. The space complexity $O(n)$ to store the averaged ratings of the products (instead of $O(n^2)$ to store the pairwise rank comparisons). Regarding calculating anchor

reviewer reliabilities, a rather loose upper bound of the time complexity is $O(|\mathcal{R}|)$, namely the time complexity to go through all ratings. However, only a small portion of the reviewers are anchor reviewers, and only their ratings need to be visited during reliability calculation. The time complexity of estimating product quality and reviewer bias using Equation 6.1 and Equation 6.2 is $O(T * |\mathcal{R}|)$ where T is the number of iterations needed for Algorithm 8 to converge. We show in the experiments that T is usually quite small and can be considered as a constant. Overall, both the time and space complexity of the proposed method is linear in $|\mathcal{R}|$.

Since the ratings keep accumulating, it is also important to consider incremental updates. It is trivial to update the product ratings for each auxiliary source. To update the reliabilities of the anchor reviewers, only their updated ratings will get involved, and that's a small number since normal reviewers don't usually add new ratings in a short period. Lastly, we only need to re-run Equation 6.1 and Equation 6.2 once to update the solutions, using the q_i and b_j from previous iterations, since the bipartite graph and the reliabilities do not change significantly from previous iterations.

6.4 Experiments

6.4.1 Datasets and Performance Metrics

We employ a rating dataset collected from multiple rating websites as our testbed. Table XXII describes the rating data of three cities, New York City (NYC), Phoenix (PHX) and San Francisco (SF) from tripadvisor.com, which is our target domain. As external domain rankings, we collect ratings of the same set of restaurants from foursquare.com and yelp.com. Similar to (51), ratings from Zagat.com are treated as ground truths. We use the number of

TABLE XXII
CHARACTERISTICS OF RATING DATASETS

	NYC	PHX	SF
# of restaurants	79	77	85
# of users	9829	5804	8183
# of ratings	12415	8050	10186

concordant pairs of products that are ordered consistently between the ground truth ranking and the ranking derived by various bias correction algorithms. A good rating bias correction algorithm should produce more concordant pairs.

6.4.2 Baselines and experimental protocol

One can simply average the ratings of each product and derive a product ranking. We denote this method by “MEAN”. This baseline does not take care of reviewer bias explicitly. We consider three more sophisticated baselines that explicitly consider rating bias. The unsupervised model proposed in (38) (denoted by “UN-SUP”) iteratively and alternatively applies Equation 6.1 and Equation 6.2 until it converges. There are two baselines that exploit transferred knowledge, either by clipping the ratings of the overlapping products to the transferred ratings (called “S-SUP”), or by enforcing the ranking of the overlapping products to be the same as the transferred ranking (**CRST**). These two baselines do not consider and model reviewer reliability.

We randomly and uniformly select half of the products as overlapping products. The performance metric is computed on the non-overlapping parts, in order to check if the transfer

knowledge can be propagated to the non-overlapping products and improve their ranking. We repeat the experiment 100 times and report the averaged performance. The proposed algorithm have a parameter (n in Equation 6.6) to cut off the reviewers who have less confident estimation of their reliability. We fix this parameter to be 3 in the following results, and study the sensitivity of this parameter in Section 6.4.3.1.

6.4.3 Results

Figure 17 compares the number of concordant pairs of products according to various rating debias methods on 3 datasets. From the figure, we can observe the followings. First, MEAN (yellow bar) has significant lower performance than other methods among 2 out of 3 tasks (NYC and SF), and is slightly better than UN-SUP (black bar) and CRST (red bar) on the other task (PHX). One possible explanation of such mixed performance is that the average rating can sometimes remove the rating bias of individual reviewers, but can also fail to do so if a product is only rated by a few biased reviewers. Second, CRST and UN-SUP have similar performance across all 3 datasets. This surprising fact indicates that the transferred ranking may be too difficult to be propagated to the non-overlapping products. The reason is that the transferred rankings are used to enforce the orderings of the overlapping products, while bias of reviewers is inferred indirectly, which is not very effective. Third, the performance of S-SUP (pink bar) is generally worse than that of UN-SUP and CRST, which is caused by the heterogeneity among different rating systems. Indeed, if the difference between rating scales is not handled carefully, simply normalizing and averaging ratings from different systems can be harmful. Lastly, we see that the proposed method (blue bar) performs the best. We conclude

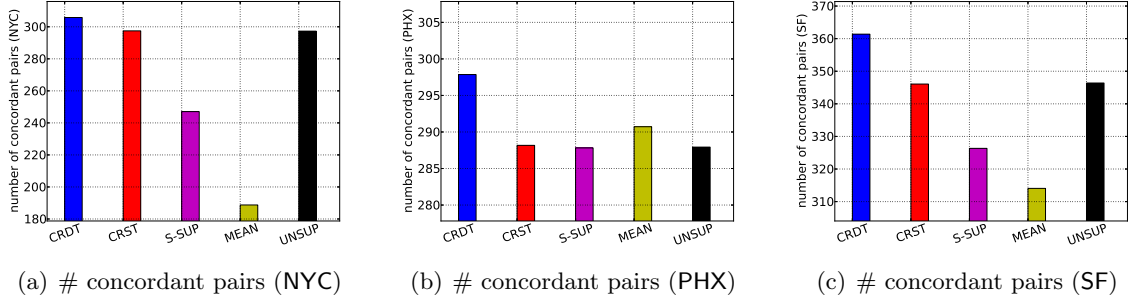


Figure 17. Overall comparisons of the proposed method and the baselines

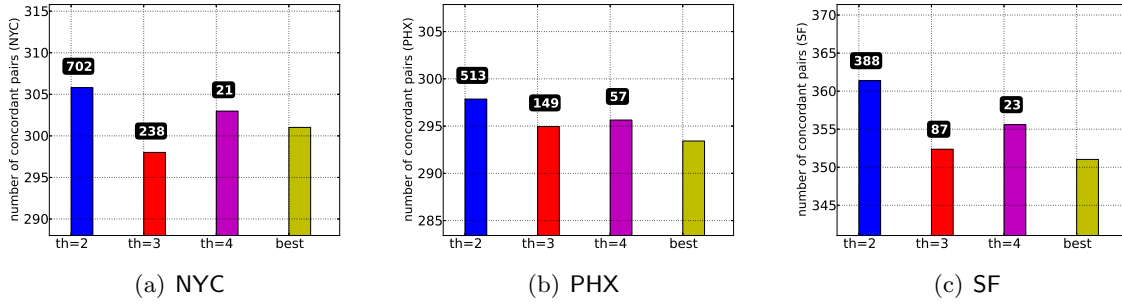


Figure 18. Sensitivity of CRDT (“best” indicates the baseline with the best performance)

that the transferred rankings can indeed be used to find out the reliability of reviewers, which can effectively adjust the ratings on the non-overlapping products to more objective ratings, leading to a better ranking of the products.

6.4.3.1 Sensitivity study

An anchor reviewer has to rate more than a certain number of overlapping products to be qualified as a confident anchor reviewer. We set this threshold to be 2, 3 and 4, and report

the performances of the proposed method, along with the performance of the best baseline, in Figure 18. We also indicate the number of anchor reviewers under each threshold on top of the bars in the figure. In general, we see that the proposed algorithm works best when the threshold is set to 2. The only setting our method is not as good as the best baseline is when the threshold is set to 3 on the NYC dataset. We have following conclusions. First, the performance of the proposed reviewer-centric approach is not that sensitive to the threshold. Even we set the threshold to as high as 5 and there are only tens of anchor reviewers, the CRDT method still outperforms the best baseline. Second, when the threshold is set to 2, and the reliability of an anchor reviewer is evaluated using the ordering of only 3 products. The resulting reliability estimation might not be very confident based on such a small sample. However, we can obtain a larger number of anchor reviewers to cover more non-overlapping products. The superior performance when the threshold is 2 indicates that the coverage of product by the anchor reviewers is more important than confidence of the reliability estimation. In practice, one can leave out a validation set to pick up the best threshold. The readers are referred to the full version¹ for more details, including the convergence of the proposed algorithm and its potential for suspicious reviewer detection.

