Authority-Shift Clustering: Hierarchical Clustering by Authority Seeking on Graphs

Minsu Cho Kyoung Mu Lee
Department of EECS, ASRI, Seoul National University,151-742, Seoul, Korea
chominsu@gmail.com kyoungmu@snu.ac.kr

Abstract

In this paper, a novel hierarchical clustering method using link analysis techniques is introduced. The algorithm is formulated as an authority seeking procedure on graphs, which computes the shifts toward nodes with high authority scores. For the authority shift, we adopted the personalized PageRank score of the graph. Based on the concept of authority seeking, we achieve hierarchical clustering by iteratively propagating the authority scores to other nodes and shifting authority nodes. This scheme solves the chicken-egg difficulty in hierarchical clustering by a semi-global bottom-up approach exploiting the global structure of the graph. The experimental evaluation demonstrates that our algorithm is more powerful compared with existing graph-based approaches in clustering and image segmentation tasks.

1. Introduction

Clustering is a major building block of modern data analysis and is widely used in important tasks in fields ranging from statistics, computer science, and biology, to psychology and the social sciences. Clustering is to partition a set of collected data into disjoint subsets based on the underlying structure of the data [1]. In general clustering problems, given data often demonstrate the hierarchical structure of clusters at multiple scales. Thus, finding salient clusters in the data is by nature a problem of multiple scales and requires a hierarchical approach. Establishing the hierarchical structure of clusters involves an inherent chicken-and-egg difficulty since local decisions in clustering should consult with the larger-scale property of data, whereas larger-scale clustering is based on local-scale measurement. This is why classic hierarchical clustering methods [1] based on local or global decisions only are not successful in many cases.

In this work, we propose a novel graph-theoretical approach for hierarchical clustering based on link analysis techniques, which are commonly used to analyze complex networks such as the World Wide Web and social networks [12, 18]. We consider the clustering problem as a node-labelling problem on graphs, and propose a method called the *authority-shift clustering* algorithm. As illustrated in Fig.1, it is formulated as an authority seeking and labeling procedure that computes the shifts toward *authority nodes* with high authority scores. We adopt the personalized PageRank score [18, 10] for the purpose, which estimates authority scores in relation to the specified nodes. Based on this concept of authority seeking, we achieve hierarchical clustering by iteratively propagating the authority scores to other nodes and seeking authority nodes again. This process reveals the hierarchical clustering structure of the given data through multiple scales, and solves the chicken-egg difficulty in the hierarchical clustering by a semi-global bottom-up approach exploiting the global structure of the graph.

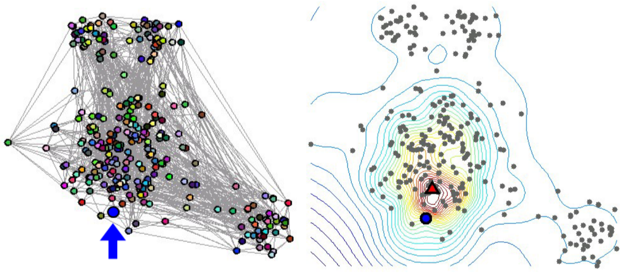

(a) A graph example (b) Authority landscape

Figure 1. Authority-shift. Given a graph, each node evaluates authority scores of other nodes. Authority landscape from the blue node at the bottom is visualized by contours. The blue node is shifted to its maximum authority node denoted by the red triangle. By this authority-shift, all nodes converge to their authority sinks and clusters emerge.

Authority seeking in the proposed algorithm is partly inspired by mode-seeking approaches [3, 22]. Unlike the mode-seeking based on kernel density estimation of samples in a feature space, however, our authority seeking is performed directly on the transition matrix of directed or undirected graphs, does not need an explicit feature space and kernel functions, and provides a semi-global shift ex-

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

ploiting the global structure of the graph by link analysis. The authority-shift is based on random walk models, and there have been several recent works on how to exploit the link structure of graph for detecting Web communities [11, 19, 13]. Markov clustering (MCL) method [24] and spectral clustering methods [17, 23] also can be explained by random walks on graphs.

Authority-shift clustering has three principal advantages over those previous methods. First, the authority-shift automatically computes the hierarchy of clusters and their stability in multiple scales without the pre-defined number of clusters. Second, the framework provides a computationally efficient algorithm applicable to large amounts of data in real-world clustering problems. Third, it naturally generalizes clustering problems for directed graphs as well as undirected graphs. The experimental evaluation demonstrates that our algorithm provides a powerful hierarchical clustering framework.

2. Clustering and Authority on Graphs

In graph-based methods for clustering, a relational graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with nodes $\mathcal{V}$, edges $\mathcal{E}$ and weights $\mathcal{W}$ is first constructed or given based on a pairwise relation between two elements. Weight $w_{ij} \in \mathcal{W}$ is associated with the graph edge from node i to node j, characterizing similarity or proximity from i to j. If $w_{ij} = w_{ji}$ is satisfied for all weights, the graph is undirected, and otherwise, it is a directed graph. Our framework can deal with directed graph as well as undirected graphs. In our approach, the clustering problem is viewed as a *node labeling problem on graph* G, in which nodes with the same label belong to the same cluster. We label each node with its *authority sink* using the authority-shift procedure explained in the following.

