Pattern Recognition 46 (2013) 3056–3065

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Agglomerative clustering via maximum incremental path integral

Wei Zhang [a], Deli Zhao [a], Xiaogang Wang [b,*]

[a] Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong
[b] Department of Electronic Engineering, The Chinese University of Hong Kong, Rm. 415, Ho Sin Hang Engineering Building, Shatin, Hong Kong

ARTICLE INFO

Article history:
Received 18 July 2012
Received in revised form
3 April 2013
Accepted 10 April 2013
Available online 24 April 2013

Keywords:
Agglomerative clustering
Path integral
Graph algorithms
Random walk

ABSTRACT

Agglomerative clustering, which iteratively merges small clusters, is commonly used for clustering because it is conceptually simple and produces a hierarchy of clusters. In this paper, we propose a novel graph-structural agglomerative clustering algorithm, where the graph encodes local structures of data. The idea is to define a structural descriptor of clusters on the graph and to assume that two clusters have large affinity if their structural descriptors undergo substantial change when merging them into one cluster. A key insight of this paper to treat a cluster as a dynamical system and its samples as states. Based on that, *Path Integral*, which has been introduced in statistical mechanics and quantum mechanics, is utilized to measure the stability of a dynamical system. It is proposed as the structural descriptor, and the affinity between two clusters is defined as *Incremental Path Integral*, which can be computed in a closed-form exact solution, with linear time complexity with respect to the maximum size of clusters. A probabilistic justification of the algorithm based on absorbing random walk is provided. Experimental comparison on toy data and imagery data shows that it achieves considerable improvement over the state-of-the-art clustering algorithms.

1. Introduction

Clustering is a classical machine learning topic with wide applications in diverse fields. It includes two major categories [1,2]: partitional clustering, which determines all clusters at once, and hierarchical clustering, which creates a hierarchy of clusters in a bottom-up (or agglomerative) process by merging small clusters or in a top-down (or divisive) process by dividing large clusters into small ones. Numerous algorithms have been proposed, such as *k*-means [2], spectral clustering [3–8] and affinity propagation [9], and achieved great success.

This work stands on the success of agglomerative clustering, which is commonly used because it is conceptually simple and produces a hierarchy of clusters. Beginning with a large number of initial small clusters, the agglomerative clustering algorithms iteratively select two clusters with the largest affinity under certain measures to merge, until some stopping condition is reached. Therefore, the affinity measure of clusters is critically important. Linkages, e.g., single linkage, complete linkage and average linkage [2], define the affinity based on inter-cluster pairwise distances. Since pairwise distances do not well capture the global structures of data, complete linkage and average linkage fail on clustering data with manifold structures. Although single linkage performs better in this case, it is very sensitive to noisy distances. Examples can be found in Fig. 3. Lossy coding theory of multivariate mixed data [10] characterizes the affinity of two clusters with the variational coding length of coding the merged cluster against coding the two clusters separately. It exhibits exceptional performance for clustering multivariate mixed Gaussian or subspace data, but is not suitable for data from other distributions. There are also approaches based on probabilistic models, such as Bayesian hierarchical clustering [11]. They all assume the forms of underlying data distributions, which are unknown in many applications.

In this paper, we propose a novel graph-structural agglomerative clustering algorithm. Although the power of graphs has been extensively exploited in clustering [3,5,12,13], semi-supervised learning [14,15], and manifold learning [16], they have received little attention in agglomerative clustering. In our algorithm the pairwise distances are only used to build a neighborhood graph, since studies [16] show the effectiveness of using local neighborhood graphs to model data lying on a low-dimensional manifold embedded in a high-dimensional space. Then a structural descriptor is defined to characterize the global structure of a cluster from the local information encoded by the graph. It is assumed that two clusters have large affinity if their structural descriptors undergo substantial change when merging them into one cluster.

