# Finding and Visualizing Graph Clusters Using PageRank Optimization

Fan Chung    Alexander Tsiatas

Department of Computer Science and Engineering
University of California, San Diego
{fan,atsiatas}@cs.ucsd.edu

**Abstract.** We give algorithms for finding graph clusters and drawing graphs, highlighting local community structure within the context of a larger network. For a given graph $G$, we use the personalized PageRank vectors to determine a set of clusters, by optimizing the jumping parameter $\alpha$ subject to several cluster variance measures in order to capture the graph structure according to PageRank. We then give a graph visualization algorithm for the clusters using PageRank-based coordinates. Several drawings of real-world data are given, illustrating the partition and local community structure.

## 1  Introduction

Finding smaller local communities within a larger graph is a well-studied problem with many applications. For example, advertisers can more effectively serve niche audiences if they can identify their target communities within the larger social web, and viruses on technological or population networks can be effectively quarantined by distributing antidote to local clusters around their origins [7].

There are numerous well-known algorithms for finding clusters within a graph, including $k$-means [20, 22], spectral clustering [26, 31], Markov cluster algorithms [11], and numerous others [17, 23–25, 27]. Many of these algorithms require embedding a graph into low-dimensional Euclidean space using pairwise distances, but graph distance-based metrics fail to capture graph structure in real-world networks with small-world phenomena since all pairs of nodes are connected within short distances. PageRank provides essential structural relationships between nodes and is particularly well suited for clustering analysis. Furthermore, PageRank vectors can be computed more efficiently than performing a dimension reduction for a large graph.

In this paper, we give clustering algorithms **PageRank-Clustering** that use PageRank vectors to draw attention to local graph structure within a larger network. PageRank was first introduced by Brin and Page [5] for Web search algorithms. Although the original definition is for the Web graph, PageRank is well defined for any graph. Here, we will use a modified version of PageRank, known as personalized PageRank [18], using a prescribed set of nodes as a seed vector.

PageRank can capture well the quantitative correlations between pairs or subsets of nodes, especially on small-world graphs where the usual graph distances are all quite small. We use PageRank vectors to define a notion of PageRank distance which provides a natural metric space appropriate for graphs.

A key diffusion parameter in deriving PageRank vectors is the jumping constant $\alpha$. In our clustering algorithms, we will use $\alpha$ to control the scale of the clustering. In particular, we introduce two variance measures which can be used to automatically find the optimized values for $\alpha$. We then use PageRank vectors determined by $\alpha$ to guide the selection of a set of centers of mass and use them to find the clusters via PageRank distances. We further apply our clustering algorithm to derive a visualization algorithm **PageRank-Display** to effectively display local structure when drawing large networks.

The paper is organized as follows: The basic definitions for PageRank are given in Section 2. In Section 3, we describe two cluster variance measures using PageRank vectors, and we give clustering algorithms in Section 4, with analysis in Section 5. A graph drawing algorithm is given in the last section and several examples are included.

## 2    Preliminaries

We consider general undirected graphs $G = (V, E)$ with vertex set $V$ and edge set $E$. For a vertex $v$, let $d_v$ denote the *degree* of $v$ which is the number of *neighbors* of $v$. For a set of nodes $T \subseteq V$, the *volume* of $T$ is defined to be $\mathrm{vol}(T) = \sum_{v \in T} d_v$. Let $D$ denote the *diagonal degree matrix* and $A$ the *adjacency matrix* of $G$.

We consider a typical random walk on $G$ with the *transition probability matrix* defined by $W = D^{-1}A$. Let $\pi$ denote the stationary distribution of the random walk, if it exists. Personalized PageRank vectors are based on random walks with two governing parameters: a seed vector $\boldsymbol{s}$, representing a probability distribution over $V$, and a jumping constant $\alpha$, controlling the rate of diffusion. The personalized PageRank $\mathrm{pr}(\alpha, \boldsymbol{s})$ is defined to be the solution to the following recurrence relation: $\mathrm{pr}(\alpha, \boldsymbol{s}) = \alpha \boldsymbol{s} + (1 - \alpha)\mathrm{pr}(\alpha, \boldsymbol{s})W$.

Here, $\boldsymbol{s}$ (and all other vectors) will be treated as row vectors. The original definition of PageRank defined in [5] is the special case where the seed vector is the the uniform distribution. If $\boldsymbol{s}$ is simply the distribution which is 1 for a single node $v$ and 0 elsewhere, we write $\mathrm{pr}(\alpha, v)$.