2.1. Authority Score by PageRank

For node labeling, we adopt the PageRank authority score which was originally developed by Brin and Page [18] to score the authority of Web pages using the hyperlink structure of the Web. In the PageRank formulation, the graph G is presented by the weight matrix $\boldsymbol{W}$, where the matrix entry $\boldsymbol{W}_{ij}$ is the weight w_{ij} from node i to node j. Assuming that all nodes in graph G have at least one outgoing edge with non-zero weight, the square row-stochastic matrix $\boldsymbol{P}$ is formed by normalizing the weight matrix $\boldsymbol{W}$ along rows to make each row sum to 1. The matrix entry $\boldsymbol{P}_{ij}$ can be interpreted to have the probability of transition from node i to node j. Assumming a random walker that follows the structure of the graph by the transition matrix $\boldsymbol{P}$ and sometimes randomly jumps to a node by the probability distribution $\boldsymbol{v}$, PageRank vector $\boldsymbol{r}$ satisfies the following equation [18]:

$$\boldsymbol{r}^T = \alpha \boldsymbol{r}^T \boldsymbol{P} + (1-\alpha)\boldsymbol{v}^T, \qquad (1)$$

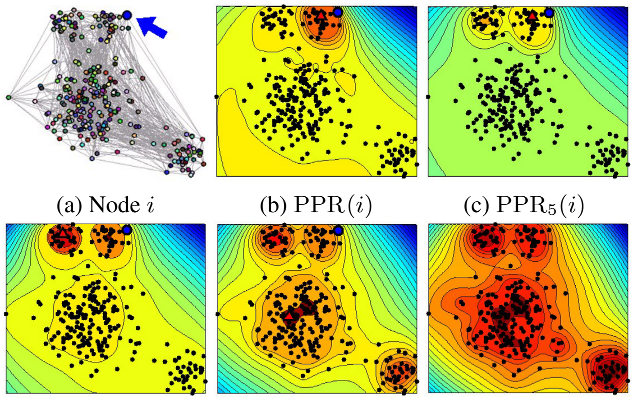


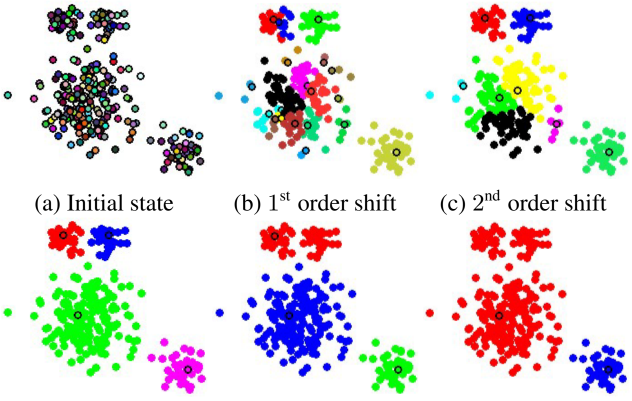
Figure 2. Landscapes of $\mathrm{PPR}_n(i)$ and GPR. The PPR order increases from (b) to (e) by PPR propagations.

in which $\boldsymbol{r}$ refers to the steady state distribution of the random walk governed by the transition matrix $\alpha\boldsymbol{P} + (1-\alpha)\boldsymbol{e}\boldsymbol{v}^T$ where $\boldsymbol{e}$ is a column of ones. If $\boldsymbol{v}$ is uniform over $\mathcal{V}$, then the steady state vector $\boldsymbol{r}$ is referred to as the global PageRank vector (GPR). For non-uniform $\boldsymbol{v}$, the steady state vector $\boldsymbol{r}$ has a weighted bias to the particular nodes represented by the personalization vector $\boldsymbol{v}$, and is referred to as the personalized PageRank (PPR) of $\boldsymbol{v}$, denoted by $\mathrm{PPR}(\boldsymbol{v})$ [7]. PPR creates the authority scores with respect to the specific nodes, and has been used in topic-sensitive web search based on the user personalization [18, 10]. In this work, we used $\boldsymbol{r}^T\boldsymbol{P}$ as PPR instead of $\boldsymbol{r}$ in (1) to suppress the effect of jumps $\boldsymbol{v}$ in the final score.

GPR and PPR can easily be computed by the power iteration or the linear system formulation [14]. In this work, we used linear system formulation for efficient computation. Note that there are many studies on efficient computation and approximation of PageRank motivated by the large size of Web graphs [2].

2.2. What is Good Clustering on the Graph?

Then, how to label each node based on the PageRank authority score? $\mathrm{PPR}(\boldsymbol{v})$ implies the authority score vector that the specific nodes weighted by the personalization vector $\boldsymbol{v}$ assign to $\mathcal{V}$, thus measuring how important each node is in relation to the nodes in $\boldsymbol{v}$. In the case when the personalization vector $\boldsymbol{v}$ has 1 at the only one node i and all other nodes have 0, the PPR will be referred to as the individual PPR vector. For notational brevity, the individual PPR vector of node i will be denoted by $\mathrm{PPR}(i)$ and the j^{th} coordinate scalar value of the vector $\mathrm{PPR}(i)$ will be denoted by $\mathrm{PPR}(i,j)$ in this paper. Then, $\mathrm{PPR}(i,j)$ is the probability that the random walker personalized to node i will be at node j in the steady state, which can be viewed as the importance of j with respect to i. Here, we can imagine the



(a) Initial state (b) 1st order shift (c) 2nd order shift

(d) 3rd order shift (e) 8th order shift (f) 73th order shift

Figure 3. Basic Authority-Shift. Iteratively propagating authority scores by PPR_n, n^{th} order authority-shift generates multi-scale clustering results. The nodes with the same color belong to the same cluster, and the authority sink for each cluster is outlined by black.

