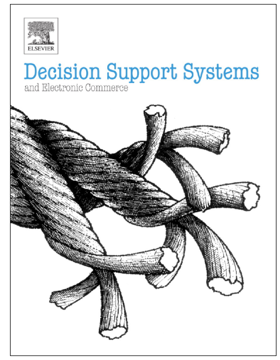

A novel probabilistic graphic model to detect product defects from social media data

Lu Zheng, Zhen He, Shuguang He

PII: S0167-9236(20)30124-X

DOI: https://doi.org/10.1016/j.dss.2020.113369

Reference: DECSUP 113369

To appear in: *Decision Support Systems*

Received date: 6 January 2020

Revised date: 20 July 2020

Accepted date: 20 July 2020

A novel probabilistic graphic model to detect product defects from social media data

Lu Zheng, Zhen He*, Shuguang He

College of Management and Economics, Tianjin University, Tianjin 300072, China

Abstract

Product defects are a major concern for manufacturers and customers. Detecting product defects is vital for manufacturers to prevent enormous product failure costs. As the surge of social media is in vogue, social media data become an important information source for manufacturers to collect defect information. In this study, we propose a novel probabilistic graphic model to discover defects from social media data. We first use three filters, namely, sentiment filter, component-symptom filter and similarity filter, to select informative data. Second, we analyze the remaining data via the proposed probabilistic graphic model and identify defect-related data. Our method provides detailed defect information including defect types, defective components and defect symptoms which is omitted by previous research. A case study in the automobile industry validates the effectiveness and superior performance of our method compared to prior approaches.

Keywords: Product defect detection, Social media data, Probabilistic graphic model, Text analysis

1. Introduction

Product defects are a major concern for manufacturers. To prevent the spread of safety-related defects, manufacturers conduct recalls which cause colossal economic costs. Non-safety related defects do not result in recalls, but can still reduce customer satisfaction

* Corresponding author.
E-mail address: zhhe@tju.edu.cn (Zhen He)

and repurchase intention, and increase within-warranty-repair costs [1]. Hence, for manufacturers, detecting product defects accurately and promptly becomes a primary task of product quality management. Traditionally, data on product defects are collected through customer feedback, complaints or warranty claims. Defect data obtained by these ways have the deficiencies of hysteresis, insufficiency and incomprehensiveness [2]. Recently, the surge of social media has brought many valuable information sources about products. For the advantages of comprehensiveness and promptness, manufacturers have developed a great interest in social media data and utilize them to identify product defects.

However, transforming unstructured and voluminous social media data into useful information on defects is a challenging task for manufacturers. It is unrealistic to handle these data only by manual tagging. To exploit defect information from social media data, lots of researchers develop effective and automated approaches. Among these approaches, the most prominent one is "Smoke Words" proposed by Abrahams et al. [3]. Before smoke words, sentiment polarity was the primary mechanism used. Online data with extremely negative sentiments are assumed to indicate the existence of defects [4]. But Abrahams et al. have verified that identifying product defects based on customer sentiments is biased and unreliable because customers usually describe defects in an objective tone. To overcome this deficiency, Abrahams et al. proposed smoke words and validated their effectiveness in various industries including toy, dishwasher, countertop appliances and medicine [5-7]. Based on smoke words, many researchers have developed different methods to detect defects. Some studies use machine learning methods to identify defect-related data [2, 8, 9]. Other studies apply probabilistic graphic models (PGMs) to discover hidden product defects [6, 10-12]. All these works prove the effectiveness of social media data in product defect detection.

Although the studies mentioned above facilitate the development of product defect detection using social media data, two distinct disadvantages still exist. The first inadequacy is the difficulty of smoke word curation. The construction of smoke words relies on expertise heavily. Only the experts familiar with the product can discern product defects and related words. It is costly for manufacturers to deal with textual data by hiring lots of experts [7]. In addition, the inherent subjectivity of experts will lead to biased results. The second inadequacy is that almost all literature above view defect detection as the task of text classification [2]. Detailed information on defects (e.g., defect types, defective components and defect symptoms) is omitted which is valuable for managerial decisions.

To tackle these research gaps, we propose a novel PGM named Product Defect Detection Model (PDDM) to identify defect-relevant data and further mine detailed information on product defects. Focusing on discussion threads from online forums, we firstly use three filters to extract informative threads. Sentiment filter is used to filter out the positive thread sentences which are irrelevant to defects. Component-Symptom filter is used to find the threads referring to product components or defect symptoms. The similarity filter is to select replies that have the same topics with their posts and to avoid the problem of "topic transfer". After filtering, the remaining informative threads are analyzed by PDDM. PDDM can transmute threads into probability distributions on different defects and provide defect-related words. Probability distributions are used to discover defect-related threads. For a certain thread, once the maximum of its defect probabilities is larger than the preset threshold, this thread is relevant to product defects. Defect-related words are used to reveal detailed defect information including defect types, defective components and defect symptoms.

To summarize, the contributions of our approach are three-fold. Firstly, as a semi-supervised method, PDDM identifies product defects without the dependence on smoke words. This advantage means manufacturers can avoid the heavy dependence on expertise when using PDDM. Secondly, PDDM discovers product defects with high accuracy. Experimental results show that PDDM processes online threads effectively and accurately and outperforms benchmark methods. Thirdly, compared to the previous research, PDDM provides detailed defect information including defect types, defective components and defect symptoms. With detailed defect information, manufacturers can take remedial actions to defects more promptly and accurately.

The remainder of this paper is organized as follows. In Section 2, we conduct a comprehensive literature review about social media data and their significant effect on product defect detection. In Section 3, we lay out the details of three filters and PDDM. Experiments are implemented to validate PDDM's effectiveness and accuracy in Section 4. Section 5 concludes our study and provides an overview of its limitations and opportunities for future work.

2. Literature review

2.1 Social media data

The worldwide surge of social media has completely changed the way customers share their opinions. Customers can express their feelings and experience freely through social media. Therefore, social media data generated by customers become a key information source to get a comprehensive understanding of products. For this reason, researchers have put attention on social media data and published many significant research findings. In the competitive analysis, social media data offer abundant information on products and their competitors which helps manufacturers make proper managerial decisions. He et al. extracted product features that customers preferred and used customer sentiments on these features to compare different products [13]. Jin et al. evaluated customer sentiments on various product features and analyzed product competitors [14]. Liu et al. exploited online reviews and ranked products based on sentiment analysis and fuzzy set theory [15]. With ensemble learning, Liu et al. identified product competitors and customer opinions towards these competitors [16]. Aside from competitive analysis, mining customer requirements and then improving product design also received lots of interest from researchers. Based on text analysis and quality function deployment, Jin et al. estimated probabilities that certain sentences belonged to specific features and then discovered the engineering characteristics to be improved [17]. In the area of Kansei design, social media data help researchers build effective frameworks to discover customers' Kansei requirements and improve product design [18, 19]. Social media data also show their value in service quality analysis [20-23], box-office prediction [24, 25], influential user discovery [26, 27], especially in the area of defect or accident prediction [3, 28-31].

2.2 Product defect detection using smoke words

Among approaches of defect discovery via social media data, the smoke words method is the most contributive one to expose defects buried in social media data. Proposed by Abrahams et al., smoke words are the words most related to product defects. The construction of smoke words includes two steps. Firstly, several experts determine which texts are related to defects and gather detailed defect information manually. Secondly, experts decide the smoke words using term-prevalence metrics (like Correlation Coefficient score (CC score), Relevance Correlation Value score, Information Gain score, etc.) [3, 5]. Although smoke words are proven to be effective in defect detection, building a smoke words lexicon involves expertise which hinders the application of smoke words. And how to curate the best smoke word lexicon may be tough for manufacturers [7]. Besides, the subjectivity of experts also influences the accuracy of smoke words. To overcome these deficiencies, many researchers leverage different approaches in the process of lexicon construction and enhance the performance of smoke words. Extending unigram words into bigram and trigram words, Law et al. proposed "Sparkle words" to improve the accuracy of smoke words [6]. Considering the huge investment of labor, Goldberg et al. used Tabu search heuristic to obtain smoke words and achieved excellent performance [7]. These approaches revive smoke words and make them more effective, but the dependence on expertise is still inevitable.

