Recognizing Zombie Fans in Microblog Based on Bayesian Theory

Yiming Ding

Supervisor: Chengpeng Zhang

No2. High School of East China Normal University

Abstract

As the Microblog experiences an explosive growth in popularity, it plays an increasingly important role in various fields of science and economics. Recently, a rampant spread of zombie fans impedes the development of Microblog. Zombie fans are malicious accounts artificially created in large quantities and manipulated by computer programs. They detract the reliability of the information in Microblog, as well as damage the Internet security. However, most of existing methods to recognize zombie fans are not effective enough and can hardly adjust to the rapid evolution of these artificial followers. Therefore, in this paper, seven zombie fan recognition models with different sample characteristics are established based on Bayesian Theory. Then, the model with best recognition capability is selected based on ROC Curve and AUC, with its performance tested by examples with Sina Microblog data. Finally, a MATLAB user interface is created. The numerical results indicate that the recognition model with combined sample characteristics of icon, following number, follower-following ratio and following-tweet ratio shows the best recognition performance, whose recall and precision reach 92.37% and 98.37% respectively under the threshold of 0.7.

Key words: Microblog, zombie fans, Bayesian Theory, recognition model

Contents

1. Introduction

1.1 Background

With the wide spread and rapid development of Internet, Microblog has experienced an explosive growth in popularity home and abroad. As reported in March 2013, Sina Microblog, the top Microblog in China, has attracted more than 500 million registered users, neck and neck with Twitter, the most influential Microblog in the world. Thus, considered as a network platform for instant communication, a new approach for marketing, and the basis of social studies, Microblog plays important roles in various fields of economics and science.

In Microblog, a fan is defined as a follower who pays constant attention to a certain user. Therefore, it is widely acknowledged that the more fans a user possesses, the more influential he is. The consensus, however, provokes a contest in number of fans between users, thus leading to the business of zombie fans[1].

Zombie fans, created in large quantities, aim at either advertising or increasing follower number of certain users. They are manipulated by computer programs, thus possessing no human emotion. As a result, the fake credit, popularity and information caused by zombie fans detract the reliability of the information in Microblog, pose negative effect on the credibility and integrity of Microblog marketing, as well as decelerate the development of burgeoning online social platform.

According to the zombie fans sellers on Taobao, there are four levels of zombie fans available for Sina Microblog: 1)primary zombie fans, which have no icon, followers or tweets (Microblog articles), easily detected by existing methods adopted; 2) junior zombie fans, which do have icons but no followers or tweets, partly detected by existing methods adopted; 3)senior zombie fans which have icon, fans and tweets, hardly detected by existing methods adopted; 4) top zombie fans, which resemble real users, even partly authorized by the official.

Although actions have been taken to cope with the rampant zombie fans[2], those artificial followers are still running their course for most methods employed are merely effective to ones with obvious clues but ineffectual to those well disguised. Moreover, adopting simple criteria to distinguish zombie fans, these methods are likely to mistake newly registered users for zombie fans[3].

Therefore, in this paper, we build mathematical models based on Bayesian theory to recognize zombie fans. The models can be applied to practice after studying the samples of real users and zombie fans, achieving both theoretical and practical significance.

1.2 Related Work

To cope with the spread of zombie fans, both officials and non-officials parties have taken actions. However, most of methods are only effectual in recognizing zombie fans with apparent traits, and bear high possibility of erroneous judgment. Thus, zombie fans have aroused the interest of scholars home and abroad, who attempt to solve the problems caused by zombie fans from different aspects.

Based on the properties of Microblog in China, a variety of proposals are put forward. Ref. [1] defines and classifies zombie fans, analyzes their behavior characteristics and propounds fundamental suggestions on zombie fans recognition. Ref. [4] proposes a method to reduce the impact of zombie fans according to the users' existing form and the characteristics of relationship between them. Ref. [5] proposes an intelligent classification method based on the characteristics of registered usernames, but it does not make full use of information other than usernames. Ref. [6] builds a classifier based on support vector machine (SVM) combined with the features selected from the user profile and tweets, and the accuracy of this method reaches 91%. However, the SVM algorithm has difficulty to train a large number of samples, which limits its capability of the application.

Scholars abroad have acquired more significant achievements in the study of zombie fans. Based on Twitter, Ref. [7] establishes a complete system to classify human (real users), bots (users controlled by automatic programs), and cyborg (users partly controlled by automatic programs) in terms of tweeting behavior, tweet content, and account properties. However, the properties of Twitter significantly differs from those of Microblogs widely used in China in various aspects such as review function, repost mode, etc.

Bayesian Theory is one of the basic statistical methods, which reduces uncertainty by sampling and revising priori information[8]. It is widely applied in fields of information classification, prediction and decision for its capability of information fusion. In spam filtering, for example, the Bayesian method proves to be effective compared with other existing methods[9].

In summary, there is a lack of effectual methods for recognizing Microblog zombie fans especially in China. Therefore, in this paper, we build models based on Bayesian Theory to recognize zombie fans and validate the models with numerical analysis.

1.3 Bayesian Theory

Bayesian Theory[8] is one of the basic statistical methods, which predicts the posterior probability by using the prior probability and the sample data. Thus Bayesian theory is a method that can predict the future by the past, whose process resembles the updating and fusion of knowledge and information.

Assume that n events do not interfere with one another and N is another event. According to Bayesian Theory, we have

$$P(M_i \mid N) = \frac{P(N \mid M_i) \cdot P(M_i)}{P(N)} \tag{1}$$

In formula (1), $P(M_i)$ $(i = 1, 2, \cdots n)$ denotes the prior probability of the occurrence of event $M_i (i = 1, \cdots, n)$. $P(N \mid M_i)$ denotes the conditional probability that event N occurs given that event M_i has occurred, which is called likelihood probability and shows the sample information. $P(N)$ denotes the probability of the occurrence of event N, where $P(N)=\sum_{i=1}^{n}P(N\mid M_i)\cdot P(M_i)$. $P(M_i\mid N)$ denotes the conditional probability that event M_i occurs given that event N has occurred. It's the posterior probability of the occurrence of M_i, which results from the combination of prior probability and likelihood probability.

2. Modeling

There are essential differences between characteristic of zombie fans and real users, for zombie fans are created for certain purposes and manipulated by computer programs. In order to model based on real data, we collect samples of zombie fans by purchasing them from online fans sellers, and collect samples of real users with the help of online questionnaire. Finally, we obtain 1425 samples of zombie fans and 413 samples of real users in Sina Microblog. By observing the data carefully, we find that among the users' fundamental information, such as icon, following number, follower number and tweet number, there is a significant difference between zombie fans and real users in icon, following number, follower-following ratio. Ref. [1] discovers a significant difference in following-tweet ratio and tweet-follower ratio. Therefore, among icon, following number, follower-following ratio, following-tweet ratio and tweet-follower ratio, we select different sample characteristics and build seven recognition models based on Bayesian theory. Then we compare the effectiveness of the seven models adopting ROC curve and AUC as criteria, select the most effective one and test its recognition performance.

2.1 Notations

（1）Define T as the sample set of real users, F as the sample set of zombie fans;

（2）Classify icon into a groups, denoted by $H_1,H_2,\cdots,H_a$;

（3）Classify following number into b groups, denoted by $G_1, G_2, \cdots, G_b$;

（4）Classify follower-following ratio into c groups, denoted by $A_1, A_2, \cdots, A_c$;

（5）Classify following-tweet ratio into d groups, denoted by $B_1, B_2, \cdots, B_d$;

（6）Classify tweet-follower ratio into e groups, denoted by $C_1, C_2, \cdots, C_e$.