second-order random walker, which starts once again from the steady state of $\mathrm{PPR}(i)$. It corresponds to the random walker personalized to $\mathrm{PPR}(i)$. Generalizing this concept, we define the $(n+1)^{\text{th}}$ order PPR by PPR propagation as

$$\mathrm{PPR}_{n+1}(i) = \mathrm{PPR}(\mathrm{PPR}_n(i)), \tag{2}$$

where the PPR vector is recursively used for high-order personalization. As shown in Fig.2, $\mathrm{PPR}_n(i)$ makes a probabilistic landscape of the authority score around node i and gradually propagate the authority score beyond node i with increasing order n. Based on the n^{th} order PPR, we assign the *authority node* to node i for each order n by

$$\mathrm{Auth}_n(i) = \arg\max_{s \in \mathcal{V}} \mathrm{PPR}_n(i,s), \tag{3}$$

which has the highest authority score for node i. The use of authority nodes to discover structure in data is natural since the authority node can be considered a good representative of node i from a random walk view [12, 14].

Using the authority nodes, we can define a *self-authoritative cluster* $\mathcal{C}$ as

$$\mathcal{C} = \{i \in \mathcal{V} | \mathrm{Auth}_n(i) \in \mathcal{C}, \forall i \in \mathcal{C}\}, \mathcal{C} \subset \mathcal{V}. \tag{4}$$

which describes the set of nodes, each of which has its authority node as a member of the set. If it cannot be further divided into smaller self-authoritative clusters, it is referred to as a *minimal* self-authoritative cluster. We propose to define a good clustering as *a set of minimal self-authoritative clusters* that naturally satisfies intra-cluser similarity and inter-cluster dissimilarity. The set of minimal self-authoritative clusters evolves with the PPR propagations as shown in Fig.3. We can see that increasing order n simulates scaling effects in clustering.

3. Authority-Shift Algorithm

In this section we propose the algorithm to find sets of minimal self-authoritative clusters presented in the previous section.

3.1. Basic Authority-Shift

The linearity of PageRank vectors provides an efficient tool to compute other PPRs since the new PPRs are easily computed by a linear combination of previous PPRs [10]. Thus, given the individual PPRs $\mathrm{PPR}(i)(i \in \mathcal{V})$, they can be used as the bases for any PPRs. Using the linearity, the n^{th} order PPR of (2) can also be efficiently computed by

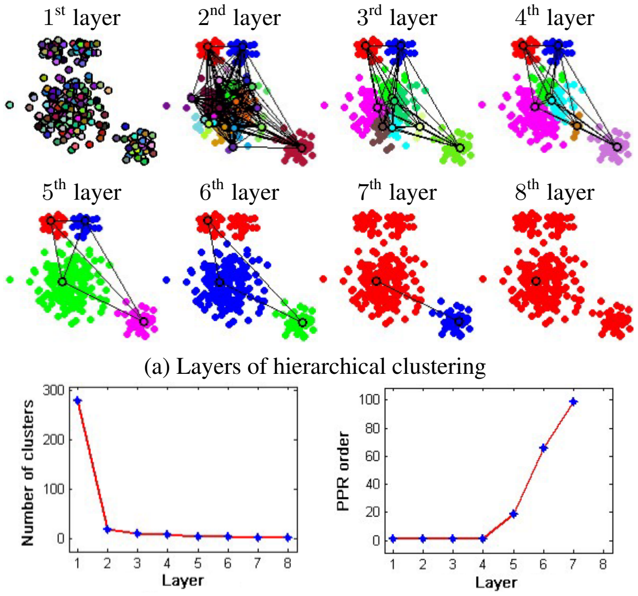
$$\mathrm{PPR}_{n+1}(i,j) = \sum_{s \in \mathcal{V}} \mathrm{PPR}(i,s)\mathrm{PPR}_n(s,j). \tag{5}$$

Assuming that $\boldsymbol{R}$ is the PPR matrix whose i^{th} row is $\mathrm{PPR}(i)$, this PPR propagation is simply expressed in terms of matrix power formulation as $\boldsymbol{R}^{n+1} = \boldsymbol{R}\boldsymbol{R}^n$.

For the given order n of PPR_n, the set of minimal self-authoritative clusters is obtained by shifting each node $i \in \mathcal{V}$ toward its authority node according to $i' \leftarrow \mathrm{Auth}_n(i)$. The shift evolves the trajectory of the node i following the authority nodes until it reaches the convergent node or find a cycle in the shifting sequence, referred to as the *authority sink*. Eventually, all nodes are linked by trajectories into a set of clusters each corresponding to a connected component, and the core of the component is its authority sink. Although this procedure is similar to mode-seeking of the meanshift [3], the authority-shift needs only to be computed once per node since the trajectories are constrained to pass through the set of nodes $\mathcal{V}$ and no special stopping rule is required unlike the meanshift. Thus, we can assign each node its authority sink by a single traversal, and the authority sinks determine the cluster membership for nodes and the total number of emergent clusters. This authority-shift procedure is iteratively performed for each PPR propagation until n^{th} PPR converges. Figure.3 shows a clustering example by the basic authority-shift. At each iteration, it requiers

