

# Detecting Influence in Blog Networks

Siavash Kazemian, Tong Wang

## Contents

<b>1</b>	<b>Problem Definition</b>	<b>2</b>
<b>2</b>	<b>The Shuffle Test</b>	<b>3</b>
<b>3</b>	<b>Data Pre-processing</b>	<b>4</b>
3.1	Detecting Language . . . . .	4
3.2	Data Structure and Serialization . . . . .	4
3.3	Tokenizing Texts . . . . .	5
<b>4</b>	<b>Building Connectivity</b>	<b>7</b>
4.1	Connectivity based on Common Outlines . . . . .	7
4.2	SVD on the Link-based Connectivity . . . . .	8
<b>5</b>	<b>Defining Activation</b>	<b>8</b>
5.1	Link-based Activation . . . . .	9
5.2	Text-based Activation . . . . .	9
<b>6</b>	<b>Experiments and Results</b>	<b>10</b>
<b>7</b>	<b>Conclusion</b>	<b>12</b>
<b>A</b>	<b>Spam Users Detected by Connectivity based on Common-Outlinks</b>	<b>12</b>

# 1 Problem Definition

Detecting influence and correlations between members in a social network is an interesting task with many important applications. For a promotion event by a retail business, for example, if only a limited number of “freebies” are available to give out to the customers, it would be most cost-effective if the knowledge of which customers are most influential is available.

There are of course more than one way of viewing influence in a social network. Particularly, it could be based on the pattern under which certain behaviors spread within the social network in question. Given a deterministic structure of connectivity, one can firstly observe how people start certain behavior (or getting *activated*) at different time. The basic idea here is, if the spreading of activation is somehow dependent on the time sequence of an observation, then we say such a behavior is due to the influence between the members in the network more than anything else. This hypothesis has been validated by simulated sequences of activation among users in an artificially-constructed social network by Anagnostopoulos et al. (2008). In the same study, the authors also claimed that the use of particular words to tag photos on the Flickr website does not exhibit influence among the users.

The two critical questions to answer about influence detection are connectivity and activation. Connectivity is how each member within a network is connected to each other. More often than not, it is helpful to view the network as a *directed* graph, with its vertices corresponding to members and edges pointing in the same direction as the possible flow of influence, if any. Such information on connectivity is sometimes readily available from the data, but this is not always the case. For data that do not explicitly specify connectivity, building one is itself a rather challenging task.

As for activation, it is worth noting that the existence and detection of influence is not only dependent on the growth pattern of activation, but also how you define the event in which a member is said to be activated. For example, if we are to detect influence from what people write in blogs, then the gradual spread of usage of a neologism is possibly due to influence, whereas discussions of the latest US election are more likely due to common interests or other source of correlation among users.

Given connectivity and a definition of activation, one can then observe the activation distribution among the data at different time stamps. Based on these observations, it is then possible to employ various statistical methods to formalize and solve the problem of influence detection.

## 2 The Shuffle Test

The shuffle test is a way of detecting influence within a social network. It requires the following three types of information: the connectivity of the network, definition of activation, and the distribution of activated users in the network at each time stamp. The basic idea is, if there exists influence of a certain behavior in the network, then changing the timing of each activation should result in a different growth pattern of activations throughout the whole network. The shuffle test is designed so that it can render a permutation of time stamps and quantify the difference of growth pattern before and after.

Specifically, for any given member in a social network, the decision of whether to get activated (i.e., taking on a certain behavior) at time  $t$  is probabilistically determined by how many activated friends (or neighbors in the graph) he/she has at that time. The probability is given by a sigmoid function as follows:

$$p(a) = \frac{e^{\alpha \ln(a+1) + \beta}}{1 + e^{\alpha \ln(a+1) + \beta}} \quad (1)$$

where  $a$  is the number of activated friends and  $\alpha, \beta$  are correlation coefficients to be estimated later. Note that although the time information is not explicit in Equation 1, it is hidden in  $a$ , i.e.,  $a = a(t)$  since the number of activated friends of a given member will vary from time to time. The basic assumption of shuffle test is that the presence of influence can be determined by the time-sensitivity of the estimation of  $\alpha$ , i.e., its estimation is significantly different for different permutations of the time stamps, then we say influence exists for the given behavior among the members.