2.3 Product defect detection using machine learning

The literature using machine learning methods to detect product defects treats defect detection as text classification. These studies emphasize the effect of text classification and categorize texts into defect-relevant texts and defect-irrelevant texts. Lo used Support Vector Machine (SVM) to classify online reviews into complaints and other reviews. Control charts were used to monitor the number of complaints and helped Lo to find out the quality issues [32]. Based on frequencies of smoke words, Zhang et al. used Chi-square to extract features. Then various classification methods were used to discover defect-related data [9]. Liu et al. extracted contextual features from online threads and used multi-view ensemble learning to classify threads [2]. Focused on numerical expressions appearing in the posts, Gruss et al. used Naïve Bayes to extract numerical features and utilized these numerical features to identify product defects [33]. Machine learning shows its performance in defect-related text classification but it cannot provide detailed information on defects. Manual analysis is required when gathering defect information.

2.4 Product defect detection using PGMs

Identifying defects with PGMs is another effective method in the studies of product defect detection or incidents detection. Chen used Latent Dirichlet Allocation (LDA) to analyze software defects and found that LDA has additional explanatory power for software quality [34]. Kinoshita and his workmates built a novel probabilistic topic model to detect real-time traffic incidents [31] while Kuhn et al. utilized Structural Topic Model to uncover hidden aviation incidents [29]. But these models just extract main topics of texts which may not be relevant to defects. Moreover, the methods above fail to provide detailed defect information. To make PGMs ingest defect information from social media data, Zhang et al. proposed a novel PGM which provides defect information including product models, years of production, detective components and symptoms [10]. Zhang et al. developed a PGM named Product Defect Latent Dirichlet Allocation which considers defect resolutions [12]. PGMs do not require large manual reading and tagging. Their performances show that they are effective tools to process social media data and discover product defects.

2.5 Summary

Table 1 summarizes previous works on product defect detection using social media data. From this table, we observe three research gaps. The first gap is that most studies depend on smoke words to identify product defects. This dependence requires the inevitable reliance on expertise. The second gap is that few studies provide detailed defect information which is helpful and valuable for managerial decisions. The third gap is that extant PGM studies do not filter the defect-unrelated texts when extracting topics. Deriving defects from a large amount of defect-irrelevant texts will lead to biased results. Given these research gaps, we propose a novel PGM with filters named Product Defect Detection Model to identify defect-related texts and gather detailed defect information.

Table 1 Summary of literature on product defect detection based on social media data

Researches	Research approaches	Using Smoke Words	Defect information	Defect-unrelated text filtering	Data
Lo, [32]	Machine learning				Online reviews
Zhang et al., [9]	Machine learning				Online threads
Liu et al., [2]	Machine learning	√			Online threads
Gruss et al., [33]	Machine learning				Online threads
Abrahams et al., [3]	Smoke words	√			Online threads
Winkler et al., [5]	Smoke words	√			Online reviews
Goldberg et al., [7]	Smoke words	√			Online reviews
Law et al., [6]	Sparkle words	√			Online reviews
Zhang et al., [10]	PGM	√	√		Online threads
Zhang et al., [12]	PGM		√		Online threads
This study	PGM		√	√	Online threads

3. Research methodology

In this section, we focus on online threads which are a typical kind of social media data and introduce the processes of data filters and PDDM. The overview of data filters and PDDM is presented in **Fig.1**.

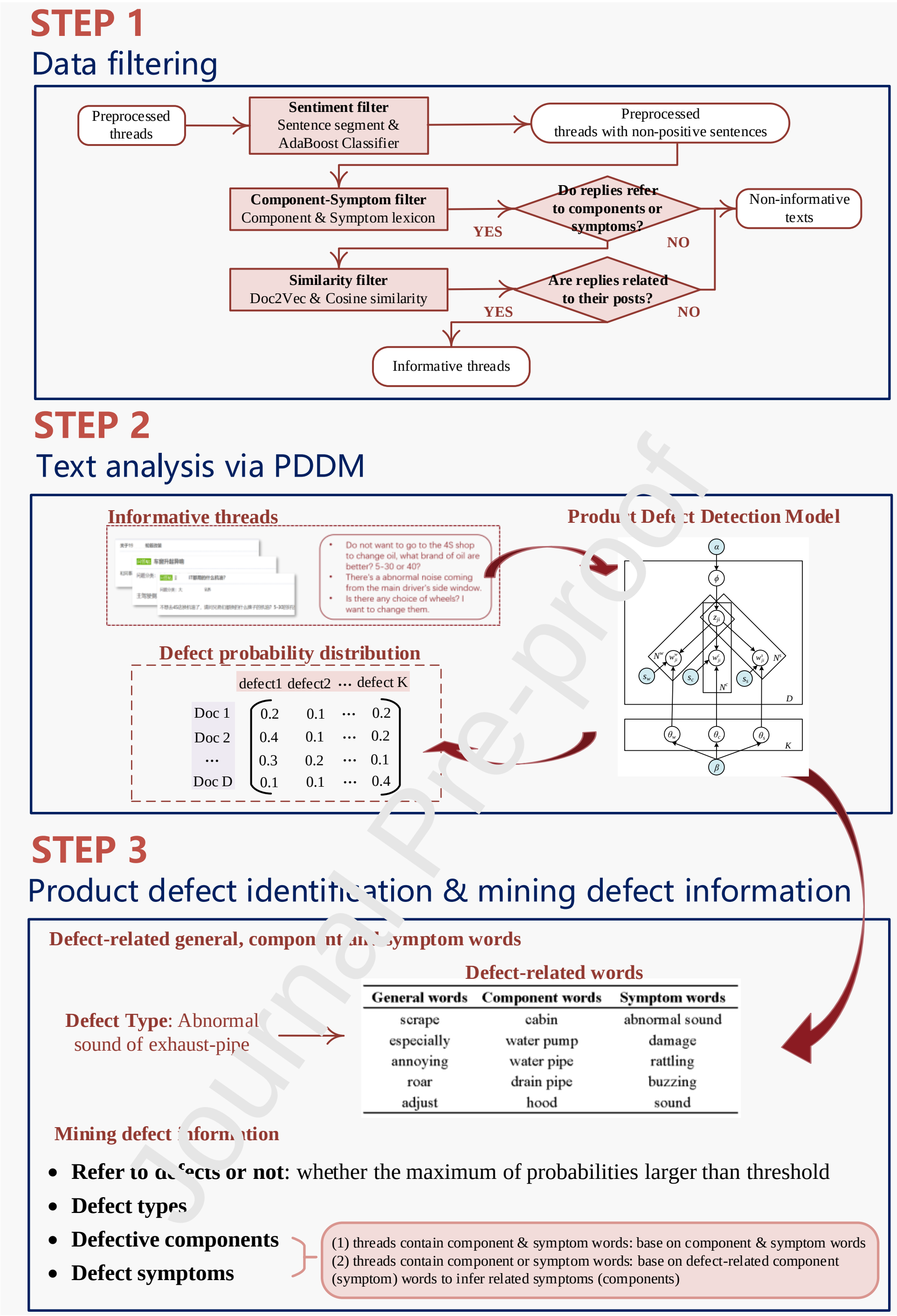

Fig. 1. The overview of data filters and PDDM

3.1 Texts preprocessing

Text preprocessing is essential in text analysis. It includes stop-word and common word removal, and special symbols filtering. Stop-words and common words are useless in defect detection because they are usually meaningless with high word frequencies. Special symbols filtering eliminates symbols like URLs or foreign language words. These symbols disrupt the discovery of defects. After preprocessing, the remaining threads will be leveraged as the input of filters for informative threads selection.

3.2 Data filtering

Diversity of contents in online threads adds noise to defect-relevant data. Filtering out the noise in threads can improve the accuracy of defect detection. Therefore, we develop three filters to select informative threads.

The first filter is the sentiment filter. Though sentiment analysis has been proven to be ineffective in defect detection, it can be used to filter out the threads irrelevant to product defects. Given that a sentence usually contains a discussed object and a kind of sentiment, we conduct sentence segments for each thread and measure the sentiment polarity of sentences. Sentiment polarity is the degree of positive or negative sentiments expressed in sentences. Based on the sentiment polarity of sentences, we exclude the positive sentences that are usually talking about customer satisfaction or other defect-irrelevant topics. In this step, machine learning methods are used to classify sentences into positive sentences and non-positive sentences and then remove the positive sentences. Section 4.3.1. gives a detailed procedure.

