Procedia Computer Science

Volume 51, 2015, Pages 2426–2434

ICCS 2015 International Conference On Computational Science

Finding Top UI/UX Design Talent on Adobe Behance

Susanne Halstead[1]*, H. Daniel Serrano[1]†, and Scott Proctor[1]‡

[1] Palo Alto, CA, U.S.A.
susannehalstead@gmail.com
[2] Adobe Systems, San Jose, CA, U.S.A.
hserrano@adobe.com
[3] Mountain View, CA, U.S.A.
scottp@gmail.com

Abstract

The Behance social network allows professionals of diverse artistic disciplines to exhibit their work and connect amongst each other. We investigate the network properties of the UX/UI designer subgraph. Considering the subgraph is motivated by the idea that professionals in the same discipline are more likely to give a realistic assessment of a colleague's work. We therefore developed a metric to assess the influence and importance of a specific member of the community based on structural properties of the subgraph and additional measures of prestige. For that purpose, we identified appreciations as a useful measure to include in a weighted PageRank algorithm, as it adds a notion of perceived quality of the work in the artist's portfolio to the ranking, which is not contained in the structural information of the graph. With this weighted PageRank, we identified locations that have a high density of influential UX/UI designers.

Keywords: Adobe Behance, PageRank, Weighted PageRank, MapReduce

1 Introduction

Professionals of all disciplines form social networks of relationships from attending the same schools, being coworkers, participating in the same local job market, attending the same conferences, and active participation in professional communities. For some professions, online communities have emerged. We are examining the community of User Experience (UX)/ User Interface (UI) designers present in Adobe's Behance social network. Behance allows creative professionals of various artistic background to share their work, receive feedback and network with each other. Importance and prestige of artists in the community is currently calculated on the whole network. We take a different approach, segmenting the network into areas of practice

*work developed during CS299 class at Stanford University
†work developed during CS299 class at Stanford University
‡work developed during CS299 class at Stanford University

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.426

and using graph techniques to calculate importance. For that, we are adapting algorithms for finding influencers in social networks to reflect the existence of several measures of influence: number of followers, number of appreciations, number of project views.

1.1 Problem Description

In the 'People' perspective, Behance currently identifies artists' creative fields, and ranks users on either views, or appreciations, or most followers. From our assessment, these rankings are calculated over the whole network, not over the network induced by applying a filter on creative field. We want to assess if segmenting the social graph by artistic field and applying a topology aware algorithm weighted with the artists' prestige to calculate influence of a node as well as considerations of locality will add insights. We want to address the following topics for the community of UX/UI designers:

1. Find influential UX/UI designers based on their position in the network and their prestige.
2. Determine significant geographic clusters of influencers.

These results can have immediate practical use: A limitation to specific artistic field for the calculation of important nodes lays the groundwork for more focused marketing campaigns or hiring. The determination of significant geographic clusters of professionals supports decisions of where to conduct user group meetings, conferences or where to physically place design shops to be able to hire top talent.

2 Related Work

2.1 On Ranking and Influence

The concepts applicable to our analysis have their origin in web mining algorithms. In an effort to discover the content of the web, there are several web mining approaches that play together, namely web content mining (information retrieval techniques), web structure mining and web usage mining [10]. Web content mining is used to group web pages by subject matter, while web structure mining approaches have been developed to determine a ranking by importance of webpages. There are two important algorithms in web structure mining: PageRank [7] and HITs [3]. While originally developed for web search engine applications, the algorithms are applicable to ranking nodes in other types of networks. C. Bento [2] illustrates how these algorithms have relevance to the task of finding highly connected nodes in social networks, showing that the concepts used for web structure mining reasonably translate to finding important nodes in social graphs. Rankings derived from graph structure mining directly translate to the notion of influence. Y. Singer [9] addresses the topic of how to best select a subset of influencers to maximize (commercial) message propagation given resource constraints of a campaign, such as paying influencers to post commercial messages, writing blog posts, or giving out free trials of new products.