2.2 Assumptions

Based on real data collected, we analyze the correlation between icon, following number, follower-following ratio, following-tweet ratio and tweet-follower ratio. The result is shown in Table 1.

Table 1：Correlation Matrix

	Icon	Following number	Follower-following ratio	Following-tweet ratio	Tweet-follower ratio
Icon	1. 0000	-0. 3328	0. 0641	0. 1719	0. 0305
Following number	-0. 3328	1. 0000	-0. 0632	-0. 1670	0. 1074
Follower-following ratio	0. 0641	-0. 0632	1. 0000	-0. 0166	-0. 0365
Following-tweet ratio	0. 1719	-0. 1670	-0. 0166	1. 0000	-0. 1863
Tweet-follower ratio	0. 0305	0. 1074	-0. 0365	-0. 1863	1. 0000

From Table 1, we find that the five sample characteristics—icon, following number, follower-following ratio, following-tweet ratio, tweet-follower ratio—are barely related. Therefore, it is reasonable to assume that the five sample characteristics are independent of each other.

2.3 Zombie Fans Recognition Models Based on Bayesian Theory

Based on Bayesian Theory, we have:

$$P(F \mid X) = \frac{P(X \mid F) \cdot P(F)}{P(X \mid F) \cdot P(F) + P(X \mid T) \cdot P(T)} \qquad (2)$$

$$P(T \mid X) = \frac{P(X \mid T) \cdot P(T)}{P(X \mid F) \cdot P(F) + P(X \mid T) \cdot P(T)} \tag{3}$$

Formula (2) and (3) calculate the posterior probability that a target user with sample characteristic X belongs to zombie fans and to real users respectively. $P(F), P(T)$ denote the prior probability that the user belongs to zombie fans and to real users respectively; $P(X \mid F), P(X \mid T)$ denote the likelihood probability of the user belonging to zombie fans and to real users respectively.

Then we select different sample characteristics to build seven zombie fans recognition models. The sample characteristics of seven models are shown in Table 2.

Table 2：Sample Characteristics of Seven Models

Model	Sample Characteristics
Model 1	icon, following number, follower-following ratio
Model 2	icon, following number, following-tweet ratio
Model 3	icon, following number, tweet-follower ratio
Model 4	icon, following number, follower-following ratio, following-tweet ratio
Model 5	icon, following number, follower-following ratio, tweet-follower ratio
Model 6	icon, following number, following-tweet ratio, tweet-follower ratio
Model 7	icon, following number, follower-following ratio, following-tweet ratio, tweet-follower ratio

Here, we take Model 1 as an example to illustrate how to calculate the likelihood probability.

In Model 1, as for a target user with sample characteristics $X_1 = \{H_i, G_j, A_k\}, \quad i \in \{1, 2, \cdots, a\}, j \in \{1, 2, \cdots, b\}, k \in \{1, 2, \cdots, c\}$, because the three sample characteristics— icon, following number , follower-following ratio—are independent of each other according to assumptions, we have：

$$P(X_1 \mid F) = P(H_i \mid F) \cdot P(G_j \mid F) \cdot P(A_k \mid F) \tag{4}$$

$$P(X_1 \mid T) = P(H_i \mid T) \cdot P(G_j \mid T) \cdot P(A_k \mid T) \tag{5}$$

In formula(4) and (5), $P(H_i \mid F)$, $P(H_i \mid T)$ denote the probability that a user's icon belongs to H_i in sample set of zombie fans and real users respectively; $P(G_j \mid F)$, $P(G_j \mid T)$ denote the probability that a user's following number belongs to G_j in sample set of zombie fans and real users respectively; $P(A_k \mid F)$, $P(A_k \mid T)$ denote the probability that a user's follower-following ratio belong to A_k in sample set of zombie fans and real users respectively.

Let $X = X_1$ in formula (2). By substituting (4), (5) into (2), (3), we can calculate the posterior probability $P(F \mid X_1)$ that a user belongs to zombie fans and the posterior probability $P(T \mid X_1)$ that a user belongs to real users.

Similarly, in Model 2, 3, 4, 5, 6 and 7, we can calculate posterior probability $P(F \mid X_t)$ that a target user belongs to zombie fans and the posterior probability $P(\mathrm{T} \mid X_t)$ that the user belongs to real users, where $t = 2, 3, \cdots 7$.

2.4 Threshold Selection Based on Risk Minimization

In the recognition model based on Bayesian theory, we have to determine whether to recognize the user as a zombie fan after calculating its posterior probability of being a zombie fan. Thus, we introduce a 'threshold'. In other words, if the user's posterior probability of being a zombie fan is no less than the threshold $\lambda, 0 \le \lambda \le 1$, we recognize the user as a zombie fan.

Here we explain how to select the threshold λ based on risk minimization[10].

As for each classification progress, there's a possibility of misjudgment. People have to bear the loss caused by misjudgment and expect minimum loss. Thus, we introduce the loss factor to show the severity of two kinds of misjudgment. Denote the loss factor of mistaking zombie fans for real users as ω_{FT};denote the loss factor of mistaking real users for zombie fans as ω_{TF}, where $\omega_{FT}, \omega_{TF} \in [0,1]$, and

$$\omega_{FT} + \omega_{TF} = 1. \tag{6}$$

As a correct judgment results in no loss, the expected loss caused by recognizing a user with sample characteristics X as a zombie fan is

$$R(F \mid X) = \omega_{TF} \cdot P(T \mid X). \tag{7}$$

Similarly, the expected loss caused by recognizing a user with sample characteristics X as a real user is

$$R(T \mid X) = \omega_{FT} \cdot P(F \mid X). \tag{8}$$

When

$$R(F \mid X) \leq R(T \mid X), \tag{9}$$

the expected loss caused by recognizing the user as a zombie fans is less severe than that caused by recognizing the user as a real user. Therefore, according to risk minimization, the user should be categorized into zombie fans to achieve minimum expected loss.

Moreover,

$$P(F \mid X) + P(T \mid X) = 1. \tag{10}$$

By substituting (6), (7), (8), (10) into (9), we find that when $P(F \mid X) \geq \omega_{TF}$, the user with sample characteristics should be classified as zombie fans. Therefore, ω_{TF} equals to the threshold λ numerically.

From the analysis above, we can infer that $\omega_{TF} = 0.5$, namely $\lambda = 0.5$ demonstrate that two kinds of misjudgments are considered with equal severity. Moreover, the larger ω_{TF} is, namely the larger λ is, the more unacceptable it is to mistake real users for zombie fans. Reversely, the smaller ω_{TF} is, namely the smaller λ is, the more unacceptable it is to mistake zombie fans for real users.

2.5 The evaluation of models

In order to select the most effective zombie fans recognition model based on its performance, we adopt ROC Curve (Receiver Operating Characteristic Curve) and AUC (Area Under the ROC Curve)[12-13] as criteria. After that, in order to test the validity of the selected model, we adopt Recall, Precision and $F-measure$ as criteria.

Here we illustrate the evaluation methods by introducing the true class and hypothesized class of a set of users as is shown in Table 3.

