# Detecting Changes in Opinion Value Distribution for Voter Model

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp

**Abstract.** We address the problem of detecting the change in opinion share over a social network caused by an unknown external situation change under the value-weighted voter model with multiple opinions in a retrospective setting. The unknown change is treated as a change in the value of an opinion which is a model parameter, and the problem is reduced to detecting this change and its magnitude from the observed opinion share diffusion data. We solved this problem by iteratively maximizing the likelihood of generating the observed opinion share, and in doing so we devised a very efficient search algorithm which avoids parameter value optimization during the search. We tested the performance using the structures of four real world networks and confirmed that the algorithm can efficiently identify the change and outperforms the naive method, in which an exhaustive search is deployed, both in terms of accuracy and computation time.

## 1 Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which has made it possible to form various kinds of large social networks, through which behaviors, ideas and opinions can spread, and our behavioral patterns are strongly affected by the interaction with these networks. Thus, substantial attention has been directed to investigating the spread of influence in these networks [9, 2, 14].

Much of the work has treated information as one entity and nodes in the network are either active (influenced) or inactive (uninfluenced), i.e. there are only two states. However, application such as an on-line competitive service in which a user can choose one from multiple choices/decisions requires a model that handles multiple states. In addition, it is important to consider the value of each choice, e.g., quality, brand, authority, etc. because this impacts our choice. We formulated this problem using a value-weighted $K$ opinion diffusion model and provided a way to accurately predict the expected share of the opinions at a future target time from a limited amount of observed data [6]. This model is an extension of the basic voter model which is based on the assumption that a person changes its opinion by the opinions of its neighbors. There has been a variety of work on the voter model. Dynamical properties of the basic model have been extensively studied including how the degree distribution and the network size affect the mean time to reach consensus [10, 12]. Several variants of the voter model are also investigated and non equilibrium phase transition is analyzed [1, 15]. Yet another line of work extends the voter model by combining it with a network evolution model [3, 2].

These studies are different from what we address in this paper. Almost all of the work so far on information diffusion assumed that the model is stationary. However, our behavior is affected not only by the behaviour of our neighbors but also by other external factors. We apply our voter model to detect a change in opinion share which is caused by an unknown external situation change. We model the change in the external factors as a change in the opinion value, and try to detect the change from the observed opinion share diffusion data. If this is possible, this would bring a substantial advantage. We can detect that something unusual happened during a particular period of time by simply analyzing the data. Note that our approach is retrospective, i.e. we are not predicting the future, but we are trying to understand the phenomena that happened in the past, which shares the same spirit of the work by Kleinberg [7] and Swan [13] in which they tried to organize a huge volume of the data stream and extract structures behind it.

Thus, our problem is reduced to detecting where in time and how long this change persisted and how big this change is. To make the analysis simple, we limit the form of the value change to a rect-linear one, that is, the value changes to a new higher level, persists for a certain period of time and is restored back to the original level and stays the same thereafter. We call this period when the value is high as "hot span" and the rest as "normal span". We use the same parameter optimization algorithm as in [6], i.e. the parameter update algorithm based on the Newton method which globally maximizes the likelihood of generating the observed data sequences. The problem here is more difficult because it has another loop to search for the hot span on top of the above loop. The naive learning algorithm has to iteratively update the patten boundaries (outer loop) and the value must also be optimized for each combination of the pattern boundaries (inner loop), which is extraordinary inefficient. We devised a very efficient search algorithm which avoids the inner loop optimization during the search. We tested the performance using the structures of four real world networks (blog, Wikipedia, Enron and coauthorship), and confirmed that the algorithm can efficiently identify the hot span correctly as well as the opinion value. We further compared our algorithm with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

## 2 Opinion Formation Models

The mathematical model we use for the diffusion of opinions is the value-weighted voter model with $K$ ($\geq 2$) opinions [6]. A social network is represented by an undirected (bidirectional) graph with self-loops, $G = (V, E)$, where $V$ and $E$ ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of $v$ in $G$, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