Algorithm 1 Basic Authority-Shift Clustering

```
Input: Weight matrix $W \in \mathbb{R}^{n \times n}$, probability $\alpha$
Construct a row-stochastic transition matrix $\boldsymbol{P}$
Compute the PPR matrix $\boldsymbol{R}$
$n = 0$
**repeat**
    $n = n + 1$
    $\mathrm{Auth}_n(i) = \arg\max_{s \in \mathcal{V}} \boldsymbol{R}^n(s,i)$ $(i \in \mathcal{V})$
    Determine the clusters by authority node traversals
    Propagate authorities by $\boldsymbol{R}^{n+1} = \boldsymbol{R}\boldsymbol{R}^n$
**until** all elements lie in a single cluster.
Output: multi-scale clusterings
```



(a) Layers of hierarchical clustering

(b) The number of nodes and the order of PPR for each layer

Figure 4. Hierarchical authority-shift. For each authority-shift, node aggregation is performed and higher order graphs with supernodes are constructed as in (a). As shown in (b), the number of nodes reduces rapidly, and the PPR order for the shift starts to soar with the emergence of reliable clustering.

worst-case computational complexity of $O(N^3)$ where N denotes the number of the nodes. The following sections deal with more advanced approaches for both of speed and performance.

3.2. Hierarchical Authority-Shift

For more efficient and stable authority-shift, we propose a hierarchical scheme to recursively aggregate nodes in each cluster into a supernode. As clusters evolve after authority-shifts, we perform node aggregation as shown in Fig.4 and create a higher-layer graph with fewer supernodes. Let $\mathcal{V}^l$ denotes the set of nodes on the l^{th} layer, and $C^l_a (\subset \mathcal{V}^l)$ a cluster of nodes generated by n^{th} order authority-shift on the l^{th} layer, then each C^l_a transformed into a supernode a on the $(l+1)^{\text{th}}$ layer. PPR relations between the supernodes are updated by linear combinations of the n^{th} order PPRs on the l^{th} layer as follows.

$$\mathrm{PPR}^{l+1}_1(a,b) \leftarrow \frac{1}{\sigma_{l+1}(a)} \sum_{i \in C^l_a, j \in C^l_b} \sigma_l(i) \mathrm{PPR}^l_n(i,j), \tag{6}$$

$$\sigma_{l+1}(a) \leftarrow \sum_{i \in C^l_a} \sigma_l(i), \tag{7}$$

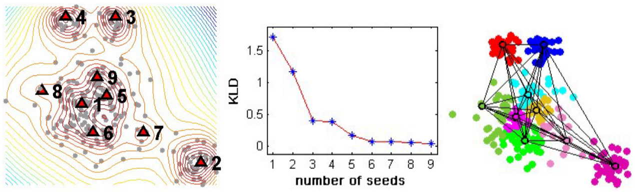
where $\sigma_1(i) = 1$ for all $i \in \mathcal{V}^1$. $\mathrm{PPR}^{l+1}_1(a,b)$ represents the probability that the random walker personalized to the nodes in C_a find themselves to stay on the nodes in C_b at its steady state. This agglomerative update rule is easily derived from the linearity of PPR [10], and significantly reduces the number of nodes and the cost of authority-shift in the subsequent process. Furthermore, it improves clustering quality and convergence speed by integrating the temporal smoothness constraint that no higher-order clustering result deviates from the lower-order ones. The overall algorithm of hierarchical authority-shift is summarized in Alg.2. As the final result, it generates a whole hierarchical representation of data as shown in Fig.4 (a).

Convergence Issue The authority propagation of (2) for authority-shifting is implemented by powers of PPR matrix $\boldsymbol{R}$, which is a stochastic matrix. Thus, according to the theory of Markov chains [15], it guarantees to converge to a steady state, which means one final cluster in the authority-shift, if PPR matrix $\boldsymbol{R}$ is irreducible. In our PPR computation of graph G, its transition matrix becomes strongly connected by additional jumps [18, 14], and the resultant PPR matrix $\boldsymbol{R}$ has irreducibility. Therefore, the authority-shift algorithm converges to one cluster. Note that this also applies to the hierarchical authority-shift algorithm since supernode aggregation of (6) and (7) preserves strong connectedness between nodes.

Selecting Stable Clustering Layers In the hierarchical structure of data obtained from our authority-shift, stable clustering resluts emerge on some layer of the hierarchy. Thus, data partitioning reduces to selecting a stable layer from the the hierarchy as shown in Fig.4. While various cluster validity indices [9] can be adopted for the purpose, we propose a simple *propagation-gap heuristic* for authority-shift clustering as illustrated in Fig.4. Here,

Algorithm 2 Hierarchical Authority-Shift Clustering

```
Input: Weight matrix $W \in \mathbb{R}^{n \times n}$, probability $\alpha$
Construct a row-stochastic transition matrix $\boldsymbol{P}$
(Optional) Initialize supernodes using authority seeds
Compute the PPR matrix $\boldsymbol{R}$
$l = 0$
repeat
    $l = l + 1$
    $n = 0$
    repeat
        $n = n + 1$
        ( Authority-shift on $l^{\text{th}}$ layer using the $n^{\text{th}}$ PPR )
        Compute authority nodes by (3)
        Determine the clusters by authority node traversals
        Propagate PPRs by (5)
    until any authority node is shifted
    ( Node aggregation for $(l+1)^{\text{th}}$ layer graph )
    Aggregate into super nodes by (6) and (7)
until all nodes lie in a single cluster.
Output: a hierarchy of clustering
```