The second filter is the component-symptom filter. Threads referring to product components or defect symptoms are more likely to discuss product defects. Therefore, the component-symptom filter is to select the threads containing component or symptom words. Component words are the words describing product components. Symptom words are the words that depict symptoms when products have quality issues. The words except for component words and symptom words are general words. Taking automobiles as an example, cylinder, brake pad, and steering wheel are component words while abnormal sound and oil leakage are symptom words describing car failures. All component words and symptom words are collected in the form of lexicons. The component lexicon is composed of all components of the product. The symptom lexicon is built by selecting words from product maintenance reports manually. We create these lexicons by reading and tagging 2000 maintenance reports manually. In addition to the construction of component and symptom lexicons, our method has no other labor requirements.

The third filter is the similarity filter. For a thread consisting of a post and several replies, its replies may deviate from what the post discusses about over time. This phenomenon is called "topic transfer". Topic transfer carries much useless information and influences our judgment of thread topics. Hence, we use the similarity filter to exclude the replies that don't share the same topics with their posts. The similarity filter is to measure the similarity between posts and their replies. For a certain post, if the similarity between the post and its reply is over the preset threshold, then the reply shares the same topics with the post. Otherwise, the reply should be removed. As for the threads without reply, we assume that these threads pass the similarity filter. To measure the similarity between the post and the reply, we first use the *Doc2Vec* model proposed in [35] to transmute posts and replies into vectors, then cosine similarity is used to measure the similarity between these vectors. Cosine similarity is calculated by:

$$sim(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}, \tag{1}$$

where **a** denotes the vector of a post and **b** denotes the vector of a reply, n is the dimension of vectors. After data filtering, we analyze the remaining threads by PDDM.

Table 2 Variable description

Notation	Description
K	The number of defects in the defect set
D	The number of sentences in the sentence set
N^w, N^c, N^s	The number of general, component and symptom words
z_{ji}	Defect assignment to word i in thread j
$z_{j,\neg i}$	Defect assignment to words in thread j excluding word i
w_w, w_c, w_s	General, component and symptom words
s_w, s_c, s_s	The label of general, component and symptom words
$\boldsymbol{\phi}$	Multinomial distribution over defects
$\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$, $\boldsymbol{\theta_s}$	Multinomial distribution over general, component and symptom words
α	Dirichlet prior for hidden variable $\boldsymbol{\phi}$
β	Dirichlet prior for hidden variable $\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$, $\boldsymbol{\theta_s}$
$n_{jk,\neg i}$	Number of words assigned to defect k in thread j excluding word i
$n^w_{kt,\neg i}$	Number of general words assigned to topic k in thread j excluding word i
$n^c_{kt,\neg i}$	Number of component words assigned to topic k in thread j excluding word i
$n^s_{kt,\neg i}$	Number of symptom words assigned to topic k in thread j excluding word i

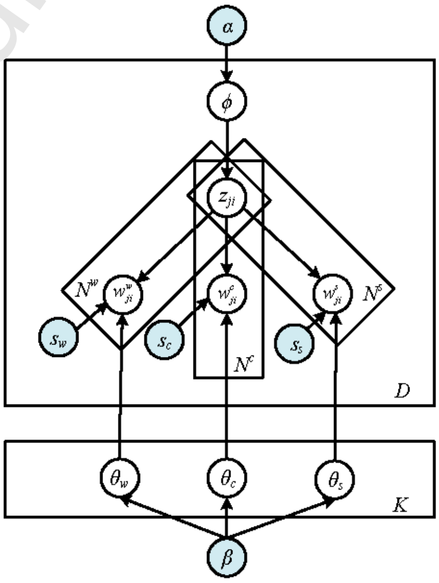

Fig.2. Plate notation of PDDM

3.3 Product Defect Detection Model

PGMs have the advantages of independence on expertise and labor and the provision of detailed defect information. Therefore, we develop a PGM named Product Defect Detection Model to analyze filtered threads and identify enclosed defects. **Fig. 2** is the plate notation of PDDM and **Table 2** describes the meanings of variables in PDDM. As shown in **Fig.2**, PDDM displays the thread generative process which is to produce the defect distribution and word distributions and then sample topics and words for each thread. The first step of PDDM is to determine the defect distribution and word distributions. For thread j, we first draw its defect *Multinomial* distribution ϕ_j which is sampled from *Dirichlet* distribution with the parameter α. Then PDDM determines the word distributions for each defect. We use $\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ to represent the general, component and symptom word distributions which are drawn from the same *Dirichlet* distribution β. The second step of PDDM is to draw a defect and related words for thread j based on these two distributions. For the i^{th} word in thread j, PDDM first samples a defect z_{ji} from ϕ_j and then chooses a word w_{ji} from word distributions related to z_{ji}. To denote which type word w_{ji} belongs to, we use three labels named s_w, s_c, s_s for indication. s_w, s_c and s_s are valued 0 or 1. If w_{ji} is a general word ($s_w = 1$), we draw w_{ji} from θ_w^{zji} (the general word distribution related to z_{ji}). If w_{ji} is a component word ($s_c = 1$), we draw w_{ji} from θ_c^{zji} (the component word distribution related to z_{ji}). If w_{ji} is a symptom word ($s_s = 1$), we draw w_{ji} from θ_s^{zji} (the symptom word distribution related to z_{ji}). When finishing the word sampling, PDDM has generated the thread j. The pseudo of PDDM is presented in **Algorithm 1**.

Algorithm 1: Generative process of threads via PDDM

Step:

```
for each thread j do
    Draw φ_j ~ Dirichlet (α)
    for each word w_ji in j do
        Draw a defect z_ji ~ Multinomial (φ_j)
        if s_w = 1 then
            Draw θ_w^zji ~ Dirichlet (β)
            Draw w_ji^w ~ Multinomial (θ_w^zj)
        end
        if s_c = 1 then
            Draw θ_c^zji ~ Dirichlet (β)
            Draw w_ji^c ~ Multinomial (θ_c^zj)
        end
        if s_s = 1 then
            Draw θ_s^zji ~ Dirichlet (β)
            Draw w_ji^s ~ Multinomial (θ_s^zj)
        end
    end
end
```

However, in the thread generative process introduced above, the variables $\boldsymbol{\phi}$, $\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ are unobserved. To obtain the defect probability distribution for each thread, we need to estimate the hidden variables first. Gibbs sampling is an effective estimation approach to infer parameters, especially in the inference of topic models. The update rule of Gibbs Sampling shows how PDDM assigns defects to general, component and symptom words in a thread. For word w_{ji}, its Gibbs updating rule is:

$$P(z_{ji}=k \mid z_{j,\neg i},\mathbf{W},\mathbf{S},\alpha,\beta)$$
$$\propto \left[s_w \cdot \frac{n^{w}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^w} n^{w}_{kt',\neg i}+N^w \cdot \beta} + s_c \cdot \frac{n^{c}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^c} n^{c}_{kt',\neg i}+N^c \cdot \beta} + s_s \cdot \frac{n^{s}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^s} n^{s}_{kt',\neg i}+N^s \cdot \beta} \right] \cdot \frac{n_{jk,\neg i}+\alpha}{\sum_{k'=1}^{K} n_{jk',\neg i}+K\alpha}. \tag{2}$$

Using Eq.(2), we sample the dataset of D threads repeatedly until we obtain the convergent sampling result. The detailed derivation process of Eq.(2) and steps of Gibbs Sampling are introduced in **Appendix A**. With the convergent sampling result, we estimate multinomial distributions $\boldsymbol{\phi}$, $\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$. $\boldsymbol{\phi}$ indicates the defect distribution of threads. $\boldsymbol{\theta_w}$, $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ indicate the word distributions for defects. The probability that thread *j* referring to defect *k* is given by:

$$\phi_{jk} = \frac{n_{jk} + \alpha}{\sum_{k'}^{K} n_{jk'} + K\alpha}. \tag{3}$$

And for defect *k*, the defect-word distributions for general, component and symptom words are given respectively by:

$$\theta_w^{kt} = \frac{n_{kt}^w + \beta}{\sum_{t'=1}^{N^w} n_{kt'}^w + N^w\beta}, \quad \theta_c^{kt} = \frac{n_{kt}^c + \beta}{\sum_{t'=1}^{N^c} n_{kt'}^c + N^c\beta}, \quad \theta_s^{kt} = \frac{n_{kt}^s + \beta}{\sum_{t'=1}^{N^s} n_{kt'}^s + N^s\beta}. \tag{4}$$

3.4 Product defect detection via PDDM

After inferring defect distributions, we decide which threads are related to product defects. For thread *j*, if the maximum of ϕ_j is over the threshold μ we set before, we deem that this thread refers to defects. Otherwise, thread *j* is irrelevant to defects.