In the model, each node of $G$ is endowed with $(K+1)$ states; opinions $1, \cdots, K$, and *neutral* (i.e., no-opinion state). It is assumed that a node never switches its state from any opinion $k$ to neutral. The model has a parameter $w_k$ ($>0$) for each opinion $k$, which is called the *value-parameter* and must be estimated from observed opinion diffusion data. Let $f_t : V \to \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time $t$, where $f_t(v)$ stands for the opinion of node $v$ at time $t$, and opinion 0 denotes the neutral state. We also denote by $n_k(t, v)$ the number of $v$'s neighbors that hold opinion $k$ at time $t$ for $k = 1, 2, \cdots, K$, i.e., $n_k(t, v) = |\{u \in \Gamma(v);\ f_t(u) = k\}|$. Given a target time $T$, and an initial state in which each opinion is assigned to only one distinct node and all other nodes are in the neutral state, the evolution process of the model unfolds in the following way. In general, each node $v$ considers changing its opinion based on the current opinions of its neighbors at its $(j-1)$th update-time $t_{j-1}(v)$, and actually changes its opinion at the $j$th update-time $t_j(v)$, where $t_{j-1}(v) < t_j(v) \leq T$, $j = 1, 2, 3, \cdots$, and $t_0(v) = 0$. It is noted that since node $v$ is included in its neighbors by definition, its own opinion is also reflected. The $j$th update-time $t_j(v)$ is decided at time $t_{j-1}(v)$ according to the exponential distribution of parameter $\lambda$ (we simply use $\lambda = 1$ for any $v \in V$)[1]. Then, node $v$ changes its opinion at time $t_j(v)$ as follows: If node $v$ has at least one neighbor with some opinion at time $t_{j-1}(v)$, $f_{t_j(v)}(v) = k$ with probability $w_k n_k(t_{j-1}(v), v)$ / $\sum_{k'=1}^{K} w_{k'}\, n_{k'}(t_{j-1}(v), v)$ for $k = 1, \cdots, K$, otherwise, $f_{t_j(v)}(v) = 0$ with probability 1. Note here that $f_t(v) = f_{t_{j-1}(v)}(v)$ for $t_{j-1}(v) \leq t < t_j(v)$. If the next update-time $t_j(v)$ passes $T$, that is, $t_j(v) > T$, then the opinion evolution of $v$ is over. The evolution process terminates when the opinion evolution of every node in $G$ is over.

Given the observed opinion diffusion data $\mathcal{D}(T_s, T_e) = \{(v, t, f_t(v))\}$ in time-interval $[T_s, T_e]$ (a single example), we consider estimating the values of value-parameters $w_1$, $\cdots$, $w_K$, where $0 \leq T_s < T_e \leq T$. From the evolution process of the model, we can obtain the following log likelihood function

$$\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e)) = \log \prod_{(v,t,k) \in \mathcal{C}(T_s, T_e)} \frac{n_k(t, v) w_k}{\sum_{k'=1}^{K} n_{k'}(t, v) w_{k'}}, \tag{1}$$

where $\boldsymbol{w} = (w_1, \cdots, w_K)$ stands for the $K$-dimensional vector of value-parameters, and $\mathcal{C}(T_s, T_e) = \{(v, t, f_t(v)) \in \mathcal{D}(T_s, T_e);\ |\{u \in \Gamma(v);\ f_t(u) \neq 0\}| \geq 2\}$. Thus, our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(T_s, T_e))$ with respect to $\boldsymbol{w}$. We find the optimal values of $\boldsymbol{w}$ by employing a standard Newton method (see [6] for more details).

## 3 Change Detection Problem

We investigate the problem of detecting the change in behavior of opinion diffusion in a social network $G$ based on the value-weighted voter model with $K$ opinions, which is referred to as the *change detection problem*. In this problem, we assume that some change has happened in the way the opinions diffuse, and we observe the opinion diffusion data in which the change is embedded, and consider detecting where in time and how long this change persisted and how big this change is.

[1] Note that this is equivalent to picking a node randomly and updating its opinion in turn $|V|$ times.

Here, we mathematically formulate the change detection problem. For the opinion diffusion data $\mathcal{D}(0, T)$ in time-interval $[0, T]$, let $[T_1, T_2]$ denote the hot (change) span of the diffusion of opinions. This implies that the intervals $[0, T_1)$ and $(T_2, T]$ are the normal spans. Let $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$ denote the value-parameter vectors for the normal span and the hot span, respectively. Note that $\boldsymbol{w}_n/||\boldsymbol{w}_n|| \neq \boldsymbol{w}_h/||\boldsymbol{w}_h||$ since the opinion dynamics under the value-weighted voter model is invariant to positive scaling of the value-parameter vector $\boldsymbol{w}$, where $||\boldsymbol{w}_n||$ and $||\boldsymbol{w}_h||$ stand for the norm of vectors $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$. Then, the change detection problem is formulated as follows: Given the opinion diffusion data $\mathcal{D}(0, T)$ in time-interval $[0, T]$, detect the anomalous span $[T_1, T_2]$, and estimate the value-parameter vector $\boldsymbol{w}_h$ of the hot span and the value-parameter vector $\boldsymbol{w}_n$ of the normal span.