Table 3：The true class and hypothesized class of a set of users

Hypothesized Class **True Class**	**Zombie Fans**	**Real users**
Zombie fans	$\mathbf{N_{FF}}$	$\mathbf{N_{FT}}$
Real users	$\mathbf{N_{TF}}$	$\mathbf{N_{TT}}$

In Table 3, $\mathbf{N_{FF}}$ denotes the number of zombie fans that are correctly recognized, $\mathbf{N_{FT}}$ denotes the number of zombie fans that are recognized as real users mistakenly, $\mathbf{N_{TT}}$ denotes the number of real users that are correctly recognized and $\mathbf{N_{TF}}$ denotes the number of real users that are recognized as zombie fans mistakenly.

2.5.1 ROC Curve and AUC

An ROC Curve depicts relative trade-offs between benefits and costs of the recognition process, thus evaluating the performance of a recognition model. In ROC graphs, $TPR(\text{True Positive Rate})$ is plotted on the Y axis and $FPR(\text{False Positive Rate})$ is plotted on the X axis, where

$$TPR = \frac{N_{FF}}{N_{FT} + N_{FF}} \tag{11}$$

$$FPR = \frac{N_{TF}}{N_{TT} + N_{TF}} \tag{12}$$

In formula (11), (12), TPR denotes the proportion of zombie fans that are correctly recognized and FPR denotes the proportion of zombie fans that are mistakenly recognized. Thus, the higher the TPR and the lower the FPR, the better the performance.

As for the recognition models built above, we vary the threshold λ to obtain different combination of TPR and FPR, so as to plot the ROC Curve in the unit square. Qualitatively, the more northwest an ROC Curve locates, the more effective the corresponding model is. As to evaluate the effectiveness of the model quantificationally, we adopt the Area under the ROC Curve, abbreviated AUC. The bigger AUC is, the more effective the corresponding model is.

2.5.2 Recall, Precision and F-measure

Recall $r = \frac{N_{FF}}{N_{FF} + N_{FT}}$ evaluates the ability of a model to recognize zombie fans thoroughly. Therefore, the higher r is, the less zombie fans are left undetected. On the other hand, Precision $p = \frac{N_{FF}}{N_{TF} + N_{FF}}$ evaluates the ability of a model to recognize zombie fans accurately. Therefore, the higher p is, the less true users are mistaken for zombie fans. $F - measure = \frac{2pr}{p+r}$ is the harmonic mean of recall and precision, taking both aspects above into consideration.

3. Model Testing and Analysis

We set $P(F) = P(T) = 0.5$ for there is little knowledge about the exact quantity of zombie fans in Sina Microblog. That is to assume that the prior probability of the appearance of zombie fans and real users are even.

Other parameters set in the numerical examples are listed below.

（1）$a = 3$. We classify icon into 3 groups: icons of human, other icons and unset icons denoted by H_1, H_2, H_3 respectively；

（2）$b = 20$. We classify following number (the maximum number of following number is 2000 in Sina Microblog) into 20 groups denoted by

$G_1, G_2, \ldots\ldots, G_{20}$, where

$$100(j-1) \le G_j < 100j, \quad 1 \le j \le 20,\ j \in Z\text{(integers)}.$$

（3） $c = 13.$ We classify follower-following ratio into 13 groups denoted by

$A_1, A_2, \ldots\ldots, A_{13}$, where

$$\begin{cases} 0.1(k-1) \le A_k < 0.1k, \quad 1 \le k \le 10, k \in Z; \\ k-10 \le A_k < k-9, \quad 11 \le k \le 12, k \in Z; \\ A_k \ge k-10, \quad k = 13. \end{cases}$$

（4） $d = 20.$ We classify following-tweet ratio into 20 groups denoted by

$B_1, B_2, \ldots\ldots, B_{20}$, where

$$\begin{cases} 0.1(l-1) \le B_l < 0.1l, \quad 1 \le l \le 10, l \in Z; \\ l-10 \le B_l < l-9, \quad 11 \le l \le 19, l \in Z; \\ B_l \ge l-10, \quad l = 20. \end{cases}$$

（5） $e = 20.$ We classify tweet-follower ratio into 20 groups denoted by

$C_1, C_2, \ldots\ldots, C_{20}$, where

$$\begin{cases} 0.1(m-1) \le C_m < 0.1m, \quad 1 \le m \le 10, m \in Z; \\ m-10 \le C_m < m-9, \quad 11 \le m \le 19, m \in Z; \\ C_m \ge m-10, \quad m = 20. \end{cases}$$

In this part of the paper, we use MATLAB[14] for programming, select the most effective model and test its performance based on samples of Sina Microblog.

3.1 Evaluation of models based on ROC Curve and AUC

Here, we test the models adopting 10-fold cross-validation [15]. In this method, the sample is partitioned into 10 subsamples, of which a single subsample is retained as the unclassified users for model testing, and the remaining 9 subsamples are used for model training. An algorithm in Ref. [13] is adopted to generate ROC Curves of the models as is shown in Fig 1. And AUC of each model is calculated as is shown in Fig 2.

