Finding Influencers in Social Networks

Carolina Bento
carolina.bento@ist.utl.pt

Instituto Superior Técnico - Lisbon Tech/ INESC-ID, Av. Professor Cavaco Silva, 2744-016 Porto Salvo, Portugal

ABSTRACT

Social networking is part of the daily routine of millions of people around the world. Modern social networking platforms provide users with tools for creating and sharing textual content, pointers to other web content, photographs or videos. From the millions of users that these platforms have, one can also acknowledge that the activities of a selected number of users are more rapidly perceived than those of others, and that the content produced by them flows swiftly through the network. We call these users the influencers. Influencers generate trends and shape opinions in social networks, being crucial in areas such as marketing, advertising or opinion mining. In this work, we studied automated techniques for discovering influential nodes in such networks, and we experimented with two different types of social networks: (1) location-based social networks (LBSN), i.e., networks that include relationships between users in the network and between users and the locations they have visited, and (2) academic citation networks (ASN), i.e., networks that relate scientific papers according to their citations. We addressed the task of identifying the most influential users in LBSN, while for ASN we addressed the task of identifying the most important papers, and developed a framework to predict the future influence scores of papers. We can conclude that these techniques really assist us when trying to find the most influential nodes in a network, and that one can make accurate predictions of future influence scores with the framework that was developed.

Keywords

[Social Networks, Network analysis, Impact Scores, Information Retrieval, Large-scale Networks, Influencers]

1. INTRODUCTION

The rise of social media platforms such as Twitter[1] and Google+[2], with their focus on user-generated content and social networks, has brought the study of authority and influence over social networks to the forefront of current research. For companies and other public entities, identifying and engaging with influential users in social networks is critical, since any opinions they express can rapidly spread far and wide. For users, when presented with a vast amount of content relevant to a topic of interest, ordering content by the source's authority or influence can also assist in information retrieval. There has been a substantial amount of recent work studying influence and the diffusion of information in social networks. Moreover, there has also been much work in the field of social network analysis that has focused explicitly on sociometry, including quantitative measures of influence, authority, centrality or prestige. These measures (e.g., degree centrality or betweenness centrality) are essentially heuristics, usually based on intuitive notions such as access and control over resources, or brokerage of information. In this context, I studied the problem of identifying the most influential nodes in a social network with two different types of social networks at hand, a location-based social network and an academic citation network. The main focus of this work was to use well-known social network analysis techniques and algorithms to address this task. Therefore, social network analysis metrics, like *degree* or *clustering coefficient*, and state-of-the-art ranking algorithms, such as PageRank and HITS, were studied in order to understand how to estimate influence in social networks.

The most important contribution of this work was a redesign of the Influence-Passivity (IP) algorithm. Initially strictly intended for Twitter data, we adapted it to be used in the context of location-based social networks, where the propagation of information is done via the locations that users visit over time.

When studying influence in academic social networks, we specifically addressed the temporal issues arising in the ranking of scientific articles. We studied techniques for estimating future influence scores. In this context, we developed a framework to predict the future PageRank scores and future download counts of scientific articles, for a specific year, through a combination of features, such as, the age of the article or previous PageRank scores.

We collected real and up-to-date data from two social networking platforms, namely Twitter and FourSquare. Then, different ranking algorithms were computed and the *top-10* highest ranked users and the *top-10* highest ranked spots

[1] `http://twitter.com/`
[2] `https://plus.google.com/`

were extracted. To assess the accuracy of our results for social networks based on location, we made an empirical analysis of our *top-10*, looking into the user profiles and spot *check-ins*, in order to understand how their profile characteristics were related to their influence in the network.

Regarding academic social networks, a citation network was built with data from the DBLP[3] digital library, and only the *top-10* highest ranked papers from the computation of PageRank algorithm were obtained. When assessing the accuracy of the results, we empirically crossed-checked the authors of the *top-10* highest ranked scientific papers in the DBLP with the recipients of various renowned scientific awards, like the Gerard Salton Award or the Turing Award. Considering the experiment for estimation of future influence scores of scientific papers and future download counts for these scientific papers, a set of evaluation metrics, including the *normalized root mean squared error* and the *spearman correlation*, was used to assess the quality of our predictions comparing to the real influence scores.

The rest of the paper is organized as follows: Section 2 describes the most significant work related to the task of finding influencers in social networks. Section 3 details the work that was developed, namely the methodology for data collection, how the networks were built, the specific implementation and adaptation of the IP algorithm, as well as, the methodology to find the influential nodes in the networks. For the experiment of prediction of future PageRank scores, Section 3 also includes the description of the features that were used and the learning regression model. Section 4 describes the validation methodology for all the experiments, the obtained results and respective discussion. Finally, Section 5 highlights the most important conclusions of this paper and presents possible future work.

2. RELATED WORK

This section presents the most important previous work related to finding influencers in social networks. Fundamental definitions, algorithms and techniques in the areas of graph theory and network analysis have been surveyed extensively in the works of Kleinberg and Easley [12] or of Cook and Holder[8]. Here we begin by presenting the *HITS* and *PageRank* algorithms, discussing how the latter evolved to more detailed and specific approaches, such as the *Weighted PageRank* algorithm. Then, we introduce the *IP Algorithm*, which determines the influence and passivity of network nodes based on their capacity to forward information. Finally, we take a deeper look at the work that has been done in order to find influencers in citation and co-authorship networks, also describing works that take into account the temporal evolution of graphs.

The *HITS* is a graph-based algorithm developed by Kleinberg [16]. Based on the notion of *authorities* and *hubs*. The *authorities*, i.e., nodes that have a greater amount of inlinks, have a *mutually reinforcing relationship* with the *hubs*, i.e., the nodes that have outlinks to many related authorities, in a way that a good *hub* is a node that points to many good *authorities*, and a good *authority* is a node that is pointed by many good *hubs*. This relationship is put into use through the iterative procedure shown in Algorithm 1, which maintains and updates the authority and hub weights of each page [16]. In his work, Kleinberg proposed to rank *Web* pages.

Algorithm 1 The Hyperlinked Induced Topic Search (HITS) Algorithm

G: A graph with n interlinked pages
k: A constant corresponding to the number of iterations
z: The vector (1,1,1,...,1) $\in \mathbb{R}^n$
Set $x_0 := z$
Set $y_0 := z$
for $i = 1, 2, ..., k$ **do**
  Apply $x_p = \sum_{q,q\to p} y_q$ to (x_{i-1}, y_{i-1}), obtaining new x-weights $x_i^{'}$
  Apply $y_p = \sum_{q,p\to q} x_q$ to ($x_i^{'}$, y_{i-1}), obtaining new y-weights $y_i^{'}$
  Normalize $x_i^{'}$, obtaining new *authority* scores x_i
  Normalize $y_i^{'}$, obtaining new *hub* scores y_i
end for

The *PageRank* algorithm is another graph-based *Web* page ranking method, which arose in the context of the development of Google's search engine [5]. PageRank is based on principles from academic citation analysis, applied to the *web*. It can be mathematically expressed as follows:

$$PR(A) = \frac{(1-d)}{N} + d\sum_i \frac{PR(T_i)}{C(T_i)} \quad (1)$$

In the PageRank model, a node (i.e., a page) A has $T_1, ..., T_n$ nodes that point to it (i.e., that cite page A) and, $C(T_1), ..., C(T_n)$ is the number of outlinks from node A to pages $T_1, ..., T_n$. The term N corresponds to the total number of nodes in the network. The free parameter d is called the *damping factor* and controls the performance of the algorithm, being usually set to 0.85. In a random *web* surfer scenario, the surfer can restart his search with probability $1 - d$ by jumping to another page that is randomly and uniformly chosen, instead of following a random link, which can be done with probability d [7]. A page can achieve a high *PageRank score* if many other pages pointing to it (i.e., if it is highly cited) or if some of the pages that point to it have themselves a high *PageRank score*.

In the realm of bibliometrics, *PageRank* often is used as a complementary method to more traditional citation analysis methods, due to mitigating citation count's drawback of not taking into account the importance of a paper. *PageRank* allows us to identify publications that are being referenced by highly cited articles [10].

Acknowledging that some links in a *web page* may be more important than others, Xing and Ghorbani proposed the *Weighted PageRank* algorithm that assigns higher scores to more important links, instead of the traditional even division among the outlinks of a page [29]. Each link is assigned with a value that is proportional to the popularity of the destination node, i.e., proportional to its number of inlinks and outlinks. In this approach, there is an inlink weight

[3] `http://www.informatik.uni-trier.de/~ley/db/`

and an outlink weight. The inlink weight of link (v, u) is based on the number of inlinks of page u and the number of inlinks from all the pages that are referenced by page v. The outlink weight is analogous. Xing and Ghorbani's studies revealed that their algorithm has a better performance than the original *PageRank*.

Applying the *Weighted PageRank* algorithm to journal citation networks, Bollen et al. took into account journal citation frequencies in the transfer of *PageRank* values, so that the prestige of a journal can be accordingly transfered along the iterations of the algorithm. They referred to this transfered value as the *Propagation Proportion*, which replaces the number of outlinks $C(T_i)$ in Equation 1.

