A Letters Journal Exploring the Frontiers of Physics

April 2015

EPL, **110** (2015) 28003 www.epljournal.org
doi: 10.1209/0295-5075/110/28003

Group-based ranking method for online rating systems with spamming attacks

Jian Gao[1], Yu-Wei Dong[1], Ming-Sheng Shang[1(a)], Shi-Min Cai[1] and Tao Zhou[1,2(b)]

[1] *Web Sciences Center, University of Electronic Science and Technology of China - 611731 Chengdu, PRC*
[2] *Big Data Research Center, University of Electronic Science and Technology of China - 611731 Chengdu, PRC*

received 6 January 2015; accepted in final form 16 April 2015
published online 7 May 2015

PACS 89.65.-s – Social and economic systems
PACS 89.20.Hh – World Wide Web, Internet
PACS 89.20.Ff – Computer science and technology

Abstract – The ranking problem has attracted much attention in real systems. How to design a robust ranking method is especially significant for online rating systems under the threat of spamming attacks. By building reputation systems for users, many well-performed ranking methods have been applied to address this issue. In this letter, we propose a group-based ranking method that evaluates users' reputations based on their grouping behaviors. More specifically, users are assigned with high reputation scores if they always fall into large rating groups. Results on three real data sets indicate that the present method is more accurate and robust than the correlation-based method in the presence of spamming attacks.

Introduction. – With the rapid development of the Internet, billions of services and objects are online for us to choose [1]. At the same time, the problem of information overload troubles us every day [2–4]. Therefore, many web sites (Amazon, Ebay, MovieLens, Netflix, etc.) introduce online rating systems, where users can give discrete ratings to objects. In turn, the ratings of an object serve as a reference and later affect other users' decisions [5,6]. Basically, high ratings can promote the consumption, while low ratings play the opposite role [7]. In real cases, some users may give unreasonable ratings since they are simply unfamiliar with the related field [8], and some others deliberately give biased ratings for various psychosocial reasons [9–13]. These widely existed distorted ratings can purposefully enhance or discredit objects' reputation, mislead users' judgments, and affect the evolution of rating systems [14–16]. Due to the negative effects of spamming attacks, how to design a robust method for online rating systems is becoming an urgent task [17–19].

To solve this problem, normally, building a reputation system for users is a good way [20–28]. Laureti *et al.* [25] proposed an iterative refinement (IR) method, where a user's reputation is inversely proportional to the difference between his ratings and the estimation of the corresponding objects' quality (*i.e.*, weighted average rating). The reputation and the estimated quality are iteratively calculated until they become stable. An improved IR method is proposed in [26], by assigning trust to each individual rating. Later, Zhou *et al.* [27] proposed the correlation-based ranking (CR) method that is robust to spamming attacks, where a user's reputation is iteratively determined by the correlation between his ratings and objects' estimated quality. Very recently, by introducing a reputation redistribution process and two penalty factors, Liao *et al.* [28] further improved the CR method.

In the majority of previous works, a single standard objects' quality [29,30] is required in determining users' reputations, with an underlying assumption that every object is associated with a most objective rating that best reflect its quality. However, in real cases, one object may have multiple valid ratings, since the ratings are subjective and can be affected by users' background [31–34]. In the presence of more than one reasonable answer to a single task, Tian *et al.* [29] analyzed the group structure of schools of thought in solving the problem of identifying reliable workers as well as unambiguous tasks in data collection. Specifically, a worker who is consistent with many other workers in most of the tasks is reliable, and a task whose answers form a few tight clusters is easy and clear. Analogously, in the online rating systems, one object's quality is clear if its ratings are centralized, while it is

(a) E-mail: msshang@uestc.edu.cn
(b) E-mail: zhutou@ustc.edu

not clear if the ratings are widely distributed. Under this framework, a single estimation of an object's quality is no longer applicable [30]. Practically, a random rating to objects with confusing quality should be acceptable since no single rating can dominate its true quality, while a biased rating to objects with clear quality is unreasonable. Users who are consistent with the majority in most ratings will form big groups and be trusted since herding is a well-documented feature of human behaviour [6,35]. Users who always give distorted ratings will form relatively small groups and be highly suspected since unreasonable or biased ratings are discordant [29]. These ideas bring us a promising way to build reputation systems based on users' grouping behaviour instead of solving the crucial problem of estimating objects' true qualities as before.