We propose *path integral* as the structural descriptor of clusters. Paths are a fundamental concept of graph theory, and are used in many graph-based algorithms. The description of paths gives rich information about the data. There has been a lot of research work

* Corresponding author. Tel.: +852 39438283; fax: +852 26035558.
E-mail addresses: wzhang009@gmail.com (W. Zhang), zhaodeli@gmail.com (D. Zhao), xgwang@ee.cuhk.edu.hk (X. Wang).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.013

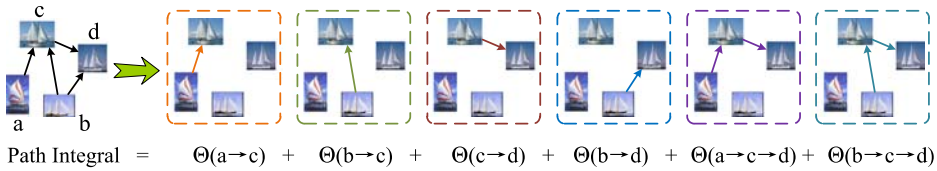

Fig. 1. A toy example on the path integral description of a cluster. There are four length-1 paths and two length-2 paths in the cluster. The path integral is computed as the sum of contributions of these paths. How to obtain each path's contribution is described in Section 3. For clarity, the vertices outside the cluster and the outer links are not shown.

on studying various aspects of paths on graphs, such as finding the shortest paths between nodes [17,18] or computing the similarity between two nodes over paths [19,20]. For example, Saerens et al. [18] proposed the randomized shortest path problem, which allows a route agent to follow different paths according to some probability distributions instead of only following the shortest path connecting a source and a destination. Their proposed model could be used to measure the dissimilarity between two nodes accounting for multiple paths. However, the purpose of this paper is to develop a structural descriptor of clusters, instead of finding the shortest path between nodes or computing the pairwise similarities between samples. The concept of path integral was first introduced in statistical mechanics and quantum mechanics [21–23], where it summed up the contributions of all possible paths to the evolution of a dynamical system. In this work, we provide our own formulation of path integral and its probabilistic interpretation based on absorbing random walk. If we treat a cluster as a dynamical system, with vertices as states and edge weights as transition probabilities between states, then the path integral measures the stability of the dynamical system, i.e. randomly starting with any state of the dynamical system, the probability of remaining within the same system after certain steps of transitions. An example is shown in Fig. 1. The affinity of two clusters is defined as the *incremental path integral* after merging them. An intuitive explanation is that if two clusters are closely connected, their stability will greatly increase after merging them. We show that the incremental path integral can be computed in a closed-form exact solution, with linear time complexity with respect to the maximum size of clusters. Experimental comparisons on toy data and imagery data show the excellent performance of the proposed algorithm and its robustness to parameter settings.

Our algorithm has several advantages compared with existing methods. First, since it measures the affinity of clusters based on the neighborhood graph instead of directly on pairwise distances between any pairs of samples, it can better cluster data on manifolds and is more robust to noisy distances compared with linkage algorithms [2] widely used in agglomerative clustering. Second, different from spectral clustering [3,5] and clustering on the manifold embedding results, it does not use any relaxation or approximation. The graph structural merging strategy also makes our algorithm more robust to noisy links than spectral clustering, because our structural descriptor involves solving a linear system, while the spectral clustering utilizes eigen-decomposition. Solving eigen-vectors is more sensitive to noise than solving a linear system [24,12]. Examples in the bottom row of Fig. 3 show that our algorithm can handle for multi-scale data, i.e., a dataset that contains structures with different densities and sizes, which is the limitation of spectral clustering [25,26]. Third, it only requires the pairwise distances or similarities of samples without any assumptions on the underlying data distributions. This is useful in the case when the vector representations of data are not available. Therefore, it has better flexibility and generalization than other agglomerative clustering methods such as lossy coding theory [10] and Bayesian hierarchical clustering [11].

The paper is organized as follows. For ease of reading, the overall clustering algorithm is first outlined in Section 2. Then, the theoretical framework of path integral and incremental path integral is presented in Section 3. Section 4 provides a probabilistic interpretation of our algorithm based on absorbing random walk. Experimental validations and conclusion are given in Sections 5 and 6, respectively.

2. Graph-structural agglomerative clustering

Our algorithm iteratively merges two clusters with maximum affinity on a directed graph.