2.1.1 The PageRank Algorithm

PageRank [7] is an algorithm developed for web structure mining in support of ranking search result by relevancy. In a first step a set of pages that match a given search query by keywords is retrieved. Then PageRank for this set is calculated to put the results in order of priority.

PageRank assumes that if a page has important links to it, its linked pages are also important; it therefore takes back links into account. A page's ranking is high if the sum of the ranks of its back links are high. This concept is expressed in the simplified PageRank formula:

$$PR(u) = c \sum_{v \epsilon B(u)} \frac{PR(v)}{N_v}$$

With u representing the webpage; $B(u)$ is the set of webpages that link to u. $PR(u)$ and $PR(v)$ are the PageRank scores of u and v; N_v is the number of outlinks of v; c is a normalization factor. The PageRank of a page is evenly distributed between its outgoing links. PageRank is calculated iteratively, until convergence is reached.

A modification of the basic formula solves the 'rank sink' problem (accumulation of rank in loops of pages with no outlink). For that, PageRank is extended with a 'teleport' function, expressed as a dampening factor in the page rank formula. The dampening factor d is the probability of a random surfer following a link on the page, while $1 - d$ is the probability to teleport to any page. In web structure mining, the dampening factor is often set to 0.85, however, it is frequently adjusted for other contexts to reflect observed behaviors in the domain.

$$PR(u) = (1 - d) + d \sum_{v \epsilon B(u)} \frac{PR(v)}{N_v}$$

2.1.2 Weighted Ranking Approaches

W. Xing and A. Ghorbani [4, 10] point out that treating all outlinks with equal weight might be a shortcoming of PageRank in many contexts. Weighted PageRank implementations modify the probability of following an outlink, and with that the PageRank of the linked pages, by weighing outedges. A general formulation of the weighted PageRank formula used by W. Xing and A. Ghorbani is:

$$PR(u) = (1 - d) + d \sum_{v \epsilon B(u)} PR(v) W_{(u,v)}$$

The definition of W will vary by context. For instance, in their paper [10], W. Xing and A. Ghorbani extend PageRank to consider the popularity of a webpage, as measured by the number of inlinks and outlinks. They achieve to produce more relevant results using this approach. Once we move from the realm of web structure mining to other types of networks, the definition of what constitutes useful weights may change. Y. Ding explains that in the context of citation networks there is a difference between popularity and prestige. In this context, popularity is defined as the number of times a researcher is cited; prestige is defined as the number of times the author is cited in highly cited papers and prestigious journals. Their finding is that prestige is most important when assigning weights in order to produce relevant results, and define a measure based on the definition of prestige as the weights specifically for citation networks.

2.2 Matrix Formulation of PageRank

The simplified PageRank algorithm can be formulated as the solution of

$$\mathbf{r} = M\mathbf{r}$$

where M is the transition matrix between nodes on the graph. Such a matrix is an nxn matrix with a row and a column for each node of the graph. M has a non-zero entry $M_{u,v}$ if v links to u. The entry corresponds to the distribution of transition probability among the outlinks of v, such that each column sums up to 1, making the matrix a stochastic matrix. In regular PageRank, all entries in a column get equal transition probability, thus the entry has value $\frac{1}{N_v}$, with N_v representing the number of outlinks of node v [6]. Weighted PageRank is modeled by altering the probability distribution across columns according to the criteria of importance chosen.

This matrix formulation describes a Markov process, where the principal eigenvector of the matrix M gives the PageRank distribution vector $\mathbf{r}$ [8]. The power method yields this vector if corresponding graph to M is strongly connected (M is irreducible) and there are no dead ends [6, 8].

Real world graphs contain dead ends thus produce a transition matrix with columns that contain all zeros, yielding a substochastic instead of a stochastic matrix. To remedy this, a possible approach is to replace all $\mathbf{0}$ columns with $\frac{1}{n}\mathbf{e}$. However, to enforce irreducibility, modify the matrix further, to obtain

$$M' = dM + (1-d)\frac{1}{n}\mathbf{e}\mathbf{e}^{\mathbf{T}} \text{ with } 0 \leq d \leq 1$$

The resulting matrix M' is both stochastic and irreducible; M' is also a primitive matrix, which means the power method is guaranteed to yield the stationary PageRank vector [6].

