Mining Negative Comment Data of Microblog Based on Merge-AP

Zhijun Chen, Weijian Jin, and Shibiao Mu

Yiwu Industrial and Commercial College, Yiwu 322000, China

Correspondence should be addressed to Zhijun Chen; roderick767@protonmail.com

Received 11 February 2020; Revised 11 April 2020; Accepted 15 April 2020; Published 25 May 2020

A new depiction method based on the merge-AP algorithm is proposed to effectively improve the mining accuracy of negative comment data on microblog. In this method, we first employ the AP algorithm to analyze negative comment data on microblog and calculate the similarity value and the similarity matrix of data points by Euclidean distance. Then, we introduce the distance-based merge process to solve the problem of poor clustering effect of the AP algorithm for datasets with the complex clustering structure. Finally, we compare and analyze the performance of K-means, AP, and merge-AP algorithms by collecting the actual microblog data for algorithm evaluation. The results show that the merge-AP algorithm has good adaptability.

1. Introduction

Social media is an important platform for users to share and acquire information. With the rapid development of social media applications in recent years, more and more users express their views through social media platforms. However, negative comments with negative effects are also emerging.

In order to find out more about the trends of social media reviews in major events, scholars have proposed a large number of evaluation methods. Wu et al. [1] proposed to extract useful sentiment knowledge from massive unlabeled messages to enhance the microblog sentiment classification, and an accelerated algorithm is proposed to tackle the most time-consuming step. Zhang et al. [2] used the network term and the wiki Chinese dataset to expand the original vocabulary with Weibo comment texts, and an optimization method according to the statement length of the pooling layer is put forward. Wang et al. [3] analyzed the process of the information dissemination in the community of Sina Weibo, and the dynamic model is improved and redefined to characterize the community. The experiment shows that this model accurately reflects the dissemination of the information community. Crokidakis et al. [4] studied opinion formation on a fully connected population participating of a public debate, and the results found that the presence of inflexible agents affects the critical behavior of the system, causing either the shift of the critical point or the suppression of the ordering phase transition, depending on the groups of opinions to which the intransigents belong. Pan and Shen [5] introduced some theories and methods of the self-organization system to the research of the diffusion mechanism of mass incidents in Weibo and found that the self-organization criticality system and dissemination bursts can be understood as one kind of self-organization behavior. Guan et al. [6] have selected 21 hot events that were widely discussed on Sina Weibo and done some statistical analyses, and the results found that messages that contain pictures and those posted by verified users are more likely to be reposted, while those with URLs are less likely. Xie et al. [7] used the probabilistic uncertain multiplicative linguistic preference relations to assess the management ways of the online public opinion. The numerical example that helped assess the valid way to manage the online public opinion was performed to check the feasibility of the proposed decision-making procedure. Zhu and Hu [8] used the probabilistic uncertain multiplicative linguistic preference relations to assess the management ways of the online public opinion, and the numerical example could help assess the valid way to manage online public opinion was performed to check the feasibility of the proposed decision-making procedure. Haihong et al. [9] explored the key issues of theme and sentiment analysis from the perspective of...
public opinion analysis with deep learning and experimented with the short text topic classification dataset TREC and the sentiment analysis dataset 1MDB to verify the validity of the proposed model. Huang and Yu [10] designed a system of network public opinion analysis and proposed a user model found on the user access behavior to scientifically classify and represent the relevant theory of the users' retrieval behavior. Wang et al. [11] proposed a supernetwork model of the online public opinion with the superedge coupling algorithm. The experiment result indicated that online opinion cases of the similar type of events had stronger coupling, while cases with opposite psychological types also couple strongly. Ruz et al. [12] considered Bayesian network classifiers to perform sentiment analysis during critical events in Twitter. The results showed the effectiveness of using the Bayes factor measure as well as its competitive predictive results when compared to support vector machines and random forests, given a sufficient number of training examples.

However, the mining accuracy of negative comment data on microblog is not high enough, and the clustering effect of nongroup datasets with the complex clustering structure is especially poor. The distance-based merge process can merge multiple categories into fewer categories according to a certain calculation method to effectively solve the problem of poor clustering effect of complex structures. Therefore, a new depiction method based on the merge-AP algorithm is proposed in this paper, and the similarity value and the similarity matrix of data points are calculated by Euclidean distance, and the damping coefficient is added for optimization to improve the mining accuracy.