¹<http://www.cs.uic.edu/~sxie/papers.html>

CHAPTER 7

A CONTEXT-AWARE APPROACH TO DETECTION OF SHORT IRRELEVANT TEXTS

(This chapter includes the paper published in *Sihong Xie, Jing Wang, Mohammad S.Amin, Baoshi Yan, Anmol Bhasin, Clement Yu, Philip Yu. “A Context-Aware Approach to Detection of Short Irrelevant Texts”. In **DSAA 2015**. ©2015 IEEE. Reprinted, with permission. (DOI: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7344831>)).*

7.1 Introduction

Popular online content providers such as LinkedIn.com and CNN.com are attracting millions of visitors per day. Meanwhile, spammers and irresponsible visitors are leaving irrelevant comments after the major contents, making the websites less attractive to visitors and reducing the websites’ traffic and revenue. It is critical to detect these irrelevant contents accurately as soon as possible. However, this is not an easy task due to the following reasons. First, comments are usually very short, and given such limited information, it is difficult to capture the semantics and relevance of the comments. Second, under different contexts, the same word can have quite different meanings. For example, given two news articles on real estate and NASA’s mars exploration plan, respectively, the term “space” used in the comments of these articles can refer either to “an area rented or sold as business premises” or “the physical universe beyond the earth’s atmosphere”, two completely different concepts. The key observation is that the

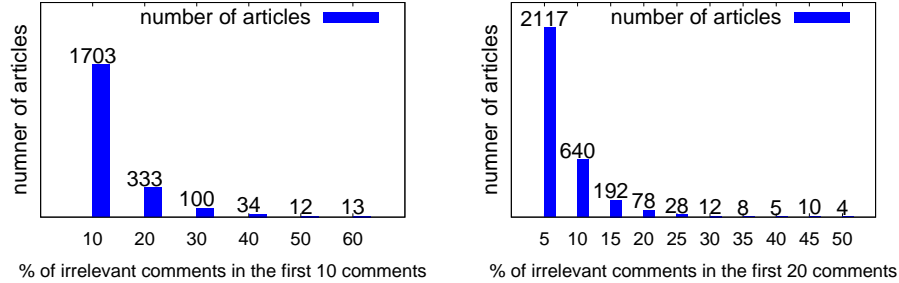


Figure 19. Early detection as a real world problem

“context” of a comment plays an important role in defining the semantics and relevance of the comment. Third, in real world applications, there are situations where irrelevant comments are posted soon after the release of an article, with only a small number of comments. In Figure 19, we plot the number of articles on LinkedIn’s news channel having various percentage of irrelevant comments at early stages. For instance, in 19(a), we count the number of articles having 10%, 20%, etc. of irrelevant comments among the first 10 posted comments. It is obvious that a large number of articles have at least one irrelevant comment among the first 10 comments. The earlier one can remove these irrelevant contents, the less the visitors will be distracted. We call this task “early detection” of irrelevant contents, where irrelevant comments have to be spotted when there are only a handful of comments following the same article. It is much more difficult to measure the context-aware semantics and relevance of a comment at an early stage, since there is less information about the context of the comment.

Previous works failed to address the above challenges in a single framework. Regarding short text mining, there are two traditional ways: topic modeling and transferring of external

data sources. (44) proposes to enhance the bag-of-word model using LDA (7). In (67; 68), the authors propose novel topic models for short texts, and yet they did not address early detection. Exploiting external corpora are also proposed to address the short text challenge, such as the works in (72; 47; 35; 27). However, under the specific setting of the paper, how to define and transfer from external sources have not been investigated. Furthermore, these works focus on handling the sparseness of individual documents, instead of mitigating the sparseness of corpus that arises in early detection. The works (58; 28; 6; 29; 37; 36) try to characterize and catch irrelevant comments via bag-of-word model, sequence mining or information theoretical approach, but they also fail to address all the above challenges. On the one hand, the above methods derive the semantics of comments in a context-agnostic way, leading to more confusing semantics and degraded irrelevant content detection performance. On the other hand, early detection of irrelevant comments, though being critical in real applications, has been overlooked so far, to the best of our knowledge.

We propose to resolve the above three challenges in a unified framework. We want to derive context-dependent (i.e. context-aware) semantics of short texts regardless of the stages of commenting activities, such that it is more accurate in relevance measurement than those derived without considering contexts (context-agnostic). The context-dependent semantics of a comment is determined by the semantic environment (surrounding texts) where the comment sits in (such as the varying meaning of the word “space” in the above example). It is essential to select proper texts that are semantically meaningful and comparable to a comment as its context. We construct the “native context” of a comment as the set of the comments posted for

the same article, since these comments are more likely to be similar to each other in terms of language, topics, etc.. The constructed native contexts can be coupled with any topic models to derive context-dependent semantics from short comments. Specifically, one can treat a native context as a corpus and employ any topic models such as LDA or SVD to find the context-dependent latent topics of the comments.

The native context constructed above assumes that there are sufficient comments posted for one article to serve as the context of a comment. However, regarding the early detection of irrelevant comments, one needs to tell irrelevant comments from only a handful of other comments. In other words, there are only a small number of comments in a native context at an early stage, posing difficulties to most topic models, which usually require a moderate number of documents for reliable topic inference. A key observation is that comments posted for articles on similar topics are more likely to have similar usages of language. For example, the comments following *articles* on “real estate” are more likely to use the term “space” in the sense of “residential/commercial space” rather than “space exploration”. We propose to transfer similar short texts from other articles of similar topics to construct “transferred contexts”, which inherit the strength of native contexts but avoid the sparseness of contextual information. Then similar topic models can derive context-dependent semantics for relevance measurement.

7.2 Irrelevant content detection

Nowadays, popular websites allow users to post their opinions, mostly in the form of text comments following articles published by the websites. For example, on news websites such as CNN.com, a visitor can express his/her opinions after reading the news about Obama’s

promotion of a new healthcare plan. Digg.com, wordpress.com and other social networks try to improve user engagements by deploying news and article sharing platforms, where their members can read the shared articles and post their opinions as responses. Due to the high visibility of the news and social network websites, spammers are joining the community to produce junk comments. Also, there are readers who are exploiting the traffic to these websites and distracting other visitors to irrelevant topics. These irrelevant comments can be detrimental to user experience of the websites, whose traffic and revenue will be affected. It is therefore an emergency task for the operators of these popular websites to detect undesirable comments and take appropriate actions. Intuitively, a normal comment should either respond to the contents of the article it follows, or sound similar to other comments following the same article (we called these comments the “surrounding comments”). Therefore, the irrelevant comments can be detected by measuring the similarity between a comment and the article it follows, and also between the comment and its surrounding comments. If either of the similarities is too low, then the comment is likely to be an irrelevant one (36; 58; 10). Indeed, content similarity is the most natural definition of relevance, as it is the way human interpret contents.

More formally, assume an article $\mathbf{w}_d \in \mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_D\}$ is followed by a set of C_d comments $\mathbf{Q}^d = \{\mathbf{q}_1^d, \dots, \mathbf{q}_{C_d}^d\}$ (see Table XXIII for a summary of notations). $\mathbf{w}_d = \{w_{dn}\}_{n=1}^{N_d}$ and $\mathbf{q}_k^d = \{q_{kn}^d\}_{n=1}^{N_k^d}$ are the vectors of words of the d -th article and the k -th comment for the article, respectively. N_d and N_k^d are the lengths of the article and the comment, respectively. Assume $f(\cdot)$ is a language model, which is a transformation from the bag-of-word vector representation of a document to another vector representation. For example, LDA (Latent Dirichlet Allocation)

maps a document to a vector of topic distribution, while an identity transformation is simply the bag-of-word vector of a document (see Section 7.3 for more details). Such a transformation might be necessary for text mining since it can potentially capture the high-level meanings of the documents, especially when the documents are short. Given a transformation $f(\cdot)$, the signals for irrelevant comment detection based on *text* can be calculated as the cosine similarity between $f(\mathbf{q}_k^d)$ (the comment) and $f(\mathbf{w}_d)$ (the article the comment follows) and the mean of $\{f(\mathbf{q}_1^d), \dots, f(\mathbf{q}_{C_d}^d)\}$ (10):

$$\cos(f(\mathbf{w}_d), f(\mathbf{q}_k^d)) = \frac{\langle f(\mathbf{w}_d), f(\mathbf{q}_k^d) \rangle}{\|f(\mathbf{w}_d)\| \cdot \|f(\mathbf{q}_k^d)\|} \quad (7.1)$$

$$\cos(\mathbf{m}_d, f(\mathbf{q}_k^d)) = \frac{\langle \mathbf{m}_d, f(\mathbf{q}_k^d) \rangle}{\|\mathbf{m}_d\| \cdot \|f(\mathbf{q}_k^d)\|} \quad (7.2)$$

where \mathbf{m}_d is the center of all transformed vectors of comments following \mathbf{w}_d

$$\mathbf{m}_d = \frac{\sum_{\mathbf{q} \in Q^d} f(\mathbf{q})}{C_d} \quad (7.3)$$

We call Equation 7.1 the “comment-to-article” irrelevance signal and Equation 7.2 the “comment-to-center” irrelevance signal.