In the context of Twitter and from the work of Weng et al. arose *TwitterRank*[28], an extension of the *PageRank* algorithm that takes both the topic similarity between users and the link structure of the social network into account. On the other hand, Romero et al. came to the conclusion that, if a Twitter user is to be considered influential, then he does not only have to be popular and get attention from his peers, but he has also to overcome passivity, a state in which a user receives information but does not propagate it through the network. They proposed the IP algorithm, that determines the influence, as well as, the passivity of a user, based on his information forwarding activity [22]. This algorithm is similar to HITS and to *PageRank*, but with the difference that the diffusion behaviour among the users is also taken into consideration. The IP algorithm assigns to every user both a passivity score and an influence score, which respectively correspond to the authority and hub scores in the HITS algorithm. The use of passivity in the algorithm comes from the evidence that Twitter users are generally passive and thus, when determining the influence of a user, taking into account the passivity of all the people that are influenced by him is also very important. The following assumptions are considered by the authors:

1. The *influence score* of a user depends on the number of people he influences, as well as on their passivity.
2. The *influence score* of a user depends on how dedicated the people that he influences are. This dedication is measured by the amount of attention a user pays to some other user, as compared to everyone else.
3. The *passivity score* of a user depends on the influence of those who he is exposed to, but not influenced by.
4. The *passivity score* of a user depends on how much he rejects some other user's influence, compared to everyone else's influence.

Given these assumptions, one should note that the network graph for this algorithm is a weighted graph $G = (N, E, W)$ with N nodes, E edges and W edge weights, where a weight w_{ij} represents the ratio of influence that node i has over node j to the total influence that i attempted to have over j. For each edge $e = (i, j) \in E$, the authors defined an *acceptance rate* that represents the amount of influence accepted by j from all users in the network and that, thus, can reflect the loyalty user j has to user i. The authors aslo defined a *rejection rate*, which is the opposite of the *acceptance rate*, because $1 - w_{ji}$ is the amount of influence user i rejects from user j. Thus, the *rejection rate* v_{ji} is the influence that user i rejected from user j, normalized by the total influence rejected from j by all other users in the network. The algorithm takes as input a weighted graph and computes the IP scores for each node in m iterations, as depicted in the pseudo-code of Algorithm 2.

Algorithm 2 The Influence-Passivity (IP) Algorithm.

$G(N, E, W)$: An influence graph with N nodes, E edges and W edge weight
$I_0 \leftarrow (1, 1, ..., 1) \in \mathbf{R}^{|N|}$
$P_0 \leftarrow (1, 1, ..., 1) \in \mathbf{R}^{|N|}$
for $i = 1 \rightarrow m$ **do**
  Update P_i using operation $P_i \leftarrow \sum_{j:(j,i)\in E} v_{ji} I_j$ and the values I_{i-1}
  Update I_i using operation $I_i \leftarrow \sum_{j:(i,j)\in E} u_{ij} P_j$ and the values P_i
  for $j = 1 \rightarrow |N|$ **do**
    $I_j = \frac{I_j}{\sum_{k\in N} I_k}$
    $P_j = \frac{P_j}{\sum_{k\in N} P_k}$
  end for
end for

The authors concluded that there is a weak correlation between popularity and influence. The *IP Algorithm* turned out to provide better indicators of popularity than the *PageRank* algorithm.

In Bibliometrics, there are essentially two classes of ranking algorithms. The class of *collection-based ranking algorithms* uses a weighted graph and its nodes correspond to the collections, e.g., journals and conference inproceedings, while the weighted edges represent the total number of citations that point from one collection to the other. In the other class, *publication-based ranking algorithms*, the nodes in the citation graph are individual publications and the edges represent citations between papers [24]. Both *PageRank* [5] and *HITS* [16] are part of the second class of ranking algorithms, while the ISI Impact Factor [3] is part of the first class.

Specifically for co-authorship networks, where the graph nodes represent authors and edges represent ties between two authors, Liu et al. proposed *AuthorRank*, a modification to the *PageRank* algorithm that is computed over a weighted directed co-authorship graph [19]. The co-authorship graph is directed and weighted in order to express the magnitude of the relationship between two authors and is, as in the *Weighted PageRank*, represented by $G = (V, E, W)$, with a set of V authors, a set of E co-author relationships, and a set W of normalized weights w_{ij} connecting authors v_i and v_j. The normalized weights w_{ij} are such that the weights of an author sum up to one.

Generally, citation networks are *static*, since a scientific article can not lose citations throughout the years, and since articles do not disappear from the network. On the other hand, social networks are generally characterized as *dynamic networks*, which change at a very fast pace, due to new users that make new connections and former users that leave the social network and break the ties they already established.

Still, even in the case of citation networks, new articles are also being constantly introduced. Therefore, time is a key factor in social network analysis, and also in the analysis of academic networks.

Sayyadi and Getoor developed FutureRank, an approach which computes the expected PageRank score of a scientific article, based on the citations it will obtain in the future [23]. This number of future citations is referred to as the *usefulness* of the article, and the authors assumed that recent articles are more useful. Nevertheless, older and highly cited articles still get a good ranking, due to being cited by recent articles. The algorithm is computed in a network that has two different types of nodes, namely, articles and authors, thus being unfold into two distinct networks (i) a citation network connecting articles through citation edges, and (ii) a authorship network connecting articles and authors through co-authorship edges. In the second network, articles can be mapped as the authorities and authors as the hubs from the HITS algorithm. In short, FutureRank runs one step of PageRank in the first network, in order to transfer authority from the articles to their references, and one step of HITS in the second network. These results are repeatedly combined until convergence is reached. The ranking of articles also involves a personalized PageRank vector, which is pre-computed with basis on the current time and the publication time of the articles, instead of being based on the number of nodes in the network as in the original PageRank algorithm.

The CiteRank algorithm [27], on the other hand, makes use of publication time in order to rank articles, where each researcher, independently of others, is assumed to start his search with recent articles, proceeding in a chain of citations until full satisfaction. The output of the algorithm can be seen as an estimate of traffic to an article, i.e., the probability of encountering an article via a path of any length, and is correlated to the number of citations in a way that the larger the number of citations, the more likely it will be for the article to be visited via one of the incoming links. CiteRank is in all similar to the PageRank algorithm, except for the fact that CiteRank initially distributes random surfers exponentially with age and with probability $\rho_i = e^{-age_i/\tau_{dir}}$, where age_i is the age of the i^{th} article and τ_{dir} is the decay of time, thus favoring recent articles.

3. FINDING INFLUENCERS

This section details the work that was developed in the context of my MSc thesis. Two distinct types of analysis were conducted, each with a different type of social network. In the first part we collected real and up-to-date data from a social networking service that is based on location, namely FourSquare, and also from Twitter, building social networks from the aforementioned collected data. The second part involved data from DBLP, a digital library containing information about academic publications, such as, paper citations, from which a citation network was built. Social network analysis algorithms and techniques were applied to the different social networks so we could explore influence in distinct contexts. On the part that involved *location-based social networks* we wanted to test how good these social network analysis algorithms and techniques were, when used to identify the most relevant nodes in a network. On the other hand, with the academic social network, we wanted to test if it was possible assess the most influential papers in the collection, and to predict the future influence scores of the nodes in the network, based on their previous influence scores.

3.1 LAW Webgraph

To perform our experiments and fulfill these tasks we used several state-of-the-art algorithms and open-source software packages for network analysis, in which is included the LAW Webgraph open-source software package. LAW Webgraph is an open source project developed by researchers from the Laboratory of Web Algorithms at the University of Milan. It contains a Java library for large-scale web graph analysis, presenting a novel approach to graph compression that enables the creation and storage of web graphs. Among metrics such as the Kendall's Tau, the LAW Webgraph package contains an implementation of the PageRank algorithm, which was the first algorithm we used for assessing the influence of nodes in our experiments. As we intended to extend this software package with the HITS and IP algorithms, the structure of LAW's PageRank algorithm implementation served as a template for our algorithmic extensions.

For the implementation of the HITS algorithm we followed the pseudo-code in Algorithm 1, in which we have to compute two different scores - the *hub score* and the *authority score*. The computation of these scores is based, respectively, on outlinks and inlinks nevertheless, through LAW Webgraph's API we could only have access to the successors of a node. To overcome this limitation, when computing the HITS algorithm we built the graph and its transpose, instead of just the graph, so we can access both the successors and predecessors of each node.

Analogously, the Influence-Passivity (IP) algorithm involves the computation of two scores - the *influence score* and the *passivity score* - therefore, two graphs were built as we followed the pseudo-code in Algorithm 2.

3.2 Characterizing Networks

To understand aspects, such as, the dimension or how well connected are the nodes in our generated graphs, we used some well-known network analysis metrics. With the average path length one can assess the average distance between the nodes in our networks, understanding how tightly connected they are, e.g., a small average path length indicates that all nodes are closely connected, which means that it will be easy to spread information through the network. The clustering coefficient allows us to assess how neighbours on our networks are close to one another, i.e., how our neighbours tend to create clusters with a large number of ties between them. On the other hand, studying the degree distribution of the nodes in a network, we can assess if we are at the presence of a large-scale network that is characterized by a power-law distribution, i.e., at presence of a network in which the majority of the nodes have few connections, but where there is a smaller set of nodes holding an extremely large number of connections.

3.3 Location-based Social Networks

A location-based social network has all the properties of a social network however, it has two types of nodes instead

of just one, (1) *user nodes*, which are the users in the network and who can be friends with other users, and (2) *location nodes*, which are the locations users have visited or mentioned in their personal messages. Therefore, one can say that a location-based social network also has two types of edges or social ties: (1) *user-user* ties, corresponding to the edges between two users and in all similar to the edges existing in social networks and (2) *user-location* ties, corresponding to the edges between users and locations, which are derived from a user mentioning or visiting a specific location. Location-based social networks yield a great amount of information, because one can look at them as if they have two layers: one where the users are connected to their friends and an underlying layer where users are connected to locations, the latter being an intersecting layer through which one can identify the most visited locations (i.e., locations that are connected to a larger number of users) and, on a location perspective, which locations exert more influence to the users they are connected to - see Figure 1.