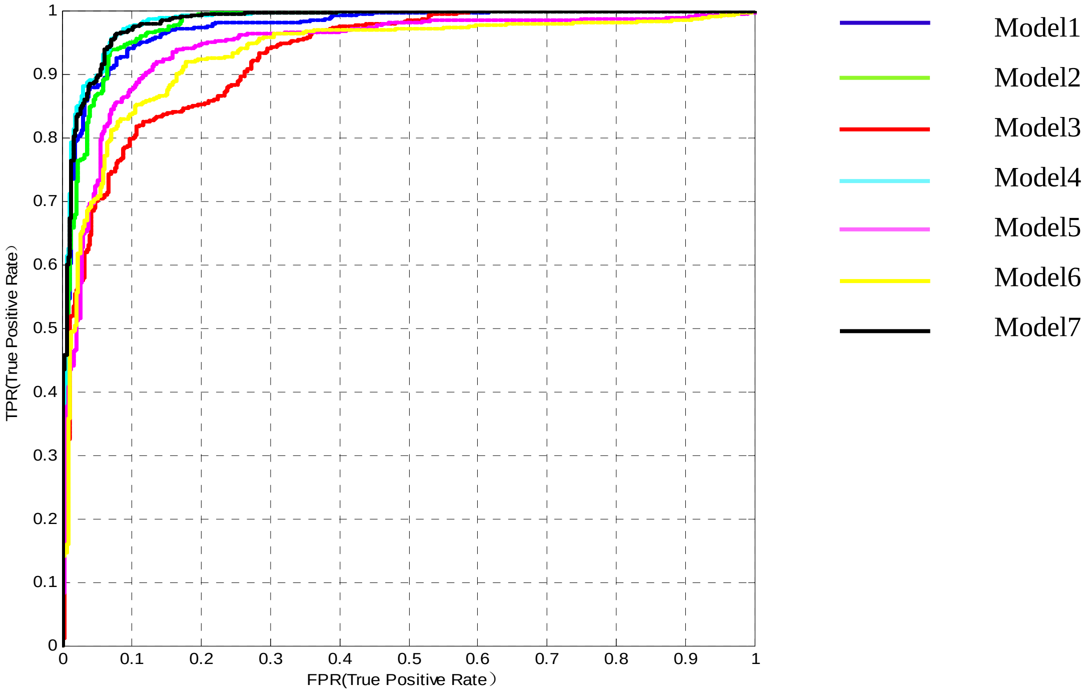


Fig 1: ROC Curve of 7 recognition models

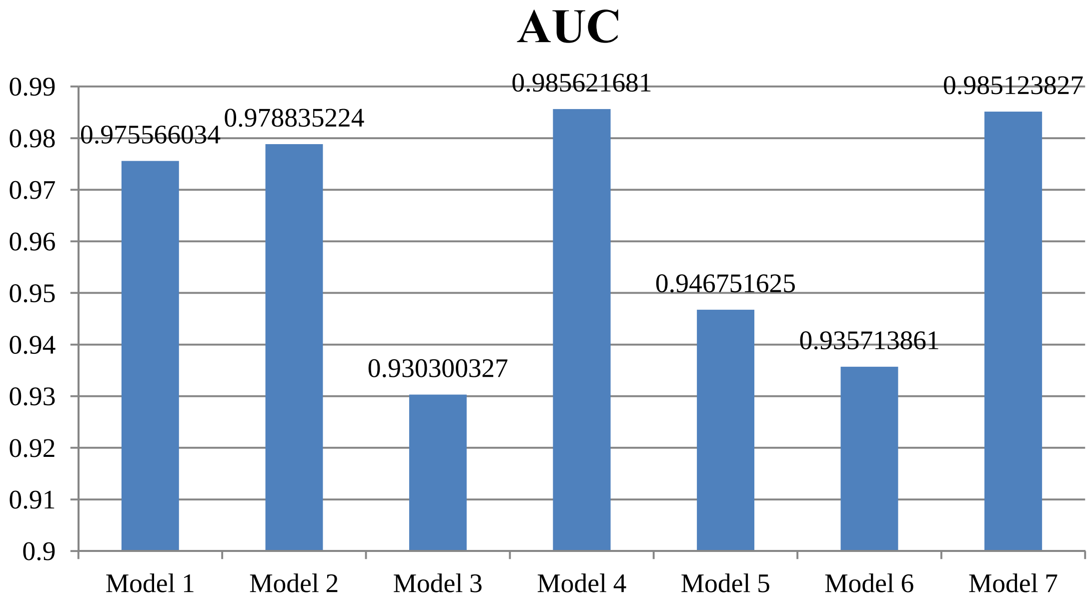


Fig 2: AUC of 7 recognition models

According to Fig 1 and Fig 2, we have:

（1）Model 4 (with sample characteristics of icon, following number, follower-following ratio, following-tweet ratio) is the most effective for its AUC is the largest among the seven models. And Model 7 (with sample characteristics of icon, following number, follower-following ratio, following-tweet ratio, tweet-follower ratio) is the second-best model.

（2）Comparing Model 1 with Model 5, Model 2 with Model 6, and Model 4 with Model 7, although the latter considers one more sample characteristic (tweet-follower ratio), it is less effective than the former. The phenomenon indicates that tweet-follower ratio, as a sample characteristic, does not show the essential difference between zombie fans and real users, thus disturbing the recognition process. Therefore, it can be concluded that the key to effective recognition model is the sample characteristic pertinent to the essential traits of a user rather than the quantity of sample characteristics.

3.2 Model Testing Based on Recall, Precision and F-measure

Generally, people consider that the consequence of mistaking a real user for a zombie fan is less severe than that of mistaking a zombie fan for a real user. Therefore, the threshold is often set at $\lambda \geq 0.5$.

Here, we test the performance of Model 4 with a brand new set of samples. We calculate recall r, precision p and $F-measure$ under different thresholds λ as is shown in Table 4 and Figure 3.

Table 4：Performance of Model 4 under varied threshold λ

Threshold λ	Recall r	Precision p	$F-measure$
0. 5	0. 9389	0. 9462	0. 9425
0. 6	0. 9237	0. 9758	0. 9490
0. 7	0. 9237	0. 9837	0. 9528
0. 8	0. 8473	0. 9823	0. 9098
0. 9	0. 7481	1. 0000	0. 8559