In general, it can be computationally expensive to compute PageRank exactly; it requires using the entire graph structure which can be prohibitive on large networks. Instead, we use an approximate PageRank algorithm as given in [3, 8]. This approximation algorithm is much more tractable on large networks, because it can be computed using only the local graph structure around the starting seed vector $\boldsymbol{s}$. Besides $\boldsymbol{s}$ and the jumping constant $\alpha$, the algorithm requires an approximation parameter $\epsilon$.

For a set of points $S = \{s_1, \ldots, s_n\}$ in Euclidean space, the *Voronoi diagram* is a partition of the space into disjoint regions $R_1, \ldots, R_n$ such that each $R_i$ contains $s_i$ and the region of space containing the set of points that are closer to $s_i$

than any other $s_j$. Voronoi diagrams are well-studied in the field of computational geometry. Here we consider Voronoi diagrams on graphs using PageRank vectors as a notion of closeness.

For two vertices $u, v$, we define the *PageRank distance* with jumping constant $\alpha$ as: $\text{dist}_\alpha(u, v) = \|\text{pr}(\alpha, u)D^{-1/2} - \text{pr}(\alpha, v)D^{-1/2}\|$.

We can further generalize this distance to two probability distributions $p$ and $q$ defined on the vertex set $V$ of $G$. Namely, the PageRank distance, with jumping constant $\alpha$, between $p$ and $q$ is defined by $\text{dist}_\alpha(p, q) = \sum_{u,v} p(u)q(v)\text{dist}(u, v)$.

With this definition, for a subset $S$ of vertices, we can generalize the notion of a center of mass for $S$ to be a probability distribution $c$. For a given $\epsilon > 0$, we say $c$ is an *$\epsilon$-center* or *center of mass* for $S$ if $\sum_{v \in S} \text{dist}_\alpha(c, v) \leq \epsilon$.

Let $C$ denote a set of $k$ (potential) centers. The goal is for each center $c$ to be a representative center of mass for some cluster of vertices. We let $R_c$ denote the set of all vertices $x$ which are closest to $c$ in terms of PageRank, provided the jumping constant $\alpha$ is given: $R_c = \{x \in V : \text{dist}_\alpha(c, x) \leq \text{dist}_\alpha(c', x) \text{ for all } c' \in C\}$.

## 3   PageRank Variance and Cluster Variance Measures

For a vertex $v$ and a set of centers $C$, let $c_v$ denote the center that is closest to $v$, (i.e., $c_v$ is the center of mass $c \in C$ such that $v \in R_c$).

We follow the approach as in $k$-means by defining the following evaluative measure for a potential set of $k$ centers $C$, using PageRank instead of Euclidean distances.

$$\mu(C) = \sum_{v \in V} d_v \|\text{pr}(\alpha, v)D^{-1/2} - \text{pr}(\alpha, c_v)D^{-1/2}\|^2 = \sum_{v \in V} d_v \text{ dist}_\alpha(v, c_v)^2 \ .$$

Selecting a set of representative centers within a graph is a hard problem, known to be NP-complete. There are many approximate and heuristic algorithms used in practice (see [30]). Here, we will develop algorithms that use personalized PageRank vectors to select the centers. In the Web graph, links between websites can be interpreted as votes for a website's importance, and PageRank vectors are used to determine which pages are intrinsically more important in the overall graph. Personalized PageRank vectors are local information quantifying the importance of every node to the seed. Thus, the $u$th component of the personalized PageRank vector $\text{pr}(\alpha, v)$ quantifies how well-suited $u$ is to be a representative cluster center for $v$.

To evaluate a set of cluster centers in a graph $G$, we consider two measures that capture the community structure of $G$ with respect to PageRank:

$$\Phi(\alpha) = \sum_{v \in V} d_v \left|\left| \mathrm{pr}(\alpha, v) D^{-1/2} - \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2} \right|\right|^2$$

$$= \sum_{v \in V} d_v \ \mathrm{dist}_\alpha(v, \mathrm{pr}(\alpha, v))^2 \ ,$$

$$\Psi(\alpha) = \sum_{v \in V} d_v \left|\left| \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2} - \pi D^{-1/2} \right|\right|^2$$

$$= \sum_{v \in V} d_v \ \mathrm{dist}_\alpha(\mathrm{pr}(\alpha, v), \pi)^2 \ .$$