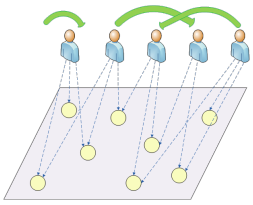

Figure 1: Example of a location-based social network (adapted from Zheng and Zhou [30]).

In FourSquare, registered users can search for other users or venues, e.g., one can search for *Indian Restaurant* near New York and a extensive list of restaurants is presented, each one with address and map, user uploaded photos, reviews by users that have had checked-in there, as well as, a list of venues that are similar to it. Venues can be associated with categories and tags. There is also an underlying *game-play* concept in this kind of social networks, encouraging continuous interaction: (i) users earn points for checking-in at venues or adding new venues to FourSquare, (ii) users earn badges if they check-in in various different venues or complete tasks, (iii) a user in FourSquare can become mayor of a specific venue if he has checked-in in that venue for more days that anyone else, in a period of 60 days. On the other hand, Twitter is a social networking and microblogging service that allows users to post messages 140 characters long - the *tweets*. It was only accessible via their website, but today one has a multitude of mobile applications at hand to can manage our account, *tweet* wherever we please and also attach links to *tweets*. Nowadays, many Twitter users *tweet* as they arrive (or check-in) at a specific location, attaching the geographical coordinates of that place to their *tweet* thus, we can associate Twitter users with locations.

To extract data about users and venues in FouSquare, for simplicity of use, an open-source Java implementation[4] of the FourSquare API was used, providing straightforward methods to make FourSquare API calls. This Java API includes all methods in the official FourSquare API however, the functionality of the *venuesSearch* was not fully implemented, so there was the need to make a simple change to the FourSquare Java API in order to extract reliable data, because even though the *venuesSearch* method allowed us to obtain a set of venues that are near the provided latitude-longitude coordinate and within a specified *radius* ranging up to 5 km, the radius functionality was not implemented in the Java API, which led to a simple addition of the radius parameter in the *venuesSearch* API call, in order to take advantage of that functionality and obtain more venues per call - see pseudo-code in Algorithm 3. Also, we have defined a bounding box for the New York City-Manhattan area to guarantee that the data we were going to collect was confined only in that geographical area, instead sparse locations around the globe.

Algorithm 3 Pseudocode for the extraction of user and friend data from FourSquare.

lat_{max}: maximum latitude for the NYC - Manhattan bounding box
$long_{max}$: maximum longitude for the NYC - Manhattan bounding box
lat_{min}: minimum latitude for the NYC - Manhattan bounding box
$long_{min}$: minimum latitude for the NYC - Manhattan bounding box
lat: current latitude
long: current longitude
$radius = 1000$ (i.e., 1km)
userSet: Set of users from a venue
for all lat $\in [lat_{min}, lat_{max}]$ and long $\in [long_{max}, long_{min}]$ **do**
  venueSet $\leftarrow$ all venues for lat, long within radius
  for all venue $\in$ venueSet **do**
    Retrieve and store venue info
    userSet $\leftarrow$ all venue's visiting users
    for all user *in* userSet **do**
      Retrieve users' friends
      Store friend information
    end for
  end for
end for

As for Twitter, we used the Twitter Public Stream API[5] that provides 1% of all the *tweets* that have been published in that *API second*. The data collection process had the following phases:

1. From that 1% of *tweets* we selected only the ones which had geographical coordinates. Also, for each *tweet* we collected information such as, *user id*, users that he is following and users that are following him. With the coordinates associated to a user's *tweet* we could establish *user-location* ties and, with the following and follower relationships, we could establish *user-user* ties.

2. From the collected user information, we selected the users which had the greater amount of connections.

3. Afterwards, similarly to what we did in FourSquare, we filtered all the collected data in order to keep only the information about *tweets* that were within the New York City-Manhattan area.

[4] `http://code.google.com/p/foursquare-api-java/`

[5] `https://dev.twitter.com/docs/streaming-apis`

In order to perform the discretization of geospatial coordinates, we used the hierarchical triangular mesh approach to divide the Earth's surface into a set of triangular regions, each roughly occupying an equal area of the Earth [25, 11]. In brief, we have that the Hierarchical Triangular Mesh (HTM) offers a multi-level recursive decomposition of a spherical approximation to the Earth's surface. It starts at level zero with an octahedron and, by projecting the edges of the octahedron onto the sphere; it creates 8 spherical triangles, 4 on the Northern and 4 on the Southern hemisphere. Four of these triangles share a vertex at the pole and the sides opposite to the pole form the equator. Each of the 8 spherical triangles can be split into four smaller triangles by introducing new vertices at the midpoints of each side, and adding a great circle arc segment to connect the new vertices with the existing ones. This sub-division process can be repeated recursively, until we reach the desired level of resolution, as shown in Figure 2. The triangles in this mesh are the regions used in our representation of the Earth, and every triangle, at any resolution, is represented by a single numeric ID. For each location given by a pair of coordinates on the surface of the Earth, there is an ID representing the triangle, at a particular resolution, that contains the corresponding point. Notice that the proposed representation scheme contains a parameter k that controls the resolution, i.e. the area of the triangular regions. With a resolution of k, the number of regions n used to represent the Earth corresponds to $n = 8 \cdot 4^k$.

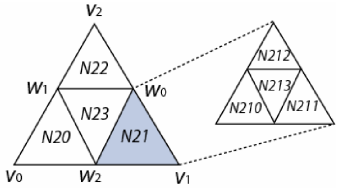

Figure 2: The HTM recursive division process (adapted from Szalay et al. [25]).

The main limitation in the FourSquare API, due to still being experimental, was that their rate limit for authenticated calls per hour is set to 500, which is a very low threshold considering that we have performed an extensive crawl and each request for the listing of a user's friends is a frequent authenticated API call. As to the Twitter API, we had a rate limit of 600 calls per hour and, exceeding that limit, we had to wait until the next hour to make more API calls, which made us disregard a great amount of *tweets* during that waiting time.

3.3.1 Adaptation of the Influence-Passivity Algorithm

A major contribution of this work was the adaptation and implementation of the aforementioned IP algorithm. This algorithm presents a novel way of quantifying the influence of nodes in a network by considering that each node has an influence score, as well as, a passivity score. For our implementation, some changes had to be conducted to the original IP algorithm, in order to perform a calculation of edge weights that was consistent with the datasets we were working with. As for our datasets from Twitter and FourSquare, we wanted to generate a weight exclusively based on *user-location* and *user-user* ties, instead of *URLs* or *retweets*, as proposed by the authors. Thus, we built a graph that rather than having two types of nodes, i.e., locations and users, would only have *user-user* ties, estimating exclusively the influence of users in the network. To calculate the weight of edges between users, we adapted the Q_i and S_{ij} parameters, having Q_i as the number of locations node i has visited and Sij as the number of locations visited by both i and j, i.e., number of common visited locations between nodes i and j, having i visited the location before j has visited it. From our adaptation of the algorithm, user influence is always dependent on the popularity of the locations a user has visited.

The original graph built from our datasets is depicted in Figure 3, i.e., the left-most graph which includes two types of nodes: (i) user nodes, represented by $U_1 ... U_4$, and (ii) location nodes, represented by $S_1 ... S_3$, and has undirected *user-location* ties and directed *user-user* ties. Also, the right-most graph in Figure 3 is the result of our adaptation of the IP algorithm, generating a network graph that only has directed and weighted *user-user* ties and has some differences regarding its structure, e.g., the original *user-user* edges no longer exist and new edges arise from common visits to locations. The connection between two nodes is associated with a non-negative, non-zero weight if they share a visited location, e.g., U_3 and U_2 both visited location S_2 so there is a new edge from U_3 to U_2, with the weight w_1, because U_3 visited S_2 after U_2 had visited it.

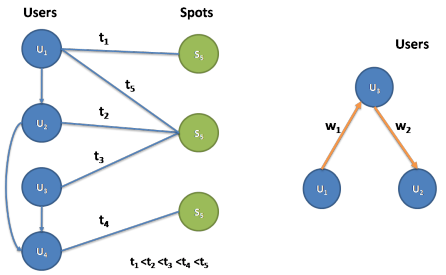

Figure 3: Transformation of the original network graph (left) to our IP algorithm graph (right).

3.4 Academic Social Networks

Alongside with the general social networks, our work focused on assessing the influence of nodes in an academic social network, which is a network where the nodes either refer to authors of scientific papers connected via co-authorship ties that form a *co-authorship network*, or to the scientific papers themselves connected through citation ties, originating a *citation network*. We wanted to assess which were the most influential papers in the scientific community, i.e., the ones that were gathering more attention either due to the importance of their author(s), due to being about a *trending* topic or an important breakthrough. To do so, we gathered the already organized data from the digital library DBLP, via the Arnetminer Project[6], which contains information about scientific papers from 1935 to 2011, including the abstract and the number of citations. From this data we built a *citation network* for set of time-stamps ranging from 2007 to 2011, as depicted in Figure 4, in order to have a record of how the network evolved over time.

[6] `http://arnetminer.org/DBLP_Citation`

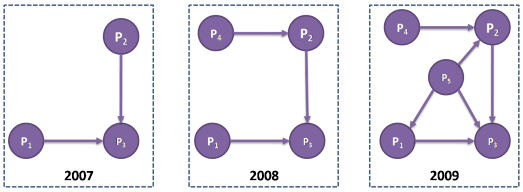

Figure 4: Citation graph for the DBLP dataset.