2. The Mining Method of Negative Comment Data

In view of the phenomenon that negative comments on microblog have led to unhealthy social atmosphere, an improved algorithm based on affinity propagation (AP) is proposed in this paper to mine negative comment data on microblog. With this method, microblog users who have been verified by their real names within the validity period cluster negative comment data on microblog according to the user type, the number of negative comments posted, whether negative comments contain some keywords, and whether the number of comments liked or forwarded is more than $n$. To acquire the category of negative comments, microblog managers can make use of the clustering results to determine which negative comments need to be processed.

2.1. The Mining Method with AP. AP algorithm clusters according to the similarity between $N$ data points, which can be symmetric or asymmetric. These similarities constitute the similarity matrix $S$ of $N \times N$ (where $N$ refers to having $N$ data points). AP algorithm does not need to specify the number of clusters in advance; instead, it takes all data points as potential clustering centers. The value $s(k, k)$ on the diagonal of the similarity matrix $S$ is taken as the criterion to judge whether the point $k$ can become the clustering center, which means that the greater the value is, the more likely the point is to become the clustering center. The value is also called the reference $p$. The number of clusters is affected by the reference $P$. If a data point is likely to become a clustering center, then $p$ should be the same value. If you take the mean value of the similarity input as the value of $P$, then you get a medium number of clusters. If you take the minimum, you get a cluster with fewer classes. Two types of messages, namely, responsibility and availability, are delivered in the AP algorithm. $r(i,k)$ represents the numerical message sent from the point $i$ to the candidate clustering center $k$ to reflect whether the point $k$ is suitable to be the clustering center of the point $i$. $a(i,k)$ represents the numerical message sent from the candidate clustering center $k$ to the point $i$ to reflect whether the point $i$ selects $k$ as its clustering center. The stronger $r(i,k)$ and $a(i,k)$ are, the more likely the point $k$ is to be the clustering center and the more likely the point $i$ is to select the point $k$ as its clustering center.

The attributes of microblog data used here mainly include the user type (normal users/users with a certain fan base), whether the number of negative comments posted is more than $n$ (yes/no), whether the negative comments contain some keywords (yes/no), and whether the number of comments liked or forwarded is more than $n$ (yes/no), which are mapped to the space, and the microblog data are represented by data points in the space.

The AP algorithm is used to mine negative comments on microblog. The specific algorithm is described as follows.

2.1.1. Initialization. Calculate the similarity value between $N$ points, construct the matrix $S$, and select the $p$ value (generally, the mean value of $S$); set a maximum number of iterations (depending on the dataset), calculate the $r$ and $a$ values for each iteration during the iteration, judge whether they are the clustering center according to the $a(i,k) + r(i,k)$ or $a(k,k) + r(k,k)$ value, and then terminate the operation when the number of iterations exceeds the maximum number of iterations or when the number of consecutive iterations of the clustering center does not change.

2.1.2. Calculate the Similarity Value of $N$ Points. The similarity value of $N$ points is calculated using the Euclidean distance shown in formula (1) and stored in the similarity matrix $S$:

$$s(i,k) = -\left\| (x_i - x_k)^2 + (y_i - y_k)^2 + \cdots + (n_i - n_k)^2 \right\|.$$  

(1)

All the similarity values between the points are generally negative; therefore, the greater the similarity value is, the closer the distance between the points will be for the convenience of comparison and calculation. At the same time, the responsibility matrix $R$ and the availability matrix $A$ are initialized to matrix 0. The parameter $P$, that is, $s(k,k)$ value (generally, the mean value of $S$), is set to determine the number of clusters. The greater the value is, the more clusters there are. It is also required to set the value of the damping coefficient $\lambda$ and the maximum number of iterations.
2.1.3. Perform the Iterative Process

(1) Update the Responsibility Matrix. After considering the availability of all candidate points and the initial similarity matrix of data points to candidate points, the responsibility value of the data point \( i \) to \( k \) is calculated as

\[
r(i, k) = \begin{cases} 
  s(i, k) - \max_{k' \neq k} \{ a(i, k') + s(i, k') \}, & i \neq k, \\
  s(k, k) - \max \{ a(k, k') + s(k, k') \}, & i = k,
\end{cases}
\]

(2) where \( s(i, k) \) is the similarity between the data points \( i \) and \( k \), that is, the similarity calculation of the point \( k \) as the clustering center of the point \( i \). \( a(i, k') \) is the effective information sent to the point \( i \) by other potential clustering centers.