The $\alpha$-*PageRank-variance* $\Phi(\alpha)$ measures discrepancies between the personalized PageRank vectors for nodes $v$ and possible centers nearest to $v$, represented by the probability distribution $\mathrm{pr}(\alpha, v)$. The $\alpha$-*cluster-variance* $\Psi(\alpha)$ measures large discrepancies between personalized PageRank vectors for nodes $v$ and the overall stationary distribution $\pi$. If the PageRank-variance $\Phi(\alpha)$ is small, then the 'guesses' by using PageRank vectors for the centers of mass give a good upper bound for the $k$-means evaluation $\mu$ using PageRank distance, indicating the formation of clusters. If the cluster-variance $\Psi(\alpha)$ is large, then the centers of masses using the predictions from PageRank vectors are quite far from the stationary distribution, capturing a community structure. Thus, our goal is to find the appropriate $\alpha$ such that $\Phi(\alpha)$ is small but $\Psi(\alpha)$ is large.

For a specific set of centers of mass $C$, we use the following for an evaluative metric $\Psi_\alpha(C)$, suggesting the structural separation of the communities represented by centers in $C$:

$$\Psi_\alpha(C) = \sum_{c \in C} \mathrm{vol}(R_c) \left|\left| \mathrm{pr}(\alpha, c) D^{-1/2} - \pi D^{-1/2} \right|\right|^2 = \sum_{c \in C} \mathrm{vol}(R_c) \ \mathrm{dist}_\alpha(c, \pi)^2 \ .$$

We remark that this measure is essentially the analog of $k$-means in terms of PageRank distance, and it has a similar flavor as a heuristic given by Dyer and Frieze [9] for the traditional center selection problem.

## 4 The PageRank-Clustering Algorithms

These evaluative measures give us a way to evaluate a set of community centers, leading to the **PageRank-Clustering** algorithms presented here. The problem of finding a set of $k$ centers minimizing $\mu(C)$ is then reduced to the problem of minimizing $\Phi(\alpha)$ while $\Psi(\alpha)$ is large for appropriate $\alpha$. In particular, for a special class of graphs which consist of $k$ clusters of vertices where each cluster has Cheeger ratio at most $\alpha$, the center selection algorithm is guaranteed to be successful with high probability.

A natural question is to find the appropriate $\alpha$ for a given graph, if such $\alpha$ exists and if the graph is clusterable. A direct method is by computing the

variance metrics for a sample of $\alpha$ and narrowing down the range for $\alpha$ using binary search. Here, we give a systematic method for determining the existence of an appropriate $\alpha$ and finding its value is by differentiating $\Phi(\alpha)$, and finding roots $\alpha$ satisfying $\Phi'(\alpha) = 0$. It is not too difficult to compute that the derivative of $\Psi$ satisfies

$$\Phi'(\alpha) = \frac{1-\alpha}{\alpha^3}\left(\|g_v(\alpha)D^{-1/2}\|^2 - 2\langle g_v(\alpha), \mathrm{pr}(\alpha, g_v(\alpha))D^{-1}\rangle\right) \qquad (1)$$

where $g_v(\alpha) = \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)(I - W))$. Here, we give two versions of the clustering algorithm. For the sake of clarity, the first PageRank clustering algorithm uses exact PageRank vectors without approximation. The second PageRank clustering algirhtm allows for the use of approximate PageRank vectors as well as approximate PageRank-variance and cluster-variance for faster performance.

**PageRank-ClusteringA**$(G, k, \epsilon)$:

- For each vertex $v$ in $G$, compute the PageRank vector $\mathrm{pr}(\alpha, v)$.
- Find the roots of $\Phi'(\alpha)$ (1).
  (There can be more than one root if the graph $G$ has a layered clustering structure.)
  For each root $\alpha$, we repeat the following process:
    - Compute $\Phi(\alpha)$.
      If $\Phi(\alpha) \leq \epsilon$, also compute $\Psi(\alpha)$.
      Otherwise, go to the next $\alpha$.
    - If $k < \Psi(\alpha) - 2 - \epsilon$, go to the next $\alpha$.
      Else, select $c\log(n)$ sets, each consisting of $k$ potential centers randomly chosen according to the stationary distribution $\pi$. (Here, $c$ is some absolute constant $c \leq 100$ to allow sampling with high probability.)
    - For each set $S = \{v_1, \ldots, v_k\}$, let $C$ be the set of centers of mass where $c_i = \mathrm{pr}(\alpha, v_i)$.
        - Compute $\mu(C)$ and $\Psi_\alpha(C)$.
          If $|\mu(C) - \Phi(\alpha)| \leq \epsilon$ and $|\Psi_\alpha(C) - \Psi(\alpha)| \leq \epsilon$, determine the $k$ Voronoi regions according to the PageRank distances using $C$.

