12-2013

A simple integration of social relationship and text data for identifying potential customers in microblogging

Guansong PANG
Singapore Management University, gspang@smu.edu.sg

Shengyi JIANG

Dongyi CHEN

A Simple Integration of Social Relationship and Text Data for Identifying Potential Customers in Microblogging

Guansong Pang[1,*], Shengyi Jiang [2], and Dongyi Chen [2]

[1] School of Management, Guangdong University of Foreign Studies, Guangzhou 510006, China
pangguansong@163.com
[2] School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510006, China
{jiangshengyi,dongyi_chen}@163.com

Abstract. Identifying potential customers among a huge number of users in microblogging is a fundamental problem for microblog marketing. One challenge in potential customer detection in microblogging is how to generate an accurate characteristic description for users, i.e., user profile generation. Intuitively, the preference of a user's friends (i.e., the person followed by the user in microblogging) is of great importance to capture the characteristic of the user. Also, a user's self-defined tags are often concise and accurate carriers for the user's interests. In this paper, for identifying potential customers in microblogging, we propose a method to generate user profiles via a simple integration of social relationship and text data. In particular, our proposed method constructs self-defined tag based user profiles by aggregating tags of the users and their friends. We further identify potential customers among users by using text classification techniques. Although this framework is simple, easy to implement and manipulate, it can obtain desirable potential customer detection accuracy. This is illustrated by extensive experiments on datasets derived from Sina Weibo, the most popular microblogging in China.

Keywords: identifying potential customers, user profiling, social relationship, text data, text classification, microblog marketing.

1 Introduction

Microblog is characterized by a huge number of users, fast message propagation and a broad range of influence, so that microblog marketing has been made as one of the most important portions of social media marketing in many companies. Nowadays, microblog marketing activities are mostly random. As a result, they are blocked and even denounced by most users. This does not produce any positive effects on companies but mostly severe negative impacts. Identifying potential customers is a fundamental problem for microblog marketing, which can enable marketers to conduct cost-effective marketing activities.

* Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part I, LNAI 8346, pp. 397–409, 2013.

One challenge in potential customer detection in microblogging is how to generate an accurate characteristic description for users, i.e., user profile generation. Some methods have been proposed to deal with this issue recently. The most relevant work is [1], where the authors focused on utilizing user's default profile features, posting behavior features, linguistic content features and social network features to construct user profiles. The gradient boosted decision trees were then applied to perform user classification tasks, including prediction of potential followers (i.e., potential customers) of Starbucks. Since this method involves a wide range of analysis over various features and is a task-specific framework, it is not easy to manipulate in real-world applications (e.g., extensive parameter tuning needed for a new task). Moreover, the best F_1-measure value in the prediction of Starbucks's potential followers was about 76%, which could be improved. User profiles can also be generated from conversational data in social media [2]. Other related work focused on some other user classification tasks, e.g., demographic attribute value prediction and communication role identification, by making use of a variety of features mentioned above [3-7]. We focus on potential customer detection by employing microblog users' self-defined tag features only.

Intuitively, the preference of a user's friends (i.e., the person followed by the user in microblogging) is of great importance to capture the characteristic of the user. Also, a user's self-defined tags are often concise and accurate carriers for the user's interests. In this paper, for identifying potential customers in microblogging, we propose a method to generate user profiles via a simple integration of social relationship and text data. In particular, our proposed method constructs self-defined tag based user profiles by aggregating tags of the users and their friends. We further identify potential customers among users by using text classification techniques such as K Nearest Neighbors (KNN), Naive Bayes (NB), Rocchio, centroid-based classification (Centroid) and Support Vector Machines (SVM). The effectiveness of our proposed framework has been illustrated by experiments on datasets derived from Sina Weibo, the most popular microblogging in China.

Our contribution is to propose a simple yet effective framework for identifying potential customers in microblogging. Although this framework is simple, easy to implement and manipulate, it can obtain desirable potential customer detection accuracy. With the aid of the fruitful social relationship in microblogging, our proposed framework can not only provide potential customer detection with a new perspective and method, but also present important reference to many classic Customer Relationship Management (CRM) problems such as customer churn.