Building the digraph. Given a set of sample vectors $\mathcal{X}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, we build a directed graph $G=(V,E)$, where V is the set of vertices corresponding to the samples in $\mathcal{X}$, and E is the set of edges connecting vertices. The graph is associated with a weighted adjacency matrix $\mathbf{W}=[w_{ij}]$, where w_{ij} is the pairwise similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ defined as

$$w_{ij}=\begin{cases}\exp\left(-\dfrac{\mathrm{dist}(i,j)^2}{\sigma^2}\right), & \text{if } \mathbf{x}_j\in\mathcal{N}_i^K,\\ 0, & \text{otherwise.}\end{cases} \tag{1}$$

$\mathrm{dist}(i,j)$ is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathcal{N}_i^K$ is the set of K-nearest neighbors of $\mathbf{x}_i$. If $\mathbf{x}_j\in\mathcal{N}_i^K$, there is an edge pointing from $\mathbf{x}_i$ to $\mathbf{x}_j$ with weight w_{ij}. σ^2 is estimated by $\sigma^2=[\sum_{i=1}^n\sum_{\mathbf{x}_j\in\mathcal{N}_i^3}\mathrm{dist}(i,j)^2]/[3n(-\ln a)]$.[1] K and a are free parameters to be set.

We define a random walk model on this directed graph. Denote the transition probability matrix as $\mathbf{P}$, whose element p_{ij} is the one-step transition probability from vertex i to vertex j. $\mathbf{P}$ is calculated as

$$\mathbf{P}=\mathbf{D}^{-1}\mathbf{W}, \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal element $d_{ii}=\sum_{j=1}^n w_{ij}$, such that $\sum_{j=1}^n p_{ij}=1$. The path integral of a cluster is computed by summing the paths within the cluster on the directed graph weighted by transition probabilities.

Affinity measure of clusters. Given two clusters $\mathcal{C}_a$ and $\mathcal{C}_b$, their structural affinity is measured as the amount of incremental path integral $\mathcal{A}_{\mathcal{C}_a,\mathcal{C}_b}$ when merging them, i.e.,

$$\mathcal{A}_{\mathcal{C}_a,\mathcal{C}_b}=(\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_a})+(\mathcal{S}_{\mathcal{C}_b|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_b}). \tag{3}$$

$\mathcal{S}_{\mathcal{C}_a}$ is the path integral descriptor of $\mathcal{C}_a$ and sums up all the paths in $\mathcal{C}_a$. $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ is the conditional path integral descriptor. All the paths to be counted lie in $\mathcal{C}_a\cup\mathcal{C}_b$. However, their starting and ending vertices must be within $\mathcal{C}_a$. If the vertices in $\mathcal{C}_a$ and $\mathcal{C}_b$ are strongly connected, merging them will create many new paths for the pairs of vertices in $\mathcal{C}_a$, and therefore $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ will be much larger than $\mathcal{S}_{\mathcal{C}_a}$. Section 4 will show that $\mathcal{S}_{\mathcal{C}_a}$ measures the cluster's stability, if $\mathcal{C}_a$ is treated as a dynamical system. An example for illustration is shown in Fig. 2. The closed-form expressions of $\mathcal{S}_{\mathcal{C}_a}$

[1] It is equivalent to setting the geometric mean of weights associated with edges pointing to 3-nearest-neighbors as a, i.e., $(\prod_{i=1}^n\prod_{\mathbf{x}_j\in\mathcal{N}_i^3}w_{ij})^{1/(3n)}=a$.

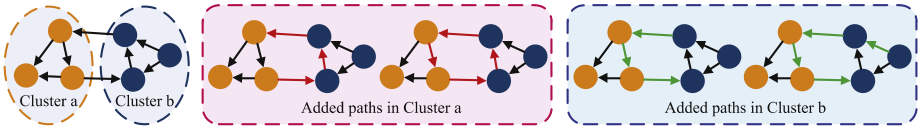

Fig. 2. An illustration of incremental path integral. After merging $\mathcal{C}_a$ and $\mathcal{C}_b$, there are two new (red) paths which are in $\mathcal{C}_a \cup \mathcal{C}_b$ and whose starting and ending vertices are both in $\mathcal{C}_a$. Similarly, there are also two new (green) paths for $\mathcal{C}_b$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

and $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ are given in Eqs. (9) and (12), respectively. The computational complexity is linear with respect to the maximum size of clusters. Refer to Section 3.4 for the details of efficient computation.