Since the value-weighted voter model is a stochastic process model, every sample of opinion diffusion can behave differently. This means that it is quite difficult to accurately detect the true hot span from only a single sample of opinion diffusion. Methods that use only the observed bursty activities, including those proposed by Swan and Allan [13] and Kleinberg [7] would not work. We believe that an explicit use of underlying opinion diffusion model is essential to solve this problem. It is crucially important to detect the hot span precisely in order to identify the external factors which caused the behavioral changes.

## 4 Detection Methods

### 4.1 Naive Method

Let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of opinion change time points of all the nodes appearing in the diffusion results $\mathcal{D}(0, T)$. We can consider the following value-parameter vector switching when there is a hot span $S = [T_1, T_2]$:

$$\boldsymbol{w} = \begin{cases} \boldsymbol{w}_n & \text{if } t \in \mathcal{T} \setminus S, \\ \boldsymbol{w}_h & \text{if } t \in \mathcal{T} \cap S. \end{cases}$$

Then, an extended objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0, T), S)$ can be defined by adequately modifying Equation (1) under this switching scheme. Clearly, the extended objective function is expected to be maximized by setting $S$ to be the true span $S^* = [T_1^*, T_2^*]$, for which $\mathcal{D}(0, T)$ is generated by the value-weighted voter model, provided that $\mathcal{D}(0, T)$ is sufficiently large. Therefore, our hot span detection problem is formalized as the following maximization problem.

$$\hat{S} = \arg\max_S \mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0, T), S), \tag{2}$$

where $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$ denote the maximum likelihood estimators for a given $S$.

In order to obtain $\hat{S}$ according to Equation (2), we need to prepare a reasonable set of candidate spans, denoted by $\mathcal{S}$. One way of doing so is to construct $\mathcal{S}$ by considering all pairs of observed activation time points. Then, we can construct a set of candidate spans by $\mathcal{S} = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}, t_2 \in \mathcal{T}\}$. Equation (2) can be solved by a naive method which has two iterative loops. In the inner loop we first obtain the maximum likelihood estimators, $\hat{\boldsymbol{w}}_n$ and $\hat{\boldsymbol{w}}_h$, for each candidate $S$ by maximizing the objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0, T), S)$ using the Newton method. In the outer loop we select the optimal $\hat{S}$ which gives the largest $\mathcal{L}(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h; \mathcal{D}(0, T), S)$ value. However, this method can be extremely inefficient when the number of candidate spans is large. Thus, in order to make it work with a reasonable computational cost, we consider restricting the number of candidate time points to a small value, denoted by $J$, i.e., we construct $\mathcal{T}_J = \{t_1, \cdots, t_J\}$ by selecting $J$ points from $\mathcal{T}$; then we construct a restricted set of candidate spans by $\mathcal{S}_J = \{S = [t_1, t_2] : t_1 < t_2, t_1 \in \mathcal{T}_J, t_2 \in \mathcal{T}_J\}$. Note that $|\mathcal{S}_J| = J(J-1)/2$, which is large when $J$ is large.

### 4.2 Proposed Method

It is easily conceivable that the naive method can detect the hot span with a reasonably good accuracy when we set $J$ large at the expense of the computational cost, but the accuracy becomes poorer when we set $J$ smaller to reduce the computational load. We propose a novel detection method below which alleviates this problem and can efficiently and stably detect a hot span from diffusion results $\mathcal{D}(0, T)$.

