Middle-East Journal of Scientific Research 23 (12): 2868-2874, 2015
ISSN 1990-9233
© IDOSI Publications, 2015
DOI: 10.5829/idosi.mejsr.2015.23.12.22777

# Candidate Transaction Rank Accuracy Algorithm for Online Social Networks

*G. Srinaganya and J.G.R. Sathiaseelan*

Department of Computer Science, Bishop Heber College,
Tiruchirapplli, Tamilnadu, India

**Abstract:** In June 2015, Social Networks reported that a whopping of 20 billion users use the social networks which has boomed and made it essential for day–to-day life. Some of the obvious advantages of social networks are worldwide connectivity, commonality of interest and real-time information sharing ability that is able to search for users to add friend circle. It is important to get the appropriate results should return according to the user, with the increasing number of users in social networks. This paper states about the accuracy of KNN based classification, prediction or recommendation depends uniquely on a data model. The specific KNN algorithm is used to identify the criteria – Parsing, Indexing, Sorting and interaction that can be used to rank search results so that more appropriate results are presented to the user.

**Key words:** Web Mining · Online Social networks · KNN Algorithm · Boxplot · Candidate Transaction Rank Accuracy Algorithm

## INTRODUCTION

Nowadays, Social networks consist of websites that allow people to interact and share social experiences by exchange of multimedia objects like text, audio and video associated with their friends to keep in touch [1]. Each and every user on a social network possesses a user profile that contains all the information about the user, ranking from basic information like name, date of birth, gender, location, educational, professional information and areas of interest [2, 3]. A major challenge is extracting the relevant results while the social network user is searching for potential friends. The need of such a search technique arises due to the inherent structure of social networks and the behavior of users. Most of the searches, on a social network queries are containing names of users and a group of users may share the same name, which makes the trivial task of searching for online friends very cumbersome [4, 5]. In this paper, social network search ranking is discussed by means of an algorithm which takes into an account of three important factors that make search results relevant to a user – Parsing, Indexing, Sorting and interaction. Based on these three factors, search ranks are prearranged the search results when one user searches by another user name. This algorithm is designed for all social networks, which are in different types. If a user searches for friends on a network of classmates, it is more relevant to rank results which are closer to the searching result in the virtual space higher in the results list. An example is a social network of contacts, where ranking on the basis of the level of interaction among the users might be a more useful benchmarks [6]. The neighbors are taken from a set of objects for which the precise classification (or, in the case of regression, the value of the property) is known. This could be thought of as the training set for the algorithm, though there is no need of explicit training step is required. In order to identify neighbors, the objects are represented by position vectors in a multidimensional feature space. Usual the Euclidean distance is used, however other distance measures, such as the Manhattan distance could be used as a replacement. The k-nearest neighbor algorithm is sensitive to the local structure of the data. This paper is categorized as follows: section 2 describes the works done by various authors in previous years. Section 3 focuses the proposed work with an algorithm. Section 4 deals with evaluation methods of algorithm. Section 5 shows the experimental results of the proposed algorithm and followed by conclusion.

**Related Work:** Recently, there has been lots of interest in the field of social network search and ranking. Monique V.Vieira *et al.* [7] focuses on the problem of how to improve the search experience of the users. It suggests

**Corresponding Author:** G. Srinaganya, Department of Computer Science, Bishop Heber College, Tiruchirapplli, Tamilnadu, India.

seed based ranking instead of text-based ranking by measuring shortest distances between the nodes in a friendship graph. This is the first work that makes use of the friendship graph in a big social network to improve the search experience. Chuan Huang *et al*. [8] presents a novel social search model for finding a friend with common interests in OSN (online Social Network) with the introduction of trust value and popularity value. The trust value is calculated by the improved shortest path algorithm with a trust threshold and the popularity value is obtained by the PageRank Algorithm iteratively. In order to ensure more accurate search results, Khuan Yew Lee *et al*. [9] demonstrates an algorithm called which has five essential components – Engagement-U, Lifetime, Impression, Timeframe and Engagement-O. Engagement-U is the affinity between users which is measured by their relationships and other related interests between them, Lifetime is a trace of users' past based on their positive, neutral and even negative interactions and actions with other users. Impression is the weight of each object determined by the number of positive responses from users, Time frame is the timeline scoring technique in which an object naturally loses its value as time passes and Engagement-O is the attraction of users to objects which is measured between objects and associated interests of users.