2 Related Work

Identifying potential customers in microblogging, which is essentially a target-specific binary user classification task, is an emerging research field. Previous work has explored user profile generation methods and their application on user classification in microblogging. In [1], the authors constructed user profiles based on users' default profile features (e.g., user name, location, number of followers and

friends), posting behavior features (e.g., total number of microblogs posted, average number of hashtags and URLs per microblog), linguistic content features (e.g., prototypical words and hashtags, sentiment words, generic and domain-specific topics generated via Latent Dirichlet Allocation [8]) and social network features (e.g., number and fraction of intersected friends between users). The user profiles were then used for user classification tasks, including a potential customer detection task. This method defined user profiles from a range of dimensions and required somewhat complicated parameter tuning for different classification tasks. Other related work is [2], which aimed at extracting terms and concepts that can reflect users' interests from social media conversional data, in order to target advertisements or conduct other commercial activities. Instead of generating user profiles from a variety of features or complex conventional data, our proposed framework focuses on making use of users and their friends' self-defined tag features only, which is easy to implement and manipulate. There have been many methods proposed for demographic attribute value prediction and communication role identification [3–7]. These methods can be further improved to use for commercial utilities, e.g., identifying potential customers, while our proposed method is especially designed for potential customer detection.

A range of previous work has aimed at identifying potential customers based on a customer database [9–12]. These work normally built potential customer detection models based on customers' demographic information. Compared to these work, we investigate potential customer detection methods in a different application context, i.e., microblogging. Moreover, we capture customers' interests from social and personal perspectives rather than personal demographic attributes. Much work has been made to develop text classification algorithms [13, 14]. In general, these work usually focused on devising more effective or efficient algorithms, while our framework is to use text classification algorithms to make better use of microblog user data for identifying a company's potential customers.

3 Identifying Potential Customers in Microblogging

A company's potential customers are conventionally defined as anyone who has a particular need for some products of the company. In this work, we define the potential customers of a company in microblogging as users who share similar interests to followers of the company's authorized microblog account, since these followers tend to be the company's customers or potential customers. Alternatively, identifying potential customers of a company in microblogging is equivalent to potential follower prediction of the company's microblog account.

3.1 Our Proposed Framework

This work aims to make use of the self-defined tags of a particular company's followers and their friends to generate tag-based user profiles, such that each user can be denoted by a tag vector, where each tag represents a dimension. Identifying potential customers among a large number of users is then transformed into a text

classification problem. Our proposed framework is shown in Fig. 1. Given a company, we first collect tags of the company's followers and their friends to construct user profiles, and then build a model based on the user profiles using text classification algorithms and classify a given user into either the potential customer class or the general user class.

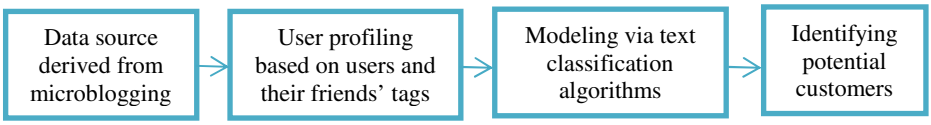

Fig. 1. Our proposed framework for identifying potential customers in microblogging

Compared to traditional methods with customer databases, our framework possesses three strengths: (i) there are abundant publicly accessible customer data in microblogging; (ii) the users' interests are denoted by self-defined tags of the users' and their friends, which are likely to reflect the users' characters from various perspectives; (iii) marketers can directly design and conduct marketing activities in microblogging other than emails or mass media.

3.2 User Profile Generation

After registering an account in microblogging, e.g., Sina Weibo, user can write multiple tags to describe their personal interests. These tags, such as "Post-80s", "data mining", "football", "housewife", "foodie" and "faculty", contain critical information for a user's personality, common interests or occupation. This enables visitors to understand the user better and further help the user socialize. Since these tags are normally self-defined, they are often concise and accurate carriers for users' interests. Thus, we focus on using tags to construct user profiles.