We first obtain the maximum likelihood estimators, $\hat{\boldsymbol{w}}$ based on the original objective function of Equation (1), and focus on the first-order derivative of the objective function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0, T))$ with respect to the value-parameter vector $\boldsymbol{w}$ at each individual opinion change time. More specifically, let $\boldsymbol{w}_t$ be the value-parameter vector at time $t \in \mathcal{T}$. Then we obtain the following formula for the maximum likelihood estimators due to the uniform parameter setting and the globally optimal condition.

$$\frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0, T))}{\partial \boldsymbol{w}} = \sum_{t \in \mathcal{T}} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0, T))}{\partial \boldsymbol{w}_t} = 0. \tag{3}$$

Now, we can consider the following partial sum for a given hot span $S = [T_1, T_2]$.

$$\boldsymbol{g}(S) = \sum_{t \in \mathcal{T} \cap S} \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0, T))}{\partial \boldsymbol{w}_t}. \tag{4}$$

Clearly, $\|\boldsymbol{g}(S)\|$ is likely to have a sufficiently large positive value if $S \approx S^*$ due to our problem setting. Namely, the hot span is detected as follows:

$$\hat{S} = \arg\max_{S \in \mathcal{S}} \|\boldsymbol{g}(S)\|. \tag{5}$$

Here note that we can incrementally calculate $\boldsymbol{g}(S)$. More specifically, let $\mathcal{T} = \{t_1, \cdots, t_N\}$ be a set of candidate time points, where $t_i < t_j$ if $i < j$; then, we can obtain the following formula.

$$\boldsymbol{g}([t_i, t_{j+1}]) = \boldsymbol{g}([t_i, t_j]) + \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(0, T))}{\partial \boldsymbol{w}_{t_{j+1}}}. \tag{6}$$

The computational cost of the proposed method for examining each candidate span is much smaller than the naive method described above. When $|\mathcal{T}| = N$ is very large, we construct a restricted set of candidate spans $\mathcal{S}_J$ as explained above. We summarize our proposed method below.

**1.** Maximize $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0, T))$ by using the Newton method.
**2.** Construct the candidate time set $\mathcal{T}$ and the candidate span set $\mathcal{S}$.
**3.** Detect a hot span $\hat{S}$ by Equation (5) and output $\hat{S}$.
**4.** Maximize $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0, T), \hat{S})$ by using the Newton method, and output $(\hat{\boldsymbol{w}}_n, \hat{\boldsymbol{w}}_h)$.

Here note that the proposed method requires likelihood maximization by using the Newton method only twice.

## 5 Experimental Evaluation

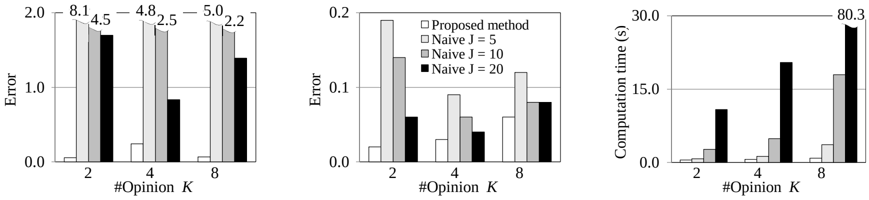
We adopted four datasets of large real networks. They are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [5], which has $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [4], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network). The third one is a network derived from the Enron Email Dataset [8] by extracting the senders and the recipients and linking those that had bidirectional communications. It has $4,254$ nodes and $44,314$ directed links (the Enron network). The fourth one is a coauthorship network used in [11], which has $12,357$ nodes and $38,896$ directed links (the coauthorship network).

For each of these networks, we generated opinion diffusion results for three different values of $K$ (the number of opinions), i.e., $K = 2$, 4, and 8, by choosing the top $K$ nodes with respect to node degree ranking as the initial $K$ nodes and simulating the model mentioned in section 2 from 0 to $T = 25$. We assumed that the value of all the opinions were initially 1.0, i.e. the value-parameters for all the opinions are 1.0 for the normal span, and further assumed that the value of the first opinion changed to double for a period of $[10, 15]$, i.e. the value-parameter of the fast opinion is 2.0 and the value-parameters of all the other opinions are 1.0 for the hot span. We then estimated the hot span and the value-parameters for both the spans (normal and hot) by the two methods (the proposed and the naive), and compared their accuracy and the computation time. We adopted $1,000$ as the value of $J$ (the number of candidate time points) for the proposed method, and 5, 10, and 20 for the naive method.

Figures 1 and 2 show the experimental results[2] where each value is the average over 10 trials for 10 distinct diffusion results. We evaluated the accuracy of the estimated hot span $[\hat{T}_1, \hat{T}_2]$ by the absolute error $|\hat{T}_1 - T_1| + |\hat{T}_2 - T_2|$, and the accuracy of the estimated opinion values $\hat{\boldsymbol{w}}$ by the mean absolute error $\Sigma_{i=1}^{K}(|\hat{w}_{in} - w_{in}| + |\hat{w}_{ih} - w_{ih}|)/K$, where $w_{in}$ and $w_{ih}$ are values of opinion $i$ for the normal and the hot spans, respectively.

