D. Cirne, Int. J. of Design & Nature and Ecodynamics. Vol. 11, No. 3 (2016) 198–207

CONVERGENCE RANK AND ITS APPLICATIONS

D. CIRNE
mParticle Inc., U.S.A.

ABSTRACT

In this paper, we explore an algorithm to determine the relevance of each item in a finite set of items in reference to each other to address an item you have to first go through a convergence or proxy item. If we imagine a media streaming company (convergence item) and all its available genres for playback (items in a finite set), how relevant is each music genre at different moments in time? Or with a sports broadcasting company and the covered sports, how does the relevance of each sport changes throughout the year as sports seasons begin and end?

As the algorithm gets developed in the paper, we introduce an artificial node to the relationship graph, which brings a disproportional weight in importance. Later, we show how to remove the artificial node from the final rank vector to obtain a ranking of items in the set without any distortions.

In addition to the ranking of a single point in time, the algorithm expands to analyze a sequence of consecutive convergence rankings, infers the behavior of trends and allows for the forecasting of near-future or cyclical ranks.

The applications of this algorithm are immediate and plentiful in possibilities. In its essence, this algorithm can help to understand the usage behavior of services by its users based on non-invasive simple data collection. From its results, it is possible to better plan the allocation of resources.

Keywords: algorithm, big data, convergence, eigenvector, graph, math, rank, reduce, stochastic.

1 INTRODUCTION

We will work with a fictional music streaming company called DVB (Dystopian Voyage Broadcaster) to develop the idea behind this algorithm and illustrate the examples. When attempting to determine the relevance of websites, *PageRank* is a particularly successful algorithm, and powers the search engine giant Google. The fundamental idea in *PageRank* is that when a website links to and from other websites it builds a degree of relevance. In addition, not all links are equal in weight; if a high traffic site links to a low traffic site, it will certainly boost is relevance, but the other way around is also true (to a much smaller magnitude).

The relevance or rank of a website will always be a non-negative real number $r \subset \mathbb{R}$, where $0 \leq r \leq 1$. The closer the value is to 1, the greater the relevance of the website.

Since not all websites are linked to each other, it leads to the formation of blocks of nodes disconnected from other blocks of nodes as shown in Fig. 1.

In order to address the disconnected blocks, weighted or modification matrixes are added in the picture to introduce an artificial bond among these blocks and permit the rank to be computed.

The case we are studying and developing on this paper introduces a convergence point or node to which all items of a set are linked to, therefore we will force a scenario with no dangling nodes. However, since the convergence node is not originally part of the set of items, it will need to be removed from the final calculation of the rank.

Going back to our imaginary company, DVB, assume a user u_1 may choose to listen to Blues and Jazz while user u_2 may choose to listen to Rock. This would lead to a graph as the one in Fig. 2.

WIT CONFERENCES

This paper is part of the Proceedings of the International Conference on Big Data (Big Data 2016)
www.witconferences.com

ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), http://www.witpress.com/journals
DOI: 10.2495/DNE-V11-N3-198-207

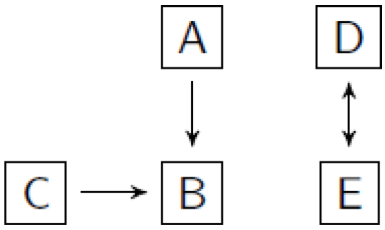

Figure 1: Graph with disconnected blocks of nodes.

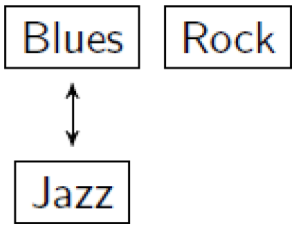

Figure 2: Listening choices from users u_1 and u_2.

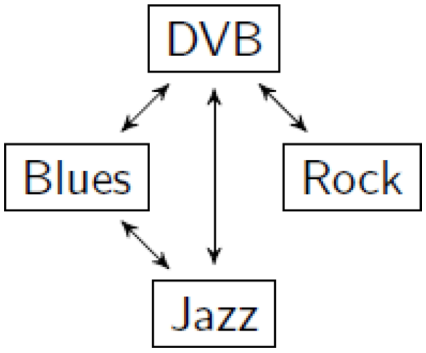

Figure 3: Graph with no dangling nodes.

The graph in Fig. 2 shows one block of connected nodes (Blues and Jazz), and one dangling node (Rock). However, we know that users u_1 and u_2 listened to those music streams via DVB. Thus, DVB is a natural choice to become the artificial proxy node not initially present in the set, and function as the "glue" to bind all other nodes together.