Now we can view a sequence of events for activation as a Bernoulli trial and estimate  $\alpha$  by maximizing the probability of the trial given by Equation 2:

$$\prod_a P(a)^{Y_a} (1 - p(a))^{N_a} \quad (2)$$

Here,  $Y_a$  is the number of users getting activated when they have  $a$  activated friends and  $N_a$  is the number of the rest. During implementation, both numbers have to be summed over all possible time stamps, i.e.,  $Y_a = \sum_t Y(a, t)$ , where  $Y(a, t)$  is the number of members with the above-mentioned activation behavior at time  $t$ . A note-worthy complication in computing  $Y(a, t)$  is that, the ‘‘history’’ of activation for users must be taken into account, since those members contributing to  $Y(a, t)$  must be ones that become activated *at* time  $t$  rather than *up until*  $t$ . This is shown in the algorithm in Figure 1.

```

48  /**
49  * get number of activated users who have a activated friends
50  */
51  public static int[] getYaNa(SubGroup group, String activationTerm, int a, TimeSequence ts) {
52      int[] result = new int[2];
53      /*
54       * Starting from t0 (under any permutation specified by ts)
55       */
56      Calendar time = ts.getStartingTime();
57      /*
58       * roll over all possible time stamps; starting from t1 instead of t0, since Ya = N.A. at t0
59       */
60      while ((time = ts.rollTimeStamp(time, 1)) != null) {
61          /*
62           * skip users that do not have 'a' friends
63           */
64          int alreadyActivatedFriends = getNumberOfAccumulatedActivatedUsers(
65              activationTerm, group, ts.rollTimeStamp(time, -1), ts);
66          if (alreadyActivatedFriends != a) {
67              continue;
68          }
69          /*
70           * get number of newly activated users at time t
71           */
72          int newlyActivatedAtTimeT = getNumberOfActivatedUsers(
73              activationTerm, group, time, ts);
74          result[0] += newlyActivatedAtTimeT;
75          result[1] += group.getMemberIdList().size()
76                      - alreadyActivatedFriends - newlyActivatedAtTimeT;
77      }
78      return result;
79  }
80  }

```

Figure 1: Accounting for activation history in computing  $Y(a, t)$

## 3 Data Pre-processing

### 3.1 Detecting Language

Upon examining the 4 Gb of blog posts that were given to us, we immediately noticed that the data was in many language. We decided to only work with the English portions as we do not know how to clean up the text in other languages. To do this, we concatenated the blog title and post title for each posting to form one UTF-8 string. We then ensured that for each character, the byte representation of the mentioned string is in the ASCII range.

### 3.2 Data Structure and Serialization

Our blog data consist of two files, one including in-links from April 25 to May 10, 2008, and the other contains all blog posts between May 1 and May 5, 2008. The first one is relatively smaller in size (about 10M) and is directly usable for constructing connectivity later in Section 4. The posts file, however, is about 4GB in size and not quite readily employable in the experiment, especially given our needs for frequent random access of a large number of users, their post contents, and the time stamps for posting. Consequently, as a pre-processing step, we make a single pass through the 4GB file, and each English blog post encountered will

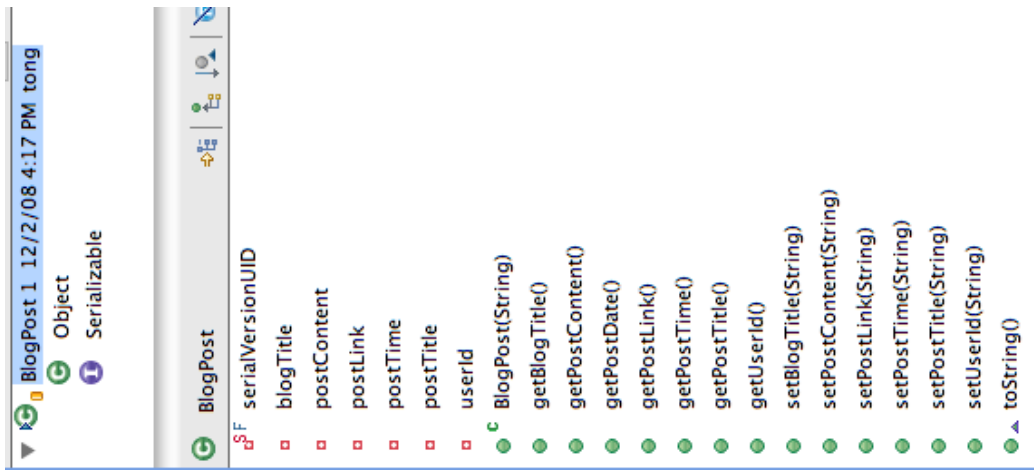


Figure 2: Customized data structure for blog posts

populate our customized data structured BlogPost (Figure 2), organized under the same author, and serialized to disk for future access.