In general, a user's personal interests or purchase behaviors tend to be affected by the user's friends, or someone who has closed relationship with; in turn, these fellow's interests also reflect the user's interests to some extent. Thus, in microblogging, the person who has a bi-direction relation with or is followed by a user (i.e., the user's friends in microblogging) is very likely to share some similar interests to the user. We therefore devise a method to construct a user's profile by a sum of tag vectors of the user and his or her friends, as detailed in Eq.(1).

$$\mathbf{up}_i = \sum_{j=1}^{s} \mathbf{friend}_j + \mathbf{user}_i \tag{1}$$

where $\mathbf{user}_i$ is a user tag vector, $\mathbf{friend}_j$ denotes the tag vector of a friend of the user $\mathbf{user}_i$ among s friends, $\mathbf{up}_i$ is the user profile vector (where each tag represents a dimension and its dimension value is the frequency occurred in the $\mathbf{user}_i$ and all the $\mathbf{friend}_j$). Each user profile vector is regarded as a document vector and weighted by a general $TF \times IDF$ method [15] (*TF* and *IDF* are short for Term Frequency and Inverse Document Frequency respectively). The weighted user profile vectors are

then used for modeling and detection. Our experiments showed that the user profile will become less effective, when we do not use the tags of the user's friends, or when we discriminate the tags of the user and the user's friends. This is why we sum tag vectors of the user and his friends with equal weights.

3.3 Modeling and Detection

We investigate various text classification algorithms for potential customer modeling and detection, including KNN, NB, Centroid, Rocchio and SVM. We briefly describe the modeling and classification processes of these five classifiers below (More detailed descriptions refer to [13, 14]).

KNN: The process of KNN text classification can be described as follows: given a test document, find the nearest neighbors for the test document among the training document set, and score class candidates for the test document based on the classes of the neighbors. KNN then assigns the class with the highest score to the test document.

NB: NB uses the Bayes Rule to determine the class of a given test document. Its independent assumption assumes term features to contribute independently for modeling and classification. Maximum likelihood estimation is usually used to compute the prior and conditional probabilities.

Centroid: In Centroid, each class is represented by its class centroid, which is the average of document vectors within the class, and a test document is assigned to the class label of its closest centroid.

Rocchio: Rocchio functions similarly to Centroid except that it uses prototype vectors to classify documents. The prototype vectors are generalized from class centroids, calculated as Eq. (2).

$$\mathbf{pv}_{\mathbf{k}} = \alpha \frac{1}{|C_k|} \sum_{\mathbf{d_m} \in C_k} \mathbf{d}_{\mathbf{m}} - \beta \frac{1}{|D - C_k|} \sum_{\mathbf{d_n} \in D - C_k} \mathbf{d}_{\mathbf{n}} \tag{2}$$

where $\mathbf{pv}_{\mathbf{k}}$ denotes the prototype vector of the class C_k, D is the entire training document set. Parameters α and β adjust the relative importance of positive and negative documents. Rocchio is equivalent to Centroid with $\alpha = 1$ and $\beta = 0$.

SVM: The main principle of SVM is to determine support vectors that maximize margins of separation between classes in a hyperplane. SVM performs well on linearly separable space, and it can use kernel methods such as polynomial and *RBF* kernels, to adapt to non-linearly separable data space. As shown in [13, 16], compared to non-linear kernels, the linear kernel can enable SVM to achieve better or very comparable text classification accuracy and perform more efficiently.

A critical step at this stage is to determine the training sets of positive and negative instances. In general, given a potential customer detection task for a specific company, the followers of the company's microblog account can be regarded as positive instances, while other users can be regarded as negative instances. However, this general rule is not without problems. This is because (i) those who do not follow the company may be the followers of the microblog accounts of this company's rivals, and those users are more appropriate to be used as positive instances rather

than negative instances, because followers of companies within the same sector tend to have similar characters and interests; (ii) there are a huge number of users who do not follow the microblog accounts of the company and its rivals, so it is a problem about how to determine the compositions of negative instances. For example, negative instances can be composed of followers of a single company from a distinct sector or multiple companies from various sectors. After determining the positive and negative instances for the given task, we then can use a text classification algorithm to construct a potential customer model and identify potential customers.