In this letter, we propose a group-based ranking (GR) method for online rating systems with spamming attacks. By grouping users according to their ratings, users' reputations are determined according to the corresponding group sizes. If they always fall into large groups, their reputations are high, otherwise, their reputations are low. Extensive experiments on three real data sets (MovieLens, Netflix and Amazon) suggest that the proposed method outperforms the CR method.

Method. – The online rating system is naturally described by a weighed bipartite network $G = \{U, O, E\}$, where $U = \{U_1, U_2, \ldots, U_m\}$, $O = \{O_1, O_2, \ldots, O_n\}$, $E = \{E_1, E_2, \ldots, E_l\}$ are sets of users, objects and ratings, respectively [36]. The degree of a user i and an object α are denoted as k_i and k_α, respectively. Here, we use Greek and Latin letters, respectively, for object-related and user-related indices to distinguish them. Considering a discrete rating system, the bipartite network can be represented by a rating matrix A [37], where the element $a_{i\alpha} \in \Omega = \{\omega_1, \omega_2, \ldots, \omega_z\}$ is the weight of the link connecting node U_i and node O_α, *i.e.*, the rating given by user i to object α. In a reputation system, each user i will be assigned a reputation, denoted as R_i. The users with very low reputations are detected as spammers.

The GR method works as follows. Firstly, we group users according to their ratings. Specifically, for an object α, we put users who gave the rating ω_s into group $\Gamma_{s\alpha}$:

$$\Gamma_{s\alpha} = \{U_i \mid a_{i\alpha} = \omega_s, i = 1, 2, \ldots, m\}. \tag{1}$$

Obviously, user i belongs to k_i different groups. Secondly, we calculate the sizes of all groups $\Lambda_{s\alpha} = |\Gamma_{s\alpha}|$, *i.e.* the number of users who gave the rating ω_s to object α. Thirdly, we establish a matrix Λ^*, named rating-rewarding matrix, by normalizing Λ per column:

$$\Lambda^*_{s\alpha} = \frac{\Lambda_{s\alpha}}{k_\alpha}. \tag{2}$$

Fourthly, referring to Λ^*, we map the original rating matrix A to a matrix A', named rewarding matrix. More specifically, the rewarding that a user i obtains from his rating $a_{i\alpha}$ is defined as $A'_{i\alpha} = \Lambda^*_{s\alpha}$, where $a_{i\alpha} = \omega_s$. $A'_{i\alpha}$ is null if user i has not yet rated object α.

Then, we assign reputations to users according to their rewarding. On the one hand, if the average of a user's rewarding is small, most of his ratings must be deviated from the majority, suggesting that he is highly suspected. On the other hand, if the rewarding varies largely, he is also untrustworthy for his unstable rating behavior. Based on these considerations, we define user i's reputation as

$$R_i = \frac{\mu(A'_i)}{\sigma(A'_i)}, \tag{3}$$

where μ and σ are functions of mean value and standard deviation, respectively. Specifically, the mean value of A'_i is defined as

$$\mu(A'_i) = \sum_\alpha \frac{A'_{i\alpha}}{k_i}, \tag{4}$$

and the standard deviation of A'_i is defined as

$$\sigma(A'_i) = \sqrt{\frac{\sum_\alpha (A'_{i\alpha} - \mu(A'_i))^2}{k_i}}. \tag{5}$$

In fact, R_i is the same with the inverse of the coefficient of variation [38] of vector A'_i, which shows the dispersion of the frequency distribution of user i's rewardings. Finally, we sort users in ascending order by reputation, and deem the top-L ones as detected spammers. A visual representation of GR method is given in fig. 1.

