Contents

[1. The Idea 2](#_Toc387639283)

[2. Motivation 2](#_Toc387639284)

[2.1 Applications: 2](#_Toc387639285)

[2.2 Use of Twitter: 2](#_Toc387639286)

[2.3 The amount of data: 3](#_Toc387639287)

[2.4 Finding the relative influence between users: 3](#_Toc387639288)

[3. Our Approach 4](#_Toc387639289)

[3.1 Defining Influence 4](#_Toc387639290)

[3.1.1 Katz relative influence: 4](#_Toc387639291)

[3.1.2 The problem with applying Katz theory directly in a Twitter network: 4](#_Toc387639292)

[3.2 Our Algorithm 5](#_Toc387639293)

[3.3 Our Implementation 7](#_Toc387639294)

[4. Experiments and Results 10](#_Toc387639295)

[4.1 Dataset: 10](#_Toc387639296)

[4.2 Experimental Setup: 10](#_Toc387639297)

[4.3 Results and Analysis: 11](#_Toc387639298)

[4.3.1 Analysis 1: Proof of Correctness 11](#_Toc387639299)

[4.3.2 Analysis 2: Effect of damping factor 12](#_Toc387639300)

[4.3.3 Analysis 3: Proof of scalability 12](#_Toc387639301)

[5. Conclusion 17](#_Toc387639302)

[6. References 18](#_Toc387639303)

**TwitterFluence**

**Finding the centers of influence in Twitter**

# 1. The Idea

There is a theory that Kevin Bacon is the most influential person in Hollywood since any individual involved in the Hollywood film industry can be linked through his or her film roles to Kevin Bacon within six steps. We aim to find the most influential people - the Kevin Bacon(s) of Twitter. In terms of centrality theory, our objective is to find the centers of influence in a social network.

# 2. Motivation

## 2.1 Applications:

There are various applications of finding the centers of influence in a given network:-

* A major application of this is to find how influential a person is in a social network. Finding an individual who can influence the most number of people is an integral part of directed advertising and brand building.
* The concept can even be extended to real networks such as a road transport network. Finding the roads that are the most used is a major challenge in traffic management.
* Search engines find the most relevant pages for a given search query. By defining the relative relevance between pages based on the relevance of pages that link to it, the problem can be translated to finding the centers of "relevance" over the internet. In fact, this is exactly how Google's PageRank algorithm works.

## 2.2 Use of Twitter:

Twitter is an online microblogging service that enables registered users to post very

short updates, comments or thoughts also known as “tweets” of up to 140 characters [1]. The main reason why we chose Twitter is its relationships between users. Twitter advocates a follower-followed relationship between users which can be visualized as a directed graph from one follower to followed. This is in stark contrast to the undirected friend-friend relationships of Facebook or Orkut. Besides most of Twitter's data is publicly available on the internet without privacy issues.

## 2.3 The amount of data:

The sheer amount of data currently on Twitter makes this an extremely interesting problem in scalability. The following data is compiled from <http://www.statisticbrain.com/twitter-statistics/> as well as recent Twitter reports:-

|  |  |
| --- | --- |
| **Twitter Statistics** | **Data (1.1.2014)** |
| Total number of active registered Twitter users | 645,750,000 |
| Number of new Twitter users signing up everyday | 135,000 |
| Average followers per user | 200 |
| Number of follower-followed relationships | 129 billion |

Finding the centers of influence in a social network of this size is a problem that can definitely not be solved using a single machine.

## 2.4 Finding the relative influence between users:

Another interesting problem is how does one compare the relative influence between users especially ones separated by billions of edges.

# 3. Our Approach

## 3.1 Defining Influence

### 3.1.1 Katz relative influence:

In Social Network Analysis (SNA) there are various measures of centrality which determine the relative importance of a node within the network. Katz centrality was introduced by Leo Katz in 1953 and is used to measure the degree of influence of an actor in a social network[2].

Katz centrality computes the relative influence of a node within a network by measuring the number of the immediate neighbors (first degree nodes) and also all other nodes in the network that connect to the node under consideration through these immediate neighbors. Connections made with distant neighbors are, however, penalized by an attenuation factor α[3].