4 Experimental Results

4.1 Datasets

Our experimental datasets derived from Sina Weibo, which is the most popular microblog platform in China. A large number of companies have registered in this microblogging, amounting to 130 thousand. These companies come from different sectors such as healthcare, education, tourist, personal services, foods and e-commerce. Our datasets consist of 8 small or medium sized companies distributed through 4 sectors, i.e., healthcare, education, tourist and personal services, as shown in Table 1. We define an ID for each company, and the columns "No. of Followers" and "No. of Friends" are the number of followers and the total number of the followers' non-overlapping friends respectively.

Table 1. A summary of the eight companies used for our experiments

Sectors	ID	No. of Followers	No. of Friends
Tourist	A1	3,397	703,607
	A2	4,308	1,006,333
Healthcare	B1	2,141	569,505
	B2	4,022	894,985
Education	C1	3,976	588,906
	C2	4,999	686,545
Personal services	D1	4,130	734,886
	D2	1,170	389,202

We focus on small or medium sized companies for two main reasons. One is that we can only gather at most 5,000 followers' information of one particular microblog account due to limitations of microblogging service providers. The number of followers of small or medium sized companies is often less than 5,000, so that we can collect all the followers of these companies. Therefore, hopefully, this kind of data can well cover the characters of the companies' customers. Another reason is that, compared to large-sized companies, it is more demanding for smaller sized companies to identify potential customers.

It is not uncommon that numerous microblogging users have been registered automatically by machines, also known as "zombie users". This kind of user is

actually noise instances, since they can be acted as "zombie followers" of many companies. In general, normal users are likely to form bi-direction relations with their friends. While "zombie users" build a huge number of one direction relations, i.e., randomly follow numerous users, yet they do not have any followers or very few followers (e.g., less than 10). In addition, normal users are likely to build a personalized user domain using their names like "http://weibo.com/" plus "user's name", while "zombie users" do not set this domain. Based on these observations, we removed the "zombie users" by filtering out the users who have less than 10 bi-direction relations and do not have a personalized user domain. The remaining data is detailed in Table 2.

Table 2. Basic information about the eight companies after removing the "zombie users"

ID	No. of Followers	No. of Friends	ID	No. of Followers	No. of Friends
A1	2080	386845	C1	2716	381613
A2	1232	337241	C2	2287	328240
B1	1726	321319	D1	1213	290805
B2	1297	295139	D2	609	213267

4.2 Parameter Settings

The parameters of the five text classifiers used, i.e., KNN, NB, Rocchio, Centroid and SVM, are set as follows:

KNN: For KNN, the parameter tuning issue is how to decide an appropriate K value [17]. Various K values has been adapted into different application domains and scenarios [13], we set $K = 50$ in our experiments after a pilot study.

NB: NB is a non-parametric classifier. We used the Laplace smoothing method to deal with the "zero probability" issue.

Rocchio: Following [18], we set $\alpha = 2\beta = 1$ after a pilot study, which enables the Rocchio classifier to obtain stable classification performance.

Centroid: Centroid is a non-parametric classifier, and it therefore does not require any parameter tuning.

SVM: We applied a widely used SVM implementation package, called LibSVM [19]. Linear kernel is used in our experiments with other settings default [13, 20].

4.3 Performance Evaluation

The datasets are split into training and test sets with the ratio 2:1. A widely used measure, F_1 measure, is used as the classification accuracy metric. F_1 is a combination measure of precision and recall, i.e., $F_1 = \frac{2 \times precision \times recall}{precision + recall}$, where

$precision = \frac{TP}{TP + FN}$ and $recall = \frac{TP}{TP + FP}$. TP, FN and FP are short for True Positive, False Negative and False Positive respectively. In our experiments, the average F_1 value of two classes has a similar trend to the F_1 value of an individual class. Since only the effectiveness of identifying positive instances, i.e., potential customers, is of our interest, we only report the F_1 value on the positive class.

4.4 Results

We examined our proposed framework on various contexts. In particular, since the composition of positive and negative instance sets is very flexible, but is a critical step for potential customer detection, we conducted extensive experiments with varying compositions of negative or positive instances.