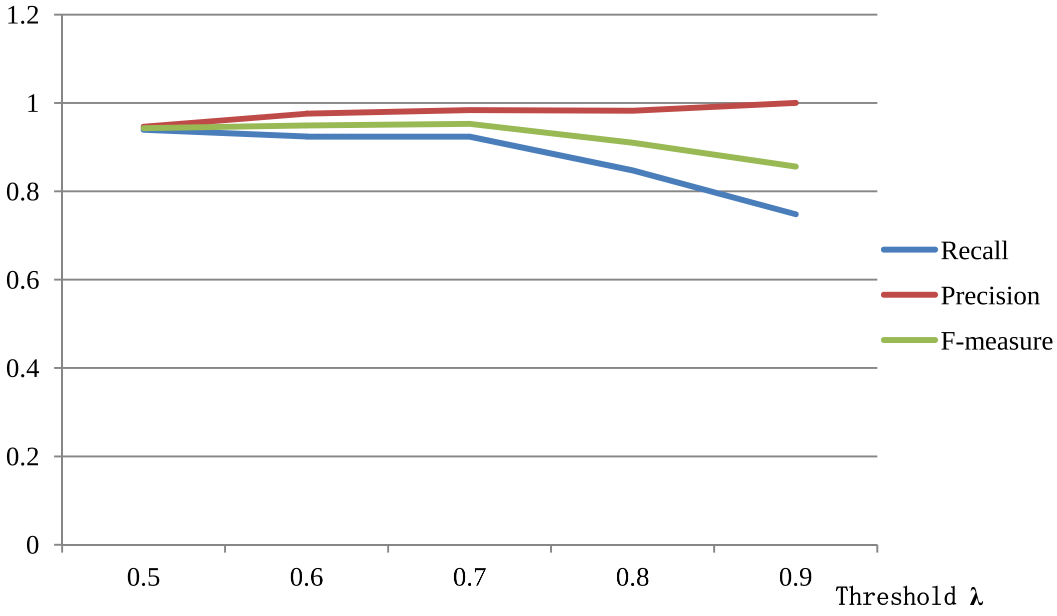


Fig 3: Performance of Model 4 under varied threshold λ

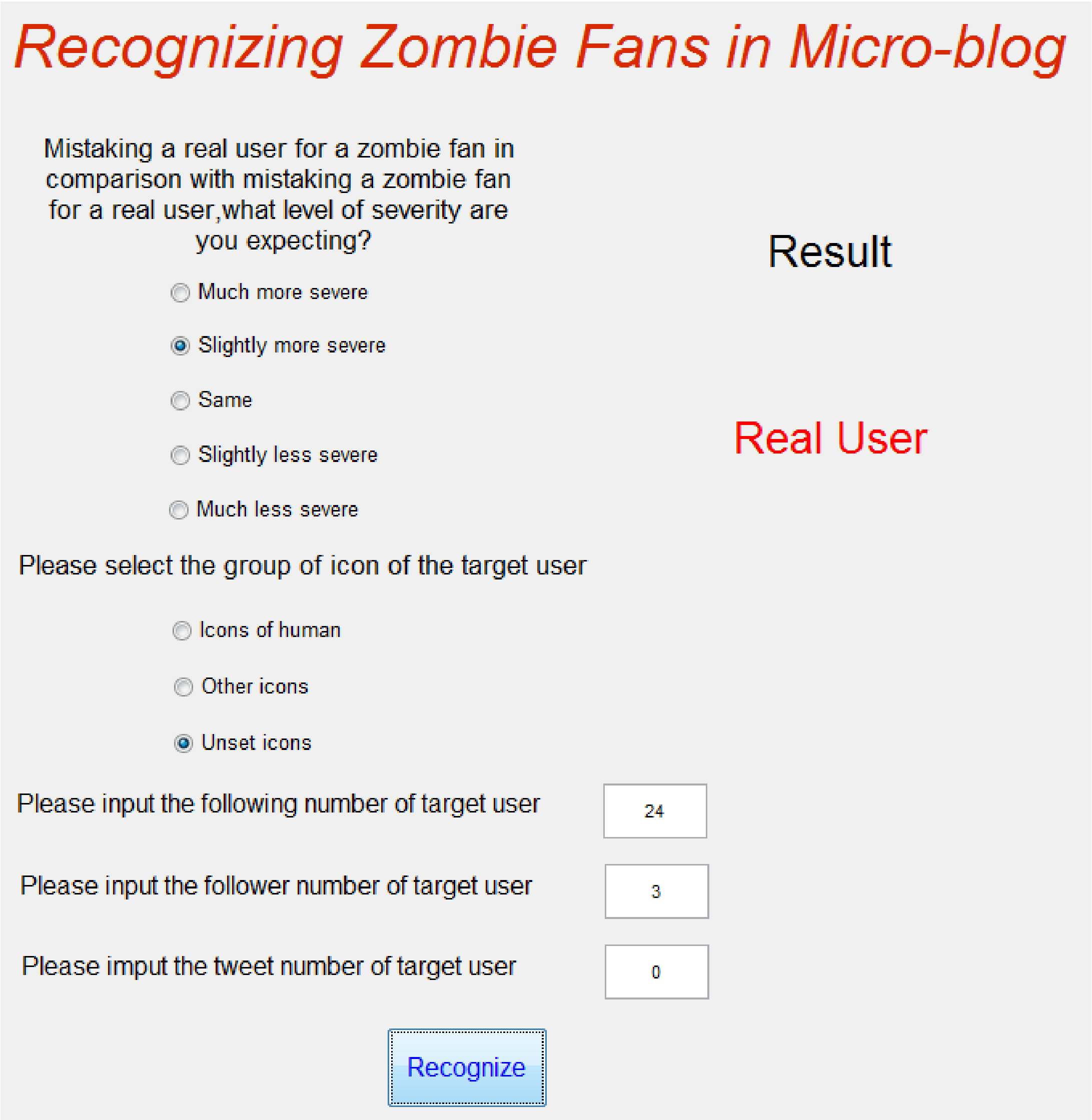
According to Table 1 and Fig 3, we have:

（1）Recall r decreases as threshold λ increases, while precision p increases as threshold λ increases. Therefore, when put into practice, threshold λ should be raised if high precision is required and should be reduced if high recall is required.

（2）$F-measure$, which takes both recall and precision into consideration, reaches its maximum under the threshold $\lambda = 0.7$, when $r = 92.37\%$, $p = 98.37\%$, and $F_1 = 95.28\%$.

4. User Interface

After modeling and model testing, we create a Microblog zombie fans recognizing user interface with GUI (Graphical User Interfaces) in MATLAB as is shown in Fig 4.



Recognizing Zombie Fans in Micro-blog

Mistaking a real user for a zombie fan in comparison with mistaking a zombie fan for a real user,what level of severity are you expecting?

○ Much more severe

● Slightly more severe

○ Same

○ Slightly less severe

○ Much less severe

Please select the group of icon of the target user

○ Icons of human

○ Other icons

● Unset icons

Please input the following number of target user: 24

Please input the follower number of target user: 3

Please imput the tweet number of target user: 0

Result

Real User

Fig 4：A Microblog zombie fans recognizing user interface

Users can recognize zombie fans within the following steps:

(1) Select the severity of two kinds of misjudgment (including mistaking real users for zombie fans and mistaking zombie fans for real users);

(2) Select the group of icon of a target user；

(3) Input following number, follower number and tweet number of a target user；

(4) Click ‘Recognize’ button, and the result will be shown on the right of the interface.

The interface is a clear and user-friendly application based on this paper, and it also meets the demands of users to some extent.

5. Conclusions and Prospects

5.1 Conclusions

With the burgeoning of Microblog come the rampant zombie fans, which pose negative effect on Internet security, as well as decelerate the development of Microblog. However, most of existing methods adopted to cope with zombie fans are not effective enough. Therefore, in this paper, we build zombie fans recognition models based on Bayesian Theory, validate the models with numerical examples and create a user interface.

The paper achieves several important objectives with both theoretical and practical significance:

1) Seven Microblog zombie fans recognition models with different sample characteristics are built and the one with most effectiveness is selected based on ROC Curve and AUC;
2) Sample characteristic pertinent to the essential traits of a user rather than the quantity of sample characteristics is the key to effective recognition model. Based on Sina Microblog, the model with sample characteristics of icon, following number, follower-following ratio, following-tweet ratio is the most effective one. Moreover, tweet-follower ratio, as a sample characteristic, does not show the essential difference between zombie fans and real users, thus disturbing the recognition process.
3) As for the model with sample characteristics of icon, following number, follower-following ratio, following-tweet ratio, recall reaches 92.37% and precision reaches 98.37% under the threshold of 0.7. As recall decreases and precision increases when threshold increases, the threshold should be raised if high precision is required and should be reduced if high recall is required.

5.2 Prospect

In this paper, sample characteristics are limited to fundamental traits including icon, following number, follower number and tweet number. Taking more information of the users into consideration, such as number of comments and number of reposts will improve the effectiveness of the model.

When the model is put into use with the help of computer techniques, data updating in sample bank can be practiced, allowing the model to adjust to the upgrading of zombie fans.

Acknowledgements

Thanks for

the research sites provided by my high school

the guide and support from Mr. Zhang

the technique support from Yu Wang

the support of families

the assistance in programming from Yiyang Guo

the precious advices from Jie Mao

all net friends that filled out my questionnaires

References

[1] Yu Guang, Yu Xuan. Microblog zombie fans behavioral characteristics and recognition method [DB/OL]. [2014-1-14]. http://www.paper.edu.cn.

[2] Wang Jing. Celebrity Microblog closed, zombie fans seized [N]. Youth Daily, 2010-9-13(A13).

[3] Wang Yang. Zombie fans cleaner [CP/OL]. Sina Microblog, [2014-1-14]. http://apps.weibo.com/jiangshifen.

[4] Yuan Fuyong, Feng Jing, Fu Xixi, etc. A method to reduce the impact of zombie fans in Microblog [J]. Analysis and study of information, 2012(5):70-75.

[5] Fang Ming, Fang Yi. A new intelligent recognition method of zombie fan [J]. Computer Engineering, 2013(4):190-193.

[6] Zhang Xiying, Che Xin, Tian Xianyun. A recognition method of zombie fans on Microblog user’s behavior [J], Journal of natural science of Heilongjiang University, 2014(2):250-254.

[7] Zi Chu, Steven Gianvecchio, Haining Wang. Detecting automation of Twitter accounts: Are you a Human, Bot, or Cyborg? [J]. IEEE transactions on dependable and secure computing, 2012.

[8] Yan Maosong. Bayesian risk decision engineering [M]. Tsinghua University Press, 1989.

[9] Zhang Mingfeng, Li Yunchun, Li Wei. Survey of application of Bayesian classifying method to spam filtering [J]. Computer Application, 2005(8):14-19.