(a) Extracted seeds (b) PageRank residual (c) Supernodes

Figure 5. Authority seed extraction for initial supernodes. Extracted authority seeds are shown in (a) by red triangle with the selected order. Increasing seeds, KLD between the GPR and the PPR of seeds decreases rapidly as in (b). Shifting all other nodes to the set of seeds, the initial graph with supernodes are constructed as in (c).

the goal is to choose the emerging layer l^* where all clusters are strongly self-authoritative, but the previous $1, 2, ..., l^* - 1$ layers all have weakly self-authoritative clusters. In the ideal case of k completely disconnected clusters $C_1, C_2, ..., C_k$, the $\mathrm{PPR}(i, j) = 0$ for $i \in C_a, i \in C_b, a \neq b$ holds, and then $\mathrm{PPR}_n(i, j) = 0$ also satisfies for any n. This means that further PPR propagations do not make any more changes in higher-order authority-shifts. It justifies that clustering stability can naturally be evaluated by the number of PPR propagations needed to merge the clusters. As shown in Fig.4, 18 PPR propagations are required on the 5th layer, but just one PPR propagation shift authority nodes on all the previous layers. This strongly implies that the reliable clusters emerges from the 5th layer.

3.3. Initial Authority Seeds by PageRank

For large graphs, running the hierarchical authority-shift directly on all the nodes can be still time and memory consuming. Therefore, instead of using all the nodes directly, we can choose to generate supernodes at the initial step by selecting the seed nodes and shifting all the other nodes to them as shown in Fig.5. Success of this approach depends on how to collect minimal but sufficient seeds for final clusters. Suppose we already have the set of k seeds S_k from the graphs, then what is the best $(k+1)^{th}$ seed s_{k+1} not covered by the current set of k seeds? Our method is inspired by two observations: (1) the important nodes of a graph have relatively high authority scores in the GPR, and (2) PPR of current seeds S_k gives higher scores to the nodes likely included in the clusters of S_k than the GPR scores do. Thus, to evaluate the important nodes not covered by the current set of seeds S_k, we define PageRank residual Δ_k as

$$\begin{aligned} \Delta_k(i) &= \mathrm{GPR}(i) - \mathrm{PPR}(S_k, i) \\ &= \mathrm{GPR}(i) - \textstyle\sum_{m=1}^{k} \mathrm{PPR}(s_m, i)/k, \end{aligned} \tag{8}$$

which means the difference between GPR and PPR of S_k. Then, the $(k+1)_{th}$ seed is selected by $s_{k+1} = \arg\max_i \Delta_k(i)$. This way, we can sequentially extract the seeds from the graph. As shown in Fig.5(a), our seed extraction algorithm successfully collects the seeds from more important to less important ones. Kullback-Leibler divergence (KLD) between two discrete probabilities GPR and $\mathrm{PPR}(S_k)$ is used to estimate the dissimilarity between GPR and $\mathrm{PPR}(S_k)$ as shown in Fig.5(b). Thus, the sufficient number of seeds is determined by KLD threshold τ in the sequential seed extraction scheme. Once the set of seed nodes S is all collected, we produce initial supernodes shifting each node i into a set of seeds by $\mathrm{Auth}_0(i) = \arg\max_{s \in S} \mathrm{PPR}(s, i)$ as shown in Fig.5(c).

4. Experiments

We evaluate our algorithm on point clustering using synthetic point sets and on image segmentation using the Berkeley image dataset.

4.1. Point set clustering

To test and evaluate our algorithm, we applied it to point set clustering problems using synthetic datasets, and compared it with several popular clustering algorithms, which are k-means, affinity propagation (AP) [8], mean-shift [3], medoid-shift [22], single-link hierarchical agglomerative clustering (HAC) [1], Markov clustering (MCL) [24], and spectral clustering [17]. To reduce the chance that a specific point pattern accidentally favors an algorithm, we use six sorts of different point patters which are all constructed by mixture of gaussian and uniform random distribution as shown in 6. For each pattern, 100 instances are generated and tested by all the algorithms.

For all the datasets, the points were normalized to occupy the $[-1, 1]^2$ space. The weight matrix $W \in R^{N \times N}$ is computed by $W_{ij} = exp(\frac{-d(i,j)^2}{\sigma^2})$ for $i \neq j$ and $W_{ii} = 0$, where $d(i, j)$ is Euclidian distance function between the point i and j. σ is a scale parameter which we fixed $\sigma^2 = 0.1$. It is commonly used for spectral clustering, MCL, and ours. Since k-mean, HAC, and spectral clustering requires the number of clusters for clustering, we provided the exact number of clusters K for each dataset to the methods. For authority-shift, we fixed $\alpha = 0.95$ and select the layer with the nearest number of clusters to K. For mean-shift, medoid-shift and MCL, we tune their parameters for the datasets. To evaluate performance, two common metrics, the normalized mutual information (NMI) [20] and Rand index (RI) [21] are used. For the details of the metrics, we refer to [20] and [21].

The exemplars are visualized in Fig.6 and the average performance over 100 point set generation for each pattern is summarized in Table.1. All the running times are given on 2.40 GHz Core2 Quad desktop PC. In Fig.6, the results of k-mean and medoid-shift are not visualized since their results are very similar to AP and mean-shift, respectively. In

Table.1, the best two results are highlighted in bold with regard to each metric. As shown in the results, authority-shift, MCL, spectral methods are ranked high, and the authority-shift outperforms the others in most cases. Note that these synthetic datasets can not be clustered in a meaningful way by methods that assume a compact shape for the data like k-means and AP. Mean-shift and method-shift also easily fails when clusters do not form clear modes in the space. HAC and spectral clustering suffer from the perturbation as shown in Fig.6.