Since blog user names are not available from the post data, the field `userId` in Figure 2 is determined by the domain name and the first part of the host name of the post link, e.g., link “`http://someone.blogspot.com`” would yield “`someone.blogspot`” as user id; this can also accommodate for blog links in the form of “`http://www.someblog.com/someone`”.

It is worth noting that, due to the enormous number of posts contained in the post file, the number of serialized files will very quickly grow unwieldy. To prevent the generated files from proliferating the directory, blog author names are hashed into  $26 \times 26$  sub-folders, each corresponding to a combination of two letters from the English alphabet, ranging from `aa` to `zz`. User id’s are then hashed into these slots according to the first two characters in the id; those beginning with numbers would be hashed to `aa`. The implementation keeps  $26 \times 26$  buffers for this hash table structure, and dumps a buffer once the number of encountered posts hashed to that particular buffer hits the buffer capacity (800 in our implementation). The code is shown in Figure 3.

### 3.3 Tokenizing Texts

Collecting accurate statistics about the term types used in the blog posts requires some preprocessing. We removed all the punctuation marks from the text. Hy-

```

152 public void add(BlogPost post) {
153     if (post == null) {
154         return;
155     }
156     if (post.getUserId() == null) {
157         return;
158     }
159     if (post.getPostTitle() == null && post.getPostContent() == null) {
160         return;
161     }
162     int hashSlot = Util.hashUserId(post.getUserId());
163     this.blogPosts[hashSlot][cursor[hashSlot]++] = post;
164
165     /*
166      * if the buffer for the userId is full
167      */
168     if (cursor[hashSlot] == BUFFER_CAPACITY) {
169         /*
170          * dump the buffer to serialization file
171          */
172         dumpBuffer(hashSlot);
173         /*
174          * clear the buffer
175          */
176         Arrays.fill(this.blogPosts[hashSlot], null);
177         /*
178          * and reset the buffer cursor.
179          */
180         cursor[hashSlot] = 0;
181     }
182 }

```

Figure 3: Buffering and hashing user id's

phens were left in the text so that words such as “bi-gram” would stay together as one type and not separated to the morphemes that constructed them. Finally, we converted all the words used in the body of text to lower-case.

It is customary to remove stopwords in many text mining tasks. Here we also saw a benefit in removing the stopwords for the following reason. Stopwords are usually very common and so it is unlikely that a person's usage of a stopword is influenced by his/her friends behavior in the network. But because stopwords are very common, there might be a correlation between the usage pattern of this class of words amongst friends in the network that is not caused by influence. The purpose of this paper is to study influence and so looking at the usage pattern of stopwords is unlikely to yield interesting results; so we have decided to remove them for our data in this study.

## 4 Building Connectivity

As stated earlier in Section 1, connectivity is one of the two major issues to address in detecting influence. Unlike the Flickr data used by Anagnostopoulos et al. (2008), the blog data we use do not endorse a easy-to-capture notion of connectivity. In other words, it is not easy to answer the question of how bloggers are connected with one another, and subsequently, how they exert influence through the available channels based on this connectivity.

Nonetheless, two paradigms in building connectivity among bloggers are introduced in this section. Both approaches use the in-link file containing in-link postings of all bloggers during a specific period of time. The first approach introduced in Section 4.1 draws on the intuition that bloggers who link to the same web pages or websites can be considered “close” in the graph; the bigger the number of such common out-links is, the closer the bloggers are in terms of connectivity.

In view of the sparseness of using specific out-links, the second approach (Section 4.2) reduces the dimension of links into a specific number of topics by applying SVD on the link structure, and then resorting back on the hypothesis from the first approach that bloggers with common interests should be grouped together.