Although any other ranking algorithm could have been used, in the case of the DBLP citation network, the most influential papers on the dataset were determined through the computation of the PageRank algorithm. The *top-10* highest ranked papers were then selected and we gathered their full information, in order to cross-check the set of authors of each paper with the recipients of renowned computer science and engineering awards such as the Gerard Salton award or the Turing award, identifying which of these authors were distinguished by the scientific community.

From the thorough state-of-the-art study we have conducted, the temporal issues related to the ranking of a scientific article arose has a future work possibility. Instead of computing future PageRank scores of scientific papers based on their future citations, as did Sayyadi and Getoor [23], we created a framework to predict the Future PageRank scores of scientific papers in a citation network for a specific year, based on their previous PageRank scores, among other features. The same principle was also applied to the prediction of download counts for scientific articles downloaded from the ACM Digital Library *website* in the year of 2011. In order to predict the future PageRank scores and download counts, we have three distinct phases:

1. **Feature Vector Creation**
   First we prepare the input for further prediction of importance scores. Having the dataset, either for paper citations or downloads counts, one generates the different features, namely the text, age and PageRank scores and store them in the database, to generate feature vectors.
2. **Prediction**
   In a second step, one creates training and test files, in order to proceed with the computation of a machine learning technique intended for predicting the future PageRank scores and future download counts.
3. **Accuracy Assessment**
   Finally, to assess the quality of the obtained results, one proceeds with the computation of various evaluation metrics.

Each aforementioned phase is a preparation to following one. To predict the PageRank scores and the download counts we rely on features that can represent the characteristics of the information in the dataset. The following types of features was considered:

1. **Absolute Scores** - Includes the *PageRank* score resulting from the computation of the algorithm for papers that were published until a specific year, inclusive. Regarding the PageRank score of a paper, we defined 5 different cumulative time-stamps, from 2007 to 2011, so we could have access to the respective PageRank scores in each k previous year.
2. **Differential Scores** - Includes the *Rank Change Rate (Racer)*, representing the change rate of PageRank score between two different years, capturing the evolution of PageRank scores. The Rank Change Rate between to time-stamps t_i and t_{i+1}, for paper p is given by the following:

$$racer(p, t_i) = \frac{rank(p, t_{i+1}) - rank(p, t_i)}{rank(p, t_i + 1)} \quad (2)$$

3. **Profile Information** - Includes the *Average PageRank Score*, that represents the average of the PageRank score of all publications that have an author in common with the paper's set of authors, and the *Maximum PageRank Score*, which represents the maximum PageRank score of all publications that have an author in common with the paper's set of authors.
4. **Age** - Includes the difference between the present year and the publication year of a paper, i.e., its age.
5. **Text** - Includes the term frequency score for the top 100 most frequent tokens in abstracts and titles of publications, not having in consideration the terms from the Standard English stop-word list.

For each aforementioned type of feature, except *age* and *text*, its value for the previous k years, with k ranging from 1 to 3 was considered, e.g., when predicting the future PageRank score for year 2010, one predicted that score only with information from the PageRank score of the previous year (k =1, i.e., 2009), then with information from the two previous years (k = 2, i.e., 2009 and 2008) and finally from the three previous years (k=3, i.e., 2009, 2008, 2007).

In order to enrich the way we made our predictions, we made a structured combination of the previously enumerated types of features, which fit into three different groups:

- **1** - In this group we used exclusively the PageRank scores of the paper as features.
- **1 + 2** - In this group we used both *PageRank* and *Racer* scores of the paper as features.
- **1 + 2 + 3** - In this group we used *PageRank* scores, *Racer* scores, *Average Author* scores and *Maximum Author* scores as features.

The remaining *text* and *age* features were separately added to the aforementioned combination of features enabling the creation of two distinct subsets of results. Thus, alongside with the different range of k used, one could assess if for that particular type of feature or group of features, adding more information about previous years would improve or deviate the accuracy of our results. Also, for a straightforward

computation of the *racer*, *average PageRank score*, *average PageRank score* an feature vectors, the PageRank scores for each paper in each time-stamp and information about the authors of the papers and the information about download counts was stored in a relational database.

3.5 The Learning Approach

To predict future PageRank scores and future download counts we used an ensemble machine learning technique included in the RT-Rank[7] package, which is an open-source project consisting in the implementation of various machine learning algorithms based on regression trees. The algorithm we used, *Initialized Gradient Boosting Regression Trees* (IGBRT) - see Algorithm 4, is a *point-wise* machine learning algorithm developed by the team from Washington University of St. Louis for the *2010 Yahoo Learning-To-Rank Challenge* and is based on *Gradient Boosting Regression Trees* (GBRT) [20]. Based on regression trees [13], GBRT is a machine learning technique also based on tree averaging, which uses a set of trees to classify a new object, instead of the single *best* tree [21]. It sequentially adds small trees (d≈ 4), each with high bias and, in each iteration, the new tree to be added focuses strictly on the objects that are responsible for the current remaining regression error. IGBRT follows the guidelines of SVM^{light} [8], proposed by T. Joachims [15, 14]

Algorithm 4 Initialized Gradient Boosted Regression Trees (Squared Loss)

Input: data set $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, Parameters: α, M_B,d, K_{RF}, M_{RF}
F ← RandomForests(D,K_{RF}, M_{RF})
Initialization: $r_i = y_i - F(x_i)$ for $i = 1 \rightarrow n$
for $i = 1 \rightarrow M_B$ **do**
  $T_t \leftarrow Cart(\{(x_1, r_1), ..., (x_n, r_n)\}, f, d)$ {Build Cart of depth d, with all f features, and targets r_i}
  for $i = 1 \rightarrow M_B$ **do**
    $r_i \leftarrow r_i - \alpha T_i(x_i)$ {Update residual of each sample x_i}
    $T(\cdot) + \alpha \sum_{t=1}^{M_B} T_t(\cdot)$ {Combine the Regression Trees $T_1, ..., T_m$ with the RF F}
  end for
end for
return $T(\cdot)$

With the intention of addressing the GBRT's weakness, i.e., the inherent trade-off between the step-size and the early stopping, Mohan et al. proposed an ensemble algorithm that starts-off at a point very close to the global minimum and refines the already good predictions [20]. Thus, instead of initializing the algorithm with an all-zero function, as occurred in GBRT, the IGBRT algorithm is initialized with the predictions of Random Forests [4], due to the latter being known as resistant towards overfitting, insensitive to parameter settings and not implying additional parameter tunning. IGBRT uses GBRT to further refine the results of Random Forests, which are regarded by the authors as a good starting point for the algorithm.

4. RESULTS AND DISCUSSION

[7] https://sites.google.com/site/rtranking/
[8] http://svmlight.joachims.org/

This section presents the results of the undertaken experiments and the evaluation methodology used to assess the veracity of the obtained results. Beginning with a concise characterization of all the datasets that were used, the evaluation methodology is then presented, comprising all the metrics that were used to assess the quality and veracity of the results. Finally, the obtained results for each experiment are presented and further discussed. The results comprise the experiments for finding influencers in Twitter, FourSquare and the citation network built upon the DBLP dataset, as well as, the experiments for predicting the future PageRank score of a scientific papers from 2010 and 2011 in the DBLP citation network and the prediction of download counts for the scientific papers published in 2011, downloaded from the ACM Digital Library.

4.1 Datasets

This section includes the dataset and network characterization of all the datasets that we used. In order to understand the structural differences between a location-based social network and a social network that only consists in relationships between users, and how this structure affects influence estimation, we created two different graphs for both FourSquare and Twitter datasets. First we considered a graph consisting in the location-based network built upon the data that was crawled, which we called the *User+Spot Graph*. Afterwards, we disregarded all the *user-location* relationships and built a graph consisting only in *user-user* ties, which we called the *User Graph*.

In the case of the DBLP dataset, the distinction between two graph was not needed, because our focus was on creating a citation network upon which we could estimate the PageRank scores of their nodes and use them as features for the algorithm that predicts future influence scores of papers and future download counts.

		FourSquare	Twitter
Spots	Total	48,257	1,358
	HTM Resolution 10	—	13
	HTM Resolution 20	—	1,277
	HTM Resolution 25	—	1,358
Users	Total	447,545	2,603,505
	Relations	970,587	3,218,997
	Visiting Spots	16,960	1,017
Arcs	PageRank & HITS (User+Spot Graph)	2,539,986	3,757,555
	PageRank & HITS (User Graph)	1,017,887	3,576,157
	IP Algorithm	1,017,887	
Nodes	PageRank & HITS (User+Spot Graph)	451,664	2,604,863
	PageRank & HITS (User Graph)	403,407	2,603,505
	IP Algorithm	447,545	
InDegree	Minimum (User+Spot Graph)	0	1
	Maximum (User+Spot Graph)	3,166	38,542
	Average (User+Spot Graph)	2.8626	5.6162
	Minimum (User Graph)	0	1
	Maximum (User Graph)	3,166	38,452
	Average (User Graph)	2.5478	5.6256
OutDegree	Minimum (User+Spot Graph)	0	1
	Maximum (User+Spot Graph)	1,000	460,466
	Average (User+Spot Graph)	74.8821	1.5615
	Minimum (User Graph)	0	1
	Maximum (User Graph)	1,000	460,466
	Average (User Graph)	60.5829	1.5618
Average Degree	Total (User+Spot Graph)	5.4640	3.8868
	Users (User+Spot Graph)	5.6714	2.8878
	Spots (User+Spot Graph)	5.7118	1.0376
	Total (User Graph)	5.0488	2.8872
Average Path Length	User+Spot Graph	4.736940	3.9776
	User Graph	4.7764	3.9823
Clustering Coefficient	User+Spot Graph	0.2987	0.1155
	User Graph	0.3718	0.1152

Table 1: Characterization of the FourSquare and Twitter networks.