Initial clusters. We use a simple *nearest neighbor merging* algorithm to obtain initial clusters. First, each sample and its nearest neighbor form a cluster and we obtain n clusters, each of which has two samples. Then, the clusters are merged to remove duplicated samples, i.e., we merge two clusters if their intersection is nonempty, until the number of clusters cannot be reduced.

The overall algorithm is presented in Algorithm 1.

Algorithm 1. Agglomerative clustering via maximum incremental path integral.

Input: a set of n sample vectors $\mathcal{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$, and the target number of clusters n_T.
Build the graph G with k-nearest-neighbors and compute its weighted adjacency matrix $\mathbf{W}$ with Eq. (1);
Get the transition probability matrix $\mathbf{P}$ with Eq. (2);
Form $n_{\mathcal{C}}$ initial clusters $\mathcal{C}_{\mathcal{C}}=\{\mathcal{C}_1,\ldots,\mathcal{C}_{n_{\mathcal{C}}}\}$, i.e., assign each sample $\mathbf{x}_i$ to a cluster, using nearest neighbor merging.
while $n_{\mathcal{C}} > n_T$ **do**
  Search two clusters $\mathcal{C}_a$ and $\mathcal{C}_b$, such that $\{\mathcal{C}_a,\mathcal{C}_b\}=\mathrm{argmax}_{\mathcal{C}_a,\mathcal{C}_b\in\mathcal{C}_{\mathcal{C}}}\mathcal{A}_{\mathcal{C}_a,\mathcal{C}_b}$, where $\mathcal{A}_{\mathcal{C}_a,\mathcal{C}_b}$ is the affinity measure between $\mathcal{C}_a$ and $\mathcal{C}_b$, computed using $\mathcal{A}_{\mathcal{C}_a,\mathcal{C}_b}=(\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_a})+(\mathcal{S}_{\mathcal{C}_b|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_b})$.
  $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ is computed as $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}=\frac{1}{|\mathcal{C}_a|^2}\mathbf{1}_{\mathcal{C}_a}^T(\mathbf{I}-z\mathbf{P}_{\mathcal{C}_a\cup\mathcal{C}_b})^{-1}\mathbf{1}_{\mathcal{C}_a}$.
  And $\mathcal{S}_{\mathcal{C}_a}$ is computed as $\mathcal{S}_{\mathcal{C}_a}=\frac{1}{|\mathcal{C}_a|^2}\mathbf{1}_{\mathcal{C}_a}^T(\mathbf{I}-z\mathbf{P}_{\mathcal{C}_a})^{-1}\mathbf{1}_{\mathcal{C}_a}$.
  $\mathcal{S}_{\mathcal{C}_b|\mathcal{C}_a\cup\mathcal{C}_b}$ and $\mathcal{S}_{\mathcal{C}_b}$ are computed in a similar way.
  $\mathcal{C}_{\mathcal{C}}\leftarrow\{\mathcal{C}_{\mathcal{C}}\backslash\{\mathcal{C}_a,\mathcal{C}_b\}\}\cup\{\mathcal{C}_a\cup\mathcal{C}_b\}\}$, and $n_{\mathcal{C}}=n_{\mathcal{C}}-1$.
end while
Output: $\mathcal{C}_{\mathcal{C}}$.

3. Incremental path integral: a structural affinity measure

A key component of the proposed algorithm is to compute path integral. In this section, we will introduce the theoretical framework of path integral, including its formal definition and efficient computation.

3.1. Path integral as a structural descriptor

Consider a subgraph $G_{\mathcal{C}}=(V_{\mathcal{C}},E_{\mathcal{C}})$ with transition probability matrix $\mathbf{P}_{\mathcal{C}}$ corresponding to a cluster $\mathcal{C}$. The path integral of a cluster is defined as follows.

Definition 3.1. The *path integral* of a cluster $\mathcal{C}$ is

$$\mathcal{S}_{\mathcal{C}}=\frac{1}{|\mathcal{C}|^2}\sum_{\gamma\in\Gamma_{\mathcal{C}}}\Theta(\gamma), \tag{4}$$

where $\Gamma_{\mathcal{C}}$ is the set of all the paths in $\mathcal{C}$, and $\Theta(\gamma)$ is the contribution or weight of a path γ.