* Mathematically,

I(A) = n(followers(A)) + α n(followers(followers(A)) + α2 n(followers(followers(followers(A))) …

* In other words,

I(A) = αk - 1 I(followeri(A)) for the kth iteration

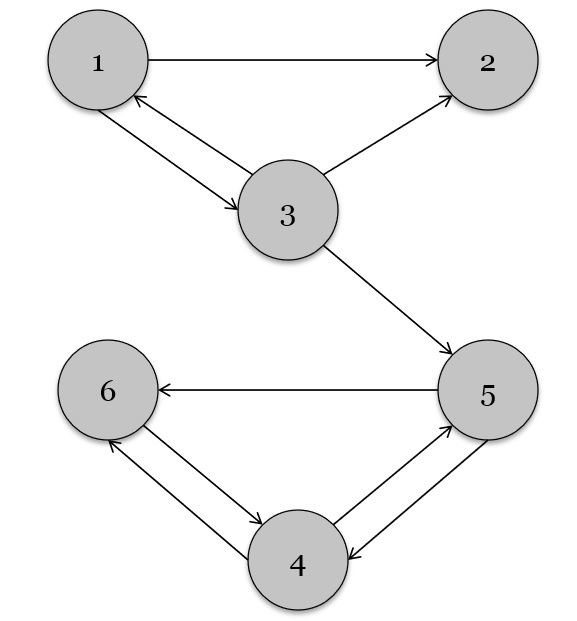
### 3.1.2 The problem with applying Katz theory directly in a Twitter network:

The main problem with applying Katz theory to Twitter is that given a user B and his follower A, if A follows n people other than B, then according to Katz theory, each of its followers gets its entire influence. However, we argue that given that B tweets, A is only 1/n likely to retweet it since A's attention is divided among n users.

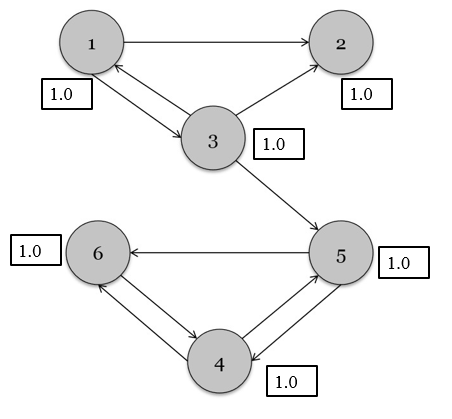
So, we define I(A) = αk - 1 I(followeri(A))/n(followers(Ai)) for the kth iteration.

## 3.2 Our Algorithm

We explain our algorithm based on the following initial graph:-



**Step 1:** Initialize all nodes of the graph with some initial influence value. For instance, in this example we initialize each node with an initial influence value of 1.0.



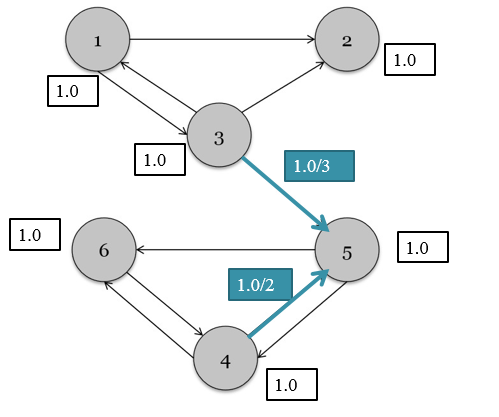
**Step 2:** Now, for each node Ai, calculate its influence on all the nodes that it follows in 3 steps:-

2.1) Count = number of nodes that Ai is following.

2.2) Influence of Ai on all its direct followers = Influence of Ai/Count

2.3) Send this value of Ai to all its direct neighbors

For instance, in the example graph, node 3 calculates its following count as 3 and its influence on its direct neighbors as 1.0/3. It sends this value to node 5. Similarly, node 4 sends the value 1.0/2 to node 5.