Emphasizing on tree based search techniques, Wookey Lee *et al*. [10] compares the efficiency of reliable searching between Maximum Reliable Tree (MRT) algorithm and Optimum Branching Tree (OBT) algorithm and proposes the use of the MRT algorithm that is newly developed based on a graph-based method, asa generic technique which facilitates effective social network search and that can be the most reliable social network search method for the promptly appearing smart phone technologies, GunWoo Park *et al*. [11] explores the correlation between preferences of web search results and similarities among users by presenting an efficient search system called SMART finder. The work provides more information about SMART finder and publishes a quantitative evaluation of how SMART finder improves web searching compared to a baseline ranking algorithm. Zongli Jiang *et al* [12] explains the concept of user tag feedback scores is employed. Based on this concept, a tag –based feedback web ranking algorithm is designed. The algorithm could efficiently use for the user's feedback.

**Proposed Work:** The one of the difficulty is that the users of web search engines are facing the quality of the results in the extracted information [13]. Google is designed to scale this difficulty by efficiently in both space and time. Classifying unknown records are relatively expensive which requires distance computation of k-nearest neighbours, computationally intensive, especially when the size of the training set grows. Accuracy could be severely degraded by the presence of noisy or irrelevant features; KNN classification expects class conditional probability to be locally constant with high dimensions. The scale of the vertical axis must be large enough to encompass the greatest value of the datasets. The horizontal axis must be large enough to encompass the number of box plots to be drawn [14]. Construct the boxes, insert median points and attach upper and lower adjacent limits. Identify outliers (values outside the upper and lower adjacent limits) with asteris. Large storage requirements computationally intensive recall with highly susceptible to the curse of dimensionality.

**The Purpose:** The purpose of the k-Nearest Neighbours (KNN) algorithm is to use web pages in which the data points are separated into several separate classes to predict the classification of a new sample point. This sort of situation is best motivated through examples. K-Nearest Neighbours classification model by minimizing, over a reasonable number of neighbourhood sizes (k) and probability cut off values (p(cutoff)), the total misclassification error percentage is based on the validation data set. This system provides users to register their various types of profile like social, personal, general, professional. This system provides users to send a scrap message, images and data files to their friends. User can maintain the scrap book whatever scraps he has send to users.

- Using the Amazon Relational Webpages Service (RDS) to obtain data from a public cloud service.
- Building a simple k-nearest neighbour (KNN) classifier to categorize instances in a given dataset.
- Evaluating the effect of neighbour set size and training set size on the accuracy of the KNN classifier built.
- It makes use of the link structure of the Web to calculate a quality ranking for each web page, called PageRank. PageRank is a trademark of Google search engine. The PageRank process has been patented by Google search engine to utilize links to improve search results.

**Problem Definition:** This paper defines three metrics for the purpose of ranking search results.

Table 1: Notation Details of the Equations used in this paper

| Symbol | Definition |
|---|---|
| X | Previous Page |
| K | Current Page |
| Ø | Randomly chosen page |
| A | The number of web pages |
| T | Linear system of equations transaction |
| W | List of Current web pages |
| B | Bottom Page |
| S | Similarity measure |
| SE | Search Engine |
| $n_y$ | http protocol |
| $\gamma^2$ | Number of nodes in $n_y$ |
| D | URL |
| $x_q$ | Secure connection |
| $x_i$ | Web Site |
| E | More than a page |
| I | Internet Page |