The name of *path integral* comes from quantum mechanics [21,22]. It was proposed as the functional integral on the path set in a general form. In our implementation, the quantity $\mathcal{S}_{\mathcal{C}}$ is discretized as the sum of weights over all the paths. But we still inherit the name to make it consistent. The path integral is a generalization of path counting, via considering the path-specific weights. If we divide the path set by selecting the starting and ending vertices of the path, we can rewrite the path integral as follows:

$$\mathcal{S}_{\mathcal{C}}=\frac{1}{|\mathcal{C}|^2}\sum_{i,j=1}^{|\mathcal{C}|}s_{ij}, \tag{5}$$

where s_{ij} is *unnormalized pairwise path integral*[2] over all the paths from i to j on $G_{\mathcal{C}}$. Generally speaking, the number of paths in a cluster $\mathcal{C}$ is proportional to $|\mathcal{C}|^2$. $\mathcal{S}_{\mathcal{C}}$ is normalized by being divided with $|\mathcal{C}|^2$, such that the clustering results based on path integral in later steps are not biased by cluster size; otherwise, the proposed algorithm prefers to merge large clusters.

3.2. Unnormalized fixed-length path integral

Given the starting vertex i and ending vertex j, the unnormalized fixed-length path integral is the simplest case for discussion. Let $\gamma=\{u_0\rightarrow u_1\rightarrow\cdots\rightarrow u_{k-1}\rightarrow u_k\}(u_0=i,u_k=j,u_1,\ldots,u_{k-1}\in V_{\mathcal{C}})$ denote any directed path of length k from i to j in $G_{\mathcal{C}}$, i.e., γ is a sequence of vertex indices from i to j and every two consecutive vertices u_s and u_{s+1} in γ are connected by an edge in the subgraph $G_{\mathcal{C}}$. We define the contribution of a path γ as

$$\Pr(\gamma)=p_{u_0u_1}p_{u_1u_2}\cdots p_{u_{k-1}u_k}, \tag{6}$$

i.e., the probability of the transition from i to j along path γ.

Definition 3.2. Given that $\Gamma_{ij}^{(k)}$ is the set of all the paths of length k from i to j on $G_{\mathcal{C}}$, the *unnormalized fixed-length path integral* over $\Gamma_{ij}^{(k)}$ is

$$s_{ij}^{(k)}=\sum_{\gamma\in\Gamma_{ij}^{(k)}}\Pr(\gamma)=\sum_{\gamma\in\Gamma_{ij}^{(k)}}\prod_{s=1}^{k}p_{u_{s-1},u_s}, \tag{7}$$

where $u_0=i$, $u_k=j$.

Remark. The value of $s_{ij}^{(k)}$ is also equal to the k-step transition probability from i to j, under an absorbing random walk model (refer to the details in Section 4).

3.3. Unnormalized pairwise path integral by integrating paths of different lengths

All the paths of possible lengths from 1 to ∞ play a role in the structural descriptor of a cluster. To integrate all the paths, we define the unnormalized pairwise path integral as the form of a generating function.

[2] When path integral was first proposed in quantum mechanics [21,22], it was normalized as shown in Eq. (4). Therefore we call s_{ij} defined in Eq. (8) as *unnormalized pairwise path integral*. Similarly, we call s_{ij}^{k} defined in Eq. (7) as *unnormalized fixed-length path integral*.

Definition 3.3. The *unnormalized pairwise path integral* over all the paths (of lengths from 1 to ∞) from i to j in $G_{\mathcal{C}}$ is defined as a generating function

$$s_{ij} = \delta_{ij} + \sum_{k=1}^{\infty} z^k s_{ij}^{(k)} = \delta_{ij} + \sum_{k=1}^{\infty} z^k \sum_{\gamma \in \Gamma_{ij}^{(k)}} \prod_{s=1}^{k} p_{u_{s-1},u_s}, \tag{8}$$

where $u_0 = i$, $u_k = j$, $0 < z < 1$ and δ_{ij} is the Kronecker delta function defined as $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

Remark. The unnormalized pairwise path integral integrates the unnormalized fixed-length path integrals of length from 1 to ∞, with weights controlled by z. The choice of $z < 1$ ensures that short paths are favored over long paths, because the vertices in a "good" cluster should be connected by many short paths.

3.4. Computing the path integral

We have the following theorem for efficient computation of the path integral.