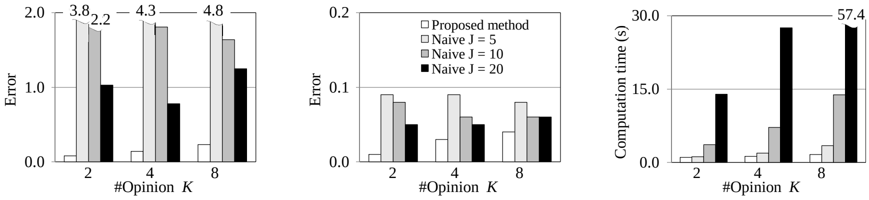
From these results, we can find that the proposed method is much more accurate than the naive method for both the networks. The average error for the naive method decreases as $J$ becomes larger. But, even in the best case for the naive method ($J = 20$), its average error in the estimation of the hot span is maximum about 30 times larger than that of proposed method (in the case of the Enron network under $K = 2$), and it is maximum about 6 times larger in the estimation of value-parameters (in the case of

[2] We only show the results for the two networks (Enron and coauthorship) due to the space limitation. In fact, we obtained similar results also for the other two networks (blog and Wikipedia).




(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 1: Comparison on the Enron network




(a) Error of $[\hat{T}_1, \hat{T}_2]$ (b) Error of $\hat{\boldsymbol{w}}$ (c) Computation time

Fig. 2: Comparison on the coauthorship network

the coauthorship network under $K = 2$). It is noted that the naive method needs much longer computation time to achieve these best accuracies than the proposed method although the number of candidate time points for the naive method is 50 times smaller. Indeed, it is about 20 times longer for the former case, about 13 times longer for the latter case, and maximum about 95 times longer for the whole results (in the case of the Enron network under $K = 8$). From these results, it can be concluded that the proposed method is able to detect and estimate the hot span and value-parameters much more accurately and efficiently compared with the naive method.

## 6 Conclusions

In this paper, we addressed the problem of detecting the unusual change in opinion share from the observed data in a retrospective setting, assuming that the opinion share evolves by the value-weighted voter model with multiple opinions. We defined the hot span as the period during which the value of an opinion is changed to a higher value than the other periods which are defined as the normal spans. A naive method to detect such a hot span would iteratively update the pattern boundaries that form a hot span (outer loop) and iteratively update the opinion value for each hot span candidate (inner loop) such that the likelihood function is maximized. This is very inefficient and totally unacceptable. We developed a novel method that avoids the inner loop optimization during search. It only needs to estimate the value twice by the iterative updating algorithm (Newton method), which can reduce the computation times by 7 to 95 times, and is very efficient. We applied the proposed method to opinion share samples generated from four real world large networks and compared the performance with the naive method that considers only the randomly selected boundary candidates. The results clearly indicate that the proposed method far outperforms the naive method both in terms of accuracy and efficiency. Although we assumed a simplified problem setting in this paper, the proposed method can be easily extended to solve more intricate problems. As the future work, we plan to extend this framework to spatio-temporal hot span detection problems.

## Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

## References

1. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear *q*-voter model. Physical Review E 80, 041129 (2009)
2. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and sociai infiuence in online communities. In: Proceedings of KDD 2008. pp. 160–168 (2008)
3. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74, 056108 (2006)
4. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08). pp. 1175–1180 (2008)
5. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, 9:1–9:23 (2009)
6. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10). pp. 1364–1370 (2010)
7. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 91–101 (2002)
8. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). pp. 217–226 (2004)
9. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). pp. 228–237 (2006)
10. Liggett, T.M.: Stochastic interacting systems: contact, voter, and exclusion processes. Spriger, New York (1999)
11. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
12. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, 178701 (2005)
13. Swan, R., Allan, J.: Automatic generation of overview timelines. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000). pp. 49–56 (2000)
14. Wu, F., Huberman, B.A.: How public opinion forms. In: Proceedings of WINE 2008. pp. 334–341 (2008)
15. Yang, H., Wu, Z., Zhou, C., Zhou, T., Wang, B.: Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80, 046108 (2009)