### 4.1 Connectivity based on Common Outlines

The proposed method groups together bloggers who share common out-links. Firstly, we believe that people’s linking behavior is indicative of their interests. That is, given a graph  $G = \langle V, E \rangle$  of inter-connected blog users, where the vertices are users and edges pointing in the direction of possible influence flow. We classify blog users into two classes, i.e., *topics* and *subscribers*. If a topical blog  $v_t$  is shared by a group of subscribers, then the set of subscribers  $\{v \in V : (v, v_t) \in E\}$  is considered connected to each other since they share common interests in the same topic  $v_t$ .

Secondly, we believe that the more common topics a certain group of subscribers share, the more closely related they are in terms of connectivity. On the one hand, there will be a great number of subscribers linking to a small number (say, one or two) of common topics; on the other hand, subscribers sharing five or more topics would be really rare. Suppose there exist subscribers sharing  $T$  topics but no two subscribers share  $T + 1$  topics, we then focus on selecting groups of subscribers who share top  $k$  numbers of topics. That is, an allowable set of common topics  $S_t$  and its corresponding group of subscribers  $S_s$  should satisfy the following properties:

- $\forall v_s \in \mathcal{S}_s, \forall v_t \in \mathcal{S}_t, (v_s, v_t) \in E$
- $|\mathcal{S}_t| \geq T - k$
- $|\mathcal{S}_s| \geq 2$

Unfortunately, (and most interestingly), what this notion gave us is, instead of groups of “normal” bloggers, a list of spam sites within the blog data. This means that, normal bloggers do not link quite often to other bloggers they are interested in; it is the spam blogs that constitute the most densely-connected part of the network. The list of detected spams can be found in Appendix A. When we removed the spams and run the experiment again, the value of  $T$  is 2, i.e., there are no two blog users linking to three common links at the same time. This led us to think of other ways of building connectivity discussed in Section 4.2.

## 4.2 SVD on the Link-based Connectivity

In the previous section, we talked about how blog users never link to more than two common out links. One of the possible reasons behind this is that the number of potential topics to link to is huge (theoretically, the number of blogs on the entire Web). This makes the connectivity quite sparse and not suitable for studying influence.

To overcome the data sparseness, we apply SVD on the graph to reduce the dimension of out-links (or topics). The intuition is to represent individual blogs by more coarse-grained topics through the use of SVD.

The linking matrix built from the in-link file is  $26720 \times 32628$  in dimension after all the pre-processing steps. We tried several Java libraries for SVD, but due to their poor support either for sparse matrix computation or for SVD, we eventually used Matlab for the task. We used 9 as the size of matrix  $\Sigma$  in SVD because it generates the best sizes of subscriber groups. This corresponds to 9 topics and results in 9 subgroups of subscribers (see Section 4.1), each believed to be completely connected due to their common interests in the specific topic they all link to.

## 5 Defining Activation

Finding a good activation definition is absolutely essential in the study of influence in a network. In fact, the ability to detect influence in a network depends on the activation definition.

Given the set of all the possible measurable actions  $\mathcal{A}$  that the users in a network can do, only a subset  $\mathcal{B}$  of  $\mathcal{A}$  may have occurred due to influence exerted on a user by his/her friends in the network. If  $\mathcal{B}$  is non empty, then we can say that for some activation  $b \in \mathcal{B}$ , the network exhibits influence. Picking  $b$  however is not very straight forward as  $\mathcal{B}$  is always unknown. It then follows that failure to pick a good activation definition does not imply lack of influence in the network but simply states that the chosen activation definition is not influenced in the network.

Anagnostopoulos (Anagnostopoulos et al., 2008) used the act of tagging photos with a certain term type for the first time as their activation function. We explored two general directions for defining activation in our blog data: link-based and text-based definitions that are explained in the following two subsections.

## 5.1 Link-based Activation

Here, we define activation based on the linking behavior of blogger subgroups. A blogger is activated once he/she links to a set of predetermined group of users  $U_r = \{u_1, u_2, u_3, \dots, u_c\}$  that have the most in-links in our data and are outside the blogger subgroups. Our rationale for picking this activation definition is the following: If a user in the blogger subgroup links to  $u_i \in U$  then it is possible for other users in the same group to take note of this event, read  $u_i$  and start talking about it and link to it in their blogs.