The graph in Fig. 3 contains no dangling nodes. DVB becomes a conver- gence node where all the to-and-from relationships with any of the other items in the set is proxied through it. Needless to say DVB will have a disproportional weight in the overall computation of the rank. We will see how to deal with it and remove it from the final rank later in this paper.

2 DEVELOPING THE ALGORITHM

Given a finite set of items P containing $(n-1) \subset \mathbb{N}$ elements, where $(n-1) > 0$. Each item in the set can be indexed by an integer i, where $1 \leq i \leq n$, the first item would be P_1, the second P_2, and so on until P_{n-1}. Each item P_i represents a subject of interest in our analysis. In our example each item would represent a music genre.

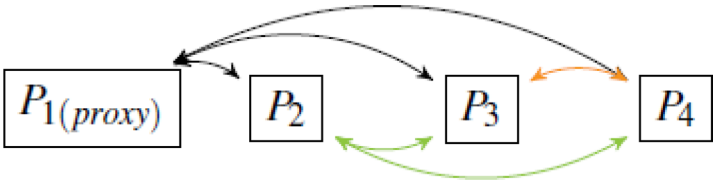

Figure 4: Bidirectional edges among visited nodes of a set.

Still working with the same set P, we will add an item with special meaning. In order to access any other item in the set one has to first access this special item, in essence a proxy element. Now our set has n elements. For example: accessing items P_3 and P_4 requires the operation to be accessing items P_n, P_3, and P_4.

The edges between any two nodes are bidirectional. Any two items in the set connected by a node contributes to each other's rank.

For the cases where the number of items m accessed in the set is greater than 1, we create not only a bidirectional edge between Pn and each of the m items, but also we will create a bidirectional edge between each of the remaining items.

To illustrate the operation described above, imagine a set with three items plus a proxy item (4 items in total) Item $P_{1(proxy)}$ has a bidirectional edge to item P_2, P_3 and P_4. The process is repeated until item P_{n-1} has a bidirectional edge to P_n. This process can be better understood if visualized as in Fig. 4.

The next step is to build a column stochastic matrix $\mathscr{S}$ representing the nodes and edges of the graph. The resulting matrix will be $n \times n$, where n equals to the number of items in the set P. Each entry $\mathscr{S}_{ij}$ is given by the number of edges from node P_i to node P_j divided by the total number of edges in node P_i. If we represent edges with the variable e we have equation (1) to compute each of the entries in the matrix.

$$\mathscr{S}_{ij} = \frac{\sum (e_i \to e_j)}{\sum e_i} \tag{1}$$

Building the column stochastic matrix for the graph displayed in Fig. 4, we end up with a simple $\mathscr{S}_{3\times3}$ matrix where the sum of each column totals exactly 1. The result from this particular example is not very exiting since each node has a total of three edges and each edge only links once to another node, thus the entries are either 0 or 1/3. However, we will compute a more interesting case later in this paper.

$$\mathscr{S} = \begin{array}{c} \\ P_{1(proxy)} \\ P_2 \\ P_3 \\ P_4 \end{array} \begin{array}{c} \begin{array}{cccc} P_{1(proxy)} & P_2 & P_3 & P_4 \end{array} \\ \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix} \end{array}$$

3 COMPUTING THE CONVERGENCE RANK

Using our fictional company DVB, let's build and compute a case that is more complex but still simple enough to understand all the details. Table 1 contains the set G of music genres. Table 2 contains the set $\mathscr{K}$ of which music genres were streamed to which anonymized users and when.

Even though this step is not strictly necessary, cross-referencing Tables 1 and 2 gives us the more comprehensive Table 3. From Table 3, we can build the music listening graph and respective column stochastic matrix.

Now, we have a table resembling Fig. 4. Repeating the steps from eqn (1), each processed row from Table 3 will become a stochastic column in the matrix $\mathscr{L}$ as in eqn (2).

In order to avoid confusion with the edge count between nodes, we colored the edges representing multiple connections between any two nodes and assigned an edge counter to it drawn in the same color as its respective edge. For example, the edge between *DVB* and *Pop* has been colored green together with its respective edge count of 2.

Table 1: G = Music genres.

Genre Id	**Genre Name**
1	Rock
2	Classical
3	Metal
4	Blues
5	Indie
6	Pop
7	Jazz
8	80's

Table 2: $\mathscr{K}$ = Played music genres.