From the above formula, one can see that similarity measurement requires a vector representation of texts, namely the transformation $f(\cdot)$. Ideally, $f(\cdot)$ should capture the meaning of the texts well for the detection signals to make sense. However, this is not an easy task and there are three challenges. First, comments are usually very short, compared to the documents processed in traditional text mining. In general, the articles published by the websites are of medium

length such that they are easy for the readers to follow. In contrast, the comments that follow are usually short, since readers are less serious and therefore unable or unwilling to produce long and organized texts. Figure 20 shows the distribution of the length of comments from a social network website, and one can see that most of the comments have less than 150 words. Due to the sparsity of the comment texts, the information provided by individual comment is very limited, and dimension reductions are usually required for this situation (44; 7; 67; 68), though it is unclear from the previous work that how effective these methods are in the irrelevant short text detection task.

Second, the semantics of comments are context-dependent. Specifically, a word in the comments might mean two different things under articles on two different topics, as the above-mentioned example shows. This variety of the semantics of words can not be fully captured by the bag-of-words representation or any other dimension reduction methods such as LDA (7), pLSA, SVD, etc., since these models ignore the contexts where a piece of text is generated. These methods are “context-agnostic”. As a result, given a comment, these models will give the same vector representation for the comment, no matter where the comment is posted. This is undesirable since under different contexts, an ideal language model should be able to capture subtle semantic difference.

Third, in real world applications, real time actions to irrelevant contents are of high priority. Spammers or promoters are more likely to post junk comments soon after an article is posted, such that a larger amount of audience can see the undesirable comments (as shown in Figure 19). Meanwhile, if too many visitors read the undesirable comments, they can have an unpleasant

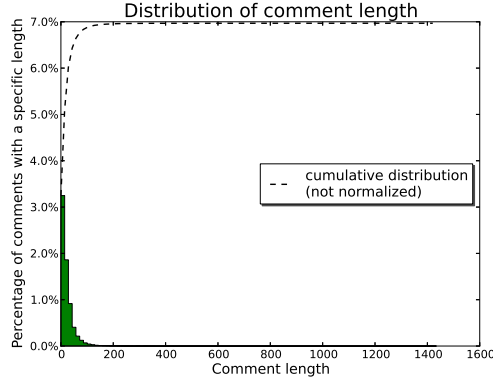


Figure 20. Distribution of length of comments

experience, leading to a lower user engagement. Therefore, it is necessary for website operators to detect irrelevant comments as soon as they show up. However, the lack of surrounding comments makes it difficult to define context for an early comment, and one might have to resort to the less effective context-agnostic approaches. To sum up, it is an important yet difficult problem to detect irrelevant short texts, with context-dependent semantics and lack of contexts. Before we address the above challenges, we first briefly review some existing context-agnostic methods.

7.3 Context-Agnostic Detection Models

7.3.1 Simple Language Model

The simplest language model is perhaps the bag-of-words representation of documents. Using this model, a document \mathbf{w} is given by a vector describing the number of occurrences of words (or the TF-IDF processed version) in the document. Then the bag-of-word vector

TABLE XXIII

NOTATIONS

Symbol	Meaning
\mathbf{W}	the collection of major posts
\mathbf{w}_d	the d -th post
\mathbf{Q}^d	the comments following the d -th post
\mathbf{q}_k^d	the k -th comment following the d -th post
C_d	the number of comments following post \mathbf{w}_d
N_d	the length of the d -th post
D	the size of the corpus
$\ \cdot\ _F$	Frobenius norm of a matrix
$\langle \cdot, \cdot \rangle$	inner product
$f(\cdot)$	a transformation defining a language model

transformation function $f_{bow}(\cdot)$ is simply an identity function. (36) adopts this language model and use the comment-to-article similarity (Equation 7.1) to detect irrelevant comments. A drawback of bag-of-words vector representation is that the vectors are usually sparse, given a large vocabulary. Indeed, in (44), it is shown that LDA (introduced next) can greatly improve the classification performance based on cosine similarity on short texts.

7.3.2 Probabilistic Topic Models

Probabilistic topic models assign a distribution of topics to a document. A popular one is the LDA (Latent Dirichlet Allocation) model. The success of LDA relies on its ability to learn topic distributions of terms and documents simultaneously. LDA assumes that a document is a mixture of topics and each word in the document is generated according to the topic of

the document and the distribution of words over topics. More formally, given a document

$$\mathbf{w}_d = \{w_{dn}\}_{n=1}^{N_d},$$

$$\theta_d \sim \text{Dir}(\boldsymbol{\alpha})$$

$$z_{dn} \sim \text{Multi}(\theta_d) \quad \forall n = 1, \dots, N_d$$

$$w_{dn} \sim \text{Multi}(\boldsymbol{\Phi}_{z_{dn}}) \quad \forall n = 1, \dots, N_d$$

where θ_d is the K dimensional topic distribution of document \mathbf{w}_d , and z_{dn} is the topic of the word w_{dn} . $\text{Dir}(\boldsymbol{\alpha})$ is the Dirichlet distribution with parameter $\boldsymbol{\alpha}$ and $\text{Multi}(\theta)$ is the multinomial distribution with parameter θ . Given a corpus \mathbf{W} , LDA infers the quantities θ_d , z_{dn} and $\boldsymbol{\Phi}$. Monte Carlo Markov Chain (MCMC) and variational methods are widely used for model inference and learning. Let $f_{lda}(\mathbf{w}_d) = \theta_d$ be the vector transformation function derived from LDA.

7.3.3 Matrix Factorization based Models

Besides LDA, matrix factorization based methods are also employed to find topics of documents. Usually, the observed corpus is modeled as a term-document matrix \mathbf{W} (here we abuse the notation), which is further factorized into the product of two or three matrices. For example, in LSI (Latent Semantic Indexing (12)) or SVD (25),

$$\mathbf{W} = U\Sigma V^\top \tag{7.4}$$

where U (V) is the left (right) matrix of singular vectors and Σ is the diagonal singular value matrix. Here U gives the topic distributions of words and V gives the topic distributions of documents. Therefore, the vector transformation function is given by $f_{svd}(\mathbf{w}_d) = V_d$, where V_d is the d -th row of V . In a similar form, non-negative matrix factorization (NMF) has also been shown to be effective in finding latent topic of documents in information retrieval (65). Formally, NMF solves the following optimization problem

$$\min_{U,V} \quad \|\mathbf{W} - UV^\top\|_F \quad (7.5)$$

$$\text{s.t. } U_{ij} \geq 0, \quad V_{ij} \geq 0 \quad \forall i, j \quad (7.6)$$

Similar to SVD, a row vector in the factor matrix V gives the topic distribution of a document and $f_{nmf}(\mathbf{w}_d) = V_d$.

7.3.4 Detection Signals based on Context-Agnostic Models

Based on the above models and Equation 7.1 and Equation 7.2, we define several irrelevant comment detection signals, which are summarized in Table XXIV. In the table, each row specifies a signal (e.g. σ_1), and the signals in the rows “Native” and “Transferred” will be defined in the next section. A check mark under the column “Mean” (“Article”) indicates that Equation 7.2 (Equation 7.1, respectively) is used to compute the signal. Note that each of $\sigma_i, i = 1, \dots, 4$ includes two similarities. These models cannot handle context-dependent semantics: none of them takes the contexts of a comment into account when computing the

TABLE XXIV

CONTEXT-AGNOSTIC IRRELEVANT COMMENT DETECTION SIGNALS

Signal	Context	Transformation	Mean	Article
σ_1	Agnostic	f_{bow} (36)	✓	✓
σ_2	Agnostic	f_{lda} (44)	✓	✓
σ_3	Agnostic	f_{svd}	✓	✓
σ_4	Agnostic	f_{nmf} (65)	✓	✓
σ_5	Native	f_{svd}^N	✓	
σ_6	Transferred	f_{svd}^G	✓	

transformations $f(\cdot)$, thus the derived signals $\sigma_1, \dots, \sigma_4$ fail to capture the context-dependent semantics when used for irrelevant comment detection.

7.4 Context-Aware Detection Signals

Below we first introduce “native context” to derive context-dependent semantics of short comments. Then we point out a practical situation where this native construction may fail, and propose a “transferred context” to handle the difficulty.

7.4.1 Native Contexts

The vector transformation function $f(\cdot)$ used in Equation 7.1 and Equation 7.2 should depend on the contexts of a comment. We observe that an article sets up the topics that are to be discussed by the comments that follow, which should have similar usages of language. Therefore, the articles naturally separate all comments into groups, each of which defines a context for the comments within. If one can learn a language model (a transformation) using such contexts for the comments, then context-dependent semantics of the comments are more likely to be well-captured.