Data and metrics. – We consider three commonly studied real data sets, MovieLens, Netflix and Amazon, to test the accuracy of the GR method. MovieLens and Netflix contain ratings on movies, provided by GroupLens project at University of Minnesota (`www.grouplens.org`) and released by the DVD rental company Netflix for a contest on recommender systems (`www.netflixprize.com`), respectively. Amazon contains ratings on products (*e.g.* books, music, etc.) crawled from `amazon.com` [39]. All the three data sets use a 5-point rating scale with 1 being the worst and 5 being the best. Herein, we sampled and extracted three smaller data sets from the original data sets, respectively, by choosing users who have at least 20 ratings and objects having been rated by these users since it is hard to tell whether small-degree users are spammers [28]. The basic statistics of data sets are summarized in table 1.

Generating artificial spammer. Two types of distorted ratings are widely found in real rating systems, namely, malicious ratings and random ratings. The malicious ratings are from spammers who always give minimum (maximum) allowable ratings to push down (up) certain target objects [9,10]. The random ratings mainly come from some naughty users or test engineers who randomly give meaningless ratings [14,15]. As spammers are unknown in real data, to test the method, we manipulate the three real data sets by adding either type of artificial spammers (*i.e.* malicious or random) at one time.

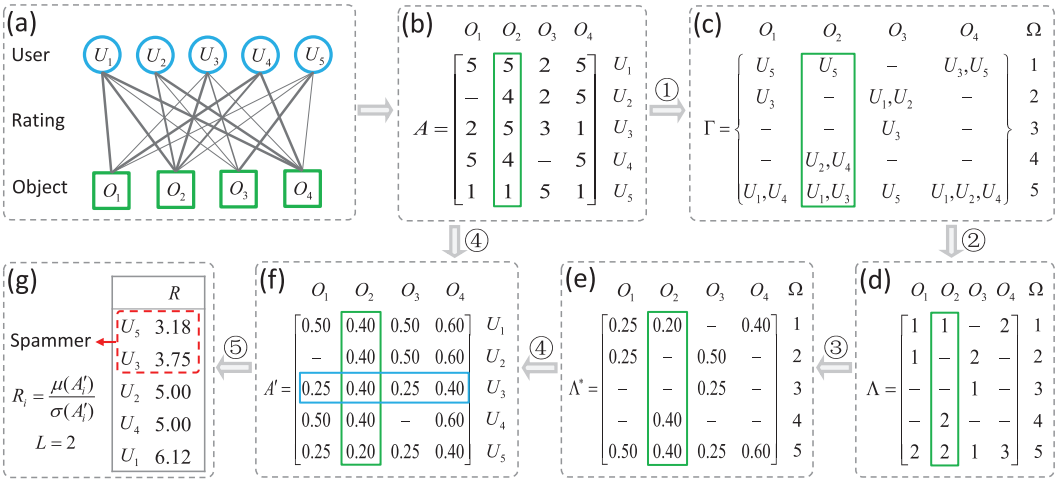

Fig. 1: (Color online) Illustrating the group-based ranking method. The number besides the gray arrow marks the step of the procedure. (a) The original weighed bipartite network, G. (b) The corresponding rating matrix, A. The row and column correspond to users and objects, respectively. The symbol "–" stands for a non-rating, which can be ignored in the calculation. (c) The groups of users, Γ, after being grouped according to their ratings. Take O_2 as an example (green vertical box). As U_2 and U_4 rated 4 to O_2, they are put into group $\Gamma_{4,2}$. (d) The sizes of groups, Λ, *e.g.* $\Lambda_{4,2} = 2$ as $\Gamma_{4,2} = \{U_2, U_4\}$. (e) The rating-rewarding matrix, Λ^*, constructed by normalizing Λ by column, *e.g.*, $\Lambda^*_{4,2} = 2/(1+2+2) = 0.40$. (f) The rewarding matrix, A', obtained by mapping matrix A referring to Λ^*, *e.g.* $A'_{4,2} = 0.40$. (g) The ranking list of users based on reputation R. Take U_3 as an example (blue horizontal box in (f)), $R_3 = \mu(A'_3)/\sigma(A'_3) = 3.75$. Setting spam list's length as $L = 2$, then U_5 and U_3 (red dashed box) are detected as spammers.