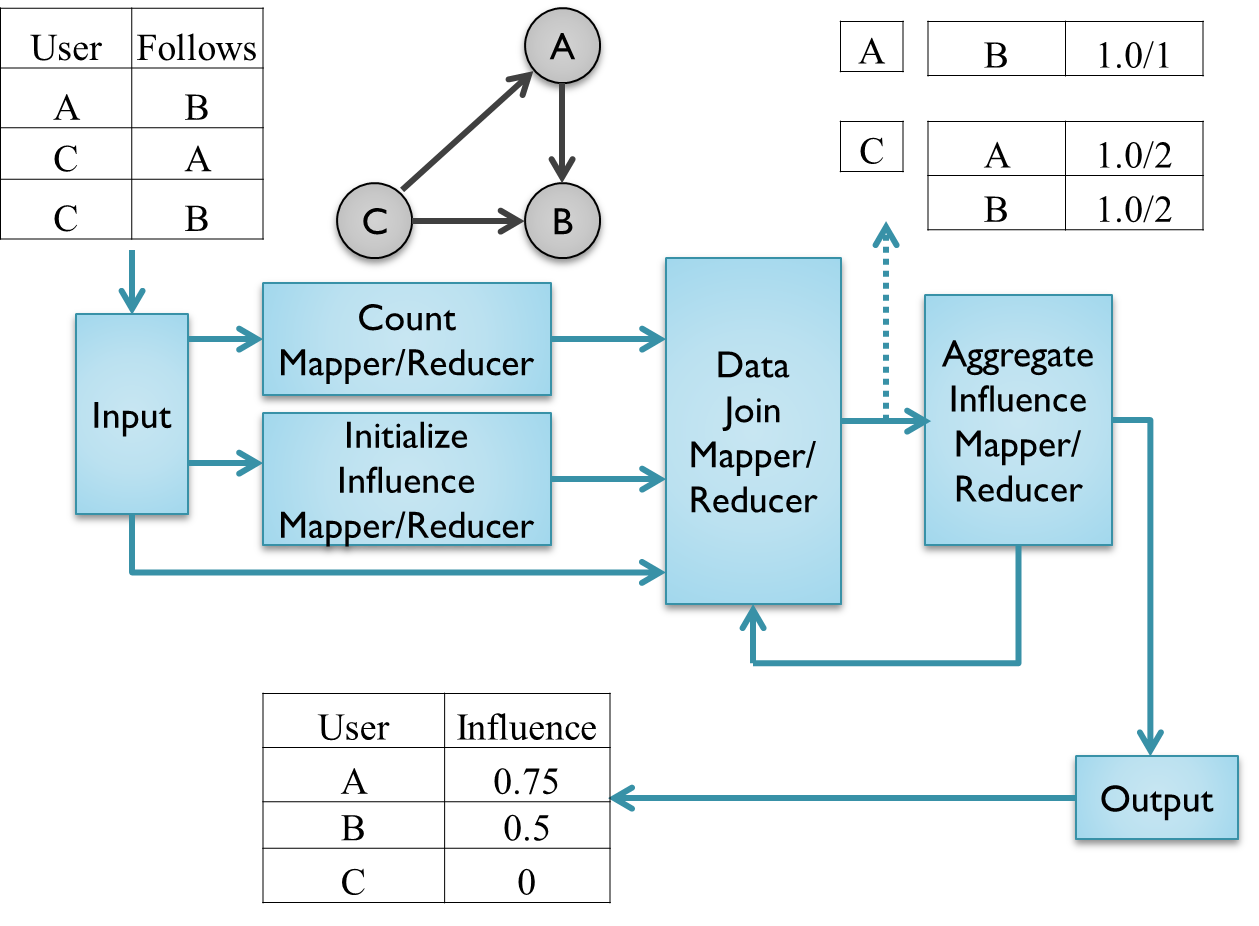


**Step 3:** Each node receives the influence values from all its followers and calculates its own influence value as 1- α + α (value from followeri). Here, α is a damping factor that reduces the effect of distant followers from the current node. 1- α is a randomness factor similar to the one used in the PageRank[4] algorithm in order to overcome the problem of cycles in the network.

For instance, in the graph, influence of 5 is calculated as 1 as 1- α + α(0.33 + 0.5)

6) Repeat from steps 2 and 3 until the maximum change in influence for all the nodes is less than some ∆.

## 3.3 Our Implementation

****

* **Initial Input data:** Our input is as follows:

Follower \t Followed\n

Eg:

|  |  |
| --- | --- |
| 12 | 1 |
| 12 | 3 |
| 12 | 5 |
| 4 | 6 |

User id 12 is follows user ids 1,3 & 5 while user id 4 follows user id 6.