**Parsing:** On a social network, a user may be linked directly and indirectly to millions of users. A social network is a myriad web of interconnections and a node which is closer to the searching node in terms of number of hops is likely to be a better search result in comparison to a node which is several hops furthers a way. Parsing measures the closeness of a node from the searching node. It is calculated by running a shortest distance algorithm and finding the minimum distance of the various nodes in the search result from the searching node. Let $d_{ab}$ be the shortest distances between nodes a and b, then proximity $p_{ab}$ calculated as:

Sample Matching Coefficient (SMC): Smc(x(i), x(j)) = da + d/n

**Similarity:** Social networks provide users a platform for interacting with other users who share similar interest, listen to the same kind of music, read books from the same author, follow the same sport, share the same hobbies, etc. on a social network all these details are captured in the user profiles. When a user issues a search for another user, a user who is more similar to the searching user is likely to be a more relevant result in comparison to a user who is less similar. The CTRAA define a user profile of a user as:

Jaccard Coeffiecient Sjc(x(i), x(j)) = k/1+ m + n

where k, l, m, n,… are the interests. Similarity S, between two nodes a and b in a list of search results with n nodes is defined as PageRank = PR(k)/ 2*2

**Interaction:** Social networks provide users different means for interacting with one another. Most online social networks allow users to interact via textual comments, exchange links through shares and like the posts of other users. When two users interact on a social network, there are two factors that can be used to gauge the closeness of these two users – frequency of interaction and recency of interaction [15]. Frequency captures volume of interaction (for each of comment, share, like) between two users within a given span of time. This spam of time is defined by window size *w*, defined further in the discussion. Recency captures the time gap between the time of issuance of the search query and the most recent interaction (for each comment, share, like) between the searching user and the user being searched. Frequency of an interaction of type T between user a and b is defined as:

$$f(x) = \sum_{k=0}^{\infty} \frac{\emptyset(k)}{k!}(-x)^k \tag{1}$$

where $T_k$ is the type of interaction (k =1 for comment, k =2 for share and k =3 for like) and $V_{ab}^{Ti}$ is the volume of interaction between users *a* and *b* of type $T_i$ Recency of interaction between users *a* and *b* is defined as

$$(x+a)^n = \left(0 < a > 1\ \left[\left(Tk\,(a^n_k)x^k a^{n(x+a)^n}\right)\right]\right)^t = \sum(k=0)^n = \ \left[\left((n\text{¦}k)\,x-k\right)\right] \tag{2}$$

where $t_o$ is the time instance at which the search query was issued, $t_{ab}^{Ti}$ is the time instance of the most recent interaction between user *a* and *b* of type T*i* and window size $w^{Ti}$ is defined as

$$PageRank\ value\ (x) = \frac{PageRank\ Connected\ (n)}{List\ link(k)} \tag{3}$$

where users x, k, n are the search results of the query. The frequency and recency metrics are then used to define interaction *i* of type $T_i$ between two users *a* and *b* as

$$PageRank\ value\ (0 < x < 1) = \frac{PageRank\ Connected\ (n)}{List\ link(k)X \propto} \tag{4}$$

where 0 ≤ α ≤ 1. A is defined as the relative importance of recendy in comparison to frequency when quantifying a particular interaction.

Euclidean distance = $d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$ (5)

The weighted interaction metric I between two users *a* and *b* is defined as the weighted sum of the three types of interactions (comment, share, like) as Manhattan distance is as follows:

$$d_{ij} = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \quad (6)$$

where β + γ + δ = 1. Here β, γ and δ define the percentage importance of the three types of interaction i.e. comment, share and like in the overall interaction metric I.