Table 1: Basic statistics of the three real data sets. m is the number of users, n is the number of objects, $\langle k_U \rangle$ is the average degree of users, $\langle k_O \rangle$ is the average degree of objects, and $S = l/mn$ is the sparsity of the bipartite network.

Data set	m	n	$\langle k_U \rangle$	$\langle k_O \rangle$	S
MovieLens	943	1682	106	60	0.063
Netflix	1038	1215	47	40	0.039
Amazon	662	1500	36	15	0.023

In the implementation, we randomly select d users and assign them distorted ratings: i) integer 1 or 5 with the same probability (*i.e.*, 0.5) for malicious spammers, and ii) random integers in $\{1, 2, 3, 4, 5\}$ for random spammers. Thus, the ratio of spammers is $q = d/m$. To study the effects of spammers' activity, we define $p = k/n$ as the activity of spammers, where k is the degree of each spammer. Here k is a tunable parameter, that is, if k is no more than a spammer's original degree, we randomly select his/her k ratings and replace them with distorted ratings and the unselected ratings are ignored. Otherwise, after replacing all the spammer's original ratings, we randomly select remaining number of non-rated objects and assign them distorted ratings.

Metrics for evaluation. To evaluate the performance of the ranking methods, we adopt two commonly used metrics: recall [40] and AUC (the area under the ROC curve) [41]. The recall value measures to what extent the spammers can be detected in the top-L ranking list,

$$R_c(L) = \frac{d'(L)}{d}, \tag{6}$$

where $d'(L) \leq d$ is the number of detected spammers in the top-L list. A higher R_c indicates a higher accuracy.

Note that R_c only focuses on the top-L ranks, and thus we also consider an L-independent metric called AUC. Provided the rank of all users, the AUC value can be interpreted as the probability that a randomly chosen spammer is ranked higher than a randomly chosen non-spammer. To calculate AUC, at each time we randomly pick a spammer and a non-spammer to compare their reputations, if among N independent comparisons, there are N' times the spammer has a lower reputation and N'' times they have the same reputation, the AUC value is

$$AUC = \frac{N' + 0.5N''}{N}. \tag{7}$$

If all users are ranked randomly, the AUC value should be about 0.5. Therefore, the degree to which the value exceeds 0.5 indicates how better the method performs than pure chance [42].

Results. – According to the single standard assumption, each object α has a true quality, denoted by Q_α. As the true quality is unknown in reality, taking the average rating as an estimation of Q_α is the most straightforward way. Then, the rating error of user i is defined as

$$\delta_i = \frac{\sum_\alpha |a_{i\alpha} - \hat{Q}_\alpha|}{k_i}, \tag{8}$$

where $\hat{Q}_\alpha = \sum_{j \in \Gamma_\alpha} a_{j\alpha}/k_\alpha$ is the average rating of object α and α runs over all objects being rated by user i. A reasonable ranking method should give high reputations to

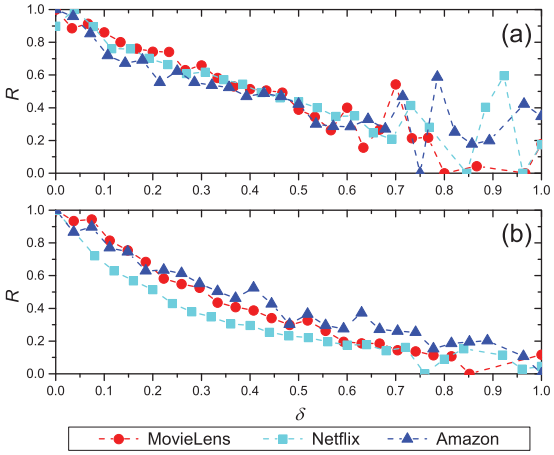

Fig. 2: (Color online) The relation between reputation R and rating error δ (bins) in different methods. Panels (a) and (b) stand for the CR and the GR method, respectively. δ and R are respectively normalized for comparison under different data sets.