An iteration of the power method of PageRank calculation can also be formulated as:

$$\mathbf{r'} = M'\mathbf{r} = dM\mathbf{r} + (1-d)\frac{1}{n}\mathbf{e}.$$

This equation holds, because $||r||_1 = 1$. This last formulation of an iteration allows for efficient implementation of the calulation, as the operation can be performed using the sparse matrix M instead of the dense matrix M' [5].

3 Data

The data for our study was obtained via web scraping of the Behance website. We started by collecting profiles of UX/UI designers from the search page. Using these search results as seeds, we then proceeded to scrape the profile information and statistics. The profile information contains the profile id, the list of artistic interests, location of the artist (city, state, country), count of followers, count of following, project views , profile views, project appreciations, project comments. The relationships of followers and following form directed links between nodes. In order to extract the subgraph of UX/UI designer, we followed all outgoing links per node ('following'). This collects all edges, including edges leading outside the subgraph. The edges leading outside the subgraph were discarded. This process yields the subgraph of UI/UX designers and their connections among each other. In a next step, we geocoded the locations of the profiles with latitudes and longitudes at city center level using the Bing Maps API. Table 1 summarizes some properties of the graph induced by follower relationships within the UX/UI subgraph. The degree distribution of the UX/UI designer graph follows a power law distribution, as expected of a social network.

Measure	
Number of Nodes	16,885
Number of Edges within UX/UI Network	240,653
Full Diameter of Network	11
Clustering Coefficient	0.168684

Table 1: Properties of the UX/UI Designer Subgraph

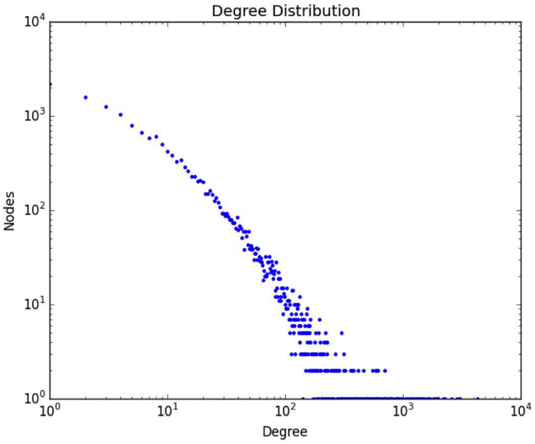


Figure 1: Degree Distribution of the UX/UI Designer Subgraph

4 Finding Important Nodes on Behance UX/UI Designer Subgraph

In the following, we develop an approach to implementing a weighted page rank for the Behance network.

4.1 Ranking with PageRank

As a first step, we used the Snapy [1] PageRank function, to append the page rank to each profile in the UX/UI designer subgraph, discarding all nodes that are not connected by at least one link to the graph. In a next step, we investigated whether the PageRank of a profile is correlated to the measures of followers, following, project views, project appreciations. Table 2 summarizes the correlation factor between the various measures.

	PageRank	Followers	Following	Project Views	Profile Views	Project Appreciations	Project Comments
PageRank	1.0000	0.8069	0.1279	0.8017	0.7048	0.7108	0.7280
Followers		1.0000	0.2409	0.7300	0.7426	0.6387	0.6450
Following			1.0000	0.1314	0.3365	0.1392	0.1850
ProjectViews				1.0000	0.7462	0.9168	0.8620
ProfileViews					1.0000	0.7476	0.7565
ProjectAppreciations						1.0000	0.8748
ProjectComments							1.0000