**Candidate Transaction Rank Accuracy Algorithm.**

Step 1: Initialize the metric Σ= I, the identity matrix.
Step 2: Spread out a nearest neighbourhood of KM points around the test point xo, in the metric Σ.
Step 3: Calculate the weighted within and between sum of squares matrices W and B using the points in the neighbourhood (partition of TSS(T=W+B)).
Step 4: Define a new metric

$$\sum = W^{-\frac{1}{2}} \left[ W^{-\frac{1}{2}} B^{-\frac{1}{2}} + \varepsilon I \right] W^{-\frac{1}{2}} \quad (7)$$

Step 5: Iterate steps 1,2 and 3.
Step 6: At completion, use the metric Σ for k-nearest neighbour classification at the test point xo.
Step 7: K is usually chosen empirically via a validation set or cross-validation by trying a range of k values.
Step 8: Distance function is crucial, but depends on applications.

The ranks of the search results are subsequently obtained by sorting the results by the weighted association values. The search with the highest weighted association value is given rank 1, the search result with the second highest weighted association value is give rank 2 and so on, until the search result with the lowest weighted association value is given rank.

$$S.E. = \sqrt{\frac{\sum_{s=1}^{m} \sum_{j=1}^{n} \gamma i s^2}{(n_y - 1)(n_y)}} \quad (8)$$

Weighting: K-Nearest Neighbour Classification may have a problem to determine the class label of testing set solely on the equal weighting voting from the k-nearest neighbours. Obviously, the closer neighbour that is more similar they are. This, weighting should be added to neighbour, which is closer to the testing instance.

**Algorithm for Computing Search Ranks:** The proposed algorithm is a ranking algorithm and, therefore, allows different search algorithms to be used to identify the unranked search results. Once the unranked search results are obtained, they are then ranked using the proposed algorithm. The association function is central to the calculation of the ranks of the search results. n potential search results are returned by the searching algorithm for a given query. Once this list is obtained, the next steps are to calculate the association and then rank the list on the basis of the association values.

**Lemma 1:** Time complexity of interest is that of the function *compute_ranks.* Let assume that the sorting algorithm used has a worst case complexity of *n log n*.

***Proof:***

```
begin
  get_window_sizes                  n steps
  get_common_interests_cardinality  n steps
  compute_association               n steps
  sort                              n log n steps
end                                 ____________
compute_ranks                       (3n+n
log n) steps
```

**Experimental Results and Performance Analysis:** The experimental study has been conducted on a ASUS-X550C laptop with an Intel Core i3-3217U, 1.8 GHz CPU and 4GB of memory, running in Windows 8.1. All programs are coded in JAVA. The dataset T1014D100K that was generated by IBM Quest Synthetic Data Generation. The real world datasets Sina Weibo and Twitter are also used to evaluate the CTRAA. The accuracy of the proposed algorithm is compared with WEAPON and CASINO algorithms by conducting experiments using the online social networks