Formally, we define the native context (NC) of a comment, say \mathbf{q}_k^d , to be the neighboring comments following the same article as \mathbf{q}_k^d , namely, all the comments in \mathbf{Q}^d :

$$\text{NC}(\mathbf{q}_k^d) = \mathbf{Q}^d$$

To learn a context-aware language model for \mathbf{q}_k^d using \mathbf{Q}^d , matrix factorizations, such as SVD, can be applied to the term-document matrix constructed from \mathbf{Q}^d :

$$\mathbf{Q}^d = U^d \Sigma^d (V^d)^\top \quad (7.7)$$

Here we abuse the notation by using \mathbf{Q}^d for both the set of comments and the term-document matrix constructed from the set. We use superscript d to emphasize that the decomposition depends only on the neighboring comments, instead of *all* comments in the corpus. The resulting factor matrix V^d gives a context-aware topic distribution of the comments:

$$f_{svd}^N(\mathbf{q}_k^d) = V_k^d \quad (7.8)$$

where V_k^d is the k -th row of V^d and $f_{svd}^N(\cdot)$ is the vector-to-vector transformation obtained by decomposing the native context using SVD. Lastly, we compute a signal (σ_5 in Table XXIV) for irrelevant comment detection by plugging $f_{svd}^N(\cdot)$ in Equation 7.2 and Equation 7.3:

$$\cos(\mathbf{m}_d, f_{svd}^N(\mathbf{q}_k^d)), \quad \mathbf{m}_d = \frac{\sum_{\mathbf{q} \in \mathbf{Q}^d} f_{svd}^N(\mathbf{q})}{C_d} \quad (7.9)$$

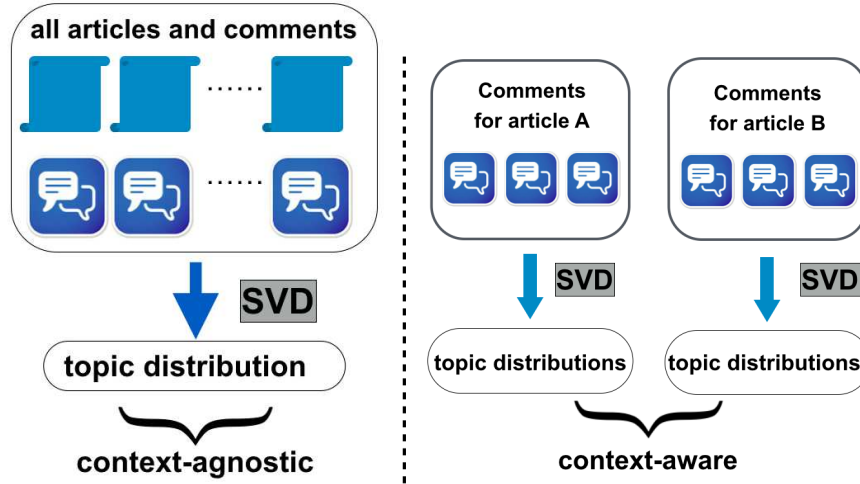


Figure 21. Context-Agnostic vs. Context-Aware methods

Note that we do not include the corresponding article \mathbf{w}_d in the decomposition in Equation 7.7, since the length of an article and a comment can differ dramatically such that the decomposition will be biased to favor the article. Indeed, we observed in the experiments, that including the article in the native context of a comment actually hurts the performance (not reported). As a result, we do not use comment-to-article similarity for detection. Nonetheless, one will soon see that the articles play a critical role in addressing the sparsity issue in early detection. In summary, the difference between context-agnostic and context-aware language models is demonstrated in Figure 21. On the left we pool all articles and comments together and apply SVD to the corresponding term-document matrix, and on the right we perform multiple SVDs on the term-document matrices derived from native contexts.

7.4.2 Early Detection of Irrelevant Comments

Although the proposed native context can define and measure context-dependent semantics and relevance in normal settings, it is insufficient for the early detection task. In particular, when there are only a small number of comments following one article, the term-document matrix (\mathbf{Q}^d in Equation 7.7) fails to provide enough information for SVD to infer meaningful topic distributions for the comments. Even if one could manage to estimate the topic distributions of the comments, the comment-to-center similarity signal would not make much sense. This is because the center \mathbf{m}_d in Equation 7.9 is the mean of a small sample and thus the variance of this estimation can be rather high according to large sample theory (2), making the signal too noisy for reliable detection. However, if one totally ignores contextual information, the context-dependent semantics cannot be sharply defined. As shown in the experiments, the lack of context leads to degenerated performance.

We propose to generalize the native contexts and add more information. The native context for a comment is defined based on the “comment-follows-article” relationship, as shown in the right panel of Figure 21. The essence of native context is to exploit the topical coherence among comments following the same article. We adopt the same idea to include more comments to define a useful context that can mitigate the sparseness of comments in early detection. The intuition is that articles of similar topics are likely to be followed by comments of the same topics, with similar usage of language. For example, the term “space” in the comments following *multiple* articles on “real estate” is likely to unambiguously refer to “a continuous area for human daily activities”, instead of “the physical universe around the earth”. Therefore,

we can transfer the comments from articles with similar topics to define a context for the comments under investigation. Such transfer is possible since popular websites store past articles and the associated comments in their databases. However, there are drifts in concepts and distributions in the comments in different articles, not all historic comments are useful for the current detection tasks. To address this issue, among the comments from similar articles, we only transfer comments that are most similar to the current ones. We define these transferred comments, together with the current comments, as the “transferred context”.

Algorithm 9 Constructing Early Detection Signal using Transferred Context

- 1: **Input:** An article \mathbf{w} with its comments $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_C\}$, a collection of past articles $\{\mathbf{w}_d\}_{d=1}^D$ and associated comments $\{\mathbf{Q}^d\}_{d=1}^D$.
 - 2: **Output:** Irrelevance detection signal σ_6 for $\mathbf{q}_k \in \mathbf{Q}$.
-
- 3: Derive LDA topics for $\{\mathbf{w}\}$ and \mathbf{Q} using trained LDA model.
 - 4: Retrieve top ℓ most similar articles to \mathbf{w} from $\{\mathbf{w}_d\}_{d=1}^D$ using LDA topics. The retrieved articles are $R = \{\mathbf{w}'_1, \dots, \mathbf{w}'_\ell\}$.
 - 5: **for** $\mathbf{q}_i \in \mathbf{Q}$ **do**
 - 6: Retrieve top 50% most similar comments to \mathbf{q}_i from the comments associated with articles in R .
 - 7: **end for**
 - 8: Define transferred context for \mathbf{Q} as the union of the retrieved comments and \mathbf{Q} .
 - 9: Apply SVD to the transferred context to find context-dependent semantics of \mathbf{Q} .
 - 10: Return σ_6 calculated using Equation 7.2 and Equation 7.3.
-

The idea of constructing transferred contexts and the corresponding detection signal is described in Algorithm 9, and is demonstrated in Figure 22. In summary, transferred contexts

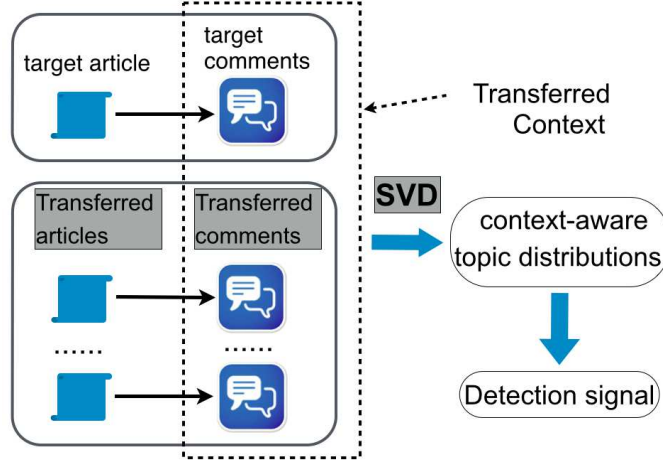


Figure 22. Transferred Contexts

address the sparsity of neighboring comments that native contexts suffer, and allow topic models to define context-aware semantics that is not available in context-agnostic methods.

Since we are focusing on early detection, efficiency becomes an issue. Here we claim that the run-time of Algorithm 9 allow the algorithm to be practically useful. First, there is no intensive computation involved in deriving topics using a trained LDA model. The retrieval of articles (step 4) can be done in parallel frameworks like MapReduce. Similarly, step 5 to 7 can be done in parallel, where each \mathbf{q}_i can be processed independently. Lastly, though in general SVD requires cubic time complexity, the matrix to be decomposed here is small and sparse. There are fast algorithms that can exploit the sparsity of the matrix. If this really becomes an bottleneck, one may resort to parallelized SVD (5).