We can further reduce the computational complexity by using approximate PageRank vectors.

**PageRank-ClusteringB**$(G, k, \epsilon)$:

- Find the roots of $\Phi'(\alpha)$ (1) within an error bound $\epsilon/2$, by using sampling techniques from [29] involving $O(\log n)$ nodes, $\log(1/\epsilon)$ values of $\alpha$ and $\delta$-approximate PageRank vectors [3, 8] where $\delta = \epsilon/n^2$.
  There can be more than one root if the graph $G$ has a layered clustering structure.
  For each root $\alpha$, we repeat the following process:
    - Approximate $\Phi(\alpha)$.
      If $\Phi(\alpha) \leq \epsilon/2$, also compute $\Psi(\alpha)$.
      Else, go to the next $\alpha$.

– If $k < \Psi(\alpha) - 2 - \epsilon/2$, go to the next $\alpha$.
Else, select $c\log(n)$ sets, each consisting of $k$ potential centers randomly chosen according to the stationary distribution $\pi$. (Here, $c$ is some absolute constant $c \leq 100$ to allow sampling with high probability.)
– For each set $S = \{v_1, \ldots, v_k\}$, let $C$ be the set of centers of mass where $c_i$ is an approximate PageRank vector for $\text{pr}(\alpha, v_i)$.
  – Compute $\mu(C)$ and $\Psi_\alpha(C)$.
  If $|\mu(C) - \Phi(\alpha)| \leq \epsilon$ and $|\Psi_\alpha(C) - \Psi(\alpha)| \leq \epsilon$, determine the $k$ Voronoi regions according to the PageRank distances using $C$.

We remark that by using the sharp approximate PageRank algorithm in [8], the error bound $\delta$ for PageRank can be set to be quite small since the time complexity is proportional to $\log(1/\delta)$. If we choose $\delta$ to be a negative power of $n$ such as $\delta = \epsilon/n^2$, then approximate PageRank vectors lead to sharp estimates for $\Phi$ and $\Phi'$ within an error bound of $\epsilon$. Thus for graphs with $k$ clusters, the **PageRank-ClusteringB** algorithm will terminate after approximating the roots of $\Phi'$, $O(k\log n)$ approximations of $\mu$ and $\Psi_\alpha$ and $O(n)$ approximate Page-Rank computations. By using approximation algorithms using sampling, this can be done quite efficiently.

We also note that there might be no clustering output if the conditions set within the algorithms are not satisfied. Indeed, there exist graphs that inherently do not have a $k$-clustered structure within the error bound that we set for $\epsilon$. Another reason for no output is the probabilistic nature of the above sampling argument. We will provide evidence to the correctness of the above algorithm by showing that, with high probability, a graph with a $k$-clustered structure will have outputs that capture its clusters in a feasible manner which we will specify further.

For a subset of nodes $H$ in a graph $G$, the *Cheeger ratio* $h(H)$ is the ratio of the number of edges leaving $H$ and $\text{vol}(H)$. We say a graph $G$ is $(k, h, \beta, \epsilon)$-clusterable if the vertices of $G$ can be partitioned into $k$ parts so that (i) each part $S$ has Cheeger ratio at most $h$ and (ii) $S$ has volume at least $\beta \text{vol}(G)/k$ for some constant $\beta$, and (iii) any subset $S'$ of $S$, with $\text{vol}(S') \leq (1-\epsilon)\text{vol}(S)$ has Cheeger ratio at least $\sqrt{h \log n}$.