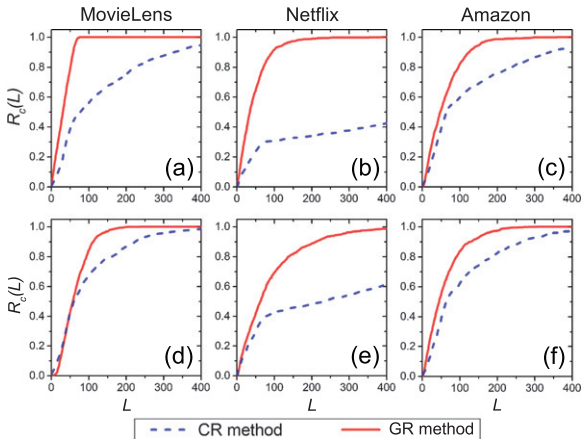

Fig. 3: (Color online) The recall R_c of different methods as a function of the length of the list L. Panels (a), (b) and (c) stand for malicious spammers; panels (d), (e) and (f) stand for random spammers with $d = 50$ being fixed. The parameter p is set as about 0.05, 0.03 and 0.02 for MovieLens, Netflix and Amazon, respectively. The results are averaged over 100 independent realizations.

the users with low rating errors, *i.e.* R_i should be negatively correlated with δ_i.

Figure 2 shows the relation between users' rating errors and reputations evaluated by CR and GR methods, respectively. As users' rating errors are continuous and with different scales, we normalize and divide them into bins with length 0.05. As one can see, the two methods both assign high reputations for users with small δ_i, while the GR method outperforms the CR method by stably assigning low reputations (see fig. 2(b)) for users with large δ_i. Furthermore, to quantify the correlation, we calculate the Pearson correlation coefficient ρ [38] between R_i and δ_i. Specifically, $\rho = -0.956$ (-0.949), -0.906 (-0.872) and -0.966 (-0.816) after applying the GR (CR) method to MovieLens, Netflix and Amazon, respectively. The larger negative correlation suggest that the GR method is better on evaluating users' reputations.

Effectiveness and efficiency. To test the effectiveness of the ranking methods, based on the three real data sets, we first generate artificial data sets with 50 spammers (*i.e.* $d = 50$). Each data set is only with one type of spammers: malicious or random. On the generated data sets, we calculate recall of different methods as a function of the spammer list's length L. As shown in fig. 3, the GR method has remarkable advantage over the CR method on detecting both types of spammers, especially when L is larger than d. We also note that R_c of ranking malicious spammers is a little higher than that of random spammers when L is smaller than d, which implies that to detect random spammers is relatively harder.

The results of AUC values are shown in table 2, where one can see that the AUC values of the GR method are higher than that of the CR method for every data set, suggesting that the GR method has significant advantage towards the CR method. It also shows that the CR method is better at detecting random spammers than

Table 2: AUC of different methods for the real data sets with $d = 50$ being fixed. The parameter p is set as about 0.05, 0.03 and 0.02 for MovieLens, Netflix and Amazon, respectively. The results are averaged over 100 independent realizations.

Data set	Malicious spammers		Random spammers	
	CR	GR	CR	GR
MovieLens	0.876	0.994	0.914	0.959
Netflix	0.543	0.977	0.668	0.930
Amazon	0.824	0.941	0.877	0.949

malicious spammers, while the GR method is inverse. In addition, it is worthy to be noticed that the AUC is generally lower in Netflix, especially for the CR method. One possible explanation is that there are more harmful spammers in Netflix and the CR method is very sensitive to "real" spammers [27,28]. Additionally, in Netflix, there are more small-degree objects whose quality will be considered higher by the CR method [28], which may also lead to the biased ranking.