* **Count Mapper/Reducer:** This module outputs the count of followed for each follower in the following format:

Follower \t Count \n

Eg:

|  |  |
| --- | --- |
| 12 | 3 |
| 4 | 1 |

User id 12 follows 3 users while user id 4 follows 1 user.

* **Initialize Influence Mapper/Reducer:** This module simply initializes the influence values to 1.0 for all users. The format is as follows:-

User \t 1.0\n

Eg:

|  |  |
| --- | --- |
| 12 | 1.0 |
| 4 | 1.0 |

Influence value of both users 12 and 4 is 1.

* **Data Join Mapper/Reducer:** This module joins the input from the count and initialize influence modules as well as the initial input module into a single record and outputs a set of user influence pairs. It consists of:-
* **Data Join Mapper:** It combines data from the 3 modules into a single record in the following format:-

UserID \t Follows Count \t Influence \t List of user this user id follows\n

Eg:

|  |  |  |  |
| --- | --- | --- | --- |
| User ID | Number of users followed | Influence | Following |
| 12 | 3 | 1 | 1, 3, 5 |
| 4 | 1 | 1 | 1 |

User 12 follows 3 user, its influence is 1.0 and it will send Influence/Follows Count (i.e. 0.33) to all the 1,3 and 5.

* **Data Join Reducer:** It uses the combined data to create pairs of user-influence keys. This effectively simulates the sending of each user's influence to its follower. The format is as follows:-

UserID \t Distributed Influence \n

Eg:

|  |  |
| --- | --- |
| 1 | 0.33 |
| 3 | 0.33 |
| 5 | 0.33 |
| 1 | 1 |

* **Aggregate Influence Mapper/Reducer:** Input to this module is the output of the Data Join module. This module simply sums up the influence value of each user while taking the damping factor into consideration. The format is as follows:-

UserID \t Final influence value\n

Eg

|  |  |
| --- | --- |
| 1 | 0.665 |
| 3 | 0.165 |
| 5 | 0.165 |

The output of this module is looped back to the data join reducer along with the initial inputs of the graph and the following count for the next iteration**.** Thiscontinues until we find little to no change in the value of the influence for each node.

# 4. Experiments and Results

## 4.1 Dataset:

We used the entire Twitter dataset as of June 2009 as the input for our algorithm.

|  |  |
| --- | --- |
| **Source** | <http://an.kaist.ac.kr/traces/WWW2010.html> |
| **Size** | 24.5 GB |
| **Number of users** | 41.7 million |
| **No. of Follower-Followed relationships** | 1.47 billion |
| **Format** | Follower \t Followed\n  Eg: 2 1  User 2 is a follower of 1. |

## 4.2 Experimental Setup:

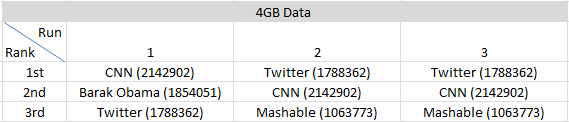
We used the HPCC cluster for our experiments with the following configuration:-

|  |  |
| --- | --- |
| **Processor on each node** | Dual Octocore Intel Xeon 2.4 GHz |
| **Memory on each node** | 64GB Memory |
| **Temporary disk size per node** | 1TB |
| **Total shared hard disk size** | 200GB |
| **Maximum Number of Node requested** | 64 |
| **Maximum Processes per node** | 2 |
| **Hadoop version** | 1.1.2 |
| **File sharing mechanism** | Using a shared file system mounted on all nodes |

## 4.3 Results and Analysis:

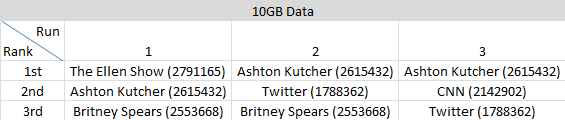
### 4.3.1 Analysis 1: Proof of Correctness

We have results with different data sizes that together prove that our implementation is correct. The first table is for 4BG of data, with1.7 million users and 0.28 billion follower-followed relationships.