Table 2: Correlation Coefficient between Profile Properties

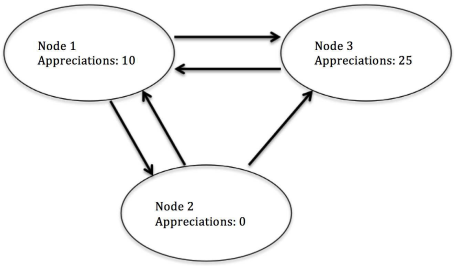
4.2 Ranking with Weighted PageRank

The 'Follower' relationship is modeled as the links in the social graph, thus the high correlation between the numbers of followers and PageRank score is expected. After the number of followers, the number of appreciations is an important measure of the prestige of an artist on Behance. The number of appreciations is not as strongly correlated to the PageRank score. We therefore extend the basic PageRank formula to contain the relative prestige among the linked nodes as weights for the outlinks at each node. Our modified PageRank formula is:

$$PR(u) = (1 - d) + d \sum_{v \epsilon B(u)} \frac{PR(v)A(u)}{TA(v)}$$

With the notations: d dampening factor, u profile page of a Behance user, $B(u)$ is the set of users that follow user u - link to user u on the graph, $PR(u), PR(v)$ are the rank scores of pages u and v, $A(u)$ is the number of appreciations for page u, $TA(v)$ is the total number of appreciations of pages linked from v.
The goal of our weighted PageRank algorithm is redistribution of the transition probability, however, we do not wish to remove links currently present in the follower relationship. We therefore, include one further modification: if a node has 0 appreciations, we replace this number by 1. This link will be assigned a very low transition probability, but will not be removed from the network, as a zero entry would imply. This also avoids division by zero

As an example, consider the toygraph G:



The sparse transition matrix M of G corresponding to standard PageRank is:

$$\begin{pmatrix} 0 & 1/2 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 \end{pmatrix}$$

The sparse transition matrix M of G corresponding to weighted PageRank is:

$$\begin{pmatrix} 0 & 10/35 & 1 \\ 1/26 & 0 & 0 \\ 25/26 & 25/35 & 0 \end{pmatrix}$$

5 Distributed Implementation in MapReduce

First prototypes on smaller subgraphs were developed using Scipy linear algebra functions. This approach does not scale once graphs and the corresponding transition matrix become too large to be held in memory. We therefore implemented a MapReduce version of the algorithm, adapting recommendations from [8].

As a first step, we need to efficiently encode the transition matrix. According to their definition, M is a sparse matrix, while M' is not sparse. Therefore, for practical computations, we use the sum formulation for one iteration:

$$\mathbf{r}' = dM\mathbf{r} + (1-d)\tfrac{1}{n}\mathbf{e}$$

Since we are using weighted PageRank, our data structure needs to record the assigned weight at each non-zero entry of the matrix. Table 3 conceptually illustrates the encoding of the sparse matrix M. Each line contains the information necessary to reconstruct a row of the matrix.

The subgraph extracted from the Behance network is small enough for $\mathbf{r}$ and subsequent $\mathbf{r}'$ to fit into memory, we therefore used a simplified approach for the implementation in MapReduce. At each mapper, a row in the matrix is read in from file and the vector $\mathbf{v}$ is multiplied, in a next step, multiply that result with d and add (1-d)/n. The output of the mapper is a component of $\mathbf{v}'$, with the position as the key, and the calculated update as the value. The reducer assembles the vector in order to redistribute to the next iteration.

Source	(Destination, Transition Probability)
Node1	(Node5, 1/3), (Node7, 2/3)
Node2	(Node1, 0.45), Node45, 0.2), (Node3, 0.35)

Table 3: Illustration of Data Structure for Transition Matrix

For larger graphs, a more robust, more scalable implementation would be required, as documented in [8].

By definition, the power method would run until a convergence condition is met. For practical purposes, running PageRank a set number of iterations anywhere between 50 -100 iterations produces sufficiently good results. [6, 8]. Our implementation therefore runs for a configurable, fixed number of iterations.

6 Where do the Influencers Live

As a concluding consideration, we investigated what locations in the US have the most influential UX/UI designers. PageRank allows us to have a clearer idea. When only considering the density of users (Figure 3, Tile 1), many cities in the US come out to have a significant number of designers. However, when considering PageRank density (Figure 3, Tile 2) as the measure, New York and San Francisco stand out as the clear epicenters of influence in UX/UI design in the US. While these results are not surprising, they are an indication that applying PageRank to measure the importance of a geographic area to an artistic community is a useful method.