User Id	**Genre Id**	**Date**
123	4	2015-06-01
123	3	2015-06-01
123	1	2015-06-01
456	4	2015-06-01
456	1	2015-06-01
789	1	2015-06-01
789	5	2015-06-01
789	2	2015-06-01
789	6	2015-06-01
321	7	2015-06-01
654	2	2015-06-01
654	4	2015-06-01
987	1	2015-06-01
987	8	2015-06-01
987	5	2015-06-01
987	2	2015-06-01
111	3	2015-06-01

Table 3: Condensed map of music genres played to anonymized users.

User Id	Entity	Genres			
123	DVB	Classical	Pop	80s	
456	DVB	Classical	80's		
789	DVB	80s	Jazz	Blues	Indie
321	DVB	Metal			
654	DVB	Blues	Classical		
987	DVB	80s	Rocks	Jazz	Blues
111	DVB	Pop			

$$\mathscr{L} = \begin{array}{c} \\ DVB \\ Classical \\ Blues \\ Pop \\ 80's \\ Jazz \\ Indie \\ Metal \\ Rock \end{array} \begin{array}{c} \begin{array}{ccccccccc} DVB & Cla & Blu & Pop & 80's & Jaz & Ind & Met & Roc \end{array} \\ \begin{pmatrix} 0 & 3/7 & 3/10 & 2/4 & 4/13 & 2/8 & 1/4 & 1 & 1/4 \\ 3/17 & 0 & 1/0 & 1/4 & 2/13 & 0 & 0 & 0 & 0 \\ 3/17 & 1/7 & 0 & 0 & 2/13 & 2/8 & 1/4 & 0 & 1/4 \\ 2/17 & 1/7 & 0 & 0 & 1/13 & 0 & 0 & 0 & 0 \\ 4/17 & 2/7 & 2/10 & 1/4 & 0 & 2/8 & 1/4 & 0 & 1/4 \\ 2/17 & 0 & 2/10 & 0 & 2/13 & 0 & 1/4 & 0 & 1/4 \\ 1/17 & 0 & 1/10 & 0 & 1/13 & 1/8 & 0 & 0 & 0 \\ 1/17 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/17 & 0 & 1/10 & 0 & 1/13 & 1/8 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{2}$$

With matrix $\mathscr{L}$ assembled, eqn (2), representing the graph in Fig. 5, we need to calculate its right eigenvectors $\mathscr{E}_R$ (or just eigenvector), more specifically, the first eigenvector. By solving eqn (3), we reference the first eigenvector of $\mathscr{L}$ as the vector $\mathscr{E}$.

$$\mathscr{L}\mathscr{E}_R = \lambda\mathscr{E}_R$$
$$\mathscr{E} = \text{First eigenvector of } (\mathscr{L}) \tag{3}$$

The eigenvector can easily be calculated numerically by using a mathematics software package such as SageMath or a linear algebra library such as LAPACK.

$$\mathscr{E} = \begin{pmatrix} 1.000000000000000 \\ 0.411764705882353 \\ 0.588235294117647 \\ 0.235294117647059 \\ 0.764705882352941 \\ 0.470588235294118 \\ 0.235294117647059 \\ 0.058823519411764 \\ 0.235294117647059 \end{pmatrix}$$