PDDM also provides defect-related general, component and symptom words for each defect. We first determine the names of defects by their defect-related words. Then detailed defect information on defect types, defective components and defect symptoms is obtained based on these defect-related words. The mining process is introduced in **Algorithm 2**.

Algorithm 2: Detailed defect information mining based on PDDM

Input: threads, defects types and their top N related component and symptom words provided by PDDM

Output: detailed defect information including defect types, defective components and defect symptoms

Steps:

1. **for each** thread **do**

2. split the thread into sentences

3. **for each** sentence **do**

4. extract component and symptom words contained in this sentence and use **W** to denote the set of all these words

5. the defect type discussed by this sentence is the defect whose defect-related words contain the most words in **W**

6. **if** the sentence contains both component and symptom words

7. the defective component is the component word

8. the defect symptom is the symptom word

9. **elseif** the sentence contains a component word

10. the defective component is the component word

11. the defect symptom is inferred from the symptom words which are related to the defect type discussed by this sentence

12. **else** the sentence contains a symptom word

13. the defect symptom is the symptom word

14. the defective component is inferred from the component words which are related to the defect type discussed by this sentence

15. **end**

16. **end**

17. the defect type discussed in the thread is the defect mentioned by most sentences of the thread

18. **end**

4. Case study: defect detection in the automobile industry

Defects of automobiles are severe threats to manufacturers and customers. Therefore, identifying product defects promptly and accurately is vital for automobile manufacturers. We use a case study in the automobile industry to validate the effectiveness of our method.

4.1 Experimental setup

We gather online threads from Autohome (www.autohome.com), a famous website of automobiles in China. 15042 Chinese threads are crawled from the Volkswagen Sagitar forum because the Sagitar model achieves very high sales in China (According to the report of China Association of Automobile Manufacturers, the Sagitar model achieves 5th place in the ranking of sedan sales and sold 307,000 cars in China in 2019[1]) but its manufacturer conducted a recall for the vital defect of broken rear suspension trailing arm. The thread dataset is tagged manually by three undergraduate students majoring in Industrial Engineering at Tianjin University and each thread was tagged by three students. The inter-annotator agreement rate is 90.13%. The description of the dataset is shown in **Table 3**. We preprocess texts and conduct data filtering and PDDM using Python 3.7. The preset threshold of the similarity filter is 0.5. Parameters of PDDM is set as Zoghbi et al. do: $\alpha = 50/D$, $\beta = 0.01$ [36].

Table 3 Description of thread dataset

Attributes of the dataset	Number
Number of threads	15042
Number of defect-related threads	907
Number of sentences in all threads	166125
Average number of comments per thread	8.26
Number of threads after filtering	5178
Number of general words in filtered threads	26932
Number of symptom words in filtered threads	340
Number of component words in filtered threads	287

4.2 Performance evaluation metrics

We use Precision, Recall and F-Measure to assess the performance of defect-related text identification. These metrics are calculated based on the confusion matrix in **Table 4**.

[1] http://www.caam.org.cn/chn/4/cate_31/con_5228406.html

Precision is calculated by *TP* / (*TP* + *FP*) while recall equals *TP* / (*TP* + *FN*). F-Measure takes a thorough consideration of precision and recall and equals 2×Precision×Recall / (Precision + Recall).

Table 4 Confusion matrix

	Actually positive	Actually negative
Predicted as positive	True Positive (TP)	False Positive (FP)
Predicted as negative	False Negative (FN)	True Negative (TN)

4.3 Experimental results

4.3.1 Comparison of classification methods in the sentiment filter

In the process of the sentiment filter, we use machine learning methods to classify threads into positive threads and non-positive threads. These machine learning methods are Random Forest (RF), Decision Tree (DT), Gradient Boosting Decision Tree (GBDT), K-Nearest Neighbor (KNN), Logistic Regression (LR), SVM, XGBoost (XGB) and AdaBoost (Ada). To train these models, we collect 2000 threads and tag them with their sentiments. Then 10-fold cross validation is conducted on these classifiers to evaluate their performance. For comparison, we also use the effective Python library named SnowNLP which specializes in Chinese text processing including sentiment analysis. **Fig.3** shows the performance of various methods on the validated dataset.

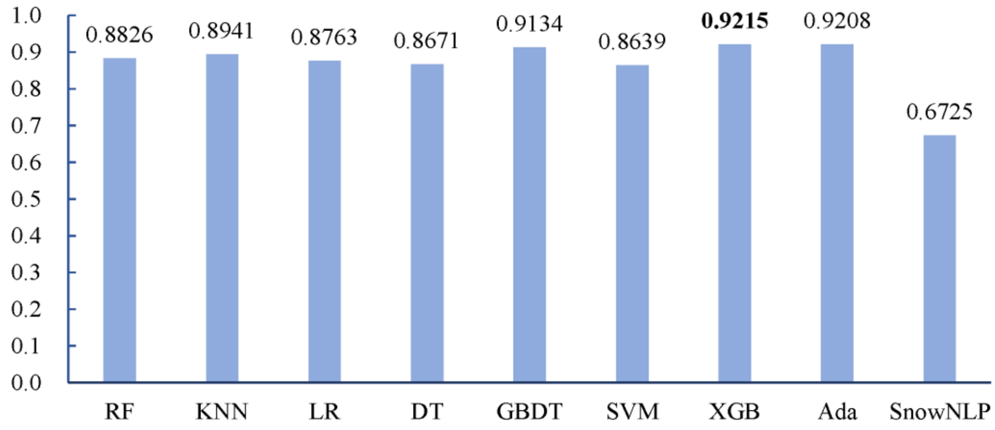

Fig. 3. F-Measure results of sentiment analysis methods

From **Fig.3** we can see that XGBoost has the highest F-Measure score and outperforms other methods. Hence, we choose XGBoost to analyze thread sentiments in the sentiment filter and train XGBoost with the 2000 threads.

To evaluate the performance of the sentiment filter and similarity filter, we select 1500 threads from 15042 threads randomly and display the performance of two filters in **Table 5**. We also apply the famous word embedding method, *Glove* [37], in the similarity filter for comparison. **Table 5** shows that our sentiment filter performs well in the sentiment classification. Moreover, the similarity filter used in our study achieves good performance and outperforms the *Glove* based similarity filter.

Table 5 Performance of sentiment and similarity filters

Filters	Precision	Recall	F-Measure
Sentiment filter	0.9800	0.9827	0.9814
Similarity filter (*Doc2Vec*)	0.9195	0.8948	0.9070
Similarity filter (*Glove*)	0.3610	0.9484	0.5229

4.3.2 Comparison of classification methods in defect-related texts identification

We analyze the performance of PDDM under the different defect numbers and threshold μ in **Fig. 4**. **Fig.4** illustrates that F-Measure scores rise gradually with the increase of μ. When μ is less than 0.05, the performance of PDDM with different defect numbers is inapparent. But when μ is larger than 0.05, F-Measure scores of different PDDM models differ significantly. When μ is 0.2, five PDDM models achieve the highest F-Measure scores. And the best F-Measure score is obtained when D = 20 and μ = 0.2. Hence, we use the performance of PDDM when D = 20 and μ = 0.2 for further analysis and comparisons.