Upon examining our data, we observed that the blogs with the most in-links are usually spams. In addition, after further examining our data, we observed that the linking behavior of users in each group was very sparse: users in a group link to outside pages an average of 1.5 times. Due to the sparseness of these connections, we can not measure influence using the methods introduced in Section 2 while utilizing an activation definition based on these connections. So we decided to pursue a different direction in order to find a good activation definition.

## 5.2 Text-based Activation

There are many choices for text-based activation definition. One could pick the first instance of using any class of term types as the activation function. Example of such classes are *named identities*, nouns, and terms with high IDF scores. It is not difficult to rationalize these choices. But we chose to pick an activation that is more directly related to our definition of influence.

We made the following observations. Terms whose usages were caused by influence exerted in the network occur more frequently in the blog posts as time moves forward. For this reason, we defined activation terms as term types with the highest *M-score*:

$$M\text{-Score} = \sum_{t=1}^{T-1} a_w(t+1) - a_w(t), \text{ for } a_w(t+1) > a_w(t) \quad (3)$$

where  $T$  is the total number of days blogs data was collected, and  $a_w(t)$  is the number of users who used term  $w$  for the first time at time  $t$ . As it can be seen, terms that will have a high *M-Score* are those whose usage increases from one day to the next amongst users that had not previously used them in their blog posts. For our experiments, we picked 20 terms with the highest *M-Score*.

## 6 Experiments and Results

Having picked 9 groups as described in Section 4.2, and 20 activation terms per group as described in Section 5.2 we now describe how influence was measured our data. We chose the same probability density function of activation as Anagnostopoulos et al. as shown in Equation (1) to represent the probability of a user being activated after he has  $a$  activated friends. We then used maximum likelihood logistic regression to estimate  $\alpha$ , the correlation coefficient that shows how well this probability density fit our data. This was done using an implementation of Brent’s method which combines a golden-section search and parabolic interpolation, which was downloaded from the web.

To detect influence, we do the shuffle as described in Section 2 (randomly assigning the activation times to the users in the network) and calculate an  $\alpha'$ . Here,  $\alpha'$  represents the how well the probability density fits our data with randomized activation times. Randomizing activation times disables the probability function to know the number of activated friends  $a$  that a user has before activation. Then  $\alpha'$  shows how well the probability density fits our data without having the correct information about  $a$ . But if the network truly exhibits influence, knowing  $a$  for each user should be essential in predicting whether a user will be activated or not. From this, we can deduce that if there is real influence in our data, then the original calculation of the probability function should fit our data much better than the shuffled version, leading to  $\alpha$  being significantly greater than  $\alpha'$ .

For our data, we calculated a total of 180 pairs of  $\alpha$  and  $\alpha'$  that resulted in two