We will provide evidence for the correctness of **PageRank-ClusteringA** by proving the following theorem:

**Theorem 1.** *Suppose a graph $G$ has an $(k, h, \beta, \epsilon)$-clustering and $\alpha, \epsilon \in (0, 1)$ satisfy $\epsilon \geq hk/(2\alpha\beta)$. Then with high probability, **PageRank-ClusteringA** returns a set $C$ of $k$ centers with $\Phi(\alpha) \leq \epsilon$, $\Psi(C) > k - 2 - \epsilon$, and the $k$ clusters are near optimal according to the PageRank $k$-means measure $\mu$ with an additive error term $\epsilon$.*

## 5  Analyzing PageRank Clustering Algorithms

We wish to show that the PageRank-clustering algorithms are effective for treating graphs which are $(k, h, \beta, \epsilon)$-clusterable. We will use a slightly modified ver-

sion of a result in [3] which provides a direct connection between the Cheeger ratio and the personalized PageRank within $S$:

**Theorem A** [3] *For any set $S$ and any constants $\alpha, \delta$ in $(0, 1]$, there is a subset $S_\alpha \subseteq S$ with volume $\mathrm{vol}(S_\alpha) \geq \delta\mathrm{vol}(S)/2$ such that for any vertex $v \in S_\alpha$, the PageRank vector $\mathrm{pr}(\alpha, v)$ satisfies $\mathrm{pr}(\alpha, v)(S) \geq 1 - \frac{h(S)}{\alpha\delta}$.*

To see that our clustering algorithm can be applied to an $(h, k, \beta, \epsilon)$-clusterable graph $G$, we will need the following condition: $\epsilon \geq \frac{hk}{2\alpha\beta}$.

Theorem A implies that in a cluster $R$ of $G$, most of the vertices $u$ in $R$ have $\mathrm{pr}(\alpha, u)(S) \geq 1 - \epsilon/(2k)$. This fact is essential in the subsequent proof that $\Psi(\alpha) \geq k - 2 - \epsilon$.

*A sketched proof for Theorem 1:*
We note that $\mathrm{pr}(0, s) = \pi$ and $\mathrm{pr}(1, s) = s$ for any distribution $s$. This implies that $\Phi(0) = \Phi(1) = \Psi(0) = 0$ and $\Psi(1) = n - 1$. It is not hard to check that $\Psi$ is an increasing function since $\Psi'(\alpha) > 0$ for $\alpha \in (0, 1]$. The function of particular interest is $\Phi$. Since we wish to find $\alpha$ such that $\Phi$ is small, it suffices to check the roots of $\Phi'$.

Suppose $\alpha$ is a root of $\Phi'$. To find $k$ clusters, we can further restrict ourselves to the case of $\Psi(\alpha) \geq k - 2 - \epsilon$ by establishing the following claim:

*Claim:* If a graph $G$ can be partitioned into $k$ clusters having Cheeger ratio at most $h$ and $\epsilon \geq hk/(2\alpha\beta)$, then $\Psi(\alpha) \geq k - 2 - \epsilon$.

Before proving the claim, we note that by sampling $c \log n$ sets of $k$ vertices from $\pi$, for sufficiently large $c$, the values $\mu(C)$ and $\Psi(C)$ for one such random set of $k$ centers are close to $\Phi(\alpha)$ and $\Psi(\alpha)$, respectively, with high probability (exponentially decreasing depending on $c$ and $\beta$) by probabilistic concentration arguments. In this context, the upper bound $\epsilon$ for $\mu(C)$ implies that the set consisting of distributions $\mathrm{pr}(\alpha, c)$ for $c \in C$ serves well as the set of centers of mass. Thus, the resulting Voronoi regions using $C$ give the desired clusters. This proves the correctness of our clustering algorithm with high probability for $(k, h, \beta, \epsilon)$-clusterable graphs.

*Proof of the Claim:*

$$\Psi(\alpha) = \sum_{v \in V} d_v \left\| \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2} - \pi D^{-1/2} \right\|^2$$

$$= \sum_{c \in C} \sum_{v \in R_c} d_v \| \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2} - \pi D^{-1/2} \|^2$$

$$= \sum_{c \in C} \sum_{v \in R_c} d_v \sum_x \left( \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2}(x) - \pi D^{-1/2}(x) \right)^2$$

$$\geq \sum_{c \in C} \sum_{v \in R_c} d_v \sum_{x \in Rc} \left( \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2}(x) - \pi D^{-1/2}(x) \right)^2$$

$$= \sum_{c \in C} \sum_{v \in R_c} d_v \sum_{x \in Rc} \left( \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v)) D^{-1/2}(x) - \pi D^{-1/2}(x) \right)^2 \sum_{x \in R_c} \frac{d_x}{\mathrm{vol}(R_c)}$$

$$\geq \sum_{c \in C} \sum_{v \in R_c} \frac{d_v}{\mathrm{vol}(R_c)} \left( \sum_{x \in Rc} \left( \mathrm{pr}(\alpha, \mathrm{pr}(\alpha, v))(x) - \pi(x) \right) \right)^2$$