Regarding the characteristics of both graphs in the FourSquare and Twitter datasets depicted in Table 1, one can acknowledge that while the first dataset is more complete in terms of *user-location* ties and quantitative spot information, the latter is more complete in terms of *user-user* ties and user friendship information. We have this *behaviour*, since FourSquare is a pure location-based network focused on sharing the locations users have visited, while Twitter is a microblogging and social network platform focused on the exchange of messages between users, thus giving priority to the relationship between the user and his friends and followers. In what regards the HTM resolution, we used a resolution of 26.

When considering the average path length and the clustering coefficient, one can assess that while the nodes in FourSquare network are more close to each other, neighbours of nodes in Twitter are more close to one another than in FourSquare. The latter phenomena has to do with the fact that we could collect a greater extent of data for friends of users in the Twitter dataset, resulting in the scenario where friends of different users can, themselves, be friends and/or have friends in common. Also, one can observe that the *User Graph* has naturally a greater average path length and a greater clustering coefficient than the *User+Spot Graph*, because the *User Graph* as less nodes and, thus, shortens the distance between users and neighbourhoods of users, previously parted by the spots between them.

Regarding the *degree* distribution in the FourSquare and Twitter networks in both *User+Spot Graph* and the *User Graph*, one can acknowledge from Figure 5 that the *degree* distribution for these datasets follows a *power-law distribution*, which a characteristic of large-scale networks, i.e., networks in which the majority of the nodes very few connections, while very few nodes have a high number of connections. Nevertheless, from the values of *average path length* and *clustering coefficient*, one can say that both FourSquare and Twitter networks are not representative of large-scales networks, because in large-scale networks, besides the *power-law* distribution for the *degree*, the *average path length* must be much smaller than the *clustering coefficient*, revealing that the nodes are very close to each other and their neighbourhoods are highly clustered.

The academic citation network built upon DBLP data comprises scientific papers from 1935 to 2011 and, from Table 2, one can also have an idea of the dimension of the dataset for each of the considered time-stamps, as well as, how complete the information about the scientific papers is, e.g., via the number of papers with abstract.

	Publications	Citations	Authors	Papers with Downloads	Papers with Abstract	Average Terms Per Paper
Overall	1,572,277	2,084,019	601,339	17,973	529,498	104
2007	135,277	1,150,195	330,001	15,516	343,837	95
2008	146,714	1,611,761	385,783	17,188	419,747	98
2009	155,299	1,958,352	448,951	17,973	504,900	101
2010	129,173	2,082,864	469,719	17,973	529,201	103
2011	8,418	2,083,947	469,917	17,973	529,498	104

Table 2: Characterization of the DBLP dataset.

On the other hand, in Table 3 one can acknowledge from this network characterization that the academic social network that was built naturally grows in each time-stamp, although this growth is not as significant in the last two time-stamps as it is in the first two. Focusing on the average path length and the clustering coefficient, one can conclude that as we introduce more papers in the network, i.e., at each time-stamp, papers are closer to one another through the existence of more citation relationships between them, even though they tend not to be as clustered together over time.

From the plots in Figure 6, one can acknowledge that the number of papers increases trough the years. However, these new paper have tend to have few citations, and so the *tail* of the plots get *ticker* throughout the years, i.e., new fewer cited papers are frequently added to the dataset, while the number highly cited paper remains almost unaltered.

4.2 Evaluation Methodology

When assessing the quality and veracity of the results for the *top-10* highest ranked users and spots in the FourSquare and Twitter datasets, we conducted an empirical analysis and relied on profile information, due to the fact that, this research area is still evolving and there are not strict parameters or ground-truth lists to truly assess the influence of a node in these networks. On the other hand, when assessing the veracity of the DBLP *top-10* highest ranked papers, we empirically analyzed our results against a list of recipients of renowned scientific awards, like the Gerard Salton Award and the Turing Award, and if they were not part of that list, we also checked their academic publication profiles[9] in order to assess if they were renowned scientists.

In the case of the experiment of future *PageRank* and future download count prediction, we used a set of error metrics. One of these metrics is *Kendall's Tau*, which corresponds to a value ranging between $[-1, 1]$ and is defined as follows:

$$\tau = \frac{2c_i}{\frac{1}{2}n_i(n_i - 1)} - 1 \tag{3}$$

In the formula, c_i is the number of concordant pairs between the produced ranked list and the ground truth list, and n_i is the length of the two lists [17]. The aforementioned software package *LAW-Webgraph* includes an implementation of this metric.

We can also assess the level of correlation between two ranked lists using *Spearman's Correlation* (i.e *Spearman's* ρ), according to the formula bellow:

$$\rho = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n^3 - n} \tag{4}$$

In the formula, $x_1, ..., x_n$ and $y_1, ..., y_n$ are the two rankings of n objects [2]. This metric was computed via its implementation in the *R-Project* open source statistical software[10]. Both Kendall's Tau and Spearman's Correlation measure the strength of the association between two ranked lists [6]. The correlation ranges between $[-1, 1]$ and, hence, if it is

[9] `http://academic.research.microsoft.com/`
[10] `http://www.r-project.org/`

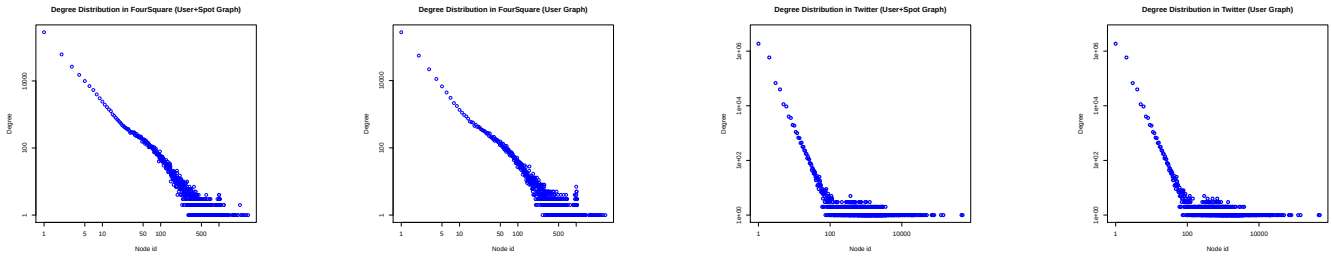

Figure 5: Degree distribution for nodes in the *User+Spot Graph* and the *User Graph*, from the FourSquare and Twitter datasets.

	In-Degree			Out-Degree			Degree			Average	Clustering
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Path Length	Coefficient
2007	0	1,508	2.9153	0	227	2.9153	0	1,508	5.8329	0.1323	6.1800
2008	0	1,875	3.5357	0	266	3.5357	0	1,875	7.0790	0.1319	6.1047
2009	0	2,207	3.6993	0	269	3.6993	0	2,207	7.4012	0.1314	6.0833
2010	0	2,306	3.7670	0	269	3.7670	0	2,306	7.5430	0.1312	6.0665
2011	0	2,311	3.7673	0	269	3.7673	0	2,311	7.5367	0.1310	6.0676

Table 3: Characterization of the DBLP network.

close to -1, one can determine the variables are *negatively correlated*, whereas if it is close to $+1$ they are *positively correlated*. To perform the Spearman's Correlation we used the R-Project for statistical computing, which a specific statistical language and open-source software package that includes various mathematical and statistical techniques, being also suitable for large amounts of data.

In order to measure the accuracy of the prediction models, we used the normalized root-mean-squared error (NRMSE) metric between our predictions and the true values, which is given by the formula:

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n}(x_{1,i}-x_{2,i})^2}{N}}}{x_{\max} - x_{\min}} \quad (5)$$

The average of absolute error, which is the average of the difference between the inferred, i.e., predicted value and the actual value, was also used and specially relevant for assessing the quality of the predictions of download counts.

4.3 Results

This section exhibits the results obtained from the various conducted experiments, alongside with their discussion. First of all, the results from the experiments for finding influencers in FourSquare and Twitter, as well as, for the BDLP citation network are presented and further discussed, where we assess the quality of these results and if the *top-10* highest ranked list of individuals and spots produced by the different algorithms really corresponds to the *top-10* of influencers and influential spots in the network.

The results for the experiment of predicting future PageRank scores and download counts are then presented, alongside with their discussion, where we compare the output of the different evaluation metrics that were computed for the different groups of features, in order to understand if the task of predicting a future PageRank score and the future download counts could be successfully accomplished with the framework that was developed.

Also, through the plot of the *indegree*, *outdegree* and *degree* distribution in FourSquare and Twitter datasets, we will be able compare the structure of both types of network graph and discuss their characteristics.

4.4 Finding Influencers

In the following sections the results of the computation of PageRank, HITS and IP algorithms for the FourSquare and Twitter datasets are presented, as well as, the results of the computation of PageRank algorithm for the DBLP dataset. While the first two datasets comprise the *top-10* highest ranked users and the *top-10* highest ranked spots in the network, the results from the DBLP highlight solely the most influential papers in the DBLP digital library dataset.