[10] Shi Xiajun, Lin yaping, Chen Zhiping. Mail filtering based on the risk minimization Bayes [J]. Computer Science, 2002(8):50-51,46.

[11] Qin Feng, Yang Bo, Chen Zekai. Research on measure criteria in evaluating classification performance [J]. Computer Technology and Development, 2006(10):85-88.

[12] Bradley A P. The use of the area under the ROC curve in the evaluation of

machine learning algorithms[J]. Pattern Recognition Society, 1997, 30(7):1145-1159.

[13] Fawcett T. ROC Graphs :Notes and practical considerations for researchers[R]. Palo Alto, CA: HP Laboratories, 2004.

[14] Zhang Zhiyong, etc. Get a good Command of MATLAB R2011a[M]. Beihang University Press, 2011.

[15] Zhang Xuenong, Zhang Licheng. Comparative analysis of English and Chinese anti-spam filtering based on naive Bayesian [J]. Computer Applications and Software, 2008(8):178-180,228.

Appendix

Appendix1: MATLAB Program for Correlation Analysis

```
A=xlsread('fansall. xls','sheet1','B2:E1839');
label=xlsread('fansall. xls','sheet1','F2:F1839');
a=1838;B=zeros(a,5);
%prepare for the data needed
for p=1:a
   B(p,1)=A(p,1);
   B(p,2)=A(p,2);
   B(p,3)=A(p,3)/(A(p,2)+1);
   B(p,4)=A(p,2)/(A(p,4)+1);
   B(p,5)=A(p,4)/(A(p,3)+1);
end
x=B(:,2);
y=B(:,3);
z=B(:,4);
w=B(:,5);
t=B(:,1);
XY=corrcoef(x,y)
XZ=corrcoef(x,z)
XW=corrcoef(x,w)
YZ=corrcoef(y,z)
YW=corrcoef(y,w)
ZW=corrcoef(z,w)
XT=corrcoef(x,t)
YT=corrcoef(y,t)
ZT=corrcoef(z,t)
WT=corrcoef(w,t)
```

Appendix2: MATLAB Program for Depicting ROC Graph

Program 1： Determine the posterior probability of every user belongs to zombie fan, and prepare for depicting ROC graph.

```
A=xlsread('fansall. xls','sheet1','B2:E1839');
label=xlsread('fansall. xls','sheet1','F2:F1839');
a=1838;B=zeros(a,5);train=zeros;N=[0 0];r=0. 5;u=0.
8;K=zeros(2,2);R=zeros(2,1);Ph=zeros(184,10);label_test=[];
%prepare for the data needed
for p=1:a
   B(p,1)=A(p,1);
   B(p,2)=A(p,2);
   B(p,3)=A(p,3)/(A(p,2)+1);
   B(p,4)=A(p,2)/(A(p,4)+1);
   B(p,5)=A(p,4)/(A(p,3)+1);
end
%divide into groups
C=B;
for p=1:a
   if B(p,2)<2000
   C(p,2)=fix(B(p,2)/100)+1;
   else C(p,2)=20;
   end
   if B(p,3)<1
      C(p,3)=fix(B(p,3)/0. 1)+1;
   else if B(p,3)<2
          C(p,3)=11;
       else if B(p,3)<3
             C(p,3)=12;
          else C(p,3)=13;
          end
       end
   end
```

```
    if B(p,4)<1
      C(p,4)=fix(B(p,4)/0. 1)+1;
    else if B(p,4)<10
            C(p,4)=fix(B(p,4))+10;
        else C(p,4)=20;
        end
    end
    if B(p,5)<1
      C(p,5)=fix(B(p,5)/0. 1)+1;
    else if B(p,5)<10
            C(p,5)=fix(B(p,5))+10;
        else C(p,5)=20;
        end
    end
end
%10-cross validation
indices = crossvalind('Kfold',a,10);
for i=1:10
    test=(indices==i);
    train=~test;
    C_train=C(train,:);
    label_train=label(train,:);
    C_test=C(test,:);
    label_test=[label_test;label(test,:)];
    [b,d]=size(label_train);
    c=a-b;
    N=[0 0];
  % count
    for p=1:b
        for j=1:2
            if label_train(p,1)==j
                N(j)=N(j)+1;
            end
        end
    end
    D=ones(2,5,20);
    for p=1:b
        for j=1:2
            if label_train(p,1)==j
                for m=1:5
                    for n=1:20
                        if C_train(p,m)==n
                        D(j,m,n)=D(j,m,n)+1;
                        end
                    end
                end
            end
        end
    end
    for j=1:2
        for m=1:5
            for n=1:20
                E(j,m,n)=D(j,m,n)/N(j);
            end
        end
    end
   %likelihood probability & posterior probability
   Psf=ones(c,1);Pst=ones(c,1);
   for p=1:c
        for k=1:5
            Pf(p,k)=E(2,k,C_test(p,k));
            Pt(p,k)=E(1,k,C_test(p,k));
        end
        for k=1:5
            if k==1|k==2|k==4|k==5
            Psf(p,1)=Psf(p,1)*Pf(p,k);
            Pst(p,1)=Pst(p,1)*Pt(p,k);
            end
        end
        Ph(p,i)=r*Psf(p,1)/((1-r)*Pst(p,1));
   end
end
P=[Ph(:,1);Ph(:,2);Ph(:,3);Ph(:,4);Ph(:,5);Ph(:,6);Ph(:,7);Ph(:,8);Ph(:,9);Ph(:,10)];
xlswrite('note1245. xls',P,'sheet1','B1:B1840');
xlswrite('note1245. xls',label_test,'sheet1','A1:A1838');
```

Program 2：Depicting ROC graphs (7 models）

```
A=xlsread('note1-3. xls','sheet1','A1:B1838');
B=xlsread('note124. xls','sheet1','A1:B1838');
C=xlsread('note125. xls','sheet1','A1:B1838');
D=xlsread('note1-4. xls','sheet1','A1:B1838');
E=xlsread('note1235. xls','sheet1','A1:B1838');
F=xlsread('note1245. xls','sheet1','A1:B1838');
G=xlsread('note1-5. xls','sheet1','A1:B1838');
tp=0;fp=0;p=0;n=0;
for i=1:1838
   if A(i,1)==1
       n=n+1;
   else p=p+1;
   end
end
for i=1:1838
   if A(i,1)==1
      fp=fp+1;
   else tp=tp+1;
   end
   plot(fp/n,tp/p,'b')
   hold on
   xlabel('FPR(True Positive Rate£©')
   ylabel('TPR(False Positive Rate£©')
   grid on
   axis square
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
   if B(i,1)==1
       n=n+1;
   else p=p+1;
   end
end
hold on
for i=1:1838
   if B(i,1)==1
      fp=fp+1;
   else tp=tp+1;
   end
   plot(fp/n,tp/p,'g')
   hold on
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
   if C(i,1)==1
       n=n+1;
   else p=p+1;
   end
end
hold on
for i=1:1838
   if C(i,1)==1
      fp=fp+1;
   else tp=tp+1;
   end
   plot(fp/n,tp/p,'r')
   hold on
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
   if D(i,1)==1
       n=n+1;
   else p=p+1;
   end
end
hold on
for i=1:1838
   if D(i,1)==1
      fp=fp+1;
   else tp=tp+1;
   end
   plot(fp/n,tp/p,'c')
   hold on
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
   if E(i,1)==1
```

```
       n=n+1;
    else p=p+1;
    end
end
hold on
for i=1:1838
    if E(i,1)==1
      fp=fp+1;
    else tp=tp+1;
    end
    plot(fp/n,tp/p,'m')
    hold on
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
    if F(i,1)==1
       n=n+1;
    else p=p+1;
    end
end
hold on
for i=1:1838
    if F(i,1)==1
      fp=fp+1;
    else tp=tp+1;
    end
    plot(fp/n,tp/p,'y')
    hold on
end
tp=0;fp=0;p=0;n=0;
for i=1:1838
    if G(i,1)==1
       n=n+1;
    else p=p+1;
    end
end
hold on
for i=1:1838
    if G(i,1)==1
      fp=fp+1;
    else tp=tp+1;
    end
    plot(fp/n,tp/p,'k')
    hold on
end
```