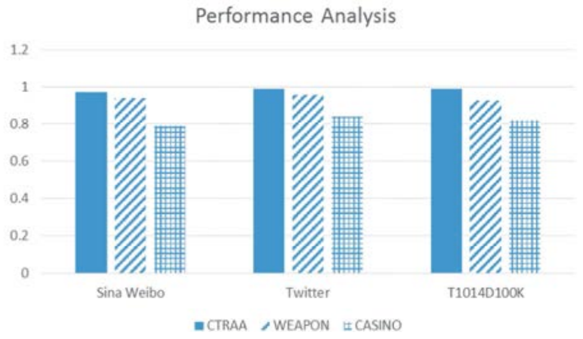

Fig. 1: Performance Analysis

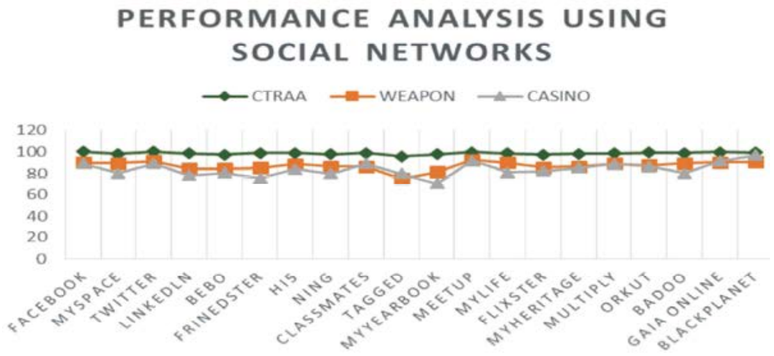

Fig. 2: Performance Analysis using Social Networks

Table 2: Comparative Experimental Results of WEAPON, CASINO and CTRAA Algorithms

| Dataset | Algorithm | Accuracy Rate |
|---|---|---|
| T1014D100K | CTRAA | 0.99 |
| | WEAPON | 0.93 |
| | CASINO | 0.82 |
| Sina Weibo | CTRAA | 0.97 |
| | WEAPON | 0.94 |
| | CASINO | 0.79 |
| Twitter | CTRAA | 0.99 |
| | WEAPON | 0.96 |
| | CASINO | 0.84 |

[15]. The performance of the CTRAA is better than WEAPON and CASINO. Table 2 explains the Comparison of algorithms - WEAPON, CASINO and CTRAA. Fig. 1 shows the performance analysis of the algorithms which Table 2 explains in a bar chat. Fig. 2 exhibits the performance analysis of the algorithms using social networks to show the algorithm CTRAA is better than WEAPON and CASINO in extracting the details according to the user's need with better accuracy.

The small simulation network which was used for computing the results of the proposed algorithm was also reconstructed on popular social network sites, Facebook and Google+. It was observed that the results obtained by issuing the same search on these networks can be replicated using the proposed algorithm by varying the various parameters, i.e., α, β, γ, δ, μ1, μ2 and μ3. Though, this algorithm is not aimed at conjecturing the logic that might have been used to obtain search results by a social network website, it can, however, be used to obtain search results that concur with the results of the website in question to a certain extent.

The algorithm discussed in [8], based on trust and popularity was also implemented on the small simulation network. The result obtained was similar to the one at serial number 3. Certain assumptions were made while implementing the algorithm in [8]. While calculating the contribution of trust in the rank, trust values for two adjacent nodes was taken as 0.5. For the calculation of the contribution of popularity in the rank, the damping factor d was taken to be 0.8 and the initial values of popularity were taken to be 0. The final values of popularity for each node with respect to the various keywords in the profile were obtained by running the circular algorithm until the values converged to a precision of 10%.

**Evaluation:** The results from the proposed algorithm show that depending on the values set for the different parameters namely, α, β, γ, δ, μ1, μ2 and μ3, suitable results can be obtained. The nature of a particular social network will dictate what values must be set for each of these parameters. The advantages of the proposed algorithm include:

**Intuitiveness:** The algorithm uses intuitive concepts like proximity, similarity and interaction to rank search results such that more relevant results may be ranked higher than less relevant ones.

**Adaptability:** The algorithm is not confined to providing suitable search ranks for any particular kind of network and can be adapted for use in any kind of social network by defining the various parameters according to the nature of the network.

**Flexibility:** The algorithm can use the search results supplied by any search algorithm. Once the association values of the elements in the search list are obtained, subsequently, any algorithm can be used to sort the elements to obtain the ranked list. Therefore, the algorithm provides flexibility of implementation as it can be easily plugged between the search and sort algorithms in a social network.

## CONCLUSION

A Boxplot shows the distribution of data. The line between the lowest adjacent limit and the bottom of the box represent one-fourth of the data. One-fourth of the data falls between the bottom of the box and the median and another one-fourth between the median and the top of the box. The line between the top of the box and the upper adjacent limit represent the final one-fourth of the data observations. Once the pattern of data variation is clear, the next step is to develop an explanation for the variation. Google is designed to be a scalable search engine. The primary goal is to provide high quality search results over a rapidly growing World Wide Web. Google employs a number of techniques to improve search quality including PageRank, anchor text and proximity information. Google is a complete architecture for gathering web pages, indexing them and performing search queries over them. Website social networks have become an essential part of the lives of internet users. The importance of these networks will only grow over time and they are likely to become more and more complex and specialized as they evolve. The work presented in this paper tries to provide a generic solution for ranking of search results over social networks and also considering the large volume of searches being performed on social networks. This work takes into account the fact that all social networks are not similar and hence, the same search result algorithm is not likely to be useful for all of them. As a result, an adaptive algorithm has been proposed which uses intuitive concepts like proximity, similarity and interaction to rank search results according to their relevance in a particular social network setting.