7.5 Experiments

7.5.1 Preparation of Datasets

We obtained two real world datasets from the news channel of LinkedIn.com (News in the sequel) and the blog service Digg.com (Blog in the sequel). For the News data, we obtain a snapshot of the news channel in May, 2013, containing a total of 200,000 comments and 5,000 articles. Since labeling a comment as relevant or irrelevant requires reading and comparing the comment and the followed article, it is very time-consuming and costly to label all comments collected, therefore we randomly sample 20,000 article-comment pairs and send them to the crowdsourcing service crowdflower.com. The crowdsourcing tasks are such designed that one task consists of an article and 10 comments, randomly picked from the pool of all following comments. A worker is instructed to first read the original article and then the comments, if he/she finds a comment is irrelevant to the article, he/she should label the comment as positive, otherwise negative. The workers are required to label all the comments to get the credit. We take several measures to ensure a certain level of label quality. Firstly, we inject an editor-labeled golds in each task, and the crowdflower platform has a mechanism to prevent a worker from further labeling the tasks if his/her competence based on the golds is lower than a pre-defined threshold. Secondly, we require that each comment is labeled by 3 workers in order to derive a confidence level of the majority voting. After harvesting the labels, we discard those comments with the lowest confidence level and keep only 6952 of them. Lastly, human experts in our corporation looked into a small amount of randomly picked labeled comments to check that the crowdsourced labels are consistent with our definition of “irrelevance”. The details of

TABLE XXV

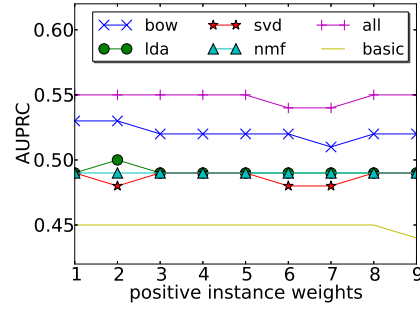
DATASET CHARACTERISTICS		
	News	Blog
# articles	363	20
# comment-article pairs	6,952	2,109
% positive instances	4.54%	28.2%

the blog dataset can be found in (58). The characteristics of these two datasets are summarized in Table XXV, from which we observe that negative instances significantly outnumber positive ones, presenting an imbalance class distribution (note that this is also true for early detection tasks, see Figure 19).

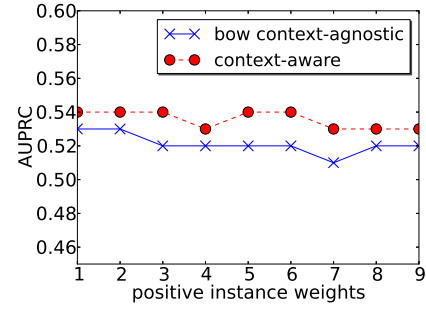
7.5.2 Experimental Settings and Results

Baselines

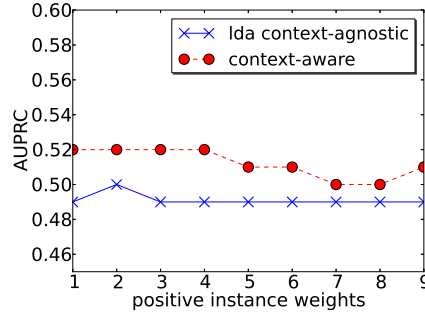
Note that the method proposed in (36) is basically σ_1 without smoothing (which requires a larger corpus retrieved from the web). σ_2 corresponds to the approach in (44) and σ_4 corresponds to that in (65). We demonstrate the effectiveness of the context-aware signals by comparing them to several enhanced baselines proposed in (44; 36; 65). Each enhanced baseline consists of two parts of features: the basic features and one of the baseline context-agnostic signals $\sigma_1, \dots, \sigma_4$. For the News dataset, a comment can be characterized by basic features based on the author’s social network connections and certain text features that are not derived from semantic relevance, such as the lengths of the comments, containment of any sensitive keywords,



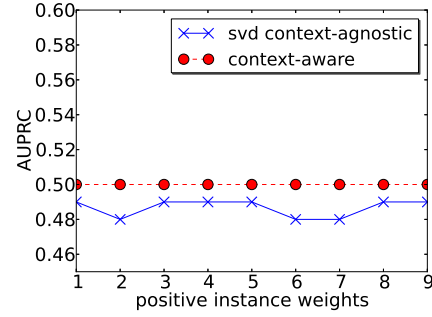
(a) Context-agnostic methods



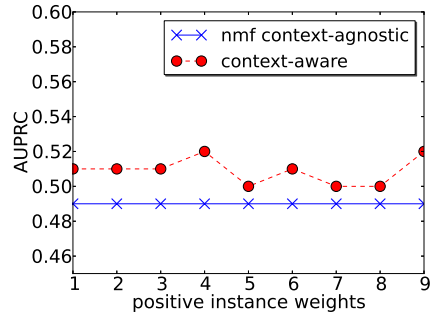
(b) BOW with native context



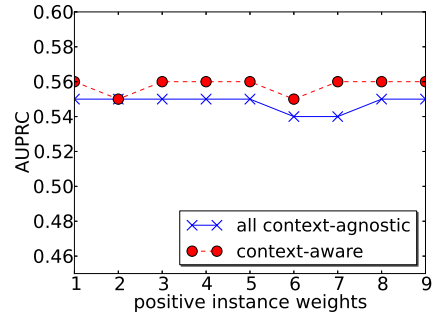
(c) LDA with native context



(d) SVD with native context

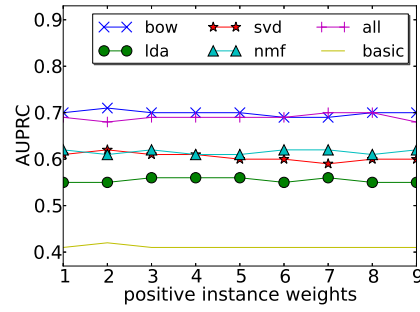


(e) NMF with native context

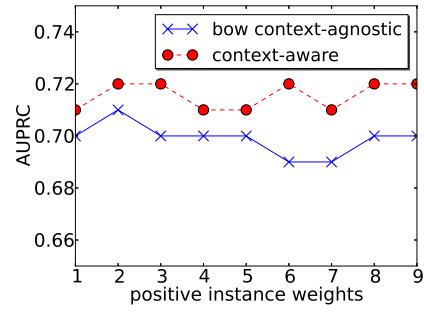


(f) All context-agnostic signals with native context

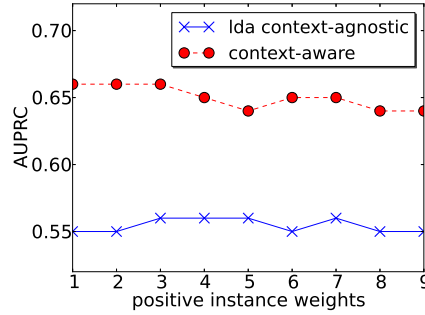
Figure 23. Effectiveness of Native Context on the News dataset



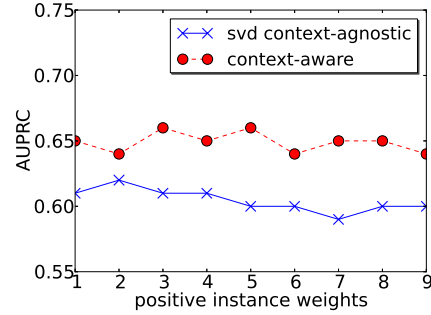
(a) Context-agnostic methods



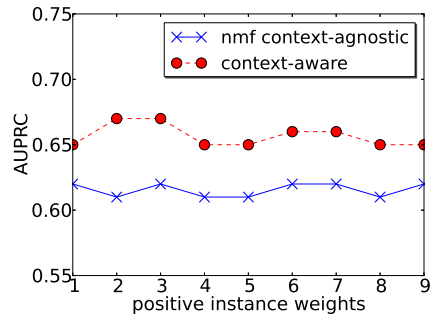
(b) BOW with native context



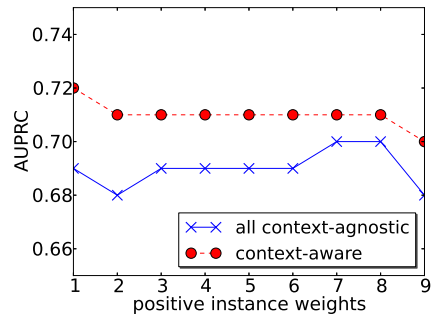
(c) LDA with native context



(d) SVD with native context



(e) NMF with native context



(f) All context-agnostic signals with native context

Figure 24. Effectiveness of Native Context on the Blog dataset

etc..¹ We also include the output of a maximum entropy text classifier as an additional basic feature. For the Blog dataset, we withhold 50% of the comment-article pairs in the Blog dataset as training data and train various classifiers (SVM, kNN, naive Bayes), whose predictions of a comment being irrelevant are treated as basic features. To derive the signals $\sigma_1, \dots, \sigma_4$: 1) we train an LDA² model using all articles, then predict the topics of all comments. 2) we construct a term-document matrix using all articles and comments, then use SVD and NMF to decompose the resulting matrix and obtain topics of articles and comments. We fix the number of topics in SVD, LDA and NMF at 50 without parameter searching.