(2) Update the Availability Matrix. The responsibility of each candidate point \( k \) is updated after considering the responsibility of all data points \( i \) to \( k \), that is, the possibility that the candidate point is finally selected as the representative point of the class:

\[
a(i, k) = \begin{cases} 
  \min \left\{ 0, r(k, k) + \sum_{i \neq (i, k)} \max(0, r(i', k)) \right\}, & i \neq k, \\
  \sum_{i \neq k} \max(0, r(i', k)), & i = k.
\end{cases}
\]

(3) Perform Optimization. The iterative process of the AP clustering algorithm is prone to oscillation, so the damping coefficient \( \lambda \) is added for optimization:

\[
r_{\text{new}}(i, k) = \lambda \cdot r_{\text{old}}(i, k) + (1 - \lambda) \cdot r(i, k),
\]

\[
a_{\text{new}}(i, k) = \lambda \cdot a_{\text{old}}(i, k) + (1 - \lambda) \cdot a(i, k).
\]

Repeat Steps (1)–(3) until the matrix is stable or reaches the maximum number of iterations.

2.1.4. Determine the Final Negative Comment Results.

The clustering center point \( k \) (that is, \( k \) that maximizes \( \{a(i, k) + r(i, k)\} \)) of the current sample data point \( i \) is acquired according to Step (3). If \( i = k \), then the sample data point \( i \) is the representative point of its own class; if not, \( i \) is a member of the class to which \( k \) belongs. Steps (1)–(4) are executed iteratively at the same time. Calculation is terminated when the number of iterations exceeds the maximum number of iterations or the clustering center does not change for several consecutive times so as to judge whether the current data are the negative comment data.

2.2. Improved Method. As the dataset with the complex (nongroup) clustering structure has poor clustering effect in the AP algorithm and the number of classes is far more than the number of clustering datasets, the distance-based merge process is introduced to merge multiple categories into fewer categories according to certain calculation methods to solve the problem of poor clustering effect of the AP algorithm.

Here, two clustering optimization indicators are given as follows: (1) the category purity in a certain category is reduced to a certain degree, and the distance threshold of this category is reduced. In the stage where the sample data are divided according to distance association, the reason for the decrease in clustering accuracy rate includes the decrease in class purity in addition to the change in the class distribution structure. And the random selection of the initial subset will also determine the initial class distribution. The error of class division is more serious than the scattered structure of the class distribution. (2) The accuracy of the overall clustering drops to a certain degree, and the overall distance threshold is adjusted. Except for the benchmark dataset, all other sample data will be associatively divided according to their distance from the nearest cluster center. The benchmark data determine the generation of the initial clustering center, and its randomness will have a certain impact on the subsequent. When the number of samples directly divided by association continues to increase, the accuracy of the overall clustering results will continue to decrease, which is manifested in the fact that the class distribution structure is becoming less and less concentrated, and the clustering deviation increases.

\[
\frac{\sum_{i \neq (i, k)} \max(0, r(i', k))}{\sum_{i \neq k} \max(0, r(i', k))} \leq \lambda
\]

(4) Given the parameter \( T \), calculate the distance between two points in two different classes \( \omega_h \) (\( h = 1, 2, \ldots, Z \)) and \( \omega_j \) (\( j = 1, 2, \ldots, Z \)), \( \omega_h \neq \omega_j \), and get the minimum value \( d_{\text{min}} \) if \( d_{\text{min}} < T \times d \) merge these two types; otherwise, do not merge.

(5) Process all \( m \) classes in order according to Step (4).
Therefore, the merge process is introduced in Step (4) of the AP algorithm. After \( m \)-class data are acquired, the distance-based merge process is called to process it to solve the problem of poor clustering effect of the AP algorithm. The specific improvement steps of the merge-AP algorithm are as follows.