$$\geq \sum_{c \in C} \sum_{v \in R_c} \frac{d_v}{\mathrm{vol}(R_c)} \left( 1 - \epsilon/2 - \frac{\mathrm{vol}(R_c)}{\mathrm{vol}(G)} \right)^2$$

$$= \sum_{c \in C} \left( 1 - \frac{\epsilon}{2k} - \frac{\mathrm{vol}(R_c)}{\mathrm{vol}(G)} \right)^2$$

$$\geq \frac{1}{k} \left( \sum_{c \in C} \left( 1 - \frac{\epsilon}{2k} - \frac{\mathrm{vol}(R_c)}{\mathrm{vol}(G)} \right) \right)^2$$

$$= \frac{1}{k} \left( k - 1 - \frac{\epsilon}{2} \right)^2 \geq k - 2 - \epsilon$$

$\square$

To illustrate **PageRank-ClusteringB**, we consider a dumbbell graph $U$ as an example. This graph $U$ has two complete graphs $K_{20}$ connected by a single edge, yielding a Cheeger ratio of $h \approx 0.0026$. Plotting $\Phi(\alpha)$ (Fig. 1) and its derivative (Fig. 2) shows that there is a local minimum near $\alpha \approx 0.018$. When $\Psi$ is large, many individual nodes have personalized PageRank vectors that differ greatly from the overall distribution. This indicates that there are many nodes that are more representative of a small cluster than the entire graph. By plotting $\Psi(\alpha)$ (Fig. 3) and its derivative (Fig. 4), we can see that there is a distinct inflection point in the plot of $\Psi$ for the dumbbell graph $U$ as well.

## 6   A Graph Drawing Algorithm Using PageRank

The visualization of complex graphs provides many computational challenges. Graphs such as the World Wide Web and social networks are known to exhibit ubiquitous structure, including power-law distributions, small-world phenomena, and a community structure [1, 6, 12]. With large graphs, it is easy for such intricate structures to be lost in the sheer quantity of the nodes and edges, which can result in drawings that reflect a network's size but not necessarily its structure.
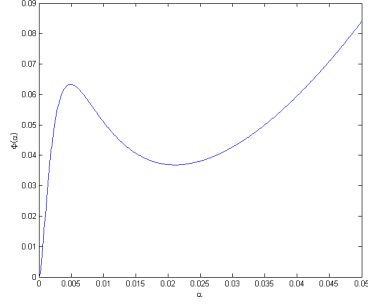
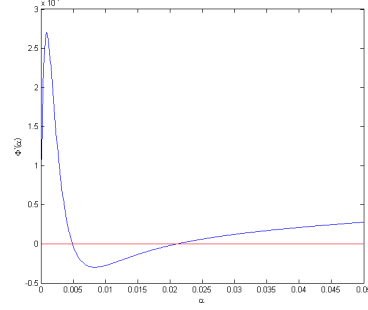**Fig. 1.** $\Phi(\alpha)$ *for the dumbell graph* $U$.



**Fig. 2.** $\Phi'(\alpha)$ *for the dumbell graph* $U$, *with the line* $y = 0$ *for reference.*
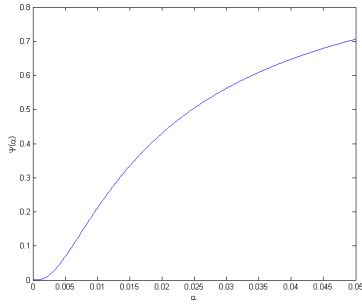


**Fig. 3.** $\Psi(\alpha)$ *for the dumbell graph* $U$.
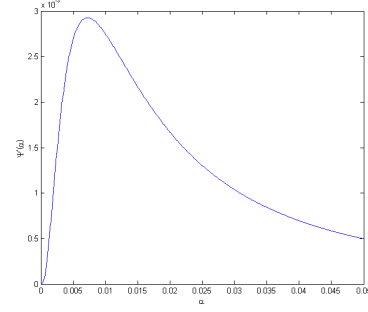


**Fig. 4.** $\Psi'(\alpha)$ *for the dumbell graph* $U$.

Given a set of nodes $S$, we can extract communities around each node and determine the layout of the graph using personalized PageRank. The arrangement can be done using a force-based graph layout algorithm such as the Kamada-Kawai algorithm [19]. The goal is to capture local communities; we can do this by assigning edges $\{s, v\}$ for each $s \in S$ and $v \in V \setminus S$ with weight inversely proportional to the personalized PageRank. This way, unrelated nodes with low PageRank will be forced to be distant, and close communities will remain close together. We also add edges $\{s, s'\}$ for $s, s' \in S$ with large weight to encourage separation of the individual communities. We use an implementation from Graphviz [14].