Negative Instances from Multiple Sectors. Microblogging consists of users with a wide range of background and character, so the variety of negative instances is important for examining the model's generalization. In this subsection, for each sector we chose one company (i.e., A1, B1, C1 and D1) as the target company, so positive instances were followers of the selected company, while negative instances were randomly selected from companies of other three sectors evenly. We employed the five classifiers to conduct the task, and each classifier runs five times for each company. Table 3 shows the average F_1 values of the classifiers and the baseline. The baseline is calculated by randomly identifying potential customers without using any classifiers. Since the datasets are made up of positive and negative instances evenly, all the baselines are 0.5.

Table 3. F1 values of five classifiers on four companies from different sectors

	KNN	*NB*	*Rocchio*	*Centroid*	*SVM*	*Baseline*
A1	0.7637	0.8749	0.7883	0.8310	0.9264	0.5000
B1	0.8183	0.9101	0.8924	0.8974	0.9445	0.5000
C1	0.6206	0.9472	0.8477	0.9008	0.9758	0.5000
D1	0.8581	0.8694	0.8535	0.7859	0.9272	0.5000
Avg.	0.7652	0.9004	0.8455	0.8538	0.9435	0.5000

The results show that all the classifiers outperform the baseline. SVM consistently outperforms other four classifiers with a very high averaging F_1 value, 0.9435, followed by NB, Centroid and Rocchio. The worst performance among these classifiers goes to KNN due to the fact that the datasets used have a number of noisy tags and KNN is sensitive to these noisy tag features [21].

Negative Instances from Single Sector. Apart from choosing negative instances from multiple sectors, negative instances can also derive from only one sector. We first used

Centroid to conduct classification on every selected company, with one of the remaining companies as negative class. This could help us decide which company would be optimal for being the negative class for the target company. The results are shown in Table 4. The best and the worst performance are bold and underline respectively. It is clear that Centroid obtains the worst performance when the positive and negative classes come from the same sector. This is because instances of the classes from the same sector are likely to distribute closely in the feature space, as they bear a number of common features. Centroid tends to achieve promising performance when the sectors of the two classes differ significantly, such as healthcare (i.e., B1 or B2) vs. education (i.e., C1 or C2). This result is consistent with the general classification intuition that instances of diversified classes are distributed separately and easier to be classified. Therefore, the company, which can enable Centroid to obtain the best classification performance, was chosen to be the negative class for a specific target company (e.g., A1-vs-B2 and A2-vs-D2). The other four classifiers were then applied to perform classification on each dataset. The results are illustrated in Table 5.

Table 4. Centroid's classification results on eight companies with one of the remaining companies as negative class

	A1	*A2*	*B1*	*B2*	*C1*	*C2*	*D1*	*D2*	*Avg.*
A1	-	0.6814	0.8931	**0.9296**	0.8635	0.8453	0.9004	0.9237	0.8624
A2	0.6390	-	0.8662	0.8795	0.8182	0.8414	0.8456	**0.9063**	0.8642
B1	0.8720	0.8919	-	0.6017	**0.9600**	0.9323	0.9015	0.9209	0.8296
B2	0.8801	0.8553	0.6357	-	0.8951	0.8568	0.8995	**0.9216**	0.9140
C1	0.8899	0.8944	**0.9746**	0.9560	-	0.6729	0.9318	0.9533	0.8682
C2	0.8748	0.9026	**0.9505**	0.9325	0.7340	-	0.9173	0.9489	0.8485
D1	0.7738	0.7766	0.8196	**0.8700**	0.8076	0.7994	-	0.7373	0.7715
D2	0.7263	0.7861	0.7857	**0.8393**	0.7990	0.7969	0.6565	-	0.8624

Table 5. F1 values of five classifiers on eight target companies

	KNN	*NB*	*Rocchio*	*Centroid*	*SVM*
A1	0.8464	0.9523	0.8899	0.9296	0.9660
A2	0.8550	0.9050	0.8997	0.9063	0.9166
B1	0.8449	0.9603	0.9643	0.9600	0.9826
B2	0.8678	0.9481	0.9318	0.9216	0.9660
C1	0.8888	0.9754	0.9774	0.9746	0.9890
C2	0.7874	0.9498	0.9601	0.9505	0.9727
D1	0.7725	0.9605	0.9174	0.8700	0.9789
D2	0.5338	0.8962	0.8626	0.8393	0.9231
Avg.	0.7996	0.9435	0.9254	0.9190	0.9619

It can be seen from Table 5 that the average F_1 values of all the five classifiers outperform their counterparts in Table 3. It also shows that SVM achieves very promising F_1 values over all the datasets, outperforming the rest of classifiers. NB,

with 0.9453 in the average F_1 value, comes in second. It is worthwhile to note that all the five classifiers' F_1 values decrease substantially on the dataset with D2 as positive class. This is mainly due to the fact that the classifiers lack sufficient instances to make predictions on this dataset.