## REFERENCES

1. Han Xiaogang, Wei Wei. Chunyan Miao and Jian-Ping Mei, 2014. Context-Aware Personal Information Retrieval From Multiple Social Networks, IEEE Computational Intelligence Magazine, pp: 18-28.
2. Anwar Tarique and Muhammad Abulaish, 2015. Ranking Radically Influential Web Forum Users, IEEE Transactions on Information Forensics and Security, 10(6): 1289-1298.
3. Interdonato Roberto and Andrea Tagarelli, 2015. Ranking Silent Nodes in Information Networks: a Quantitative Approach and Applications, Elsevier, Physics Procedia, 62: 36-41.
4. Xung-Truong Vu, Marie-Helene Abel and Pierre Morizet-Mahoudeaux, 2015. A user-centered approach for integrating social data into groups of interest, Elsevier, Data & Knowledge Engineering, pp: 43-56.
5. Hamed Ahmed Abdeen, Sindong Wu, Robert Erickson and Tamer Fandy, 2015. Twitter K-H networks in action: Advancing biomedical literature for drug search, Elsevier, Journal of Biomedical Informatics, 56: 157-168.
6. Wang Guojun, Wenjun Jiang, Jie Wu and Zhengli Xiong, 2014. Fine-grained Feature-based Social Influence Evaluation in Online Social Networks, IEEE Transactions on Parallel and Distributed Systems, 25(9): 2286-2296.
7. Vieira Monique V., Bruno M.Fonseca, Rodrigo Damazio, Paulo B.Golgher and Davi de Castro Reis, 2007. Efficient Search Ranking in Social Networks, Lisboa Portugal, ACM, CIKM, 07: 563-572.
8. Huang Chuan, Yinzi Chen, Wendong Wang, Yidong Cui, Hao Wang and Nan Du, 2010. A Novel Social Search Mode Based on Trust and Popularity IEEE, IC-BNMT, pp: 1-5.
9. Lee Khuan Yew and Jer Lang Hong, 2012. ELITE – A Novel Ranking Algorithm for Social Networking Sites, IEEE 9th Internation Conference on Fuzzy Systems and Knowledge Discovery, pp: 1-5.

10. Lee Wookey, James Jung-Hun Lee, Justin Jong-Su Song and Chris Soo-Hyn Eom, 2011. Maximum Reliable Tree for Social Network Search, IEEE International Conference on Depedable, Autonomic and Secure Computing, pp: 1-5.
11. Park GunWoo, SooJin Lee and Gang Hoon Lee, 2009. To Enhance Web Search based on Topic Sensitive Social Relationship Ranking Algorithm in Social Networks, IEEE International Joint Conferences on Web Intelligence and Intelligent Agent Technology, pp: 1-5.
12. Zongli Jiang and Jingheng Li, 2012. A Tag Feedback Based Sorting Algorithm for Social Search, IEEE, ICSAI, pp: 1-5.
13. Chen Yan, Xiaoming Zhang, Zhoujm Li and Jun-Ping Ng, 2015. Search Engine reinforced semi-supervised classification and graph-based Summariza tion of microblogs, Elsevier, Neurocomputin, 152: 274-286.
14. Tassa Tamir and Dror J. Choen, 2013. Anonymization of Centralized and Distributed Social Networks by Sequential Clustering, IEEE Transaction on Knowledge and Data Engineering, 25(2): 311-324.
15. Hui Li, SHEN Bingqing, CUI Jaingtao and MA Jianfeng, 2014. UGC-Driven Social Influence Study in Online Micro-Blogging Sites, ICT Management, China Communications, pp: 141-151.