We begin by exposing and discussing the results from the experiments with, respectively, the FourSquare and Twitter datasets, then we present and discuss the influence estimation for the DBLP dataset, closing this section with the results from the future PageRank scores and download counts experiment.

In order to identify the most influential users and spots in FourSquare and Twitter datasets, average anonymous users and spots (e.g., streets) are identified, respectively, by $Person - XXXX$ and $Spot - YY : ZZ$, where $XXXX$ corresponds to the real *user id*, YY corresponds the latitude and ZZ to the longitude associated with that *spot id* in the network, while publicly well-known companies, locations/venues and people are identified by their real name, e.g., Ellen DeGeneres for users and Dunkin' Donuts for spots.

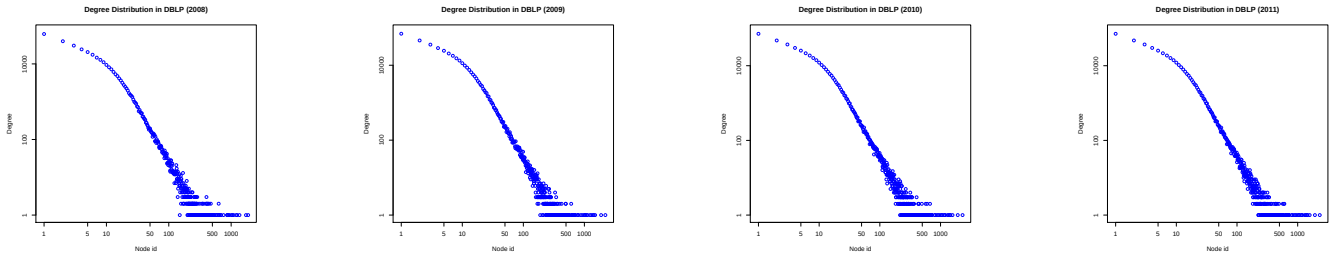

Figure 6: Degree distribution for the DBLP dataset from 2008 to 2011.

4.4.1 *Location-based social networks: FourSquare & Twitter*

From the user influence scores for PageRank and HITS algorithm depicted in Table 4, one can acknowledge that the addition of spots to the network reveals well-known influentials, such as worldwide celebrities, TV channels or magazines.

PageRank			HITS - Authority			HITS - Hub		
Name	Friends	Likes	Name	Friends	Likes	Name	Friends	Likes
TimeOut NY	—	122,172	ZAGAT	—	328,189	ZAGAT	—	328,189
Lucky Mag.	—	164,323	TimeOut NY	—	122,172	MTV	—	731,067
ZAGAT	—	328,189	MTV	—	731,067	Bravo TV	—	375,363
NYPL	—	61,132	Bravo Tv	—	375,363	History Chnl	—	541,847
MTV	—	731,067	History Chnl	—	541,847	The NY Times	—	367,008
Person-12935563	956	20	Starbucks	—	929,915	Starbucks	—	929,915
Bravo TV	—	375,363	The NY Times	—	367,008	VH1	—	380,987
Person-1478079	981	96	Lucky Mag.	—	164,323	People Mag.	—	372,008
NYC Parks	—	17,429	VH1	—	380,987	TimeOut NY	—	122,172
History Chnl	—	541,847	NYPL	—	61,132	The WSJ	—	227,894

Table 4: User influence scores for PageRank and HITS algorithms, for the *User+Spot Graph*, from the FourSquare dataset.

Meanwhile, when we have the *User Graph*, as depicted in Table 5, the average users of social platforms are distinguished both in the PageRank and the HITS algorithms, the latter when ordered by *hub scores*. In this case, average users are highlighted through their great amount of mayorships, checkins, tips about locations and friends. Mostly through their *outlinks*, they become network users that other users want to follow and *listen to*.

PageRank			HITS - Authority			HITS - Hub		
Name	Friends	Likes	Name	Friends	Likes	Name	Friends	Likes
Person-11890308	794	84	ZAGAT	—	328,189	Person-2630685	110	817
Person-449480	1,000	374	MTV	—	731,067	Person-1127366	39	749
Person-1544684	987	144	Bravo TV	—	375,363	Person-4148169	77	899
Person-619656	823	8	History Chnl	—	541,847	Person-634270	216	755
Person-4071912	1,004	860	Starbucks	—	929,915	Person-42695	128	775
NYCHA	807	59	The NY Times	—	367,000	Person-1011520	39	723
Person-6935835	990	275	VH1	—	380,987	Person-3231666	14	713
Person-6004767	958	319	Ellen DeGeneres	—	457,155	Person-7991820	3	767
Person-10934560	1,001	64	TimeOut NY	—	122,172	Person-3290360	62	632
Person-10554269	985	4	People Mag.	—	372,008	Person-6483868	95	765

Table 5: User influence scores for PageRank and HITS algorithms, for the *User Graph*, from the FourSquare dataset.

When the location-based network was reshaped to connect only the users that have visited at least one location in common, for the IP algorithm, the average user of FourSquare is distinguished, yet again due to a combination of factors that include their great amount of mayorships, checkins, tips about locations and friend counts, as one can acknowledge from Table 6.

In brief, the fact that worldwide TV channels, magazines, and celebrities are highlighted in a network that contains both users and spots reveals a strict connection between these well known influentials and the spots, through a continuous activity that is intended to gather and retain their followers. When these ties are removed, the connections between *real users* prevail.

Name	Friends	Likes
Person-9797197	52	10
Person-9726342	5	—
Person-9615360	25	9
Person-9578554	34	—
Person-9553862	4	—
Person-9450025	47	7
Person-9264407	43	—
Person-8956766	28	—
Person-8916830	47	4
Person-884020	95	32

Table 6: User influence scores for the IP algorithm, from the FourSquare dataset.

As for the most influential spots in the FourSquare dataset, the *top-10* highest ranked spots resulting from the computation of both PageRank and HITS algorithms, either with *authority* or *hub* sort, was the same. Focusing on the type of spots that were highlighted, they mainly include bars, boardwalks and other spots near the New York coastline due to the fact that the data collection was done during the months of August and early September of 2012.

Name	Checkins
Tattoo Shot Lounge	227
Dunkin' Donuts	970
Gargiulo's Restaurant	697
The Freak Bar	540
Ruby's Bar & Grill	2,025
Coney Island Beach & Boardwalk	36,206
Cha Cha's	1,142
Denny's Delight	84
Coney Island Sound	280
Coney Island Polar Bear Club	85

Table 7: Spot influence scores for PageRank and HITS algorithms (that present the exact same *top-10*), for the *User+Spot Graph*, from the FourSquare dataset.

When finding influencers in the Twitter dataset, one must acknowledge that users *tweet* wherever they are, may it be at home, while waiting for a doctor's appointment, etc. Therefore many of the locations that we could identify are not necessarily venues, i.e., the geographic coordinates associated with a *tweet* may *point* to a street or avenue, and not

a theater, museum or restaurant like it happened in the FourSquare experiment. Nevertheless, this is only due to the inner characteristics of the Twitter social network, which is *content and user-centered* and not *location-centered* like FourSquare. Due to the fact that social networks have a *dynamic* behaviour, i.e., they can change over time with the addition or loss of users and relationship ties, the third highest ranked user for HITS - Authority, from Tables 8 and 9 had a profile on Twitter and was active during our crawl, between July and August of 2012, nevertheless, he no longer has a Twitter profile thus, being marked with a *, after the *user id*. Also, for this dataset, the results from the computation of IP algorithm are not be presented, because the obtained results were not coherent and not nearly comparable with the ones that were obtained in FourSquare.

From Table 8, we can observe that HITS algorithm, with influence sorted by authority or hub score, reveals Twitter users that are well-known to the public and whom exert significant influence due to their roles on society, e.g., by being an entrepreneur, a journalist or an actor. Also, due to their professional activity and media exposure, one can say that they can *shape conversations*, they are users other network users want to listen to. Conversely, from the *top-10* generated by PageRank algorithm, one can acknowledge that friendship ties among anonymous (to public) users are highlighted.

Regarding the *User Graph*, we can see that the output from HITS an PageRank algorithms, depicted in Table 9, is exactly the same as in the *User+Spot Graph*. This enhances the fact that in this particular dataset there is a greater number of relationships among users than between users and locations, so when these location ties are disregarded the strong ties between users naturally prevail. Also, one can see from Tables 8 and 9 that, yet again, the total number of follower and friends is not necessarily correlated with influence on Twitter.

As one can observer from Table 10, a great majority of the *top-10* highest ranked scores are not venues *per se*, the geographical locations associated with these *tweets* correspond to streets or avenues, due to the use of Twitter in various mobile applications. Nevertheless, some well known spots like Times Square and JFK are naturally highlighted. Also, one can acknowledge that, in this particular case, the spots with greater number of checkins turn out to be the most influential spots in the dataset.

4.4.2 Academic social network: DBLP

In Table 11 are the *top-10* highest ranked papers from the citation network built upon DBLP data, where recipients of scientific awards are highlighted in bold. From this table one can acknowledge that the *top-10* remained unaltered for scientific papers published until 2010 and until 2011, and that the majority of these publications are authored by recipients of one or more of the renowned awards, like the Gerard Salton Award and the Turing Award.

Focusing on the title of these scientific papers, one can also verify that this *top-10* comprises publications that can be considered breakthroughs in a specific research area, e.g., Gerard Salton's leading work in information retrieval, or inevitable textbook references, e.g., Cormen et al.'s *Introduction to Algorithms*. Nevertheless, even if the authors aren't recipients of renowned scientific awards, the fact that they collaborate with many other authors lead them to be cited in a greater number of publications, reinforcing their PageRank score.