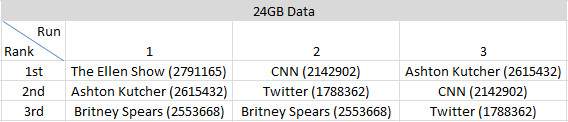


Here in the first iteration, we can clearly see that the CNN has the maximum followers and hence, correctly comes up as the most influential profile while in the second iteration, we are considering the 2nd level followers of each profile and in this case, Twitter's own profile trumps CNN.

The second table shows the same analysis for 10 GB of data with 4.4 million users and 0.6 billion relationships.



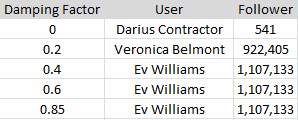
Finally, the following data is for the entire twitter dataset as of July 2009, with 41.7 million users and 1.47 billion relationships.



Based on the data from these 3 tables and comparing our results with the metadata received from the same source, we find that our algorithm correctly returns the influence values for each dataset.

### 4.3.2 Analysis 2: Effect of damping factor

We further analyzed the effect of varying the damping factor with an input size of 5, 4767 users and 10 million relationships:



The lesser the damping factor, the more the probability of a random user having an increased influence. This is exactly what we found in our experiment. For a damping factor of 0, the most influential user is Darius with merely 541 followers. However, as we increase the damping factor, the data tends to be more based on the number of followers and less on a random factor.

### 4.3.3 Analysis 3: Proof of scalability

#### 4.3.3.1 Scalability based on increased dataset size

We conducted timing experiments on our algorithm with increasing data sizes. Note that our implementation relies mainly on the number of relationships that it has to process and so, we conducted experiments with increasing relationships.

*Configuration parameters*

Compute Nodes: 16

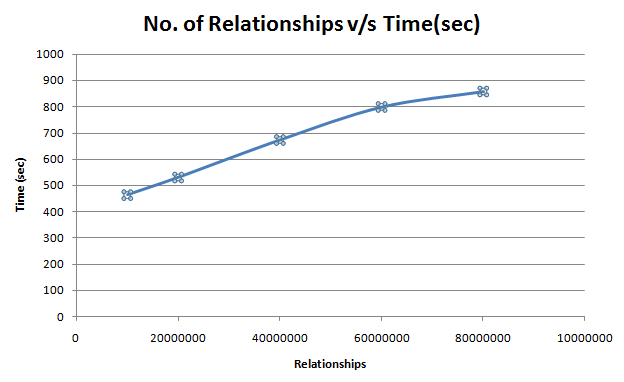
Processes per node: 2

Reducers: 32

Iterations: 5

**Experiment results:**

|  |  |  |  |
| --- | --- | --- | --- |
| **User** | **Relationships** | **Time(sec)** | **Factor** |
| 4767 | 10 M | 463 | 1.00 |
| 21377 | 20 M | 530 | 1.14 |
| 53982 | 40 M | 673 | 1.45 |
| 120212 | 60 M | 799 | 1.73 |
| 194830 | 80 M | 858 | 1.85 |



Our observation is that as we increase the number of relationships, the time taken by our implementation does not increase exponentially or even linearly. In fact, with an increase in data size by a factor of 8, the time increases by a mere factor of 1.85. This certainly proves that our algorithm is scalable with respect to data size.

#### 4.3.3.2 Scalability based on increased number of nodes

We then conducted timing experiments on our algorithm by increasing the number of compute nodes.

*Configuration parameters*

No. of Users: 4767

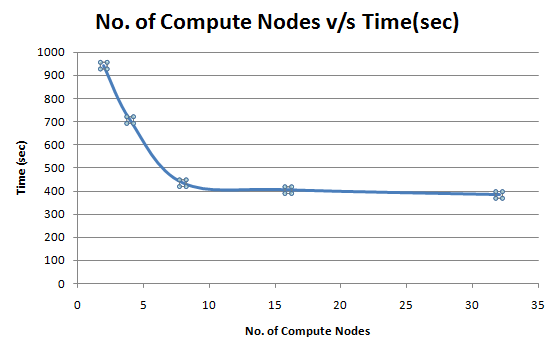
No. of relationships: 10,000,000

Processes per node: 2

No. of Runs: 5

**Experiment Results:**

|  |  |  |
| --- | --- | --- |
| **Nodes** | **Time(sec)** | **Speedup** |
| 2 | 942 | 1.00 |
| 4 | 707 | 1.33 |
| 8 | 434 | 2.17 |
| 16 | 404 | 2.33 |
| 32 | 383 | 2.46 |



Our observation is that as we increase the number of compute nodes, the time taken by our implementation initially decreases linearly, but, after a certain value, stops decreasing by a large extent. We attribute this to the fact that irrespective of the number of nodes, there is a certain minimum amount of work that needs to be done in each MapReduce iteration consisting of:-

* + Initial job setup on each compute node. Note that this increases with the number of compute nodes.
  + Communication of a particular amount of data over the network. Note that the network is a bottleneck in this regard.