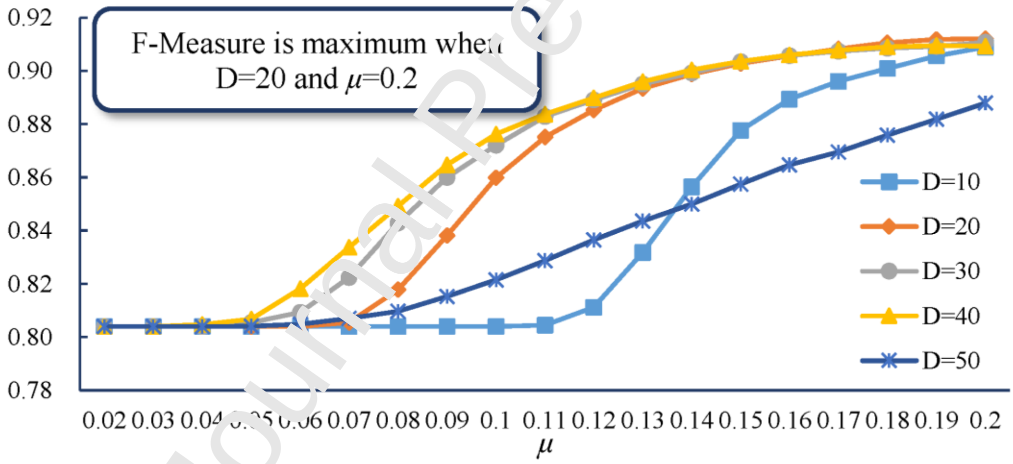

Fig.4. The F-Measure of PDDM with various defect numbers and μ

To evaluate the defect detection performance of PDDM, we compare PDDM with machine learning methods mentioned in *4.3.1*. We retag the 2000 thread dataset in *4.3.1* to find which threads are relevant to defects. And we retrain these machine learning methods using the retagged dataset for defect discovery. To select the best size of the training and the holdout dataset, we display the performance of machine learning methods on various sizes of the holdout dataset in **Fig.5**. The abscissa of **Fig.5** indicates the proportion of the holdout dataset to the whole dataset. For 8 machine learning methods, the optimal sizes are 0.3, 0.3, 0.1, 0.3, 0.1, 0.2, 0.2 and 0.1 respectively. Therefore, based on these optimal sizes, we train machine learning methods to identify defect-related threads. In addition, we compare PDDM with famous clustering methods including K-means and LDA. As the most prevalent approach in defect detection, smoke words are also used for comparison. We obtain smoke words of automobiles via CC scores according to the procedure in Winkler et al. [5] and identify defect-related threads by summing up the CC scores of smoke words contained in the thread. If the aggregate is over 0, the thread is related to defects. Otherwise, the thread is unrelated to defects.

Table 6 describes the performance of various methods in defect-related thread identification. It is worth noting that filters exclude 9864 non-informative threads and select 5178 informative threads from the thread dataset. These results indicate that filters reduce the number of defect-unrelated threads to a large extent. From **Table 6** we can see clearly that PDDM with filters has an excellent performance in Recall. And it has the highest F-Measure of 0.9125, which even outperforms the rest machine learning methods. Comparing to K-means, LDA and smoke words, PDDM with filters enhances the performance over 3%, 25% and 78% respectively. Overall, PDDM with filters improves the performance significantly in comparison with K-means, LDA and smoke words, even is comparable with machine learning methods.

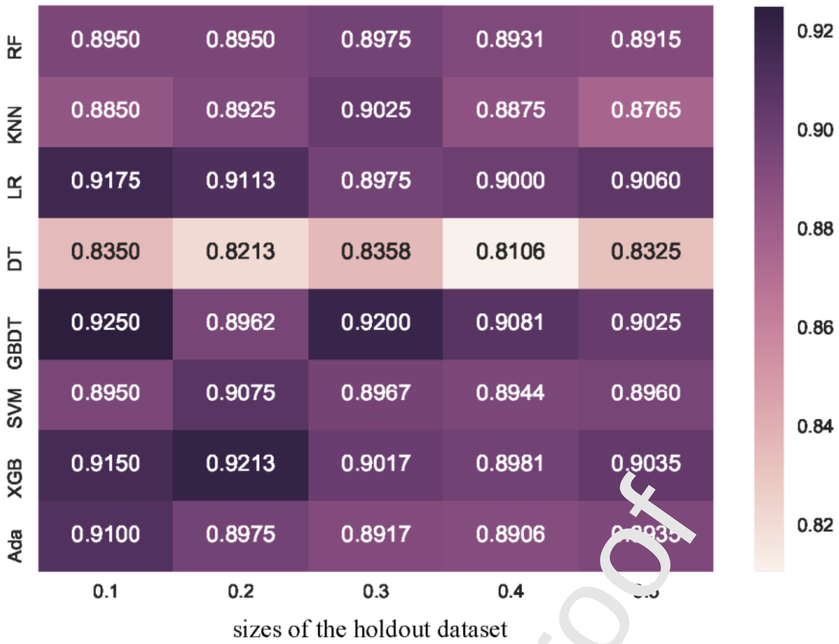

Fig.5 The F-Measure of machine learning methods on various sizes of the holdout dataset

Table 6 Performance of classification methods in defect-relevant threads identification

Methods	Precision	Recall	F-Measure
RF	0.9226	0.8811	0.9014
KNN	0.9232	0.8761	0.8991
LR	**0.9282**	0.8547	0.8899
DT	0.9102	0.7683	0.8333
GBDT	0.9261	0.8587	0.8911
SVM	0.9222	0.8920	0.9068
XGB	0.9272	0.8695	0.8974
Ada	0.9253	0.8557	0.8891
K-means	0.8910	0.8838	0.8874
LDA	0.9212	0.6024	0.7284
Smoke Words	0.5492	0.4810	0.5128
PDDM with filters	0.8891	**0.9372**	**0.9125**

4.3.3 Defect information obtained via PDDM

PDDM provides defect-related words for each product defect (shown in **Appendix B**) and we gather detailed defect information depending on these words. We extract 20 defects from the thread dataset and decide the defect names by the meanings of defect-related words. Defect names are translated into English in **Table 8**. Then we use threads in **Table 7** to illustrate how PDDM mine detailed defect information.

Table 7 Examples of defect information mining

Thread A *Post: Sometimes there is an abnormal sound at the left rear of the sunroof when I drive, like the sound of iron sheets or glasses. It is annoying.* *Reply 1: Me too. How to solve it?* *Reply 2: Go to the 4S shop* quickly.*	***Annotated Sentences in Thread A*** *Post:* (1) *Sometimes there is an* ***abnormal sound*** **[S]***at the left rear of the* ***sunroof*** **[C]** *when I drive.* (2) *like the sound of iron sheets or* ***glasses*** **[C]**. (3) *It is annoying.* *Replies:* (4) *Me too.* (5) *How to solve it?* (6) *Go to the 4S shop quickly.*
Thread B *Post: What's wrong with my car's door?* *Reply: Check the switch.*	***Annotated Sentences in Thread B*** *Post:* (1) *What's wrong with my car's* ***door*** **[C]***?* *Reply:* (2) *Check the* ***switch*** **[C]**.
Thread C *Post: Why does my car always squeak?* *Reply: You'd better go to the 4S shop.*	***Annotated Sentences in Thread C*** *Post:* (1) *Why does my car always* ***squeak*** **[S]***?* *Reply:* (2) *You'd better go to the 4S shop.*

*4S shop is an automobile sales service shop. 4S shop integrates the functions of automobile sales, maintenance, accessories and information collection services. 4S means sale, spare part, service and survey.

We first determine defect types for each thread. Following the process of **Algorithm 2**, we segment threads into sentences and bold the component (using **[C]** for annotation) and symptom words (using **[S]** for annotation). Taking Thread A as an example, Thread A consists of six sentences, two of which contain defect-related words. Based on defect-related words in **Appendix B**, we find that D19's defect-related words contain component and symptom words appearing in sentence (1) and (2). Hence, the defect type of Thread A is D19. Similarly, defect types of Thread B and Thread C is D20 and D8 respectively.

After determining defect types for each thread, we derive defective components and defect symptoms. For the sentences (like Thread A sentence (1)) containing both component and symptom words, the defective component is the component word and the defect symptom is the symptom word. Thread A sentence (1) contains the component word “sunroof” and the symptom word “abnormal sound”. So, the defective component is the sunroof and the defect symptom is the abnormal sound. For the other sentences only containing component words (like Thread B sentence (1)), the defective component is the component word. Therefore, the defective component mentioned in Thread B sentence (1) is the door. Given that Thread B refers to D20, the potential symptoms may be “cannot open” or “abnormal sound”. For the sentences only contain symptom words (like Thread C sentence (1)), the symptom is the symptom word relevant to the defect type discussed by the thread. Hence, the symptom mentioned by Thread C sentence (1) is “squeak”. The defective components may be the glove compartment according to component words related to D8.