Increasing Positive Instances with Followers of the Rival Company. One obvious advantage for identifying potential customers in microblogging is that it is easy to augment positive instances. This can be done by collecting information of followers of the rival's microblog account. Augmenting positive instances is very likely to improve classification accuracy. We examined this strategy in this subsection. The classification task is the same as the one in the first subsection with the only exception that we increased the positive instances with the instances of another company within the same sector in Table 2. The F_1 values of the five classifiers are shown in Table 6. The "origin" column directly derives from Table 3, and the "augment" column denotes the classifiers' results on each dataset with augmented positive instances. On average, NB, Rocchio, Centroid and SVM with augmented positive instances slightly outperform that with original positive instances. Specifically, these four classifiers with the augmented strategy obtain improvements on the first two datasets. KNN with the augmented positive instances only performs well in the first dataset. It indicates that the augmented positive instances do not always help improve the classifiers' performance. The main reason for this result, we conjecture, is that the instances within A1 and A2, B1 and B2 are closed to each other, while the instances within C1 and C2, D1 and D2 tend to be diversified. This can be observed from Table 4, where the averaging F_1 values of C1-vs-C2 and D1-vs-D2 (0.7035 and 0.6969 respectively) are much larger than that of B1-vs-B2 and A1-vs-A2 (0.6187 and 0.6602 respectively), indicating that the class pairs A1, A2 and B1, B2 are more likely to share an underling latent feature space than the pairs C1, C2 and D1, D2.

Table 6. Classification results with augmented positive instances

	KNN		*NB*		*Rocchio*		*Centroid*		*SVM*	
	Origin	Augment	Origin	Augment	Origin	Augment	Origin	Augment	Origin	Augment
A1	0.7637	**0.7780**	0.8749	**0.8906**	0.7883	**0.7885**	0.8310	**0.8405**	0.9264	**0.9373**
B1	0.8183	0.7956	0.9101	**0.9281**	0.8924	**0.9211**	0.8974	**0.9191**	0.9445	**0.9489**
C1	0.6206	0.5997	0.9472	0.9415	0.8477	0.8297	0.9008	**0.9125**	0.9758	0.9733
D1	0.8581	0.8269	0.8694	0.8584	0.8535	0.8436	0.7859	0.7489	0.9272	0.9254
Avg.	0.7652	0.7501	0.9004	**0.9047**	0.8455	**0.8457**	0.8538	**0.8553**	0.9435	**0.9462**

Computation Time Comparisons. Table 7 shows the averaging running time of 5 runs of the five classifiers on the datasets. All the results were obtained from one computer with configuration: Intel (R) Core (TM) 2 Duo CPU 3.00GHz 1.85GB. NB achieves the best classification efficiency, followed by Centroid, Rocchio, SVM and KNN. NB runs about 28-fold and 37-fold faster than SVM and KNN. SVM runs faster than KNN, and Centroid runs slightly faster than Rocchio.

Table 7. Running time (in seconds) of the five classifiers on the four datasets

	KNN	*NB*	*Rocchio*	*Centroid*	SVM
A1	1091	19	75	70	685
B1	605	10	63	59	532
C1	1623	62	109	103	1105
D1	335	8	37	35	422
Avg.	914	25	71	67	686

5 Conclusions

We proposed a framework of identifying potential customers in microblogging. Specifically, we first devised a user profile generation method via a simple integration of users' self-defined tags based on their social relations. This generated a tag vector to represent a user. We further proposed a potential customer detection method by using text classification algorithms. The promising performance of our proposed framework has been illustrated by a series of experiments on datasets with varying compositions of positive or negative instances. In particular, negative instances derived from a single diversified sector enable classifiers to perform more accurately than negative instances from multiple sectors. However, it should be noted that, in the former case, models are likely to overfitting. Augmented positive instances from a company's rivals can improve the detection accuracy, when customers of the company and its rivals share a large number of features, i.e., an underlying latent feature space.