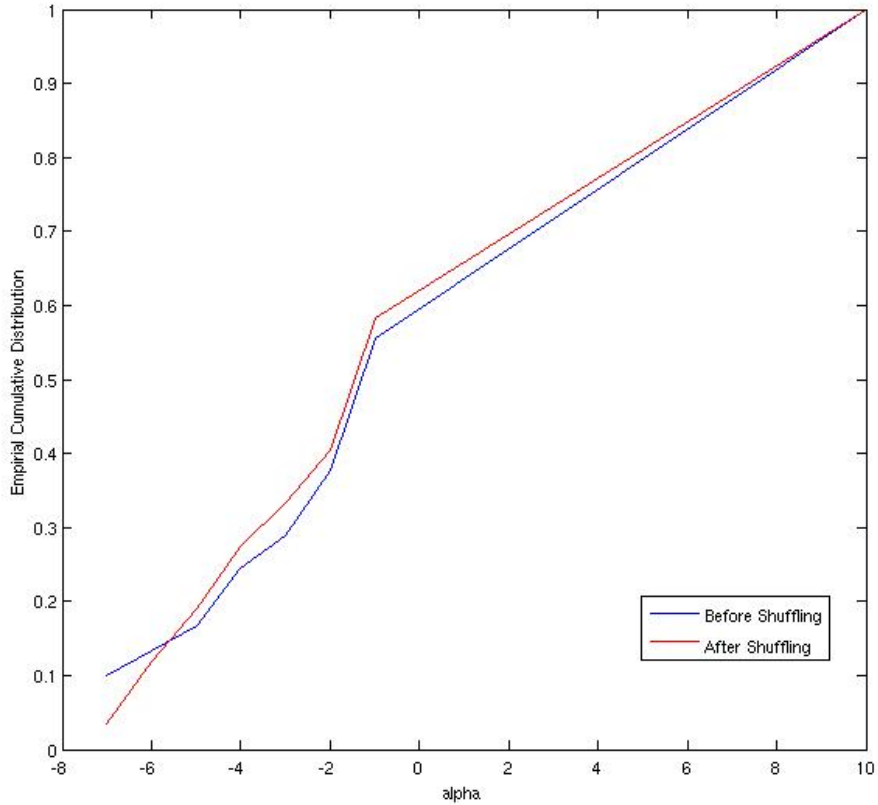


Figure 4: Empirical Cumulative Distribution of  $\alpha$  before and after shuffling.

distributions. We then derived the Empirical Cumulative Distribution function for both  $\alpha$  and  $\alpha'$ . Figure 4 shows these results.

As you can see in Figure 4, the two curves follow each other rather closely. If there was influence in our data, the CDF curve referring to the shuffled data would increase faster when  $\alpha$  is smaller while the curve representing the original data would increase faster at higher  $\alpha$  values which is not the case in Figure 4. This suggests that there is no influence in our data given our user grouping and activation definition. However, one point should be made about our data before making any conclusions about our activation definition. Our data is the collection of only 5 days of blogs posted on the web. It can be argued that it takes much

longer for bloggers to influence each other. If this argument holds, then these experiments should be repeated on blog data collected in a longer time span. In any case, we can safely argue that for our current data set, our choice of activation did not exhibit influence.

## 7 Conclusion

Studying influence in blogs' networks has many complications. Unlike Flickr data used by Anagnostopoulou et al., our blog data did not connect bloggers in *friend networks*. We had to come up with our own definition of connectivity based on the links present in different blogs. Furthermore, there are many choices available for definition of activation. Our choice of activation function did not seem to exhibit influence in our data, however one can not draw the conclusion that our data does not have influence.

We have learned that coming up with an activation definition can be very tricky. One should carefully understand the underlying data before coming up with an activation definition. For instance, good activation definition will be different for data that is collected over a long span of time as opposed to data that is collected in a short time. Our data was collected only over 5 days. This time span maybe too short for exhibiting influence but a more careful study is required to justify this claim.

## A Spam Users Detected by Connectivity based on Common-Outlinks

businessduatujuh.blogspot	jayden9787.blogspot
leeannfxrz.livejournal	scottgletji36.blogspot
izaiahkelgn67.blogspot	denunciatordetb.blogspot
emmettkeloi53.blogspot	jorgeregite22.blogspot
kalinara.blogspot	kaleighhmj3.livejournal
businessempatlima.blogspot	businessstigapuluh.blogspot
businessstujuhbelas.blogspot	rljg119.blogona
plsa955.mastersubmit	lizetteixab.livejournal
sweetperdition.wordpress	kaedencavsc96.blogspot
jayden1794.blogspot	beautychick101.blogspot

lincolnpeyc78.blogspot  
businesslimapuluh.blogspot  
jeanettevpnf74.blogspot  
businesslima.blogspot  
bakingncooking.wordpress  
dad-baker.blogspot  
bloggerview.com/qint401  
ronaldtrto178.blogspot  
max4971.blogspot  
sadiereywsz56.blogspot  
historiasruivas.blogspot  
nataliedz5.livejournal  
businessempatdelapan.blogspot  
cedricsbigmix.blogspot  
ai-meus-sais.blogspot  
daniellepd3283.blogspot  
makena5qw1.livejournal

karleyadhnf66.blogspot  
thomasfriedmanisagreatman.blogspot  
blogking.ch/tpmb927  
makeuplovesme.com  
leonardo9646.blogspot  
fashionablekiffen.blogspot  
businessenam.blogspot  
ian6806.blogspot  
businessstigabelas.blogspot  
dorientafpr53.blogspot  
businesssembilan.blogspot  
businessenambelas.blogspot  
pigd363.blogpear  
businessduapuluh.blogspot  
amyb587.livejournal  
businessduatiga.blogspot

## References

- A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. 2008.