Robustness against spammers. We then study the robustness of different methods by varying p (the ratio of objects rated by spammers) and q (the ratio of spammers). In the following, we set the length of the detected spam list being equal to the number of artificial spammers, namely, $L = d$. Figure 4 shows the recall obtained by the GR method. The ranges of p and q are personalized set for different data sets referring to their sparsity. One can observe that, overall, the GR method has better performance on detecting malicious spammers, especially when p is small (*i.e.* spammers are of small degree). Moreover, when q is small, the recall values of ranking random spammers are low for MovieLens and Netflix data sets

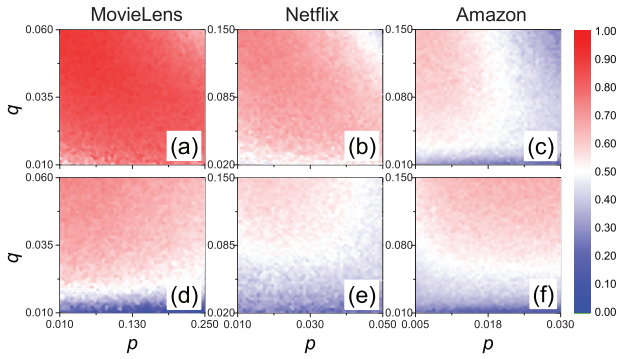

Fig. 4: (Color online) The effectiveness of the GR method. The color marks recall R_c. q and p are ratio of spammers and ratio of objects rated by spammers, respectively. Panels (a), (b) and (c) stand for malicious spammers. Panels (d), (e) and (f) stand for random spammers. The parameter is set as $L = d$. The results are averaged over 100 independent realizations.

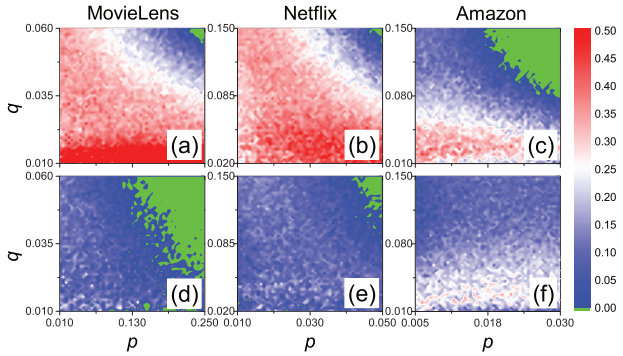

Fig. 5: (Color online) The comparison of GR and CR methods. The color marks ΔR_c if $\Delta R_c > 0$, except for the green color, meaning that the CR method is better. Panels (a), (b) and (c) stand for malicious spammers. Panels (d), (e) and (f) stand for random spammers. The parameter is set as $L = d$. The results are averaged over 100 independent realizations.

(see figs. 4(d) and (e)), while for Amazon data set, the recall values of ranking both two types of spammers are low (see figs. 4(c) and (f)). In addition, the recall value positively increases with q. These observations suggest that i) detecting malicious spammers is easier than random spammers, ii) MovieLens and Netflix may contain more "real" spammers, and iii) the GR method is powerful to detect spammers who only rate a small number of objects.

To comprehensively compare the performance of GR and CR methods, we calculate the difference of recall values between the two, formulateed as $\Delta R_c = R_c^{GR} - R_c^{CR}$. As shown in fig. 5, ΔR_c is above 0 in most area, suggesting that the overall performance of the GR method is better. In detail, the GR method has remarkable advantage in detecting malicious spammers, while the advantage is not obvious for random spammers. In addition, ΔR_c is big when p and q are small, implying that GR method is more robust against a small number of small-degree spammers, which are usually difficult to be detected out.