The clustering center point \( k \) (that is, \( k \) that maximizes \( \{a(i,k) + r(i,k)\} \)) of the current sample data \( i \) is acquired according to Step (3). If \( i = k \), then the sample data point \( i \) is the representative point of its own class; if not, \( i \) is a member of the class to which \( k \) belongs. After \( m \)-class data are acquired, it is required to calculate the average distance \( d \) between all points in the whole dataset according to formulas (5)–(7), judge whether to include it in the same class by comparing the size of \( d_{\text{min}} \) and \( T \times d \), and thus judge whether the current data are the negative comment data. Steps (1)–(4) are executed iteratively at the same time. Calculation is terminated when the number of iterations exceeds the maximum number of iterations or the clustering center does not change for several consecutive times.

For the AP algorithm, the number of operations required mainly includes the following: the number of operations required to calculate data similarity is \( pN^2 \), the operations required to initialize the parameters are \( 2N^2 + N \), and the iterate operations are \( 2WN^2 \) (\( W \) is the number of iterations), so the total time required by the algorithm is \( pN^2 + 2N^2 + N + 2WN^2 \), and the time complexity is approximately \( O(N^3) \).

The merge method has a minimum comparison number of \( N \log N/2 \) and a maximum comparison number of \( N \log N – N + 1 \), so the average time complexity is \( O(N \log N) \). This paper introduces the merge method in the fourth step of the AP algorithm. In the fourth step, steps (1)–(4) need to be performed iteratively, so the average time complexity is \( O(N^3 \log N) \).

### 3. Data Analysis

In view of the mining method for negative comment data on microblog established above, microblog data are collected here for performance evaluation. Here, we use the microblog posts and comment data published on October 8, 2017, in https://blog.csdn.net/karamos/article/details/80132231 as the data source of our algorithm. The original data system has a large amount of data, but due to the low value density, we preprocess the data (noise removal, cleaning, filtering, etc.) to get 50,000 effective comments on microblog. We also set the category, attribute, and attribute value of negative comments and determine the number of categories, attributes, and attribute values; set the number of iterations, and give the reference \( P \), damping coefficient \( \lambda \), and parameter \( T \).

The experimental environment used in this paper is as follows: Windows 10 operating system, 4 GB memory and Intel Core i5 CPU. The performance of \( K \)-means algorithm, AP algorithm, and merge-AP algorithm proposed in this paper is also compared.

Figure 1 shows the relationship between the accuracy of \( K \)-means, merge-AP, and AP algorithms and the number of negative comments on microblog. As can be seen from Figure 1, the accuracy of the three algorithms increases with the number of negative comments on microblog. When the number of negative comments on microblog increases to a certain value, the accuracy of the three algorithms gradually remains unchanged. This is because the more negative comments on microblog are, the more complete the classification information obtained by the algorithm will be, so the classification of negative comments on microblog will be more accurate. When enough classification information is obtained, the classification accuracy of the algorithm is the highest. When the number of negative comments continues to increase, the accuracy of the algorithm will remain basically unchanged. It can also be seen from the figure that the accuracy of the merge-AP algorithm is always higher than that of the other two algorithms. This is because the distance-based merge process is introduced in the merge-AP algorithm. The clustering effect of AP algorithm is not good, and \( K \)-means algorithm is limited by the initial value of the parameter \( k \), so its classification performance is poor.

Figure 2 shows the relationship between the accuracy of merge-AP, \( K \)-means, and AP algorithms and the number of iterations. As can be seen from Figure 2, the accuracy of the three algorithms first increases with the number of iterations. When the number of iterations increases to a certain extent, the accuracy of the three algorithms remains basically unchanged. This is because as the number of iterations increases, the convergence effect of the algorithm is better, and the accuracy of the algorithm increases. When the number of iterations is larger than the number of iterations required for the complete convergence of the algorithm, the accuracy of the algorithm will remain basically unchanged. It can also be seen from the figure that the merge-AP algorithm has the highest accuracy, and the \( K \)-means algorithm has the lowest accuracy. This is because the merge-AP algorithm introduces the distance-based merge process on the basis of the AP algorithm, which merges the categories with high
similarity to solve the problem of poor clustering effect of the AP algorithm. The clustering effect of AP algorithm is not good, and K-means algorithm is limited by the initial value of the parameter $k$.

Figure 3 shows a comparison of the convergence time between the merge-AP algorithm and the mixing evolution model [13]. From Figure 3, we can see that the convergence time of the merge-AP algorithm is better than the mixing evolution model. The merge-AP algorithm merges categories with high similarity by introducing a distance-based merge process, thereby effectively reducing the convergence time.