Our proposed framework is simple and easy to implement and manipulate, which hopefully can be mastered by marketers quickly. Our results also showed SVM is the best choose in terms of accuracy, while NB is preferable to the other four classifiers when taking both effectiveness and efficiency into consideration.

We continue to further investigate a combination of dimension reduction methods with our proposed framework. We are also interested in improving our proposed framework to deal with other CRM issues.

Acknowledgements. We thank the anonymous reviewers for their fruitful suggestions. This paper was completed when Guansong Pang was a visiting student in the Web Sciences Center at University of Electronic Science and Technology of China. He wishes to thank his supervisor Prof. Mingsheng Shang in the Web Sciences Center for his support on this work. This work was supported in part by the Natural Science Foundation of China under Grant No. 61070061 and No. 61202271, and by the Social Science Foundation of China under Grant No. 13CGL130, and by the National Key Technologies R&D Program Project under Grant No. 2012BAH02F03.

References

[1] Pennacchiotti, M., Popescu, A.: Democrats, republicans and starbucks afficionados: user classification in twitter. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 430–438 (2011)
[2] Konopnicki, D., Shmueli-Scheuer, M., Cohen, D., Sznajder, B., Herzig, J., Raviv, A., Zwerling, N., Roitman, H., Mass, Y.: A statistical approach to mining customers' conversational data from social media. IBM Journal of Research and Development 57(3/4), 11–14 (2013)
[3] Burger, J.D., Henderson, J., Kim, G., Zarrella, G.: Discriminating gender on Twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1301–1309 (2011)
[4] Fink, C., Kopecky, J., Morawskib, M.: Inferring Gender from the Content of Tweets: A Region Specific Example. In: Proceedings of the 6th International AAAI Conference on Weblogs and Social Media (2012)
[5] Rao, D., Yarowsky, D., Shreevats, A., Gupta, M.: Classifying latent user attributes in twitter. In: Proceedings of the 2nd International Workshop on Search and Mining User-generated Contents, pp. 37–44 (2010)
[6] Tinati, R., Carr, L., Hall, W., Bentwood, J.: Identifying communicator roles in twitter. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 1161–1168 (2012)
[7] Yu, S., Kak, S.: A Survey of Prediction Using Social Media. arXiv preprint:1203.1647 (2012)
[8] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
[9] Drew, J.H., Mani, D.R., Betz, A.L., Datta, P.: Targeting customers with statistical and data-mining techniques. Journal of Service Research 3(3), 205–219 (2001)
[10] Kim, Y., Street, W.N.: An intelligent system for customer targeting: a data mining approach. Decision Support Systems 37(2), 215–228 (2004)
[11] Kim, Y., Street, W.N., Russell, G.J., Menczer, F.: Customer targeting: A neural network approach guided by genetic algorithms. Management Science 51(2), 264–276 (2005)
[12] Berry, M.J., Linoff, G.S.: Data mining techniques: for marketing, sales, and customer relationship management. Wiley Computer Publishing (2004)
[13] Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 42–49 (1999)
[14] Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. Mining Text Data, pp.163–222. Springer US (2012)
[15] Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
[16] Kim, H., Howland, P., Park, H.: Dimension reduction in text classification with support vector machines. The Journal of Machine Learning Research 6, 37–53 (2005)
[17] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: Using kNN model for automatic text categori-zation. Soft Computing 10(5), 423–430 (2006)
[18] Pang, G., Jiang, S.: A generalized cluster centroid based classifier for text categorization. Information Processing & Management 49(2), 576–586 (2013)
[19] Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)

[20] Guan, H., Zhou, J., Guo, M.: A class-feature-centroid classifier for text categorization. In: Proceedings of the 18th International Conference on World Wide Web, pp. 201–210 (2009)
[21] Han, E., Karypis, G., Kumar, V.: Text categorization using weight adjusted k-nearest neigh-bor classification. In: Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 53–65 (2001)