Effectiveness of Native Contexts

Recall that the information in the constructed native contexts is given by the signal σ_5 in Table XXIV. To demonstrate that the proposed native context can enhance various context-agnostic methods, we compare the classification performance of the basic features with and without signal σ_5 . Without searching the parameter, we set the number of topics in Equation 7.7 to 20, as there are less documents in native contexts. Since there are several context-agnostic methods (BOW, LDA, SVD and NMF), we add σ_5 to each of the signals in $\{\sigma_1, \dots, \sigma_4\}$ corresponding to the above methods. For example, σ_5 can be combined with σ_1 and other basic features. We also add σ_5 to all of $\{\sigma_1, \dots, \sigma_4\}$ and other basic features. In sum, we have 5 different combinations of σ_5 with the other signals. If the combinations of features with σ_5

¹Due to corporation privacy, we are unable to discuss the details of these features

²use the implementation GibbsLDA++, with default parameters except the number of topics

outperform the same sets of features without σ_5 , then it is demonstrated that the native context does capture context-dependent semantics, which would otherwise be unavailable through context-agnostic methods.

We use the random forest implementation in `sklearn`¹ to evaluate each set of features, since random forest has been proven to be effective for imbalance two-class problems, as it is the case in this paper. Regarding the forests, we use 100 random trees, each of which grows to its full depth. The performance of random forest is evaluated using 10-fold cross validation. We choose AUPRC (Area Under Precision-Recall Curve) as our performance metric, as in real world applications like spam detection, one usually wants to achieve high precisions with low recalls. Note that one can adjust the cost of false negatives in imbalance classification problems. Therefore, with the weight of negative instances fixed at 1, we give different weights to positive instances, ranging from 1 to 9 with stepsize 1. Random decision trees can gracefully take care of the weights.

In Figure 23 (News dataset) and Figure 24 (Blog dataset), we demonstrate the performance of various signal combinations. In 23(a) and 24(a), one can observe that the signals $\sigma_i, i = 1, \dots, 4$ improve the detection performance based on the rest of the basic features. This shows that the similarity between the usage of words or topics of a comment and the proceeding article or surrounding comments can significantly improve the performance. Surprisingly, on both datasets, f_{bow} outperforms any other single dimension reduction methods (f_{lda} , f_{svd} or

¹`scikit-learn.org`

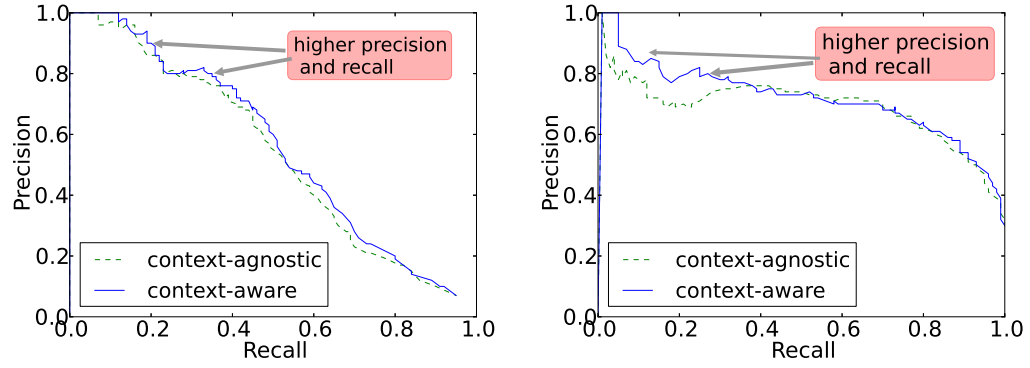
f_{nmf}) that try to capture the topics of the comments. This is because comments are usually too short to provide sufficient information for topic modeling. In 23(a), we observe that by combining all context-agnostic signals, one can obtain a significant improvement on the News dataset, though not so on the Blog dataset in 24(a). We improve the performance of context-agnostic signals consistently by including a context-aware signal σ_5 , as shown in 23(b)-23(e), and 24(b)-24(e). For example, on the News dataset, the native context maximally improves LDA and NMF by 6.1%. On the Blog dataset, the improvements are even more significant, where the native context improves LDA by 20% (24(c)). More importantly, the improvements are consistent regardless of the cost of false negatives, eliminating the time-consuming process of tuning the cost parameter in real world applications.

In 23(f) and 24(f), we show the improvements due to native contexts on the combination of all context-agnostic signals. The improvements are 1.8% on the News dataset and 4.3% on the Blog dataset. Note that using all 4 context-agnostic models gives the best performance on the News dataset (23(a)), and the proposed native context brings the AUPRC even higher. In real world applications, it is more important to locate certain points on the precision-recall curve where precisions are high. In 25(a) and 25(b), we plot the PRCs when bundling all context-agnostic models with and without σ_5 for both datasets. The areas where precisions are at least 80% are annotated using arrows. It is clear that native contexts consistently improve the performance over the combined context-agnostic models by achieving higher recalls in the critical regions.

Effectiveness of Transferred Contexts

For each irrelevant comment, we randomly sample a certain number (2, 4, and 6) of relevant comments following the same article, then we treat the irrelevant comment and the sampled relevant comments as the only available comments for the article. We run Algorithm 1 to construct transferred contexts and derive detection signal σ_6 in Table XXIV. σ_6 is then added to the combination of all context-agnostic signals $\sigma_1, \dots, \sigma_4$, since the combined signals have the best performance on this dataset (23(a)). We do not include the comment-to-center similarity for $\sigma_1, \dots, \sigma_4$, since there are only a very small number of comments at an early stage and the estimated center is inaccurate. The context-agnostic signals are generated as follows: SVD and NMF are used to decompose the term-document matrices derived from articles and the associated positive/sampled negative comments; LDA and BOW are the same as they were in the last experiment. Since there is a source of randomness due to sampling, we repeat the experiment 10 times for each parameter setting and report the mean AUPRC. We perform this experiment only on the News dataset, since there are only 20 articles in the Blog dataset, based on which the results might not be significant.

The mean of AUPRC of the methods with and without σ_6 are compared in Figure 26. Each of the figures (from left to right) is obtained using different number of sampled normal comments. In 26(a), one can see that transferred contexts only slightly change the AUPRC, when the detection task is relatively easy (smaller number of comments to distinguish). However, when there are more negative samples but insufficient contexts, the detection tasks become much more difficult. In such situations, the transferred contexts start to serve as a good source



(a) PRC for all context-agnostic signals with and without native context on the News dataset (b) PRC for all context-agnostic signals with and without native context on the Blog dataset

Figure 25. Precision-Recall Curves for the context-agnostic and context-aware detections

for detection signal σ_6 . In 26(b) and 26(c), one can see that σ_6 improves the AUPRC more than it does in 26(a). In particular, in 26(c), the improvements are most obvious.

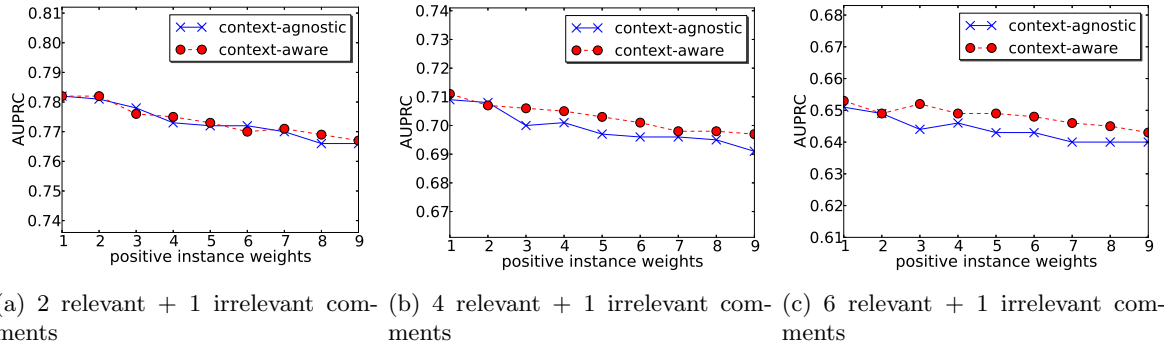



Figure 26. Effectiveness of Transferred Contexts on the News dataset

COPYRIGHTS

ACM Information for Authors

Author Rights	FAQ
<h3>ACM Author Rights</h3> <p>ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:</p> <ul style="list-style-type: none"> • Affordably priced publications • Liberal Author rights policies • Wide-spread, perpetual access to ACM publications via a leading-edge technology platform • Sustainability of the good work of ACM that benefits the profession 	
<h3>CHOOSE</h3> <p>Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.</p> <ul style="list-style-type: none"> • Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM's traditional copyright transfer agreement. • Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties. • Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works. 	
<h3>POST</h3> <p>Authors can post the accepted, peer-reviewed version prepared by the author-known as the "pre-print"-to the following sites, with a DOI pointer to the Definitive Version of Record in the ACM Digital Library.</p> <ul style="list-style-type: none"> • On Author's own Home Page <i>and</i> • On Author's Institutional Repository <i>and</i> • In any repository legally mandated by the agency funding the research on which the work is based <i>and</i> • On any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise 	

profit from serving articles.