We note that because force-based algorithms are simulations, they do not guarantee the exact cluster structure, but we will illustrate that it works well in practice. Additionally, there are algorithms specifically designed for clustered graph visualization [10, 28] and highlighting high-ranking nodes [4], but they

impose a lot of artificial hierarchical structure onto the drawing and often require precomputing the clusters. Once we have a layout for all the nodes in the graph, we can partition them by using a Voronoi diagram. We compute the Voronoi diagram efficiently using Fortune's algorithm [13].

We tie together personalized PageRank and Voronoi diagrams in the following graph visualization algorithm:

**PageRank-Display**$(G, S, \alpha, \epsilon)$

Input: a graph $G = (V, E)$, a seed set $S \subseteq V$, a jumping constant $\alpha \in (0, 1]$, and an approximation factor $\epsilon > 0$.

1. For each $s \in S$, compute an approximate PageRank vector $p(\alpha, s)$.
2. Construct a new graph $G'$ with vertex set $V$ and edges as follows:
   - $\{s, v\}$ for $s \in S$ and $v \in V \setminus S$ with weight $1/p_s(v)$, as long as $p_s(v) > 0$.
   - $\{s, s'\}$ for $s, s' \in S$ with weight $10 \times \max_{s,v} 1/p_s(v)$.
3. Use a force-based display algorithm on $G'$ to determine coordinates $c_v$ for each $v \in V$.
4. Compute the Voronoi diagram on $S$.
5. Draw $G$ using the coordinates $c_v$, highlighting $S$, and overlaying the Voronoi diagram.

The jumping constant $\alpha$ is associated with the scale of the clustering. We can determine $\alpha$ either by trial and error or by optimizing $\Phi$ and $\Psi$ as in section 4. As long as $G$ is connected, the PageRank vector will be nonzero on every vertex. Using the algorithms from [3, 8], the approximation factor $\epsilon$ acts as a cutoff, and any node $v$ with PageRank less than $\epsilon d_v$ will be assigned zero. This is advantageous because the support of the approximate PageRank vector will be limited to the local community containing its seed. In **PageRank-Display**, we give weights to the edges equal to $1/p_s(v)$, but this is problematic if $p_s(v) = 0$. In that case, we omit the edge from $G'$ entirely.

We remark that the selection of $\epsilon$ will influence the size of the local communities: the subset of nodes with nonzero approximate PageRank has volume at most $\frac{2}{(1-\alpha)\epsilon}$ (see [3]). This implies that a good selection of $\epsilon$ is $O\left(\frac{|S|}{(1-\alpha)\mathrm{vol}(G)}\right)$.

We also remark that the selection of $S$ is important. If $S$ contains vertices that are not part of communities or two nodes in the same community, then there will be no structure to display. In general, the selection of $S$ is similar to the geometric problem of finding a set of points with minimum covering radius, which can be intractable (see [16]). There are several algorithms that can automatically choose $S$, including **PageRank-Clustering** as presented here.

We used **PageRank-Display** to demonstrate the existence of local structure in two real-world datasets. The first is a social network of 62 dolphins [21], and one can see in Fig. 5 that they can be divided into two communities.

A more interesting example is shown in Fig. 6. The graph represents games between 114 NCAA Division I American collegiate football teams [15] in 2000. The league is divided into smaller conferences; for each team, about half of its games are against conference opponents. An appropriate selection of the 8 teams in Fig. 6 reveal a partition that separates their conferences, and others are placed on the periphery of the drawing.
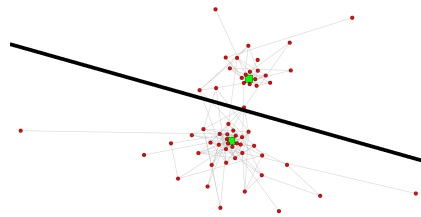
**Fig. 5. *PageRank-Display*** *(α = 0.03) on the dolphin social network [21], separating the dolphins into two communities.*
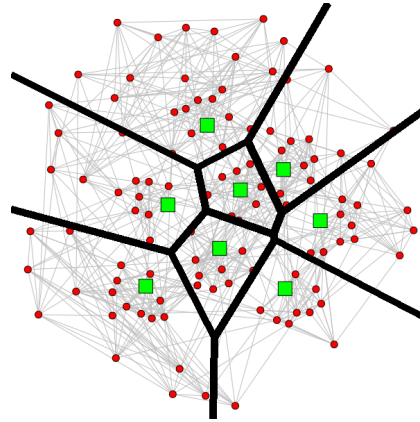
**Fig. 6. *PageRank-Display*** *(α = 0.1) on the football game network [15], highlighting 8 of the major collegiate conferences.*