Theorem 3.1. *s_{ij} always converges, and $s_{ij} = [(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1}]_{ij}$, i.e., the (i,j)-element of $(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1}$, where $\mathbf{P}_{\mathcal{C}}$ is the submatrix of $\mathbf{P}$ by selecting the samples in $\mathcal{C}$. If we define $\mathbf{S}_{\mathcal{C}} = [s_{ij}]_{i,j \in \mathcal{C}}$, we have $\mathbf{S}_{\mathcal{C}} = (\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1}$. Then, we can compute the path integral as the structural descriptor of cluster $\mathcal{C}$ as follows*:

$$\mathcal{S}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|^2} \mathbf{1}^T \mathbf{S}_{\mathcal{C}} \mathbf{1} = \frac{1}{|\mathcal{C}|^2} \mathbf{1}^T (\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1} \mathbf{1}, \tag{9}$$

where $\mathbf{1}$ is all-one column vector.

Proposition 3.2. *$(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})$ is a strictly diagonally dominant matrix with the ∞-norm condition number no more than $(1+z)/(1-z)$.*

Refer to Appendix A and Appendix B for the proofs of Theorem 3.1 and Proposition 3.2.

Efficient computation. Note that the inverse of the matrix $(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})$ *does not* need to be explicitly computed. The computation of $\mathcal{S}_{\mathcal{C}}$ only involves solving a linear system

$$(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})\mathbf{y} = \mathbf{1}, \tag{10}$$

and then

$$\mathcal{S}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|^2} \mathbf{1}^T \mathbf{y}. \tag{11}$$

For a large cluster, $(\mathbf{I} - z\mathbf{P}_{\mathcal{C}})$ is sparse.[3] As the sparse linear system has the nice property in Proposition 3.2 (empirically we choose a small z in experiments), it can be efficiently solved by iterative methods [27], with a complexity of $O(|\mathcal{C}|)$.

Incremental path integral. Given two clusters $\mathcal{C}_a$ and $\mathcal{C}_b$, their incremental path integral is computed from Eq. (3). Similar to Theorem 3.1, the conditional path integral in Eq. (3) is computed as

$$\mathcal{S}_{\mathcal{C}_a | \mathcal{C}_a \cup \mathcal{C}_b} = \frac{1}{|\mathcal{C}_a|^2} \mathbf{1}_{\mathcal{C}_a}^T (\mathbf{I} - z\mathbf{P}_{\mathcal{C}_a \cup \mathcal{C}_b})^{-1} \mathbf{1}_{\mathcal{C}_a}, \tag{12}$$

where $\mathbf{1}_{\mathcal{C}_a}$ is the vector in which the elements corresponding to the vertices in $\mathcal{C}_a$ are all one and other elements are zero.

Finding exemplars of clusters. When the agglomerative clustering stops, the exemplar of each cluster $\mathcal{C}$ can be found by selecting the sample i with the largest value of

$$\sum_{j \in \mathcal{C}} (s_{ji} + s_{ij}) = (\mathbf{1}_{\{i\}}^T (\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1} \mathbf{1} + \mathbf{1}^T (\mathbf{I} - z\mathbf{P}_{\mathcal{C}})^{-1} \mathbf{1}_{\{i\}}). \tag{13}$$

This quantity is the path integral on the paths from any vertex to i and from i to any vertex in $\mathcal{C}$. It reflects the vertex i's incoming and outgoing connections to samples in $\mathcal{C}$.

3.5. Discussions

Connection and comparison with diffusion kernels and connectivity kernels. s_{ij} in Eq. (8) can be viewed as the structural similarity between samples i and j if cluster $\mathcal{C}$ is equal to the whole dataset. This view brings the connection to the von Neumann kernel [28,29], which is one of the diffusion kernels [19] defined on the whole graph and has been successfully applied to computing similarities between vertices [20]. This kernel view has profound theoretical and practical significance, yet it is not the focus of this paper. We focus on a novel perspective of characterizing the structure of a cluster instead of similarities of samples. Note that our clustering algorithm, from a novel graph structural view of agglomerative clustering, is significantly different from directly using the similarities derived from the von Neumann kernel or any other path-based similarity [30,31] (such as the connectivity kernel [31]) in an existing clustering algorithm.