DISTRIBUTE

Authors can post an [Author-Izer](#) link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library

- On the Author's own Home Page or
- In the Author's Institutional Repository.

REUSE

Authors can reuse any portion of their own work in a new work of *their own* (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is *not* the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

- Commercially produced course-packs that are *sold* to students require permission and possibly a fee.

CREATE

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

RETAIN

Authors retain all *perpetual rights* laid out in the [ACM Author Rights and Publishing Policy](#), including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM

[Have more questions? Check out the FAQ.](#)

[back to top](#)

About the Proceedings

PROCEEDINGS OF THE TWELFTH SIAM INTERNATIONAL CONFERENCE ON DATA MINING

Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California,
April 26 - 28, 2012.

Copyright © 2012 by the Society for Industrial and Applied Mathematics.

10 9 8 7 6 5 4 3 2 1

All rights reserved. Printed in the United States of America. No part of this book may be reproduced, stored, or transmitted in any manner without the written permission of the publisher. For information, write to the Society for Industrial and Applied Mathematics, 3600 Market Street, 6th Floor, Philadelphia, PA 19104-2688 USA.

ISSN info to come
ISBN: 978-1-61197-232-0

In order for SIAM to include your paper in the SDMI12 proceedings, the following Copyright Transfer Agreement must be completed during the paper upload process.

COPYRIGHT TRANSFER AGREEMENT

Title of Paper:

Author(s):

Copyright to this paper is hereby irrevocably assigned to SIAM for publication in *Proceedings of the Twelfth SIAM International Conference on Data Mining*, April 26-28, 2012 at the Disney's Paradise Pier Hotel, Anaheim, California. SIAM has sole use for distribution in all forms and media, such as microfilm and anthologies, except that the author(s) or, in the case of a "work made for hire," the employer will retain:

The right to use all or part of the content of the paper in future works of the author(s), including lectures, textbooks, reviews, and articles.

The right to refuse permission to third parties to republish all or part of the paper or translation thereof.

The right to reprint the paper or parts thereof to the extent the Fair Use Provisions of the Federal Copyright Act apply.

It is affirmed neither this paper nor portions of it have been published elsewhere. * For multi-author works, the signing author agrees to notify all co-authors of his/her action.

Signature:** _____

Organization: _____

Date: _____

* ☐ Check here if portions have been published elsewhere and enclose appropriate credits and permissions to republish.

** ☐ Check here if signature is on behalf of employer in the event article is "work made for hire."



RightsLink®

[Home](#)
[Create Account](#)
[Help](#)


Title: Multilabel Consensus Classification

Conference Proceedings: 2013 IEEE 13th International Conference on Data Mining

Author: Sihong Xie; Xiangnan Kong; Jing Gao; Wei Fan; Philip S. Yu

Publisher: IEEE

Date: 7-10 Dec. 2013

Copyright © 2013, IEEE

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your copyright.com credentials. Already a RightsLink user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[BACK](#)
[CLOSE WINDOW](#)



RightsLink®

[Home](#)
[Create Account](#)
[Help](#)


Title: Robust crowd bias correction via dual knowledge transfer from multiple overlapping sources

Conference Proceedings: Big Data (Big Data), 2015 IEEE International Conference on

Author: Sihong Xie; Qingbo Hu; Jingyuan Zhang; Jing Gao; Wei Fan; Philip S. Yu

Publisher: IEEE

Date: Oct. 29 2015-Nov. 1 2015

Copyright © 2015, IEEE

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your copyright.com credentials. Already a RightsLink user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[BACK](#)
[CLOSE WINDOW](#)



RightsLink®

[Home](#)
[Create Account](#)
[Help](#)


Title: A context-aware approach to detection of short irrelevant texts

Conference Proceedings: Data Science and Advanced Analytics (DSAA), 2015. 36678
2015. IEEE International Conference on

Author: Sihong Xie; Jing Wang; Mohammad S. Amin; Baoshi Yan; Anmol Bhasin; Clement Yu; Philip S. Yu

Publisher: IEEE

Date: 19-21 Oct. 2015

Copyright © 2015, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already a RightsLink user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

[BACK](#)
[CLOSE WINDOW](#)

CITED LITERATURE

1. A, M., B, L., J, W., N, G., AND N, J. Detecting group review spam. WWW '11.
2. ANDERSON, T. *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley, 2003.
3. BARTLETT, P. L. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theor.* (2006).
4. BEN-HUR, A., HORN, D., SIEGELMANN, H. T., AND VAPNIK, V. Support vector clustering. *Journal of Machine Learning Research* (2002).
5. BERRY, M., MEZHER, D., PHILIPPE, B., AND A, S. *Parallel algorithms for the singular value decomposition*. 2006.
6. BHATTARAI, A., RUS, V., AND DASGUPTA, D. Characterizing comment spam in the blogosphere through content analysis. Computational Intelligence in Cyber Security.
7. BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation.
8. BOYD, S., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, 2004.
9. C, E., AND J, W. E. bcp: An r package for performing a bayesian analysis of change point problems. *Journal of Statistical Software* (2007).
10. CHANG, Y., WANG, X., MEI, Q., AND LIU, Y. Towards twitter context summarization with user influence models. WSDM.
11. CORTES, C., AND MOHRI, M. Auc optimization vs. error rate minimization. In *NIPS* (2003).
12. DEERWESTER, S. Improving information retrieval with latent semantic indexing. In *Proceedings of the 51st ASIS Annual Meeting* (1988).

13. DEMBCZYNSKI, K., CHENG, W., AND HÜLLERMEIER, E. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML* (2010).
14. DEMBCZYŃSKI, K., WAEGEMAN, W., CHENG, W., AND HÜLLERMEIER, E. On label dependence and loss minimization in multi-label classification.
15. DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* (2006).
16. DHILLON, I. S., GUAN, Y., AND KULIS, B. Kernel k-means: spectral clustering and normalized cuts. In *SIGKDD* (2004).
17. DUCHI, J., SHALEV-SHWARTZ, S., SINGER, Y., AND CHANDRA, T. Efficient projections onto the l_1 -ball for learning in high dimensions. *ICML*.
18. E-P, L., V-A, N., N, J., B, L., AND H W, L. Detecting product review spammers using rating behaviors. *CIKM '10*.
19. ELISSEEFF, A., AND WESTON, J. A kernel method for multi-labelled classification. In *NIPS* (2001).
20. FAN, R.-E., CHANG, K.-W., HSIEH, C.-J., WANG, X.-R., AND LIN, C.-J. LIBLINEAR: A library for large linear classification. *JMLR* 9 (2008), 1871–1874.
21. FEI, W., XIN, W., AND TAO, L. Generalized cluster aggregation. In *IJCAI* (2009).
22. G, W., S, X., B, L., AND P, S. Identify online store review spammers via social review graph. In *ICDM'11*.
23. GAO, J., FAN, W., TURAGA, D., VERSCHEURE, O., MENG, X., SU, L., AND HAN, J. Consensus extraction from heterogeneous detectors to improve performance over network traffic anomaly detection. In *INFOCOM* (2011).
24. GAO, J., LIANG, F., FAN, W., SUN, Y., AND HAN, J. Graph-based consensus maximization among multiple supervised and unsupervised models. *NIPS '09*, pp. 585–593.
25. GOLUB, G., AND LOAN, C. *Matrix Computations*, 3rd ed. John Hopkins University Press, 1996.