Here, the average classification accuracy is defined as the sum of the accuracy of each algorithm tested divided by the number of tests. Figure 4 shows the relationship between the average classification accuracy of merge-AP, $K$-means, and AP algorithms and the number of tests. As can be seen from Figure 4, the average classification accuracy of the three algorithms oscillates continuously with the increase in the number of tests. The oscillation amplitude of merge-AP and AP algorithms is smaller than that of the $K$-means algorithm. This is because the damping coefficients are introduced into merge-AP and AP algorithms, which reduce the oscillation in the iterative process, while the $K$-means algorithm is easy to produce oscillation, leading to the low stability of the algorithm, so its average classification accuracy curve will oscillate greatly with the increase in the number of tests. It can also be seen from the figure that the merge-AP algorithm has the highest average classification accuracy, while the $K$-means algorithm has the lowest average classification accuracy. This is because the merge-AP algorithm introduces the distance-based merge process on the basis of the AP algorithm, which has high accuracy.

Figure 5 shows that the accuracy of the merge-AP algorithm varies with the parameter $T$ under the influence of the number of negative comments on microblog. Regardless of the number of negative comments on microblog, the accuracy of the algorithm increases first and then decreases with the increase of parameter $T$. This is because the setting of the parameter $T$ affects the effect of the algorithm. When $T$ is very small, the merge-AP algorithm degenerates into an AP algorithm, and the accuracy of the algorithm is low; when $T$ is 0.5–2, the algorithm has the best effect and the highest accuracy. When $T$ is very large, it is easy to merge different classes and thus leads to bad results and lower accuracy. However, when the number of negative comments on microblog is 1,000, the accuracy of the algorithm is always greater than that when the number of negative comments is 500. This is because the more the negative comments on microblog are, the more stable
and mature the algorithm is and the more obvious the clustering effect and the higher the accuracy will be.

Figure 6 shows the relationship between the similarity value between sample points and the number of iterations under different damping coefficients $\lambda$. As can be seen from Figure 6, the larger $\lambda$, the better the effect of eliminating oscillations and the smoother the iteration curve. The smaller $\lambda$, the worse the effect of eliminating oscillations and the steeper the iterative curve. The algorithm oscillates greatly when $\lambda = 0.7$ and slightly when $\lambda = 0.9$. As can be seen from the iterative curve, the larger $\lambda$ is, the slower the convergence rate of the algorithm is. This is because the damping coefficient $\lambda$ has a great influence on the operation efficiency of the merge-AP algorithm, and the convergence performance of the algorithm is sensitive to the selection of the initial value of the damping coefficient $\lambda$, so the iteration curve changes obviously.

Figure 7 shows the relationship between the convergence time of the merge-AP algorithm and the similarity value between sample points under different negative comments on microblog. As can be seen from Figure 7, when the similarity between sample points increases, the convergence time of the algorithm will decrease regardless of the number of negative comments on microblog. This is because the higher the similarity between sample points, the better the clustering effect of the algorithm and the shorter the time required for the algorithm to reach stability, so the convergence time of the algorithm will decrease. It can also be seen from the figure that the convergence time of the algorithm increases with the increase of negative comments on microblog. This is because the increase in the number of negative comments on microblog results in the increase of sample points to be classified, and the number of iterations required for the algorithm to reach stability will also increase, so the iteration time of the algorithm will increase.

4. Conclusions

Negative comments with negative effects will have negative social consequences. A mining method based on the merge-AP algorithm is proposed in this paper to effectively improve the mining accuracy of negative comment data on microblog. In this method, we first employ the AP algorithm to analyze negative comment data on microblog and calculate the similarity value and the similarity matrix of data points by Euclidean distance. Then, we introduce the distance-based merge process to solve the problem of poor clustering effect of the AP algorithm for datasets with the complex clustering structure. Finally, we set the category and attribute value of negative comments by collecting the actual
microblog data for algorithm evaluation, determine key parameters such as the number of categories, attributes, and attribute values, compare key factors affecting the algorithm such as sample similarity value, convergence time, and average classification accuracy, and analyze the performance of K-means, AP, and merge-AP algorithms. The results show that the merge-AP algorithm has good adaptability.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant no. 61802123 and 2020 Key Research Project of YWICC (ZD2020JD188-02).

References