4.5 Predicting Future PageRank scores and Download counts

In this section, the experiment regarding the prediction of future influence scores and future download counts is detailed and thoroughly discussed. For a better understanding, we call the model for predicting future PageRank scores and download counts that includes the age of each article the *age model* and the model that includes age of the article and the term frequency of the 100 most frequent words in the abstract and title of each paper the *text model* - see Table 12.

From Table 12 and considering the experience of predicting the PageRank scores for the year of 2010, both models have provided very similar results, both improving as we added more information, i.e, comparing the three groups of features (PageRank Scores, PageRank scores with *racer* scores, and PageRank scores with *Racer* scores, Average PageRank score of the author and Maximum PageRank score of the author) and also comparing within the same groups, the quality of the results improves consistently. Only for the set of features that combine the PageRank score of one previous year with its respective *Racer* and the author's Average and Maximum PageRank score, the *age* model is outperformed by the *text* model. Comparing the *error rate* for the same year, one can assess that, for both models, as we add more information the *error rate* increases, resulting in the deviation of the results. Nevertheless, for the first two groups of features, the *text* model has a lower *error rate* than the *age* model, while the opposite happens for the third group of features.

Having computed the *absolute error* for all the groups of features in both models, the results show that, on average, the *text* model has always a lower *absolute error* than the *age* model.

For the year of 2011, as we add more information to the models, the *text* model outperforms the *age* model, as shown in the last two sets of features from the third group. Also, in the scenario in which the models only have the information about the immediately previous PageRank score, the *age* model is again outperformed by the *text* model. Nevertheless, when considering the *error rate* for both models for this year, the *text* model has an overall higher *error rate* than the *age* model showing that, even though the quality of the predicted results is lower in the *age* model, the results are more accurate.

As occurred for the computation of the *absolute error* for the year 2010, in all groups of features in both models, the results for the year of 2011 show that, on average, the *text* model has a lower *absolute error* than the *age* model.

Regarding the prediction of download counts depicted in Table 13, one can acknowledge that using a *text* model in-

PageRank			HITS - Authority			HITS - Hub		
Name	Followers	Following	Name	Followers	Following	Name	Followers	Following
Person-67779865	45,702	41,870	Jenna Wortham (NY Times Tech Reporter)	463,772	3,424	Jonah Lupton	301,965	276,780
Jeff Keni Pulver (Entrepreneur)	469,092	38,542	Jeff Keni Pulver (Entrepreneur)	469,092	38,542	NOH8 Campaign	426,079	251,158
JobsDirectUSA.com	17,075	18,782	Person-325410549*	—	—	Person-25915690	595,404	192,241
Person-479562736	16,703	16,241	Baratunde Thurston (Comedian, Actor)	124,722	5,707	Mike Allen (Journalist)	144,540	55,678
America Hires	11,824	13,006	StumbleUpon	72,133	10,370	Person-203455506	188,527	41,190
Person-52306188	9,989	9,878	DL Hughley(Actor Comedian)	73,835	886	New York Daily News	85,821	10,681
Person-35844123	10,030	9,761	John Rampton (Entrepreneur)	47,593	578	Person-18704291	19,212	21,098
Person-24883913	11,191	9,583	Person-51560438	103,721	14.766	Jason Calacanis (Entrepreneur)	151,155	112,248
Person-213105865	8,531	9,965	Person-67779865	45,699	41,868	92YTribeca	13,015	10,560
Person-30735143	7,837	8,513	Person-1536651	34,216	456	C.C. Chapman	34,512	28,505

Table 8: User influence scores for PageRank and HITS algorithms, for the *User+Spot Graph*, from the Twitter dataset.

PageRank			HITS - Authority			HITS - Hub		
Name	Followers	Following	Name	Followers	Following	Name	Followers	Following
Person-67779865	45,702	41,870	Jenna Wortham (NY Times Tech Reporter)	463,772	3,424	Jonah Lupton	301,965	276,780
Jeff Keni Pulver (Entrepreneur)	469,092	38.542	Jeff Keni Pulver (Entrepreneur)	469,092	38,542	NOH8 Campaign	426,079	251,158
JobsDirectUSA.com	17,075	18,782	Person-325410549*	—	—	Person-25915690	595,404	192,241
Person-479562736	16,703	16,241	Baratunde Thurston (Comedian, Actor)	124,722	5,707	Mike Allen (Journalist)	144,540	55,678
America Hires	11,824	13,006	StumbleUpon	72,133	10,370	Person-203455506	188,527	41,190
Person-52306188	9,989	9,878	DL Hughley(Actor Comedian)	73,835	886	New York Daily News	85,821	10,681
Person-35844123	10,030	9,761	John Rampton (Entrepreneur)	47,593	578	Person-18704291	19,212	21,098
Person-24883913	11,191	9,583	Person-51560438	103,721	14.766	Jason Calacanis (Entrepreneur)	151,155	112,248
Person-213105865	8,531	9,965	Person-67779865	45,699	41,868	92YTribeca	13,015	10,560
Person-30735143	7,837	8,513	Person-1536651	34,216	456	C.C. Chapman	34,512	28,505

Table 9: User influence scores for PageRank and HITS algorithms, for the *User Graph*, from the Twitter dataset.

creases the quality of our results. In the *age* model, we can verify that adding information about the *Racer* to the previous PageRank scores affects the results negatively, while combining previous PageRank scores with *Racer*, and the author's Average and Maximum PageRank scores provides better results with a lower *error rate*. From this fact, we can conclude that the *age* model provides a more accurate prediction as it becomes more complete. The opposite happens in all groups of the *text* model, i.e., as we, within the same group, add more information to the model, one can acknowledge that the quality of the results decreases, even though they are far better than the corresponding results in the *age* model.

We can also verify that the *age* model, for the groups of features that only include previous PageRank scores, and for the ones that combine previous PageRank scores with *Racer* and author's Average and Maximum PageRank scores, have a lower *error rate* than the corresponding groups in the *text* model. And even though *text* model has better overall results, the *error rate* is greater than in the *age* model for download counts prediction.

As for the *absolute error* the results showed that, generally, the *text* model has a lower *absolute error* rate than the *age* model in all groups, except the third.

In brief, from the results in Tables 12 and 13, we can acknowledge that predicting the number of downloads is an harder task than predicting the future PageRank scores. We can also see that, when predicting future PageRank scores, as more information is added to the model, the more the results deviate. Nevertheless, the opposite happens when we are trying to predict the number of downloads.

Comparing the years of 2010 and 2011, we can acknowledge that predicting the PageRank scores of a more recent year is easier than if we progressively go back in time to predict the PageRank score of a more distant year.

	Features	ρ	τ	NRMSE
Age	Rank k = 1	0,3864814	0,2742998	0,0080585
	Rank k = 2	0,4221492	0,3001470	0,0029377
	Rank k = 3	0,4323201	0,3080974	0,0028074
	Racer + Rank k = 1	0,4396605	0,3076576	0,0076713
	Racer + Rank k = 2	0,3370149	0,4747241	0,0078403
	Racer + Rank k = 3	0,3313412	0,4612442	0,0088301
	A + R + Rank k = 1	0,3377553	0,2558403	0,0147155
	A + R + Rank k = 2	0,5335481	0,3894899	0,0088093
	A + R + Rank k = 3	0,5406937	0,3962472	0,0078576
Text	Rank k = 1	0,5250188	0,3837016	0,0086955
	Rank k = 2	0,5261168	0,3849615	0,0087775
	Rank k = 3	0,5060003	0,3674801	0,0091976
	Racer + Rank k = 1	0,5325432	0,3887987	0,0085328
	Racer + Rank k = 2	0,5224018	0,3822982	0,0089440
	Racer + Rank k = 3	0,5087407	0,3703400	0,0091979
	A + R + Rank k = 1	0,5709764	0,4234845	0,0076071
	A + R + Rank k = 2	0,5651282	0,4180070	0,0079000
	A + R + Rank k = 3	0,5608946	0,4148554	0,0088935

Table 13: Results for the prediction of download numbers for papers in the DBLP dataset.

5. CONCLUSIONS AND FUTURE WORK

With our experiments we could perform a detailed characterization of the aforementioned social networks, and verify that social network analysis techniques can be used to assess the most influential nodes of a network. As for the prediction of future influence scores, we can conclude that the framework that was developed for academic citation networks provides reliable and accurate estimations, very close to the real values.

A major limitation of this work resides in the evaluation of the results regarding location-based networks. Unlike academic social networks, where one can either assess the validity of the most influential authors or the most influential articles through an extensive list of renowned scientific awards that have been earning prestige throughout the years, social network analysis and, most specifically, location-based networks is a recent area of studies in which one does not yet have a list of characteristics that indicate without flaws that a user or a spot is influential, or a series of public prizes that

PageRank		HITS - Authority		HITS - Hub	
Name	Checkins	Name	Checkins	Name	Checkins
Broadway - Times Square	4	Pace University	8	Spot40.71498749:-73.95485289	2
JFK Airport	2	Spot40.679254:-73.8632521	1	Spot40.7827699:-73.95211752	1
JFK Airport (Subway Station)	1	Spot40.67982674:-73.86344992	1	Spot40.76619859:-73.91322359	1
Spot40.80567362:-73.91862858	1	Spot40.6792906:-73.8622276	1	Skin Magic Ltd	1
Spot40.66931554:-74.20359207	1	Park Lane Hotel	1	Spot40.76614592:-73.91323331	1
Spot40.73262798:-73.98359375	1	Astoria Bowl	1	Spot40.76616717:-73.91319381	1
Rosa Mexicano (Restaurant)	1	Spot40.7166368:-73.9543937	1	Broadway - Times Square	1
The Abyssinian Baptist Church	1	Columbus Circle	1	Spot40.75612638:-73.90477465	1
St Luke's School	1	Spot40.86745661:-74.12978901	1	Spot40.76113205:-73.97952078	1
Spot40.742727:-73.994372	1	Spot40.89064994:-73.89948689	1	JFK Airport	1

Table 10: Spot influence scores for PageRank and HITS algorithms, for the *User+Spot Graph*, from the Twitter dataset.