Figure 3, Tile 3 illustrates the appreciation density of UX/UI designers within the continental US. Tile 4 shows the density of weighted PageRank. Weighted PageRank conservers much of the information contained in the graph structure. Cities that have high PageRank density also appear when plotting weighted PageRank, while adjustments of rank distribution according to the distribution of appreciation are visible.

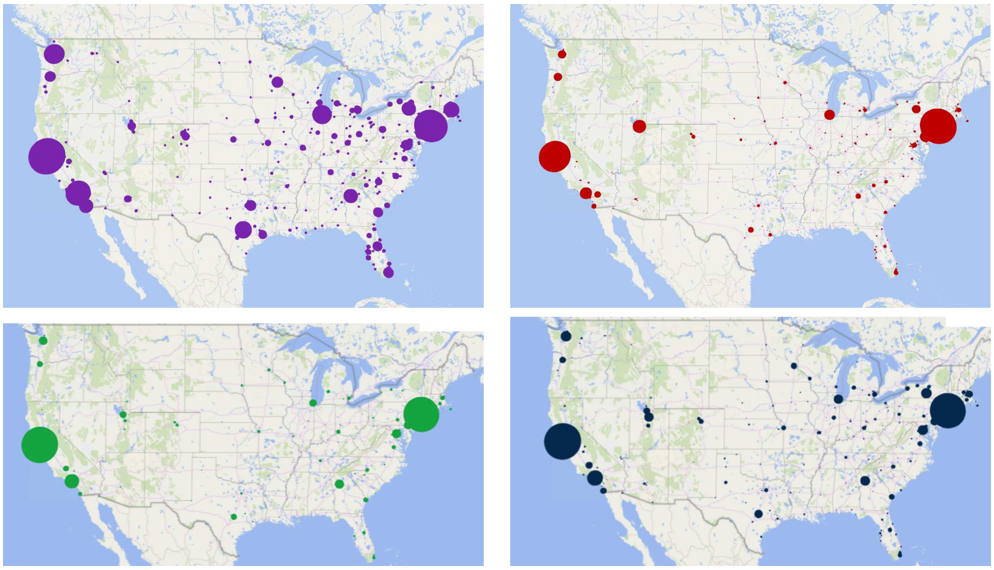


Figure 2: Geographic Distribution of Behance Users: 1) By User Density 2) By Page Rank Density 3) By Appreciation Density 4) By Weighted PageRank

7 Discussion and Future Work

Our work was able to show that PageRank is a useful measure of influence in the Behance social network, and that specifically extending PageRank to contain scores of appreciation allow the identification of the most important UX/UI designers. Future work would include a more robust evaluation of the relevance of the adjusted PageRank measure as compared to regular PageRank scores.

When calculating the PageRank scores, we used the dampening factor of 0.85, which is the reported dampening factor for the Google PageRank implementation. An adaptation to the dampening factor based on actual surfing behavior of visitors to the Behance social network is thinkable.

References

[1] Snap.py - snap for python. `http://snap.stanford.edu/snappy/index.html`. Accessed: 2014-12-30.

[2] Carolina Bento. Finding influencers in social networks.

[3] Soumen Chakrabarti, Byron E Dom, S Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and Jon Kleinberg. Mining the web's link structure. *Computer*, 32(8):60–67, 1999.

[4] Ying Ding. Applying weighted pagerank to author citation networks. *Journal of the American Society for Information Science and Technology*, 62(2):236–245, 2011.

[5] Taher Haveliwala. Efficient computation of pagerank. 1999.

[6] Amy N Langville and Carl D Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[7] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[8] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[9] Yaron Singer. How to win friends and influence people, truthfully: influence maximization mechanisms for social networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 733–742. ACM, 2012.

[10] Wenpu Xing and Ali Ghorbani. Weighted pagerank algorithm. In *Second Annual Conference on Communication Networks and Services Research CNSR04*, pages 305–314. IEEE, 2004.