Thus, our algorithm is definitely scalable to a certain extent with increasing number of nodes.

#### 4.3.3.3 Scalability based on increased number of iterations

Finally, we conducted timing experiments on our algorithm by increasing the number of iterations.

*Configuration Parameters*

Compute Nodes: 16

No. of Users: 4767

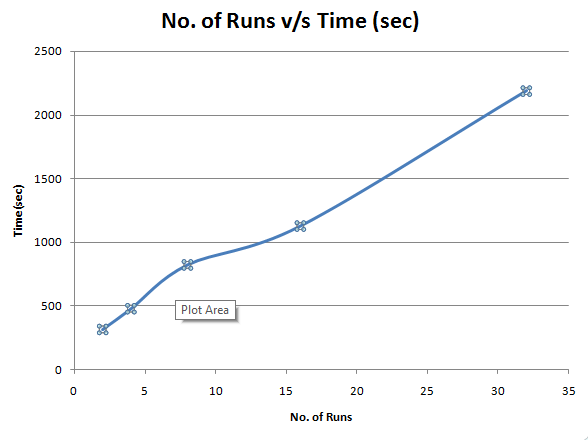
No. of relationships: 10,000,000

Processes per node: 2

Reducers: 32

No. of Runs: 5

|  |  |  |
| --- | --- | --- |
| No. of Runs | Time(sec) | Factor |
| 2 | 315 | 1.00 |
| 4 | 478 | 1.52 |
| 8 | 822 | 2.61 |
| 16 | 1127 | 3.58 |
| 32 | 2187 | 6.94 |



Our observation in this case is that as we increase the number of runs, the time taken by our implementation does not increase exponentially but linearly. With an increase in number of iterations by a factor of 16, the time increases by a factor of 6.94. We attribute this to the fact that as one increases the number of iterations, the work done by each iteration remains the same. Consequently, the total work done by our algorithm must increase linearly with the number of iterations. The slope of increase is not necessarily 1 since in each iteration, the initial setup such as the calculation of count and initialization are not duplicated.

# 5. Conclusion

* In conclusion, based on our experiments, our implementation not only works correctly but even scales very close to our expectations with increase in data, number of nodes and iterations. We plan to explore avenues for performance optimization especially for higher number of compute nodes as part of our future research.

# 6. References

[1] Twitter ofﬁcial website.

[2] Katz, L. (1953). A New Status Index Derived from Sociometric Index. Psychometrika, 39-43

[3] Aggarwal, C. C. (2011). Social Network Data Analysis. New York, NY: Springer

[4] The PageRank Citation Ranking: Bringing Order to the Web <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf>

[5] Facts about Google and Competition [<http://googlecompetition.blogspot.com/>]

[6] Katz, L. (1953). A New Status Index Derived from Sociometric Index. Psychometrika, 39-43

[7] Hanneman, R. A., & Riddle, M. (2005). Introduction to Social Network Methods. Retrieved from [http://faculty.ucr.edu/~hanneman/nettext/](http://faculty.ucr.edu/%7Ehanneman/nettext/)

[8] Koschützki, Dirk; Katharina A. Lehmann; Leon Peeters; Stefan Richter; Dagmar Tenfelde-Podehl; Oliver Zlotowski (2005). "Centrality Indices". In Ulrik Brandes, Thomas Erlebach. Network Analysis – Methodological Foundations. LNCS 3418. Springer Verlag, Heidelberg, Germany. pp. 16–60. ISBN 978-3-540-24979-5.4

[9] <http://hadoop.apache.org/>. Hadoop ofﬁcial website.

[10]  <http://www.ams.org/samplings/feature-column/fcarc-pagerank>