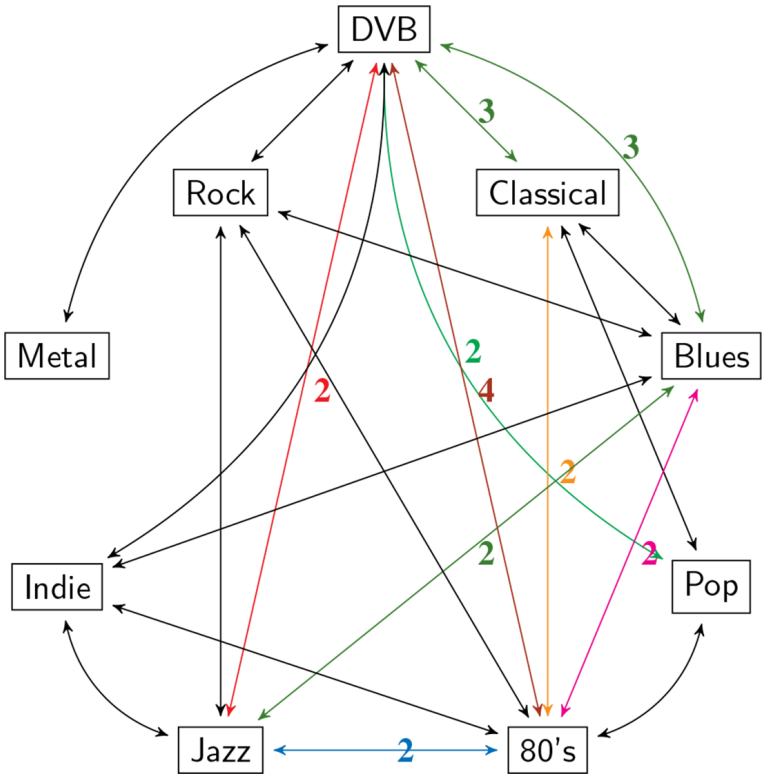

Figure 5: Streamed music graph.

The eigenvector $\mathscr{E}$ is not stochastic. Furthermore, we can clearly see the dis- proportional weight entry E_1 has. It represents our proxy item, and since it is connected to all other items, its magnitude should come with no surprise. We need now to remove entry $\mathscr{E}_1$ from vector $\mathscr{E}$ and transform it into a stochastic vector $\mathscr{P}$ containing the results of computed convergence rank.

Even though $\mathscr{E}_1$ has a disproportional weight compared to the other entries of $\mathscr{E}$, the proportions of the weights of the remaining items when measured against each other are still correct. We can just remove entry $\mathscr{E}_1$ from vector $\mathscr{E}$ and create a new vector $\mathscr{E}'$ as shown in eqn (4).

$$\mathscr{E}' = \begin{pmatrix} 0.411764705882353 \\ 0.588235294117647 \\ 0.235294117647059 \\ 0.764705882352941 \\ 0.470588235294118 \\ 0.235294117647059 \\ 0.058823519411764 \\ 0.235294117647059 \end{pmatrix} \tag{4}$$

Vector $\mathscr{E}'$ is transformed into a stochastic vector by first calculating the sum $\mathscr{S}$ of all its entries, eqn (5), then dividing each entry $\mathscr{E}'_1$ by $\mathscr{S}$ as in eqn (6). The result of this operation becomes vector

$\mathscr{R}$, which is our column stochastic vector and also the convergence rank of the items from the original set.

$$\mathscr{P} = \sum_{i=1}^{n-1} \mathscr{E}_i, \tag{5}$$

$$\mathscr{R}_i\left(\frac{\mathscr{E}_i}{\mathscr{P}}\right), \text{ where } i \in [1..(n-1)] \tag{6}$$

$$\mathscr{R} = \begin{matrix} Classical \\ Blues \\ Pop \\ 80's \\ Jazz \\ Indie \\ Metal \\ Rock \end{matrix} \begin{pmatrix} 0.137254901960784 \\ 0.196078431372549 \\ 0.078431372549019 \\ 0.254901960784314 \\ 0.156862745098039 \\ 0.078431372549019 \\ 0.019607843437254 \\ 0.078431372549019 \end{pmatrix} \tag{7}$$

We have successfully computed the convergence rank, eqn (7), of each of the music genres. In our example 80s, Blues, and Jazz are the most popular music genre with about 25%, 19%, and 15% of listeners' preferences, respectively.

4 APPLICATIONS

Once enough data have been collected over a period of time, we can compute historical convergence ranks, separated by a desired time interval of Δt and store the computed values in a set $\mathscr{C} = \{\mathscr{R}_1, \mathscr{R}_2, \ldots, \mathscr{R}_n\}$. By analyzing a set of consecutive time-distributed convergence rank values, we can learn more about the data; for instance, if we compute the convergence rank for music genre playback for every day over a period of a few weeks, we will see the historical preferences for DVB. If we narrow the focus to a particular subset ($\mathscr{C}_{genre} \ni \mathscr{C}$), let's say Rock, we can calculate how *sticky* or *viral* this particular music genre is.