Paper	Authors	PageRank 2010	PageRank 2011
A Unified Approach to Functional Dependencies and Relations	**Philip A. Bernstein**, J. Richard Swenson, Dennis Tsichritzis	0,000903919	0,000903646
On the Semantics of the Relational Data Model	Hans Albrecht Schmid, J. Richard Swenson	0,000891394	0,000891123
Database Abstractions: Aggregation and Generalization	John Miles Smith, Diane C. P. Smith	0,000860181	0,00085993
Smalltalk-80: The Language and Its Implementation	**Adele Goldberg**, David Robson	0,000763314	0,000763174
A Characterization of Ten Hidden-Surface Algorithms	**Ivan E. Sutherland**, Robert F. Sproull, Robert A. Schumacker	0,000716136	0,000716507
An algorithm for hidden line elimination	R. Galimberti	0,000706674	0,000707118
Introduction to Modern Information Retrieval	**Gerard Salton**, Michael McGill	0,000699671	0,000699584
C4.5: Programs for Machine Learning	**J. Ross Quinlan**	0,000635416	0,000636705
Introduction to Algorithms	**Thomas H. Cormen**, **Charles E. Leiserson**, **Ronald L. Rivest**	0,000592198	0,000592414
Compilers: Princiles, Techniques, and Tools	**Alfred V. Aho**, Ravi Sethi, **Jeffrey D. Ullman**	0,000528325	0,000528235

Table 11: PageRank scores for top-10 highest ranked papers of the DBLP dataset.

award people, companies or spots due to their relevance and influence in a specific context. Therefore, this task had to be done by comparison to well known state-of-the-art social network analysis metrics. Also, social networks are dynamic, so that set of users or spots that can be considered influent or trendy today, might be different if we make the same estimation, within the same conditions, in a couple of months or a year.

In terms of future work, it would be important to address all the tasks that I initially intended fulfill, namely conduct rank aggregation in the aforementioned experiments. It would also be very interesting to find the most influential users and spots for more complete datasets, which could result in much richer networks and subsequent analysis.

Taking advantage of the fact that this research area is still in its infancy, we could combine the work of my MSc thesis with the work of Lima and Musolesi, which adapts well known local and global social network analysis metrics like *degree* or *clustering coefficient* that are location-agnostic, giving them a spatial context, e.g., to calculate the *degree* of a node in the network, but only considering the friends of this node that are associated with a specific geographical location, such as a city or a state [18].

Also, due to the fact that social networks are *dynamic networks*, i.e, its structure can change overtime with the addition or loss of nodes and relationships, we could integrate state-of-the-art frameworks and algorithms in order to include the passage of time in the networks we have studied. Even though *dynamic networks* have been frequently addressed regarding network visualization [9], works such as of [1] break away from conventional networks analysis, by proposing a mathematical framework for *dynamic network* analysis.

On the other hand, we could also extend our work with the implementation of temporal distance metrics proposed by Tang et al., that could be applied to networks that change over time and allow us to capture the properties of these time-varying graphs, such as *delay*, *duration* and *time order* of interactions between nodes [26].

	Features	PageRank 2010 ρ	PageRank 2010 τ	PageRank 2010 NRMSE	PageRank 2011 ρ	PageRank 2011 τ	PageRank 2011 NRMSE
Age	Rank k = 1	0,9725065	0,9163994	0,0003224	0,9929880	0,9837121	0,0001057
	Rank k = 2	0,9836493	0,9381865	0,0006161	0,9999050	0,9994758	0,0000995
	Rank k = 3	0,9890716	0,9506366	0,0006391	0,9999002	0,9993787	0,0004768
	Racer + Rank k = 1	0,9724540	0,9173649	0,0003469	0,9998887	0,9994037	0,0002322
	Racer + Rank k = 2	0,9837098	0,9387564	0,0006520	0,9999004	0,9992955	0,0001634
	Racer + Rank k = 3	0,9888725	0,9493687	0,0006605	0,9952435	0,9866206	0,0005492
	A + R + Rank k = 1	0,9675213	0,9098510	0,0005354	0,9998529	0,9994497	0,0002530
	A + R + Rank k = 2	0,9840530	0,9355465	0,0008336	0,9998353	0,9993422	0,0002962
	A + R + Rank k = 3	0,9892456	0,9468673	0,0006986	0,9938021	0,9828511	0,0005317
Text	Rank k = 1	0,9708719	0,9101722	0,0003608	0,9992124	0,9979693	0,0002479
	Rank k = 2	0,9831039	0,9310399	0,0006268	0,9997962	0,9992362	0,0004543
	Rank k = 3	0,9886945	0,9451537	0,0006276	0,9995012	0,9983375	0,0005800
	Racer + Rank k = 1	0,9711170	0,9098901	0,0005515	0,9994290	0,9984499	0,0001590
	Racer + Rank k = 2	0,9832037	0,9314405	0,0006747	0,9997300	0,9990720	0,0001919
	Racer + Rank k = 3	0,9887959	0,9470102	0,0006667	0,9994104	0,9980729	0,0006416
	A + R + Rank k = 1	0,9705230	0,9984499	0,0001590	0,9997019	0,9990583	0,0002480
	A + R + Rank k = 2	0,9837012	0,9990720	0,0001919	0,9998617	0,9993443	0,0002800
	A + R + Rank k = 3	0,9888386	0,9980729	0,0006416	0,9998793	0,9993885	0,0006987

Table 12: Results for the prediction of impact PageRank scores for papers in the DBLP dataset.

References

[1] T. Y. Berger-Wolf and J. Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[2] D. J. Best and D. E. Roberts. Algorithm as 89: The upper tail probabilities of spearman's rho. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(3), 1975.

[3] J. Bollen, M. A. Rodriguez, and H. Van de Sompel. Journal status. *Scientometrics*, 69(3), 2006.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1), 2001.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, 1998.

[6] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 2010 International AAAI Conference on Weblogs and Social Media*, 2010.

[7] P. Chen, H. Xie, S. Maslov, and S. Redner. Finding scientific gems with google's pagerank algorithm. *Journal of Informetrics*, 1(1), 2007.

[8] D. J. Cook and L. B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[9] B. S. Demoll and D. Mcfarland. The Art and Science of Dynamic Network Visualization. *Journal of Social Structure*, Volume 7, 2005.

[10] Y. Ding, E. Yan, A. Frazho, and J. Caverlee. Pagerank for ranking authors in co-citation networks. *Journal of the American Society for Information Science and Technology*, 60(11), 2009.

[11] G. Dutton. Improving locational specificity of map data - a multi-resolution, metadata-driven approach and notation. *International Journal of Geographical Information Science*, 10(3), 1996.

[12] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[13] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 2000.

[14] T. Joachims. Advances in kernel methods. chapter Making large-scale support vector machine learning practical. MIT Press, 1999.

[15] T. Joachims. Learning to classify text using support vector machines. Kluwer, 2002. Dissertation.

[16] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[17] H. Li. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers, 2011.

[18] A. Lima and M. Musolesi. Spatial dissemination metrics for location-based social networks. In *Proceedings of the 4th ACM International Workshop on Location-Based Social Networks (LBSN 2012). Colocated with ACM UbiComp 2012*, 2012.

[19] X. Liu, J. Bollen, M. L. Nelson, and H. Van de Sompel. Co-authorship networks in the digital library research community. *Information Processing and Management*, 41(6), 2005.

[20] A. Mohan, Z. Chen, and K. Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *Journal of Machine Learning Research - Proceedings Track*, 14, 2011.

[21] J. J. Oliver and D. J. Hand. On pruning and averaging decision trees. In *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995.

[22] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th International Conference Companion on World Wide Web*, 2011.

[23] H. Sayyadi and L. Getoor. Futurerank: Ranking scientific articles by predicting their future pagerank. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, 2009.

[24] A. Sidiropoulos and Y. Manolopoulos. A citation-based system to assist prize awarding. *ACM SIGMOD Record*, 34(4), 2005.

[25] A. S. Szalay, J. Gray, G. Fekete, P. Z. Kunszt, P. Kukol, and A. Thakar. Indexing the sphere with the hierarchical triangular mesh, 2007. Techinical Report.

[26] J. Tang, M. Musolesi, C. Mascolo, and V. Latora. Temporal distance metrics for social network analysis. In *Proceedings of the 2nd ACM workshop on Online social networks*, 2009.

[27] D. Walker, H. Xie, K.-K. Yan, and S. Maslov. Ranking scientific publications using a simple model of network traffic. *Journal of Statistical Mechanics*, (6), 2007.

[28] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, 2010.

[29] W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *Proceedings of the 2004 Annual Conference on Communication Networks and Services Research*, 2004.

[30] Y. Zheng and X. Zhou, editors. *Computing with Spatial Trajectories*. Springer, 2011.