Table 8 20 defects extracted from the thread dataset

Defects	Defect types	Defects	Defect types
D1	Abnormal sound of exhaust-pipe	D11	Shifting problems
D2	Engine defects	D12	Abnormal sound when braking
D3	Dashboard defects	D13	Car shuddering
D4	Fuel leakage	D14	Electronic device defects
D5	Abnormal sound when driving	D15	Smoking
D6	Console defects	D16	Water leakage
D7	Fuel caps cannot be closed or opened	D17	Air conditioning defects
D8	Abnormal sound of the interior	D18	Difficult to start
D9	Underpowered vehicles	D19	Window defects
D10	Transmission defects	D20	Door defects

5. Conclusion

In this study, we propose a novel PGM named PDDM to discover product defects from online threads. We first use three filters, namely the sentiment filter, the component-symptom filter and the similarity filter, to select informative threads. The sentiment filter is to remove the threads with positive sentiments. The component-symptom filter is to find out threads referring to product components or defect symptoms. The similarity filter is to remove the replies irrelevant to their posts and solve the problems of “topic transfer”. Then we analyze remaining threads via PDDM. PDDM identifies the threads related to defects and provides detailed defect information including defect types, defective components and defect symptoms. Finally, a case study is used to validate the effectiveness of our method. PDDM with filters achieves excellent performance and offers detailed defect information which is overlooked by baseline methods. With detailed defect information, manufacturers can make prompt and proper remedial decisions and prevent the spread of product defects. Our work contributes a novel way to detect product defects from social media data and can be implemented in practice by manufacturers to sift through voluminous data.

Limitations still exist in our work. Firstly, the number of defects is preset manually. How to select a proper number of defects still needs further research. Secondly, defect types are determined by the meanings of words extracted from threads. How to overcome the inherent subjectivity is an urgent task for PGMs. Thirdly, due to the limited component and symptom lexicons, we cannot discover the defects which have never occurred before. In future research, several extensions of research interests can be explored. First, defect solutions and causes of defects can be added into PDDM to provide more valuable defect information. Second, to identify defects more promptly, time labels can be considered in PDDM to predict defects. Based on PDDM and time labels, defect early warning systems can be constructed to catch the defects hidden in social media data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.71661147003 and Grant No.71532008).

Appendix A Derivation of the Gibbs Sampling rule and steps of Gibbs Sampling

By applying Bayes' rule to $P(z_{ji}=k\,|\,\mathbf{z}_{j,\neg i},\mathbf{W},\mathbf{S})$, we obtain:

$$P(z_{ji}=k\,|\,z_{j,\neg i},\mathbf{W},\mathbf{S},\alpha,\beta)\propto P(W_{ji}\,|\,z_{ji}=k,\mathbf{z}_{j,\neg i},\mathbf{W}_{j,\neg i},\mathbf{S}_{ji},\beta)P(z_{ji}=k,|\,\mathbf{z}_{j,\neg i},\alpha). \tag{5}$$

Given that *Dirichlet* distribution is conjugate to *Multinomial* distribution, the first term of Eq.(5) can be written as Eq.(6),

$$\begin{aligned} P(W_{ji}\,|\,z_{ji}=k,\mathbf{z}_{j,\neg i},\mathbf{W}_{j,\neg i},\mathbf{S}_{ji},\beta) &= \int_{\theta_{\mathbf{S}k}} P(W_{ji}\,|\,z_{ji}=k,\mathbf{z}_{j,\neg i},\theta_{\mathbf{S}k})P(\theta_{\mathbf{S}k}\,|\,\beta) \\ &= \int_{\theta_{\mathbf{S}k}} \theta_{\mathbf{S}k}\cdot Dir(\theta_{\mathbf{S}k}\,|\,n^{\mathbf{S}}_{kt,\neg i}+\beta)d\theta_{\mathbf{S}k} \\ &= E[Dir(\theta_{\mathbf{S}k}\,|\,n^{\mathbf{S}}_{kt,\neg i}+\beta)] \\ &= \frac{n^{\mathbf{S}}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{\mathbf{W}}} n^{\mathbf{S}}_{kt',\neg i}+N^{\mathbf{W}}\cdot\beta}, \end{aligned} \tag{6}$$

where $n^{\mathbf{S}}_{kt,\neg i}$ means the number of times that term t was assigned to defect k without considering the current word. Noting that $\mathbf{S}$ comprises three types of words, we can extend Eq.(6) and obtain:

$$P(W_{ji}\,|\,z_{ji}=k,\mathbf{z}_{j,\neg i},\mathbf{W}_{j,\neg i},\mathbf{S}_{ji},\beta)=s_w\cdot\frac{n^{w}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{w}} n^{w}_{kt',\neg i}+N^{w}\cdot\beta}+s_c\cdot\frac{n^{c}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{c}} n^{c}_{kt',\neg i}+N^{c}\cdot\beta}+s_s\cdot\frac{n^{s}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{s}} n^{s}_{kt',\neg i}+N^{s}\cdot\beta}. \tag{7}$$

Similarly, the rightmost term of Eq.(5) can be yielded by integrating out $\boldsymbol{\phi}$,

$$P(z_{ji}=k\,|\,\mathbf{z}_{j,\neg i},\alpha)=\int_{\phi_j} P(z_{ji}=k\,|\,\phi_j)P(\phi_j\,|\,\alpha)d\phi_j=\frac{n_{jk,\neg i}+\alpha}{\sum_{k'=1}^{K} n_{jk',\neg i}+K\alpha}, \tag{8}$$

where $n_{jk,\neg i}$ denotes the number of times thread j is assigned to defect k, without considering the current word. Hence, combining Eq.(7) and Eq.(8), Eq.(5) yields:

$$\begin{aligned} &P(z_{ji}=k\,|\,z_{j,\neg i},\mathbf{W},\mathbf{S},\alpha,\beta) \\ &\propto\left[s_w\cdot\frac{n^{w}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{w}} n^{w}_{kt',\neg i}+N^{w}\cdot\beta}+s_c\cdot\frac{n^{c}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{c}} n^{c}_{kt',\neg i}+N^{c}\cdot\beta}+s_s\cdot\frac{n^{s}_{kt,\neg i}+\beta}{\sum_{t'=1}^{N^{s}} n^{s}_{kt',\neg i}+N^{s}\cdot\beta}\right]\cdot\frac{n_{jk,\neg i}+\alpha}{\sum_{k'=1}^{K} n_{jk',\neg i}+K\alpha}. \end{aligned}$$

With the equation above, we conduct Gibbs Sampling on our thread dataset and estimate the hidden variables. The process of Gibbs Sampling is (1) Initialization: for each word in thread j, assign the thread a defect number randomly. (2) Probability calculation: for each word in thread j, calculating the probability that the word assigned to defect k using the equation above. (3) Defect number re-assignment: re-assign the defect number for each word in thread j based on the maximum of probabilities on K defects. (4) Repeat step(2)-(3) until the number of iterations exceeds the preset threshold and obtain the final defect assignment for each word in each thread. (5) Variable estimation: estimating $\boldsymbol{\phi}$, $\boldsymbol{\theta}_w$, $\boldsymbol{\theta}_c$ and $\boldsymbol{\theta}_s$ based on the final defect assignment and Eq.(4). To make Gibbs Sampling more understandable, we also use a simple example (shown in **Fig.6**) to illustrate the process of sampling and layout the detailed calculation process.