Appendix3: MATLAB Program for Model Testing of Recognition Performance

```
A=xlsread('fansall. xls','sheet1','B2:E1839');
label=xlsread('fansall. xls','sheet1','F2:F1839');
A_test=xlsread('fansalltest. xls','sheet1','B2:E205');
label_test=xlsread('fansalltest. xls','sheet1','F2:F205');
a=1838;b=204;B=zeros(a,5);B_test=zeros(b,5);N=[0 0];r=0. 5;u=0. 6;K=zeros(2,2);
%prepare for the data needed
for p=1:a
    B(p,1)=A(p,1);
    B(p,2)=A(p,2);
    B(p,3)=A(p,3)/(A(p,2)+1);
    B(p,4)=A(p,2)/(A(p,4)+1);
    B(p,5)=A(p,4)/(A(p,3)+1);
end
for p=1:b
    B_test(p,1)=A_test(p,1);
    B_test(p,2)=A_test(p,2);
    B_test(p,3)=A_test(p,3)/(A_test(p,2)+1);
    B_test(p,4)=A_test(p,2)/(A_test(p,4)+1);
    B_test(p,5)=A_test(p,4)/(A_test(p,3)+1);
end
%分组
C=B;C_test=B_test;
```

```
for p=1:a
   if B(p,2)<2000
       C(p,2)=fix(B(p,2)/100)+1;
   else C(p,2)=20;
   end
   if B(p,3)<1
      C(p,3)=fix(B(p,3)/0. 1)+1;
   else if B(p,3)<2
          C(p,3)=11;
       else if B(p,3)<3
             C(p,3)=12;
          else C(p,3)=13;
          end
       end
   end
   if B(p,4)<1
      C(p,4)=fix(B(p,4)/0. 1)+1;
   else if B(p,4)<10
          C(p,4)=fix(B(p,4))+10;
       else C(p,4)=20;
       end
   end
   if B(p,5)<1
      C(p,5)=fix(B(p,5)/0. 1)+1;
   else if B(p,5)<10
          C(p,5)=fix(B(p,5))+10;
       else C(p,5)=20;
       end
   end
end
for p=1:b
   if B_test(p,2)<2000
   C_test(p,2)=fix(B_test(p,2)/100)+1;
   else C_test(p,2)=20;
   end
   if B_test(p,3)<1
      C_test(p,3)=fix(B_test(p,3)/0. 1)+1;
   else if B_test(p,3)<2
          C_test(p,3)=11;
       else if B_test(p,3)<3
             C_test(p,3)=12;
          else C_test(p,3)=13;
          end
       end
   end
   if B_test(p,4)<1
      C_test(p,4)=fix(B_test(p,4)/0. 1)+1;
   else if B_test(p,4)<10
          C_test(p,4)=fix(B_test(p,4))+10;
       else C_test(p,4)=20;
       end
   end
   if B_test(p,5)<1
      C_test(p,5)=fix(B_test(p,5)/0. 1)+1;
   else if B_test(p,5)<10
          C_test(p,5)=fix(B_test(p,5))+10;
       else C_test(p,5)=20;
       end
   end
end

N=[0 0];
  % count
   for p=1:a
       for j=1:2
          if label(p,1)==j
             N(j)=N(j)+1;
          end
       end
   end
   D=ones(2,5,20);
   for p=1:a
       for j=1:2
          if label(p,1)==j
             for m=1:5
                for n=1:20
                   if C(p,m)==n
```

```
                    D(j,m,n)=D(j,m,n)+1;
                    end
                 end
             end
          end
       end
   end
   for j=1:2
       for m=1:5
          for n=1:20
             E(j,m,n)=D(j,m,n)/N(j);
          end
       end
   end
  %likelihood probability & posterior probability
  Psf=ones(b,1);Pst=ones(b,1);
  for p=1:b
       for k=1:5
          Pf(p,k)=E(2,k,C_test(p,k));
          Pt(p,k)=E(1,k,C_test(p,k));
       end
       for k=1:4
          Psf(p,1)=Psf(p,1)*Pf(p,k);
          Pst(p,1)=Pst(p,1)*Pt(p,k);
       end
       Ph(1,p)=r*Psf(p)/(r*Psf(p)+(1-r)*Pst(p));
       if Ph(1,p)>=u
          label_test1(p,1)=2;
       else label_test1(p,1)=1;
       end
    %evaluate
       for m=1:2
          for n=1:2
             if label_test(p,1)==m&&label_test1(p,1)==n
                    K(m,n)=K(m,n)+1;
             end
          end
       end
   end
recall=K(2,2)/(K(2,1)+K(2,2))%召回率
precision=K(2,2)/(K(1,2)+K(2,2))%准确率
F1=2*recall*precision/(recall+precision)%F1值
```