# References

1. R. Albert, A.-L. Barabási and H. Jeong. Diameter of the World Wide Web. *Nature* **401** (1999), 130–131.
2. R. Andersen and F. Chung. Detecting sharp drops in PageRank and a simplified local partitioning algorithm. *Proceedings of the 4th International Conference Theory and Applications of Models of Computation* (2007), 1–12.
3. R. Andersen, F. Chung and K. Lang. Local graph partitioning using PageRank vectors. *Proceedings of the 47th Annual IEEE Symposium on Foundation of Computer Science* (FOCS 2006), 475–486.
4. U. Brandes and S. Cornelsen. Visual ranking of link structures. *Proceedings of the 7th International Workshop on Algorithms and Data Structures* (2001), 222–233.
5. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **30** (1998), 107–117.
6. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J. Wiener. Graph structure in the Web. *Computer Networks* **33** (2000), 1–6.
7. F. Chung, P. Horn and A. Tsiatas. Distributing antidote using PageRank vectors. *Internet Mathematics* **6**:2 (2009), 237–254.
8. F. Chung and W. Zhao. A sharp PageRank algorithm with applications to edge ranking and graph sparsification. Preprint, http://www.math.ucsd.edu/∼fan/wp/sharp.pdf.
9. M.E. Dyer and A.M. Frieze. A simple heuristic for the *p*-centre problem. *Operations Research Letters* **3**:6 (1985), 285–288.
10. P. Eades and Q. Feng. Multilevel visualization of clustered graphs. *Proceedings of the International Symposium on Graph Drawing* (1996), 101–112.
11. A.J. Enright, S. Van Dongen and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research* **30**:7 (2002), 1575–1584.

12. M. Faloutsos, P. Faloutsos and C. Faloutsos. On power-law relationships of the Internet topology. *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (SIGCOMM 1999), 251–262.

13. S. Fortune. A sweepline algorithm for Voronoi diagrams. *Proceedings of the Second Annual Symposium on Computational Geometry* (1986), 313–322.

14. E. Gansner and C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience* **30**:11 (2000), 1203–1233.

15. M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**:12, (2002), 7821-7826.

16. V. Guruswami, D. Micciancio and O. Regev. The complexity of the covering radius problem on lattices and codes. *Computational Complexity* **14**:2, (2005), 90–120.

17. D. Harel and Y. Koren. Graph drawing by high-dimensional embedding. *Proceedings of the 10th International Symposium on Graph Drawing* (2002), 207–219.

18. G. Jeh and J. Widom. Scaling personalized Web search. *Proceedings of the 12th International Conference on World Wide Web* (2003), 271–279.

19. T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters* **31**:1 (1989), 7–15.

20. S. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory* **28**:2 (1982), 129–137.

21. D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten and S.M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **54**:4 (2003), 396–405.

22. J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (1967), 281–297.

23. S. Mancoridis, B.S. Mitchell, Y. Chen and E.R. Gansner. Bunch: a clustering tool for the recovery and maintenance of software system structures. *Proceedings of the IEEE International Conference on Software Maintenance* (1999), 50–59.

24. J. Moody. Peer influence groups: identifying dense clusters in large networks. *Social Networks* **23**:4 (2001), 261–283.

25. M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E* **69** (2004), 026113.

26. A. Ng, M. Jordan and Y. Weiss. On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems* **14**:2 (2002), 849–856.

27. A. Noack. Modularity clustering is force-directed layout. *Physical Review E* **79** (2009), 026102.

28. G. Parker, G. Franck and C. Ware. Visualization of large nested graphs in 3D: navigation and interaction. *Journal of Visual Languages and Computing* **9**:3 (1998), 299–317.

29. M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM* **54**:4 (2007), Article 21.

30. S.E. Schaeffer. Graph clustering. *Computer Science Review* **1**:1 (2007), 27–64.

31. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**:8 (2000), 888–905.