The difference exists in three aspects. Firstly, these methods first re-compute similarities of samples over graphs and then apply the refined similarities to an existing clustering algorithm. The strategy of splitting clustering into two steps could be suboptimal, while our approach directly compares the structural affinity between clusters without computing similarities of samples. Since the ultimate goal is to compute the affinity of clusters, there is no need to have an extra step of re-computing the similarities of samples. Notice that the objective of diffusion kernels or connectivity kernels is to optimize sample similarity instead of cluster affinity. Secondly, diffusion kernels and connectivity kernels compute sample similarities from the whole graph. When they are used to compare the affinity of two clusters, samples outside the clusters get involved. Our s_{ij} is computed on a single cluster. Considering the subgraphs of clusters is enough, since the affinity between clusters is mainly determined by local structures. Thirdly, in our efficient implementation, s_{ij} is actually not computed and the kernel matrix $\mathrm{S} = (\mathbf{I} - z\mathbf{P})^{-1}$ is dense. Instead, the path integral is directly obtained by efficiently solving a sparse linear system in Eqs. (10) and (11). Experimental results in Section 5 show that our approach outperforms both the diffusion kernel and the connectivity kernel.

Comparison with commute/hitting time based algorithms. The commute time [32] and hitting time [33] of a random walk have been used for clustering. However, they are all implemented in two steps: computing affinities between samples using commute time or hitting time, and then applying an existing clustering algorithm. As previously discussed, this strategy could be suboptimal. Our approach computes the affinity between clusters directly by measuring the structural changes, and our incremental path integral has significant difference with their affinities.

Comparison with zeta function of a graph. Cycles, i.e., self-connecting paths, were exploited in zeta function based clustering [34]. Since our path integral considers all the paths within clusters, it captures more information of cluster structures.

Deciding the number of clusters. For some clustering tasks in real-world applications, it is sometimes required to automatically determine the number of clusters from data. The accurate determination of cluster numbers is a difficult problem and also a specific research topic in the field of pattern clustering. For hierarchical agglomerative clustering, a commonly used idea is to build the complete hierarchy of clusters. Initialized by viewing each data point as a cluster, it merges samples to be clusters until all the samples merge as one cluster. Investigating the largest gaps between adjacent layers in the dendrogram rationally determines the number of clusters. For instance, dissimilarity increments are

[3] Graph G, which is built by K-nearest-neighbors, is not fully connected and only has a relatively small number of edges. Therefore, its transition probability matrix $\mathbf{P}$ is sparse according to Eq. (1). $\mathbf{P}_{\mathcal{C}}$ is the submatrix of $\mathbf{P}$ by selecting the samples in $\mathcal{C}$, and is also sparse.