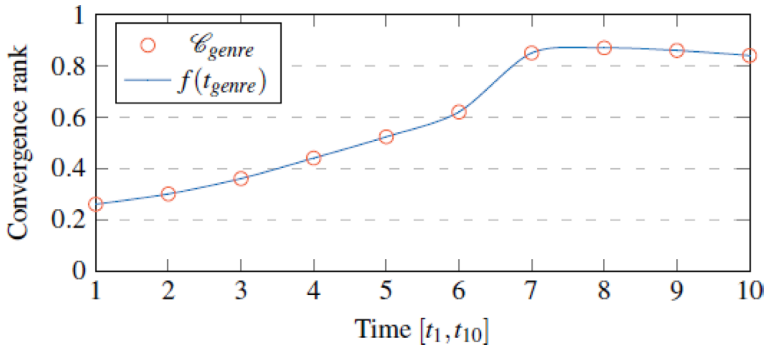

Figure 6: Convergence rank plotted over time.

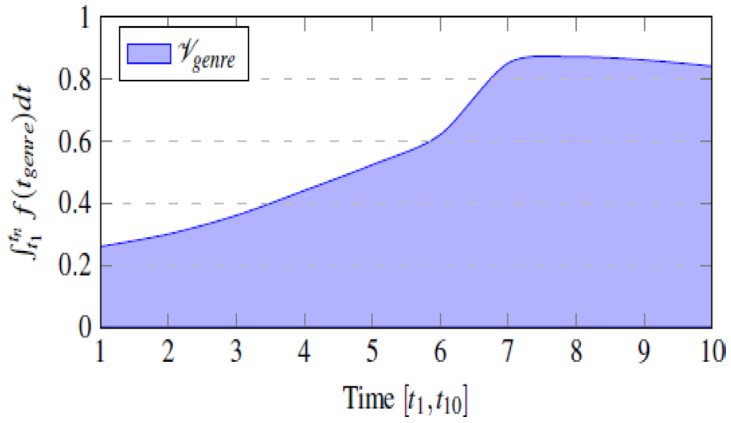

Figure 7: Viral factor $\mathscr{V}$ plotted over time.

Plotting all the values from set $\mathscr{C}_{genre}$ on a *XY* Cartesian chart (Fig. 6), we can interpolate those points using a *Piecewise Cubic Hermite Interpolating Polynomial* (PCHIP) and obtain a continuous function $f(t_{genre})$ for the time interval $[t_1, t_n]$, where n is the number of samples of computed convergence ranks.

4.1 Viral factor

Integrating $f(t_{genre})$ over the time interval $[t_1, t_n]$ we get a relative value, eqn (8), whose meaning can be assigned to or assumed to be how viral/popular the music genre is when compared to other genres during the observed period of time.

$$\mathscr{V}_{genre} = \int_{t_1}^{t_n} f(t_{genre})\, dt \tag{8}$$

After computing and sorting the viral factor $\mathscr{V}$ for several genres, we will end up with a set of values that will resemble the following distribution:

$$\mathscr{V}_{genre(x)} \geq \mathscr{V}_{genre(y)} \geq \ldots \geq \mathscr{V}_{genre(z)}$$

4.2 Popularity factor

The derivative of the continuous function $f(t_{genre})$ over the time interval $[t_1, tn]$ will indicate the growth in popularity, eqn (9), of a particular genre. Once again we get a value which can be used to compare the relative growth of this genre compared to others.

$$\mathscr{P}_{genre} = \frac{\partial(f(t_{genre}))}{\partial t} \tag{9}$$

Computing and sorting the growth in popularity factor $\mathscr{P}$ of several genres would give as a set of values telling us, which genres are growing in popularity and which genres are declining.

$$\mathscr{P}_{genre(a)} \geq \mathscr{P}_{genre(b)} \geq \ldots \geq \mathscr{P}_{genre(c)}$$

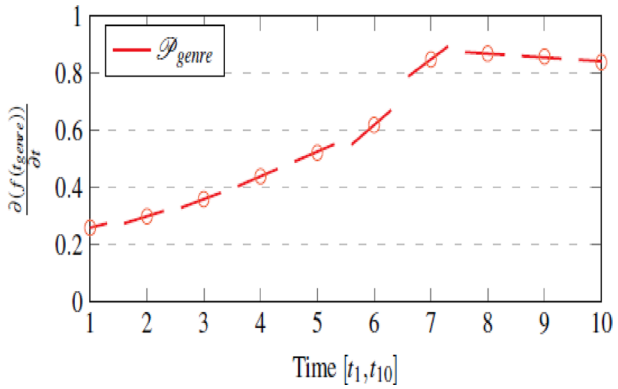

Figure 8: Popularity $\mathscr{P}$ plotted over time.

Table 4: Strategies for resources management.