The time cost of hierarchical authority-shift is mainly governed by the complexity of matrix powers. The worst case with fully connected graph is $O(N^3)$ in naive implementation where N is the num of nodes. However, our supernode aggregation significantly reduces the costs in subsequent iterations. Thus, as shown in Table.1, the practical average time of authority-shift is less than those of MCL, AP, and spectral method that all have also $O(N^3)$ worst-case complexity. MCL and AP takes much longer time than others since they iteratively propagate messages or probabilities between all the neighboring nodes at each iteration, while authority-shift performs fast due to the hierarchical shifting scheme. In the case of sparse graphs, the complexity decreases much further. Moreover, when dealing with a large graph, the initial seeding technique also can be adopted for reducing the compuatitional costs of authority-shift.

4.2. Image Segmentation

In the second experiment, we applied our algorithm on image segmentation problem, and compared ours with state-of-the-art segmentation methods. The graph is constructed by simple 8-neighborhood system MRFs on the image grid. The sparse weight matrix is computed using only pixel color cues by $W_{ij} = exp(-\lambda d(i,j)^2)$ for $i \neq j$ and $W_{ii} = 0$, where $d(i,j)$ is normalized RGB distance function between node i and j. We fixed $\lambda = 95$ in our experiments. For large amount of nodes in images, we used the seeding technique by PageRank residual with the KLD threshold $\tau = 0.02$.

We test our method on the well-known Berkeley image segmentation dataset [16] It provides 300 images and corresponding ground truth data. For visual comparison, we compare our results with the results of the NCut segmentation method [23] in Fig.7. We used the MATLAB implementation of NCut algorithm provided by the author of [23] and the same weight matrix is used as ours for this experiment. In Fig.7(a), the test image, the extracted seeds, the initial supernodes, and four layers among our clustering hierarchy are shown. The following two results are obtained by NCut with the same number of segment as our final two results in the fifth and the sixth column. In Fig.7(b), test image, two salient layers in our results, and one NCut result are shown. The examples in Fig.7 compare coarse-level segmentation results of ours and NCut's given the same number of clusters. It demonstrates that the authority-shift preserves the good segmentation quality until the coarse-level in its segmentation hierarchy.

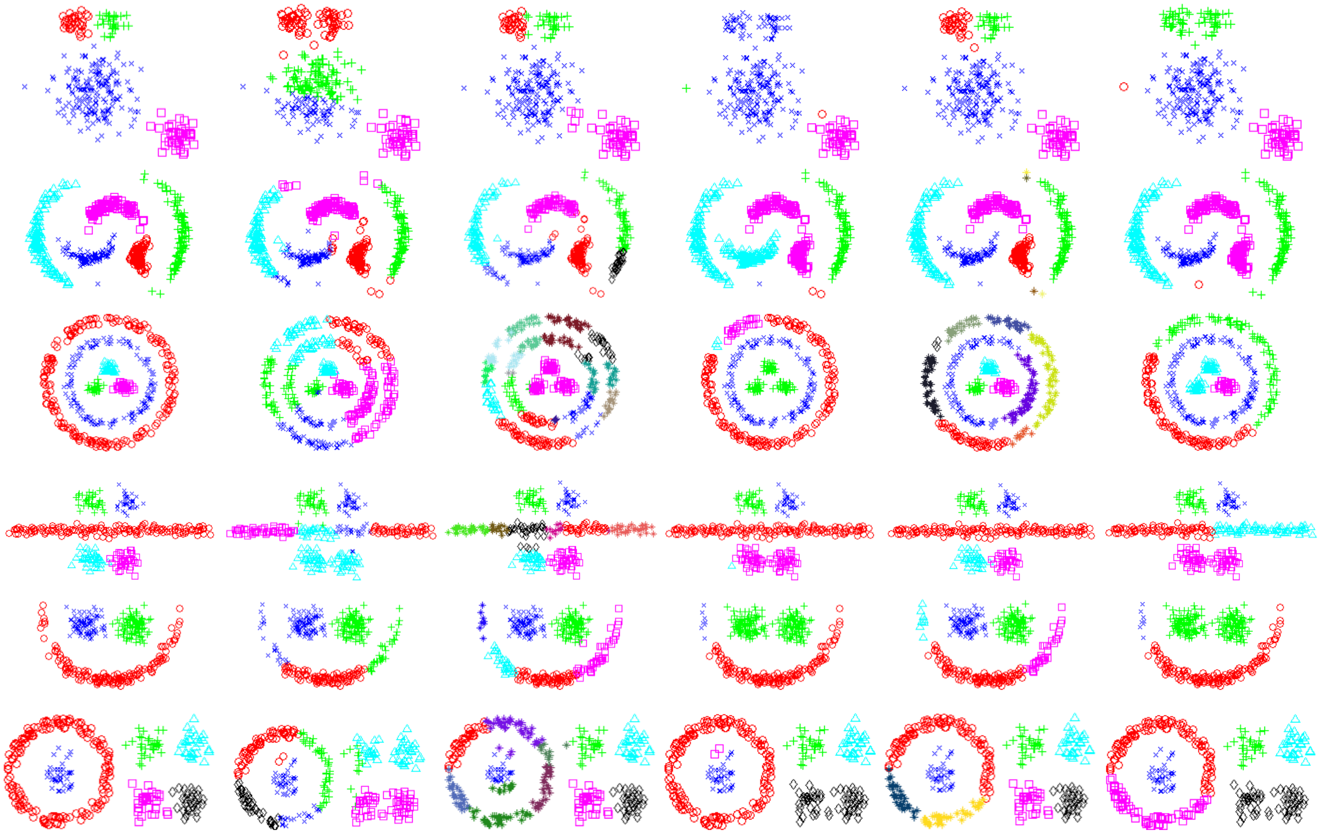
Method/ Score	PRI	VoI	GCE	BDE
NCuts [23]	0.7242	2.9061	0.2232	17.15
MCSpec [26]	0.7357	2.6336	0.2469	15.40
MNcut [4]	0.7559	2.4701	0.1925	**15.10**
NormTree [25]	0.7521	2.4954	0.2373	16.30
GraphBased [6]	0.7139	3.3949	**0.1746**	16.67
MShift [3]	**0.7958**	**1.9725**	**0.1888**	**14.41**
Ours	**0.7738**	**2.0292**	0.2225	16.09