Conclusions and discussion. – In summary, we have proposed a group-based ranking (GR) method to solve the ranking problem in online rating systems with spamming attacks. By grouping users according to their ratings, we construct a rating-rewarding matrix according to the corresponding group sizes, based on which we map a user's rating vector to rewarding vector. Then, this user's reputation is assigned according to the inverse of dispersion of frequency distribution of his rewarding vector. Extensive experiments showed that the proposed method is effective on evaluating users' reputations, especially for those with high rating errors. When tested on the generated data sets with two types of artificial spammers, the GR method gives higher performance in both accuracy and robustness compared with the correlation-based ranking (CR) method, especially on resisting small-degree spammers. Interestingly, the accuracy of spam detection on Netflix data set is low for both the two methods, indicating that there are more original distorted ratings in Netflix data sets, which is in accordance with some previous studies [27,28].

The proposed method has several distinguishing characteristics, differentiating it from current users' reputation allocation procedures: i) The method assigns users' reputation by considering the grouping behavior of users instead of being based on the estimation of objects' true qualities. ii) The method shows high performance in both accuracy and robustness, especially when dealing with small-degree spammers' attacks. iii) The method is very efficient, as its time complexity [43] is $O(m^2)$, which is significantly lower than most of previously proposed iterative methods. As further improvement, we could consider introducing this method to an iterative process [25–28], applying it to continuous rating systems [44,45], and considering the effect of long-term evolution of online rating systems [16].

* * *

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 61370150, 91324002, 61433014 and 11222543. JG acknowledges support from Tang Lixin Education Development Foundation by UESTC. TZ acknowledges the Program for New Century Excellent Talents in University under Grant No. NCET-11-0070, and Special Project of Sichuan Youth Science and Technology Innovation Research Team under Grant No. 2013TD0006.

REFERENCES

[1] Lü L., Medo M., Yeung C. H., Zhang Y.-C., Zhang Z.-K. and Zhou T., *Phys. Rep.*, **519** (2012) 1.
[2] Zhang F. and Zeng A., *EPL*, **100** (2012) 58005.
[3] Zeng A., Vidmer A., Medo M. and Zhang Y.-C., *EPL*, **105** (2014) 58002.

[4] Guo Q., Song W.-J. and Liu J.-G., *EPL*, **107** (2014) 18001.

[5] Jindal N. and Liu B., in *Proceedings of the 16th International Conference on World Wide Web* (ACM Press) 2007, pp. 1189–1190.

[6] Muchnik L., Aral S. and Taylor S.-J., *Science*, **341** (2013) 647.

[7] Yu X., Liu Y., Huang X. and An A., *IEEE Trans. Knowl. Data Eng.*, **24** (2012) 720.

[8] Pan W., Xiang E.-W., Liu N.-N. and Yang Q., in *Proceedings of the 24th AAAI Conference on Artificial Intelligence* (AAAI Press) 2013, pp. 39–55.

[9] Wang G., Xie S., Liu B. and Yu P. S., in *Proceedings of the IEEE 11th International Conference on Data Mining (ICDM)* (IEEE Press) 2011, pp. 1242–1247.

[10] Chung C.-Y., Hsu P.-Y. and Huang S.-H., *Decis. Support Syst.*, **55** (2013) 314.

[11] Yang Z., Zhang Z.-K. and Zhou T., *EPL*, **100** (2012) 68002.

[12] Huang J., Cheng X.-Q., Shen H.-W., Zhou T. and Jin X., in *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)* (ACM Press) 2012, pp. 573–582.

[13] Zhang Y.-L., Ni J., Guo Q. and Liu J.-G., *Physica A*, **417** (2015) 261.

[14] Zeng A. and Cimini G., *Phys. Rev. E*, **85** (2012) 036101.

[15] Zhang Q.-M., Zeng A. and Shang M.-S., *PLoS ONE*, **8** (2013) e62624.

[16] Zhao D.-D., Zeng A., Shang M.-S. and Gao J., *Chin. Phys. Lett.*, **30** (2013) 8901.