26. HANLEY, J. A., AND MCNEIL, B. J. The meaning and use of the area under a receiver operating characteristic (roc) curve.
27. HU, X., SUN, N., ZHANG, C., AND CHUA, T.-S. Exploiting internal and external semantics for the clustering of short texts using world knowledge. CIKM.
28. KANT, R., SENGAMEDU, S. H., AND KUMAR, K. Comment spam detection by sequence mining. WSDM.
29. KANTCHELIAN, A., MA, J., HUANG, L., AFROZ, S., JOSEPH, A., AND TYGAR, J. D. Robust detection of comment spam using entropy rate. AISec.
30. KARLIN, S., AND TAYLOR, H. M. *A First Course in Stochastic Processes*, 2 ed. Academic Press, 1975.
31. KENDALL, M. G. A new measure of rank correlation. *Biometrika* (1938).
32. LAUW, H. W., LIM, E.-P., AND WANG, K. Bias and controversy: Beyond the statistical deviation. KDD.
33. LAUW, H. W., LIM, E.-P., AND WANG, K. Summarizing review scores of unequal reviewers. SDM.
34. LI, T., AND DING, C. Weighted consensus clustering. SDM.
35. LONG, G., CHEN, L., ZHU, X., AND ZHANG, C. Tcsst: transfer classification of short & sparse text using external data. CIKM.
36. MISHNE, G., CARMEL, D., AND LEMPEL, R. Blocking blog spam with language model disagreement. In *AIRWeb* (2005).
37. MISHNE, G., AND GLANCE, N. Leave a reply: An analysis of weblog comments. WWW.
38. MISHRA, A., AND RASTOGI, R. Semi-supervised correction of biased comment ratings. WWW.
39. N, J., AND B, L. Opinion spam and analysis. WSDM '08.
40. N, J., B, L., AND E-P, L. Finding unusual review patterns using unexpected rules. CIKM '10.

41. NG, A. Y., JORDAN, M. I., AND WEISS, Y. On spectral clustering: Analysis and an algorithm. In *NIPS* (2001).
42. PAUGAM-MOISY, H., ELISSEEFF, A., AND GUERMEUR, Y. Generalization performance of multiclass discriminant models. In *Neural Networks, 2000. IJCNN 2000* (2000).
43. PETTERSON, J., AND CAETANO, T. Reverse multi-label learning. In *NIPS* (2010).
44. PHAN, X.-H., NGUYEN, L.-M., AND HORIGUCHI, S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. WWW.
45. RAYKAR, V. C., YU, S., ZHAO, L. H., JEREBKO, A., FLORIN, C., VALADEZ, G. H., BOGONI, L., AND MOY, L. Supervised learning from multiple experts: whom to trust when everyone lies a bit. ICML, ACM.
46. READ, J., PFAHRINGER, B., HOLMES, G., AND FRANK, E. Classifier chains for multi-label classification. ECML PKDD.
47. SAHAMI, M., AND D. HEILMAN, T. A web-based kernel function for measuring the similarity of short text snippets. WWW.
48. SCHAPIRE, R. E., AND SINGER, Y. Boostexter: A boosting-based system for text categorization.
49. SHI, C., KONG, X., YU, P. S., AND WANG, B. Multi-label ensemble learning. ECML/PKDD.
50. STREHL, A., AND GHOSH, J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* (2003).
51. TAN, C., CHI, E. H., HUFFAKER, D., KOSSINETS, G., AND SMOLA, A. J. Instant foodie: Predicting expert ratings from grassroots. CIKM.
52. TOKDAR, S. T., AND KASS, R. E. Importance sampling: a review. *Wiley Interdisciplinary Reviews Computational Statistics* 2, 1 (2010), 54–60.
53. TSOUMAKAS, G., KATAKIS, I., AND VLAHAVAS, I. P. Random k-labelsets for multilabel classification. *IEEE Trans. Knowl. Data Eng.* (2011).

54. V, M., H, M., G, D., AND K, E. Indexing multi-dimensional time-series with support for multiple distance measures. KDD '03.
55. VAPNIK, V. *Statistical learning theory*. Wiley, 1998.
56. VON AHN, L., MAURER, B., McMILLEN, C., ABRAHAM, D., AND BLUM, M. reCAPTCHA: Human-based character recognition via web security measures.
57. WANG, H., SHAN, H., AND BANERJEE, A. Bayesian cluster ensembles. In *SDM* (2009).
58. WANG, J., YU, C. T., YU, P. S., LIU, B., AND MENG, W. Diversionary comments under political blog posts. CIKM.
59. XIE, S., FAN, W., AND YU, P. S. An iterative and re-weighting framework for rejection and uncertainty resolution in crowdsourcing. In *SDM* (2012).
60. XIE, S., GAO, J., FAN, W., TURAGA, D., AND YU, P. S. Class-distribution regularized consensus maximization for alleviating overfitting in model combination. In *SIGKDD* (2014).
61. XIE, S., HU, Q., ZHANG, J., GAO, J., FAN, W., AND YU, P. S. Robust crowd bias correction via dual knowledge transfer from multiple overlapping sources. In *Big Data (Big Data), 2015 IEEE International Conference on* (2015).
62. XIE, S., KONG, X., GAO, J., FAN, W., AND YU, P. Multilabel consensus classification. In *ICDM* (2013).
63. XIE, S., WANG, G., LIN, S., AND YU, P. S. Review spam detection via temporal pattern discovery. KDD, ACM, pp. 823–831.
64. XIE, S., WANG, J., AMIN, M. S., YAN, B., BHASIN, A., YU, C., AND YU, P. S. A context-aware approach to detection of short irrelevant texts. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on* (2015), pp. 1–10.
65. XU, W., LIU, X., AND GONG, Y. Document clustering based on non-negative matrix factorization. SIGIR.
66. YAN, R., TESIC, J., AND SMITH, J. R. Model-shared subspace boosting for multi-label classification. KDD.

- 67. YAN, X., GUO, J., LAN, Y., AND CHENG, X. A biterm topic model for short texts. WWW.
- 68. YAN, X., GUO, J., LIU, S., CHENG, X., AND WANG, Y. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. SDM.
- 69. YAN, Y., ROSALES, R., FUNG, G., AND DY, J. Modeling multiple annotator expertise in the semi-supervised learning scenario.
- 70. YAN, Y., ROSALES, R., FUNG, G., SCHMIDT, M. W., VALADEZ, G. H., BOGONI, L., MOY, L., AND DY, J. G. Modeling annotator expertise: Learning when everybody knows a bit of something.
- 71. YI, J., YANG, T., JIN, R., JAIN, A., AND MAHDAVI, M. Robust ensemble clustering by matrix completion. ICDM.
- 72. YIH, W.-T., AND MEEK, C. Improving similarity measures for short segments of text. AAAI.
- 73. YU, G., DOMENICONI, C., RANGWALA, H., ZHANG, G., AND YU, Z. Transductive multi-label ensemble classification for protein function prediction. KDD.
- 74. ZHANG, M.-L., AND ZHANG, K. Multi-label learning by exploiting label dependency. KDD.
- 75. ZHANG, X., YUAN, Q., ZHAO, S., FAN, W., ZHENG, W., AND WANG, Z. Multi-label Classification without the Multi-label cost. In *SDM* (2010).

VITA

Sihong Xie

Education

Ph.D., Computer Science, University of Illinois at Chicago, Chicago, Illinois	2016
M.E., Software Engineering, Sun Yat-Sen University	2010
B.E., Software Engineering, Sun Yat-Sen University	2008

Honors

SIAM SDM 2012 Student Travel Award	05/2012
ACM SIGKDD 2012 Student Travel Award	08/2012
IEEE ICDM 2008 Data Mining Contest Crown Award (highest award)	2008
Outstanding Graduate Student Scholarship, Sun Yat-Sen University	2008 - 2009
Outstanding Student Scholarship First Prize, Sun Yat-Sen University	2006 - 2007

Working Experience

Research Intern, Exploratory Stream Analytics Group, IBM Research , Hawthorne, NY.	
05/2012 - 08/2012	
Applied Researcher, Search and Network Analysis Group, LinkedIn Corporation , Mountain View, CA.	05/2013 - 08/2013
Research Assistant, Department of Computer Science, University of Illinois at Chicago , Chicago, IL.	08/2012 - 07/2016

Selected Publications

- Sihong Xie, Wei Fan, Jing Peng, Olivier Verscheure, and Jiangtao Ren. Latent space domain transfer between high dimensional overlapping distributions. In: WWW. ACM, 2009.
- Sihong Xie, Wei Fan, Olivier Verscheure, and Jiangtao Ren. Efficient and numerically stable sparse learning. In: ECML/PKDD. 2010.
- Sihong Xie, Guan Wang, Shuyang Lin, and Philip S.Yu. Review spam detection via temporal pattern discovery. In: KDD. ACM, 2012.
- Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, and Philip S Yu. Multilabel consensus classification. In: ICDM. IEEE, 2013.
- Sihong Xie, Wei Fan, and Philip S.Yu. An iterative and re-weighting framework for rejection and uncertainty resolution in crowdsourcing. In: SDM. SIAM, 2012.
- Sihong Xie, Jing Gao, Deepak Turaga, Wei Fan, and Philip S. Yu. Class- distribution regularized consensus maximization for alleviating overfitting in model combination. In: KDD. ACM, 2014.
- Sihong Xie, Qingbo Hu, Weixiang Shao, Jingyuan Zhang, Jing Gao, Wei Fan, and Philip S. Yu. Effective Crowd Expertise Modeling via Cross Domain Sparsity and Uncertainty Reduction. In: SDM. SIAM, 2016.