Table 2. Comparison of different methods on Berkeley image database by Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE) and Boundary Displacement Error (BDE). The best two results are highlighted in bold. Our method shows competitive results compared to state-of-the-arts

For quantitative evaluation of the overall segmentation quality we follow a recent trend using four different quality measures: the Probabilistic Rand Index (PRI), the Variation of Information (VoI), the Global Consistency Error (GCE) and the Boundary Displacement Error (BDE) [25]. For this benchmarks, we simply select the lowest layer clustering which is stable over 5th order PPR. We compared the scores calculated for our algorithm to the popular state-of-the-art segmentation algorithms: the Mean Shift method (MShift) [3], the normalized cut algorithm (NCut) [23], the multiclass spectral (MCSpec) [26], the graph based segmentation (GraphBased) [6], the multi-scale normalized cut approach (MNCut) [4], and the normalized partitioning tree (NormTree) [25]. The Results were taken from [25] and [5]. Table.2 summarizes the scores for all algorithms. Our proposed algorithm shows competitive results compared to state-of-the-art. Although our method is based on simple local-neighborhood graph and pixel color value, it is ranked in the second place in PRI and VOI and outperforms all the graph-based image segmentation methods.

5. Conclusion

We presented a novel hierarchical clustering method, called authority-shift clustering, that exploits the global structure of graphs based on link analysis techniques. Experiments demonstrated that it provides robust performance comparable to state-of-the-art methods in point clustering and image segmentation. Although we did not demonstrate in this paper, it can also be applied to clustering on directed graphs since the algorithm is formulated as an authority seeking procedure on directed graphs as well as undirected graphs. In future work, we will extend our work for this



(a) authority-shift (b) AP (c) mean-shift (d) HAC (e) MCL (f) spectral

Figure 6. Point set results

	k-means		AP		mean-shift		medoid-shift		HAC		MCL		spectral		authority-shift	
	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI	NMI	RI
1^{st} data	0.653	0.733	0.669	0.750	0.876	0.956	0.896	0.951	0.535	0.695	**0.899**	**0.958**	0.841	0.930	**0.913**	**0.965**
2^{nd} data	0.881	0.961	0.881	0.963	0.926	0.975	0.920	0.971	0.874	0.895	**0.984**	**0.995**	0.957	0.967	**0.989**	**0.995**
3^{rd} data	0.269	0.629	0.242	0.626	0.354	0.669	0.434	0.683	0.853	0.879	0.824	0.858	**0.874**	**0.911**	**0.930**	**0.953**
4^{th} data	0.589	0.726	0.562	0.719	0.716	0.784	0.710	0.784	0.895	0.926	**0.983**	**0.991**	0.805	0.834	**0.852**	**0.876**
5^{th} data	0.640	0.808	0.652	0.816	0.776	0.877	0.762	0.873	0.293	0.523	**0.945**	**0.975**	0.608	0.731	**0.875**	**0.916**
6^{th} data	0.585	0.738	0.606	0.751	0.722	0.786	0.673	0.767	0.909	0.947	0.917	0.949	**0.928**	**0.951**	**0.954**	**0.967**
Avg. Time	0.09s		5.89s		0.03s		0.08s		0.03s		5.86s		1.16s		0.22s	

Table 1. Performance on Synthetic datasets. Each pattern consists of mixture of gaussian and uniform point distribution, and the algorithms are tested on 100 instances of the pattern. The average NMI and RI are shown here for each method, and the best two values are highlighted in bold. The last row shows the average running time for each method.

direction and apply it to various vison problems requiring hierarchical approaches.

Acknowledgements

This research was supported in part by the IT R&D program of MKE/IITA (2008-F-030-01), and in part by the ITRC program of MKE/NIPA through 3DRC (NIPA-2009-C1090-0902-0018), Korea.