Viral factor	**Popularity growth factor**	**Action**
Small	Large and Positive	*Invest* resources
Large	Positive and Near Zero	*Hold* resources
Large	Negative	*Withdraw* resources

By knowing the convergence rank of each music genre at any given time t, their respective viral $\mathscr{V}$ and popularity $\mathscr{P}$ factors, one becomes empowered to make educated decisions regarding resource allocations and strategic planning over time.

Table 4 shows a few suggestions of actions to be taken to optimize the strategy for the allocation of resources. Potential applications include: Maximization of exposure of an advertisement; Coverage given to a sport at pre, mid, and post season (when compared to other sports); Sponsorship of artists and athletes; and much more.

5 APPLICATION TO A REAL CASE

The data points shown in Fig. 9 were collected from the first 100,000 users of ESPN's mobile app selecting their favorite sports between 2009–06–01 and 2009–11–27 (initial 180 days). Each graphic contains the historical convergence rank for the four most popular sports in U.S.A.

Utilizing convergence rank, we were able not only to compare the rank of sports not having the same fan base in common, but also we can see that MLB's viral factor was larger than the NFL's during that period ($\mathscr{V}_{MLB} > \mathscr{V}_{NFL}$). However, the popularity factor ($\mathscr{P}_{MLB} < 0$ and $\mathscr{P}_{NFL} > 0$) suggests that it would probably be better to shift resources from MLB to NFL around 30 days (early July) from the beginning of the data set (see Table 4 for suggested strategies).

Another example is applying convergence rank to NBA and NHL, both begin growing their respective viral and popularity factors at around 90 days past 2009–06–01. Even though strategically it makes sense to invest in both sports during at that point, allocating more resources towards the NBA may result in better returns given that the magnitude of its viral factor is larger, and the fact that the popularity factor for the NHL becomes negative around the 130-day mark.

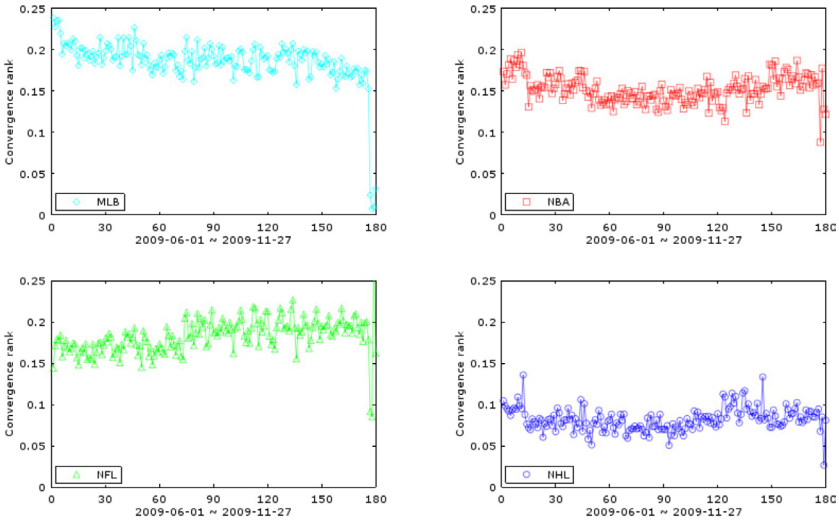

Figure 9: Historical convergence rank for MLB, NBA, NFL, and NHL.

6 CONCLUSION

The applications of the convergence rank algorithm are immediate and plentiful in possibilities. In its essence, it can help someone utilizing it to understand more about the users to whom services and/products are being provided, allow for better and more educated decisions making, and aid how to strategically manage and deploy limited resources.

REFERENCES

[1] Bryan, K. & Leise, T., The 25,000,000,000 eigenvector: the linear algebra behind google. *SIAM Review*, 48(3), pp. 569–581, 2006. http://dx.doi.org/10.1137/050623280

[2] Faires, J.D. & Burden, R., *Numerical Methods*, Thomson Brooks/Cole, 3rd edn., 2003.

[3] Strang, G., *Linear Algebra and Its Applications*, Thomson Learning, 3rd edn., 1998.

[4] Rosen, K.H., *Discrete Mathematics and Its Applications*, McGraw-Hill, 7th edn., 2012.

[5] Joyner, D., Nguyen, M.V. & Cohen, N., *Algorithmic Graph Theory*, Free Software Foundation, version 0.7-r1984 edn., 2012.

[6] Wrede, R. & Spiegel, M.R., *Advanced Calculus*, McGraw-Hill, 2nd edn., 2002.