Appendix4: MATLAB Program for UI Producing

```
function varargout = gui(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',      mfilename, . . .
                '_gui_Singleton',  gui_Singleton, . . .
                'gui_OpeningFcn', @gui_OpeningFcn, . . .
                'gui_OutputFcn',  @gui_OutputFcn, . . .
                'gui_LayoutFcn',  [] , . . .
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State. gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
end
% End initialization code - DO NOT EDIT


% --- Executes just before gui is made visible.
function gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui (see VARARGIN)
% Choose default command line output for gui
handles. output = hObject;

% Update handles structure
guidata(hObject, handles);
end

% UIWAIT makes gui wait for user response (see UIRESUME)
% uiwait(handles. figure1);


% --- Outputs from this function are returned to the command line.
function varargout = gui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles. output;
end

function radiobutton1_Callback(hObject, eventdata, handles)
global e1;
set(handles. text8,'string','');
set(handles. radiobutton1,'value',1);
set(handles. radiobutton2,'value',0);
set(handles. radiobutton3,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   e1=1;
%end
end
function radiobutton2_Callback(hObject, eventdata, handles)
global e1;
set(handles. text8,'string','');
set(handles. radiobutton1,'value',0);
set(handles. radiobutton2,'value',1);
set(handles. radiobutton3,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   e1=2;
%end
end
function radiobutton3_Callback(hObject, eventdata, handles)
global e1;
set(handles. text8,'string','');
set(handles. radiobutton1,'value',0);
set(handles. radiobutton2,'value',0);
set(handles. radiobutton3,'value',1);
%if (get(hObject,'value')==get(hObject,'Max'));
   e1=3;
%end
end
function radiobutton4_Callback(hObject, eventdata, handles)
global u;
set(handles. text8,'string','');
set(handles. radiobutton4,'value',1);
set(handles. radiobutton5,'value',0);
set(handles. radiobutton6,'value',0);
set(handles. radiobutton7,'value',0);
set(handles. radiobutton8,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   u=0. 9;
%end
end
function radiobutton5_Callback(hObject, eventdata, handles)
global u;
set(handles. text8,'string','');
set(handles. radiobutton4,'value',0);
set(handles. radiobutton5,'value',1);
set(handles. radiobutton6,'value',0);
set(handles. radiobutton7,'value',0);
set(handles. radiobutton8,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   u=0. 7;
%end
end
```

```
function radiobutton6_Callback(hObject, eventdata, handles)
global u;
set(handles. text8,'string','');
set(handles. radiobutton4,'value',0);
set(handles. radiobutton5,'value',0);
set(handles. radiobutton6,'value',1);
set(handles. radiobutton7,'value',0);
set(handles. radiobutton8,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   u=0. 5;
%end
end
function radiobutton7_Callback(hObject, eventdata, handles)
global u;
set(handles. text8,'string','');
set(handles. radiobutton4,'value',0);
set(handles. radiobutton5,'value',0);
set(handles. radiobutton6,'value',0);
set(handles. radiobutton7,'value',1);
set(handles. radiobutton8,'value',0);
%if (get(hObject,'value')==get(hObject,'Max'));
   u=0. 4;
%end
end
function radiobutton8_Callback(hObject, eventdata, handles)
global u;
set(handles. text8,'string','');
set(handles. radiobutton4,'value',0);
set(handles. radiobutton5,'value',0);
set(handles. radiobutton6,'value',0);
set(handles. radiobutton7,'value',0);
set(handles. radiobutton8,'value',1);
%if (get(hObject,'value')==get(hObject,'Max'));
   u=0. 3;
%end
end
function edit3_Callback(hObject, eventdata, handles)
set(handles. text8,'string','');
input = str2num(get(hObject,'String'));
guidata(hObject, handles);
end
function edit3_CreateFcn(hObject, eventdata, handles)
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
end
function edit4_Callback(hObject, eventdata, handles)
set(handles. text8,'string','');
input = str2num(get(hObject,'String'));
guidata(hObject, handles);
end
function edit4_CreateFcn(hObject, eventdata, handles)
% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
end
function edit5_Callback(hObject, eventdata, handles)
set(handles. text8,'string','');
input = str2num(get(hObject,'String'));
guidata(hObject, handles);
end
function edit5_CreateFcn(hObject, eventdata, handles)

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
f=get(handles. edit3,'String');
g=get(handles. edit4,'String');
h=get(handles. edit5,'String');
```

```
f1=str2num(f);
g1=str2num(g);
h1=str2num(h);
global e1;
global u;
%check
if (get(handles. radiobutton4,'value')+get(handles. radiobutton5,'value')+get(handles.
radiobutton6,'value')+get(handles. radiobutton7,'value')+get(handles.
radiobutton8,'value')==0)||(get(handles. radiobutton1,'value')+get(handles.
radiobutton2,'value')+get(handles.
radiobutton3,'value')==0)||(f1<0)||(g1<0)||(h1<0)||(f1>2000)
   set(handles. text8,'string','error! ');
else A_test=[e1,f1,g1,h1];
   A=xlsread('fansall. xls','sheet1','B2:E1839');
   label=xlsread('fansall. xls','sheet1','F2:F1839');
   a=1838;b=1;r=0. 5;
   %prepare for the data needed
   for p=1:a
      B(p,1)=A(p,1);
      B(p,2)=A(p,2);
      B(p,3)=A(p,3)/(A(p,2)+1);
      B(p,4)=A(p,2)/(A(p,4)+1);
      B(p,5)=A(p,4)/(A(p,3)+1);
   end
   for p=1:b
      B_test(p,1)=A_test(p,1);
      B_test(p,2)=A_test(p,2);
      B_test(p,3)=A_test(p,3)/(A_test(p,2)+1);
      B_test(p,4)=A_test(p,2)/(A_test(p,4)+1);
      B_test(p,5)=A_test(p,4)/(A_test(p,3)+1);
   end
   %divide into groups
   C=B;C_test=B_test;
   for p=1:a
      if B(p,2)<2000
         C(p,2)=fix(B(p,2)/100)+1;
      else C(p,2)=20;
      end
      if B(p,3)<1
      C(p,3)=fix(B(p,3)/0. 1)+1;
      else if B(p,3)<2
         C(p,3)=11;
         else if B(p,3)<3
            C(p,3)=12;
            else C(p,3)=13;
            end
         end
      end
      if B(p,4)<1
         C(p,4)=fix(B(p,4)/0. 1)+1;
      else if B(p,4)<10
         C(p,4)=fix(B(p,4))+10;
         else C(p,4)=20;
         end
      end
      if B(p,5)<1
      C(p,5)=fix(B(p,5)/0. 1)+1;
      else if B(p,5)<10
         C(p,5)=fix(B(p,5))+10;
         else C(p,5)=20;
         end
      end
   end
   for p=1:b
      if B_test(p,2)<2000
   C_test(p,2)=fix(B_test(p,2)/100)+1;
      else C_test(p,2)=20;
      end
      if B_test(p,3)<1
         C_test(p,3)=fix(B_test(p,3)/0. 1)+1;
      else if B_test(p,3)<2
         C_test(p,3)=11;
         else if B_test(p,3)<3
               C_test(p,3)=12;
            else C_test(p,3)=13;
            end
         end
```

```
        end
        if B_test(p,4)<1
            C_test(p,4)=fix(B_test(p,4)/0. 1)+1;
        else if B_test(p,4)<10
                C_test(p,4)=fix(B_test(p,4))+10;
            else C_test(p,4)=20;
            end
        end
        if B_test(p,5)<1
       C_test(p,5)=fix(B_test(p,5)/0. 1)+1;
        else if B_test(p,5)<10
                C_test(p,5)=fix(B_test(p,5))+10;
            else C_test(p,5)=20;
            end
        end
    end

    N=[0 0];
  % count
    for p=1:a
        for j=1:2
            if label(p,1)==j
                N(j)=N(j)+1;
            end
        end
    end
    D=ones(2,5,20);
    for p=1:a
        for j=1:2
            if label(p,1)==j
                for m=1:5
                    for n=1:20
                        if C(p,m)==n
                        D(j,m,n)=D(j,m,n)+1;
                        end
                    end
                end
            end
        end
    end
    for j=1:2
        for m=1:5
            for n=1:20
                E(j,m,n)=D(j,m,n)/N(j);
            end
        end
    end
   Psf=ones(b,1);Pst=ones(b,1);
   for p=1:b
        for k=1:5
            Pf(p,k)=E(2,k,C_test(p,k));
            Pt(p,k)=E(1,k,C_test(p,k));
        end
        for k=1:4
            Psf(p,1)=Psf(p,1)*Pf(p,k);
            Pst(p,1)=Pst(p,1)*Pt(p,k);
        end
        Ph(1,p)=r*Psf(p)/(r*Psf(p)+(1-r)*Pst(p));
        if Ph(1,p)>=u
            label_test(1,p)=2;
        else label_test(1,p)=1;
        end
   end
   if label_test(1,1)==1
      set(handles. text8,'string','Real user');
   else set(handles. text8,'string','Zombie fan');
   end
   guidata(hObject, handles);
end
```