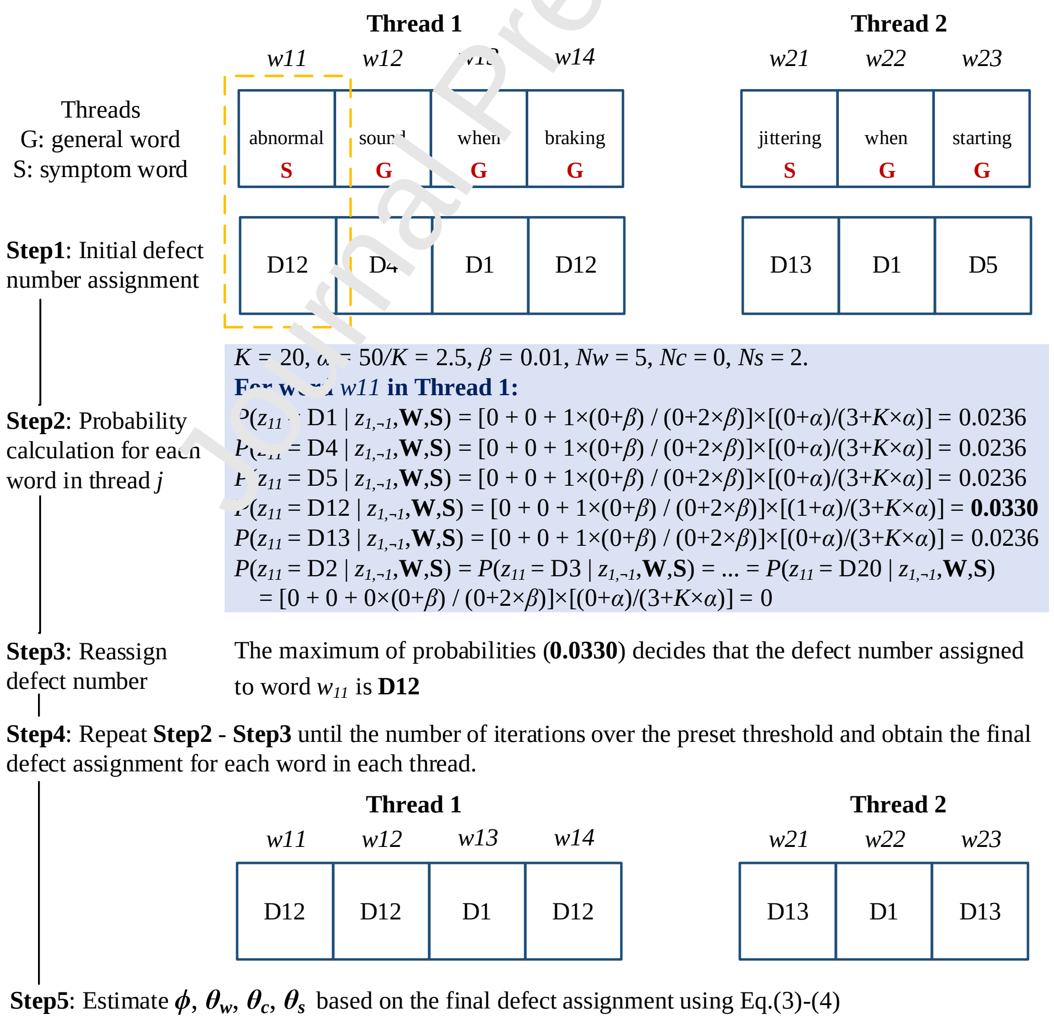

Fig.6 An example to illustrate the process of Gibbs Sampling in PDDM

Appendix B. Top keywords, for each Defect Type, extracted by PDDM

Defects	GWs	CWs	SWs	Defect-related word illustration
D1: Abnormal sound of exhaust-pipe	scrape, annoying, roar	rear-wheel, water pipe, exhaust- pipe	abnormal sound, damage, rattling	Cars with D1 often have annoying roar and customers choose to scrape exhaust-pipes. Customers often hear the sound from rear-wheels. And the abnormal sound of exhaust-pipes is usually accompanied by ponded water which may be caused by water pipes.
D2: Engine defects	hear, 1.4T, louder	turbine, engine, nozzle	rattling, strenuous, abnormal sound	Customers usually hear loud sound from engines in 1.4T cars. D2 is caused by the defect of turbines or nozzles. And D2 often has abnormal sound and strenuous to start.
D3: Dashboard defects	turn off, cannot see, dealer	electric wire, dashboard, screen	short circuit, out of control, beep	Customers usually difficult to turn off or see the information on screens clearly. They are likely to ask dealers for help. And D3 is often caused by electric wire defects due to the short circuit.
D4: Fuel leakage	just bought a week, fuel consumption	nozzle, cv joint, pipe	sealing, petrol smell, leakage	D4 often happens in new cars and cause abnormal fuel consumption. CWs and SWs indicate D4 usually caused by nozzles, cv joints or fuel pipes with symptoms of poor sealing or leakage.
D5: Abnormal sound when driving	dealer, front, complaint	body, wiper bar, windshield	shuddering, whistle, abnormal sound	Customers suffer D5 usually to complain to dealers and they often hear the abnormal sound in the front of cars especially in wiper bars and windshields. D5 may occur with car shuddering.
D6: Console defects	fuel consumption, auto, dealer	locks in console, screen, switch	resonance, crack, water leakage	D6 often occurs in AT cars with an abnormal display of fuel consumption. It usually includes problems of console locks, screens

				and switches which has symptoms of resonance, crack, and water leakage.
D7: Fuel caps cannot be closed or opened	warranty, worry, complaint	fuel cap, fuel pipe, cv joint	leakage, burst, damage	Customers suffering D7 usually worry about whether the warranty can cover repair fees in dealers. And D7 is often caused by the failure (like leakage, burst or damaged) of fuel caps, fuel pipes and cv joints.
D8: Abnormal sound of the interior	liquid crystal, 1.3T, dealer	glove compartment, air intake, right door	loud, rattling, squeak	D8 often happens in 1.3T cars and the loud, rattling or squeaking sound usually occurs in glove compartments, air intakes or right doors.
D9: Underpowered vehicles	replace, suddenly, sometimes	electricity meter, radiator, valve	underpowered, overheating, smoking	D9 happens sometimes and suddenly for defects of electricity meters, radiators or valves which may lead to the overheating of engines and smoking in the exhaust-pipes.
D10: Transmission defects	slow down, AT, starting	cylinder, transmission, transmission gear	out of gear, liquid leakage, insensitivity	D10 usually occurs when AT cars slow down or start. D10 is caused by defects (like leakage) of cylinders, transmissions and transmission gears which lead to the insensitivity and out of gear.

Defects	GWs	CWs	SWs	Defect-related word illustration
D11: Shifting problems	problem, driving, slow down	control lever, pedal, transmission gear	shudder, unsmooth, stuck	D11 often occurs when driving or slowing. And customers suffer the unsmooth or stuck shifting due to defects of control levers, pedals or transmission gears. D11 also happens with a car shudder.
D12: Abnormal sound when braking	maintain, reason, safety	brake disc, ABS, rear-wheel	insensitive, rattling, noise	Customers ask for reasons of D12 in forums for safety and choose to maintain cars. D12 is caused by braking disc or ABS problems and often happens in rear-wheels with noise or insensitive response.
D13: Car shudder	annoying, braking, driving	fuel pump, engine cabin, rear seat	smoking, shudder, underpowered	Customers often suffer from D13 when braking or driving and feel the shudder in the engine cabin or rear

				seat. D13 may be caused by damaged fuel pumps with symptoms of smoking and underpowered.
D14: Electronic device defects	1.4T, starting, inaccurate	temperature sensor, control button, solenoid valve	dim, noise, insensitivity	D14 usually happens in 1.4T cars when staring with symptoms of insensitive, inaccurate, noise and dim. D14 is caused by defects of temperature sensors, control buttons or solenoid valves.
D15: Smoking	protect rights, sign, starting	valve, alarm, fuel pump	fuel stain, smoking, misfire	Customers suffering D15 often ask how to protect their rights. D15 is usually caused by defects of valves or fuel pumps and raise to the alarm. Cars with D15 often have fuel stains, smoking or misfire.
D16: Water leakage	long time, repairman, replace	window, chassis, door	water leakage, spray, dripping	D16 is a long-term defect that often happens in windows, chassis or doors. Customers bearing D16 usually ask repairmen to replace defective components.
D17: Air conditioning defects	local, check, stop	cooling system, fan, air conditioning	heating, wind leakage, poor	D17 is usually caused by defects of cooling systems or fans and has symptoms of the poor performance of heating or wind leakage. Customers ask for local repair shop recommendation to check their cars.
D18: Difficult to start	embarrass, suddenly, 100 km	engine, transmission, power system	cannot start, stall, lights always on	D18 often happens suddenly on cars whose mileage over 100 km and is usually caused by engine, transmission or power system problems. D18 is prone to make cars stall and alarm lights always on.
D19: Window defects	maintenance, disappointed, design	glass, elevator, sunroof	stuck, abnormal sound, leakage	D19 often makes customers disappointed who regard D19 as the design defect. And D19 happens in car windows or sunroofs with symptoms of window stuck, abnormal sound and wind/water leakage.
D20: Door	common	door, hinge,	cannot open,	D20 is a common failure of the

defects	failure, cause, open	switch	too tight, abnormal sound	Sagitar model and is usually caused by problems of hinges or switches. Doors with D12 are often too tight to be opened and have abnormal sound.