[17] Mukherjee A., Liu B., Wang J., Glance N. and Jindal N., in *Proceedings of the 20th International Conference on World Wide Web* (ACM Press) 2011, pp. 93–94.

[18] Fei G., Mukherjee A., Liu B., Hsu M., Castellanos M. and Ghosh R., in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM)* (AAAI Press) 2013, pp. 175–184.

[19] Lin Y., Zhu T., Wang X., Zhang J. and Zhou A., in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion* (WWW) 2014, pp. 341–342.

[20] Masum H. and Zhang Y.-C., *First Monday*, **9** (2004) 7.

[21] Resnick P., Kuwabara K., Zeckhauser R. and Friedman E., *Commun. ACM*, **43** (2000) 45.

[22] Jøsang A., Ismail R. and Boyd C., *Decis. Support Syst.*, **43** (2007) 618.

[23] Allahbakhsh M., Ignjatovic A., Motahari-Nezhad H. R. and Benatallah B., *World Wide Web*, **18** (2015) 73.

[24] Ling G., King I. and Lyu M. R., in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence* (AAAI Press) 2013, pp. 2670–2676.

[25] Laureti P., Moret L., Zhang Y.-C. and Yu Y.-K., *Europhys. Lett.*, **75** (2006) 1006.

[26] de Kerchove C. and Van Dooren P., arXiv:0711.3964.

[27] Zhou Y.-B., Lei T. and Zhou T., *EPL*, **94** (2011) 48002.

[28] Liao H., Zeng A., Xiao R., Ren Z.-M., Chen D.-B. and Zhang Y.-C., *PLoS ONE*, **9** (2014) e97146.

[29] Tian Y. and Zhu J., in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM Press) 2012, pp. 226–234.

[30] Raykar V. C., Yu S., Zhao L.-H., Valadez G. H., Florin C., Bogoni L. and Moy L., *J. Mach. Learn. Res.*, **11** (2010) 1297.

[31] Shi X., Zhu J., Cai R. and Zhang L., in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM Press) 2009, pp. 777–786.

[32] Welinder P., Branson S., Belongie S. and Perona P., in *Advances in Neural Information Processing Systems (NIPS)* (MIT Press) 2010, pp. 2424–2432.

[33] Reed M. S., Evely A. C., Cundill G., Fazey I., Glass J., Laing A., Newig J., Parrish B., Prell C., Raymond C. and Stringer L. C., *Ecol. Soc.*, **15** (2010) r1.

[34] Ross J., Irani L., Silberman M., Zaldivar A. and Tomlinson B., in *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (ACM Press) 2012, pp. 2863–2872.

[35] Raafat R. M., Chater N. and Frith C., *Trends Cognit. Sci.*, **13** (2009) 420.

[36] Shang M.-S., Lü L., Zhang Y.-C. and Zhou T., *EPL*, **90** (2010) 48006.

[37] Zhou T., Ren J., Medo M. and Zhang Y.-C., *Phys. Rev. E*, **76** (2007) 46115.

[38] Lin L. I., *Biometrics*, **45** (1989) 255.

[39] Jindal N. and Liu B., in *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM)* (ACM Press) 2008, pp. 219–230.

[40] Herlocker J. L., Konstan J. A., Terveen L. G. and Riedl J. T., *ACM Trans. Inf. Syst.*, **22** (2004) 5.

[41] Hanley J. A. and McNeil B. J., *Radiology*, **143** (1982) 29.

[42] Lü L. and Zhou T., *Physica A*, **390** (2011) 1150.

[43] Sipser M., *Introduction to the Theory of Computation* (Cengage Learning, Boston) 2012.

[44] Stanley N. and Jenkins S., in *Challenges of a Changing World. Proceedings of the 5th International Conference of the Association for Survey Computing* (Association for Survey Computing, Berkeley) 2007, pp. 81–92.

[45] Treiblmaier H. and Filzmoser P., in *Proceedings of the International Conference on Information Systems (ICIS)* (Association for Information Systems, Utrecht) 2011, Paper 1.