References

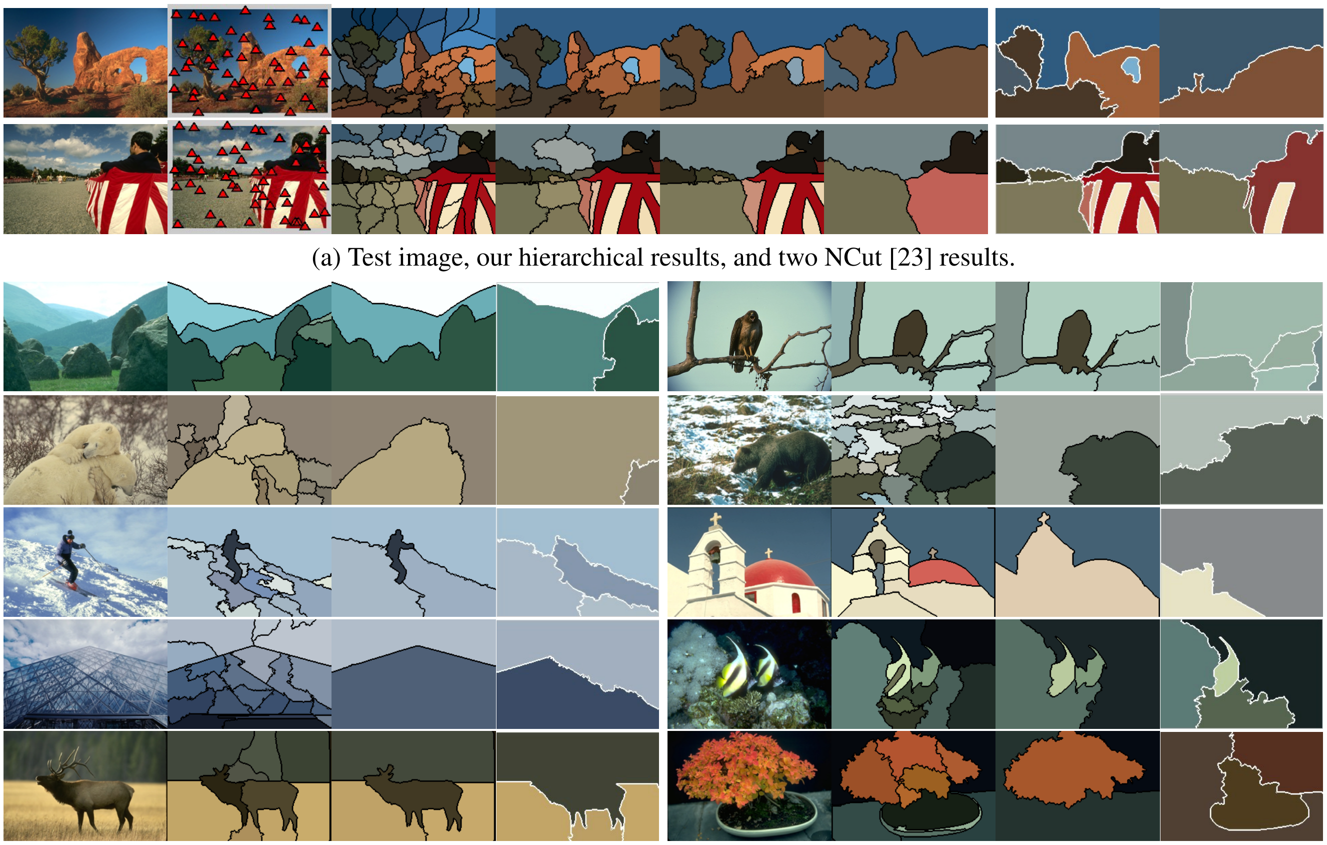
[1] A.K.Jain and R.C.Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1998.

[2] P. Berkhin. A survey on pagerank computing. *Internet Mathematics*, 2005.

[3] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, pages 24, 603–619, 2002.

[4] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. *CVPR*, 2005.

[5] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. *ICCV*, 2009.



(a) Test image, our hierarchical results, and two NCut [23] results.

(b) Test image, two salient layers in our result, and a NCut [23] result compared to ours.

Figure 7. Image segmentation results compared with the normalized cut (NCut) method of [23] on the Berkeley dataset. Our hierarchical clustering results are segmented by black lines, and the results of NCut by white lines. Note that the number of segments in the results of NCut is set to the same as ours. For details, see the text.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[7] D. Fogaras, B. Racz, K. Csalogany, and T. Sarlos. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2005.

[8] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[9] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *JIIS*, 2001.

[10] T. H. Haveliwala. Topic-sensitive pagerank. *WWW*, 2002.

[11] J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. *ECML/PKDD*, 2006.

[12] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, pages 46:5 604–632, 1999.

[13] V. D. Konstantin Avrachenkov and D. Nemirovsky. Pagerank based clustering of hypertext document collections. *SIGIR*, 2008.

[14] A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 2003.

[15] J. S. Liu. Monte carlo strategies in scientific computing. *Springer*, 2001.

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *ICCV*, 2001.

[17] A. Ng, M. Jordan, , and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2001.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical Report, Stanford University*, 1998.

[19] P. Pons and M. Latapy. Computing communities in large networks using random walks. *JGAA*, 2006.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical recipes in c. *Cambridge University Press*, 1988.

[21] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

[22] Y. A. Sheikh, E. A. Khan, and T. Kanade. Mode-seeking by medoidshifts. *ICCV*, 2007.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, page 22(8), 2000.

[24] S. van Dongen. Graph clustering by flow simulation. *PhD thesis, University of Utrecht*, 2000.

[25] T. Wang, Y. Jia, X.-S. Hua, C. Zhang, , and L. Quan. Normalized tree partitioning for image segmentation. *CVPR*, 2008.

[26] S. X. Yu and J. B. Shi. Multiclass spectral clustering. *ICCV*, pages 313–319, 2003.