*GWs: General words, CWs: Component words, SWs: Symptom words. “T” in 1.3T and 1.4T means turbocharged straight injection engines

References

[1] M. Alkahtani, A. Choudhary, A. De, J.A. Harding, A decision support system based on ontology and data mining to improve design using warranty data, Computers & Industrial Engineering, (2018).

[2] Y. Liu, C. Jiang, H. Zhao, Using contextual features and multi-view ensemble learning in product defect identification from online discussion forums, Decision Support Systems, 105 (2018) 1-12.

[3] A.S. Abrahams, J. Jiao, G.A. Wang, W. Fan, Vehicle defect discovery from social media, Decision Support Systems, 54 (2012) 87-97.

[4] A. Zavala, J.E. Ramirez-Marquez, Visual analytics for identifying product disruptions and effects via social media, International Journal of Production Economics, 208 (2019) 544-559.

[5] M. Winkler, A.S. Abrahams, R. Gruss, J.P. Ehsani, Toy Safety Surveillance from Online Reviews, Decision Support Systems, 90 (2016) 23-32.

[6] D. Law, R. Gruss, A.S. Abrahams, Automated defect discovery for dishwasher appliances from online consumer reviews, Expert Systems with Applications, 67 (2017) 84-94.

[7] D.M. Goldberg, A.S. Abrahams, A Tabu search heuristic for smoke term curation in safety defect discovery, Decision Support Systems, 105 (2018) 52-65.

[8] A.S. Abrahams, J. Jiao, W. Fan, G.A. Wang, Z. Zhang, What's buzzing in the blizzard of buzz? Automotive component isolation in social media postings, Decision Support Systems, 55 (2013) 871-882.

[9] X. Zhang, S. Niu, D. Zhang, G.A. Wang, W. Fan, Predicting Vehicle Recalls with User-Generated Contents: A Text Mining Approach, in: Intelligence and Security Informatics: Pacific Asia Workshop, PAISI 2015, Ho Chi Minh City, Vietnam, 2015, pp. 41-50.

[10] X. Zhang, Z. Qiao, L. Tang, P.W. Fan, Edward Fox, A.G. Wang, Identifying product defects from user complaints a probabilistic defect model, in: The 22nd Americas Conference on Information Systems (AMCIS), 2016.

[11] Q. Liang, K. Wang, Monitoring of user-generated reviews via a sequential reverse joint sentiment-topic model, Quality and Reliability Engineering International, 35 (2019) 1180-1199.

[12] X. Zhang, Z.L. Qiao, A. Ahuja, W.G. Fan, E.A. Fox, C.K. Reddy, Discovering Product Defects and Solutions from Online User Generated Contents, Web Conference 2019: Proceedings of the World Wide Web Conference (Www 2019), (2019) 3441-3447.

[13] W. He, H. Wu, G. Yan, V. Akula, J. Shen, A novel social media competitive analytics framework with sentiment benchmarks, Information & Management, 52 (2015) 801-812.

[14] J. Jin, P. Ji, R. Gu, Identifying comparative customer requirements from product online reviews for competitor analysis, Engineering Applications of Artificial Intelligence, 49 (2016) 61-73.

[15] Y. Liu, J.-W. Bi, Z.-P. Fan, Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory, Information Fusion, 36 (2017) 149-161.

[16] Y. Liu, C. Jiang, H. Zhao, Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media, Decision Support Systems, 123 (2019) 113079.

[17] J. Jin, P. Ji, Y. Liu, S.C. Johnson Lim, Translating online customer opinions into engineering characteristics in QFD: A probabilistic language analysis approach, Engineering Applications of Artificial Intelligence, 41 (2015) 115-127.

[18] Y.-H. Hsiao, M.-C. Chen, W.-C. Liao, Logistics service design for cross-border E-commerce using Kansei engineering with text-mining-based online content analysis, Telematics and Informatics, 34 (2017) 284-302.

[19] W.M. Wang, Z. Li, Z.G. Tian, J.W. Wang, M.N. Cheng, Extracting and summarizing affective features and responses from online product descriptions and reviews: A Kansei text mining approach, Engineering Applications of Artificial Intelligence, 73 (2018) 149-162.

[20] N. Korfiatis, P. Stamolampros, P. Kourouthanassis, V. Sagiadinos, Measuring service quality from unstructured data: A topic modeling application on airline passengers'

online reviews, Expert Systems with Applications, 116 (2019) 472-486.

[21] X. Xu, Y. Li, The antecedents of customer satisfaction and dissatisfaction toward various types of hotels: A text mining approach, International Journal of Hospitality Management, 55 (2016) 57-69.

[22] X. Xu, X. Wang, Y. Li, M. Haghighi, Business intelligence in online customer textual reviews: Understanding consumer perceptions and influential factors, International Journal of Information Management, 37 (2017) 673-683.

[23] Y. Zhao, X. Xu, M. Wang, Predicting overall customer satisfaction: Big data evidence from hotel online textual reviews, International Journal of Hospitality Management, 76 (2019) 111-121.

[24] M. Hur, P. Kang, S. Cho, Box office forecasting based on sentiments of movie reviews and Independent subspace method, Information Sciences, 372 (2016) 608-624.

[25] J. Du, H. Xu, X. Huang, Box office prediction based on microblog, Expert Systems with Applications, 41 (2014) 1680-1689.

[26] X. Fang, P.J.-H. Hu, Top Persuader Prediction for Social Networks, MIS Quarterly, 42 (2018) 63-82.

[27] L. Liu, J. Tang, J. Han, S. Yang, Learning influence from heterogeneous social networks, Data Mining and Knowledge Discovery, 25 (2012) 511-544.

[28] A.S. Abrahams, W. Fan, G.A. Wang, Z.J. Zhang, J. Jiao, An Integrated Text Analytic Framework for Product Defect Discovery, Production and Operations Management, 24 (2015) 975-990.

[29] K.D. Kuhn, Using structural topic modeling to identify latent topics and trends in aviation incident reports, Transportation Research Part C: Emerging Technologies, 87 (2018) 105-122.

[30] Z. Zhang, Q. He, J. Gao, M. Ni, A deep learning approach for detecting traffic accidents from social media data, Transportation Research Part C: Emerging Technologies, 86 (2018) 580-596.

[31] A. Kinoshita, A. Takasu, J. Adachi, Real-time traffic incident detection using a probabilistic topic model, Information Systems, 54 (2015) 169-188.

[32] S. Lo, Web service quality control based on text mining using support vector machine, Expert Systems with Applications, 34 (2008) 603-610.

[33] R. Gruss, A.S. Abrahams, W. Fan, G.A. Wang, By the numbers: The magic of numerical intelligence in text analytic systems, Decision Support Systems, 113 (2018) 86-98.

[34] T.-H. Chen, W. Shang, M. Nagappan, A.E. Hassan, S.W. Thomas, Topic-based software defect explanation, Journal of Systems and Software, 129 (2017) 79-106.

[35] Q. Le, T. Mikolov, Distributed Representations of Sentences and Documents, in: Proceeding ICML'14 Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, 2014, pp. 1188-1196.

[36] S. Zoghbi, I. Vulić, M.-F. Moens, Latent Dirichlet allocation for linking user-generated content and e-commerce data, Information Sciences, 367-368 (2016) 573-599.

[37] J. Pennington, R. Socher, C.D. Manning, Glove: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.

Highlights

1) A novel probabilistic graphic model named Product Defect Detection Model (PDDM) is proposed to reveal product defects based on social media data.
2) PDDM exposes the defects buried in social media data and provides detailed defect information without the deficiency of huge labor investment.
3) PDDM is validated to outperform the existing methods of defect discovery based on social media data.

Lu Zheng: Conceptualization, Methodology, Software, Visualization, Writing - Original Draft

Zhen He: Writing - Review & Editing, Validation, Funding acquisition

Shuguang He: Data Curation, Software

Lu Zheng is a Ph.D. student in the Department of Management Science and Engineering at College of Management and Economics at Tianjin University. Her current research interests include social media data, business intelligence and analytics and product defect detection.

Zhen He is a professor in the Department of Management Science and Engineering at College of Management and Economics at Tianjin University. Dr. He's primary research interest is quality management. He is the Changjiang Scholarship Distinguished Professor and the academician of International Academy for Quality.

Shuguang He is a professor in the Department of Management Science and Engineering at College of Management and Economics at Tianjin University. Dr. He's primary research interest is supply chain management and warranty policy design.