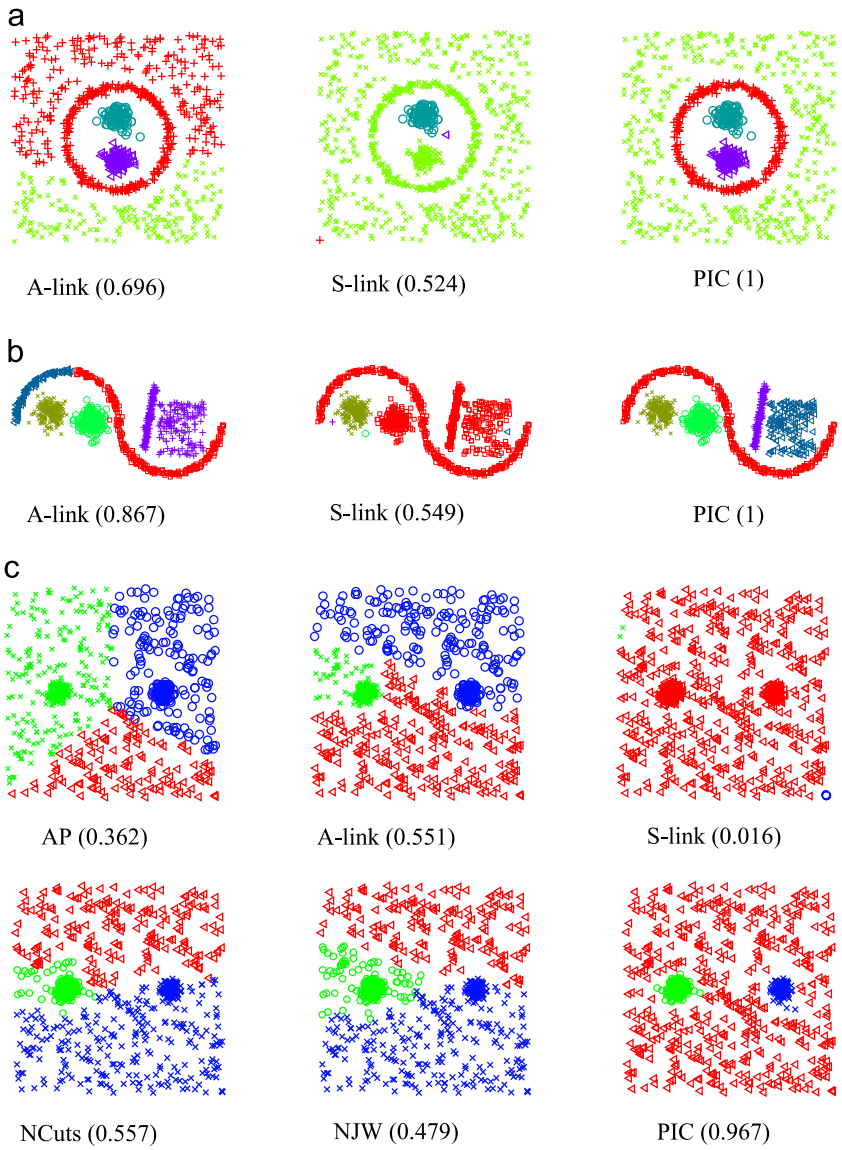

Fig. 3. Clustering results on three synthetic datasets (best viewed on screen) (a)–(c). The NMI results are shown in the brackets. The best values are in bold. (a) Synthetic dataset I, (b) synthetic dataset II, (c) synthetic dataset III.

applied in [35] and a variational cluster descriptor quantized by the zeta function of a graph is adopted in our previous work [34]. In complex networks, a quantizer called Q-function is frequently used to determine the number of communities (clusters) [36]. The Q-function is the difference between the number of edges within communities and the expected number of such edges. The number of clusters is specified at the maxima of Q-function. Interested readers may refer to these papers for further investigation. These techniques can be well applied to our approach to decide the number of clusters. They are not our contribution, and therefore not evaluated in this paper.

4. Absorbing random walk: a probabilistic view

An absorbing random walk is a special Markov chain which has absorbing states, i.e., states which once reached cannot be transitioned out of. It provides a probabilistic view of our algorithm. For a cluster $\mathcal{C}$, we construct an absorbing random walk by setting all the samples outside $\mathcal{C}$ as absorbing states, i.e., $p_{ii}=1$, $p_{ij}=0$, for all $i\notin\mathcal{C}$ and $j\neq i$.

Theorem 4.1. *Let Y_k be the state of the random walk at time k. Given that the random walk starts with a uniform distribution over states in $\mathcal{C}$, i.e., $\Pr(Y_0=i)=1/|\mathcal{C}|$, for all $i\in\mathcal{C}$, and $\Pr(Y_0=i)=0$, for all $i\notin\mathcal{C}$, we have*

$$\mathcal{S}_{\mathcal{C}}=\frac{1}{|\mathcal{C}|}\sum_{k=0}^{\infty}z^k\Pr(Y_k\in\mathcal{C}|Y_0\in\mathcal{C}). \tag{14}$$

See the proof in Appendix C. Here $\Pr(Y_k\in\mathcal{C}|Y_0\in\mathcal{C})$ is the probability of remaining within the cluster after k steps. The absorbing probability after k steps is

$$\Pr(Y_k\notin\mathcal{C}|Y_0\in\mathcal{C})=1-\Pr(Y_k\in\mathcal{C}|Y_0\in\mathcal{C}). \tag{15}$$

According to the description of Section 3, from the path integral point of view, a good cluster with many paths inside the cluster should maximize $\mathcal{S}_{\mathcal{C}}$. According to Theorem 4.1, we can understand it from another perspective that a good cluster should keep the state not to be easily absorbed by states outside the cluster. In this sense, if a cluster is treated as a dynamical system, $\mathcal{S}_{\mathcal{C}}$ measures its stability. The conditional path integral $\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}$ can be understood in the same way.

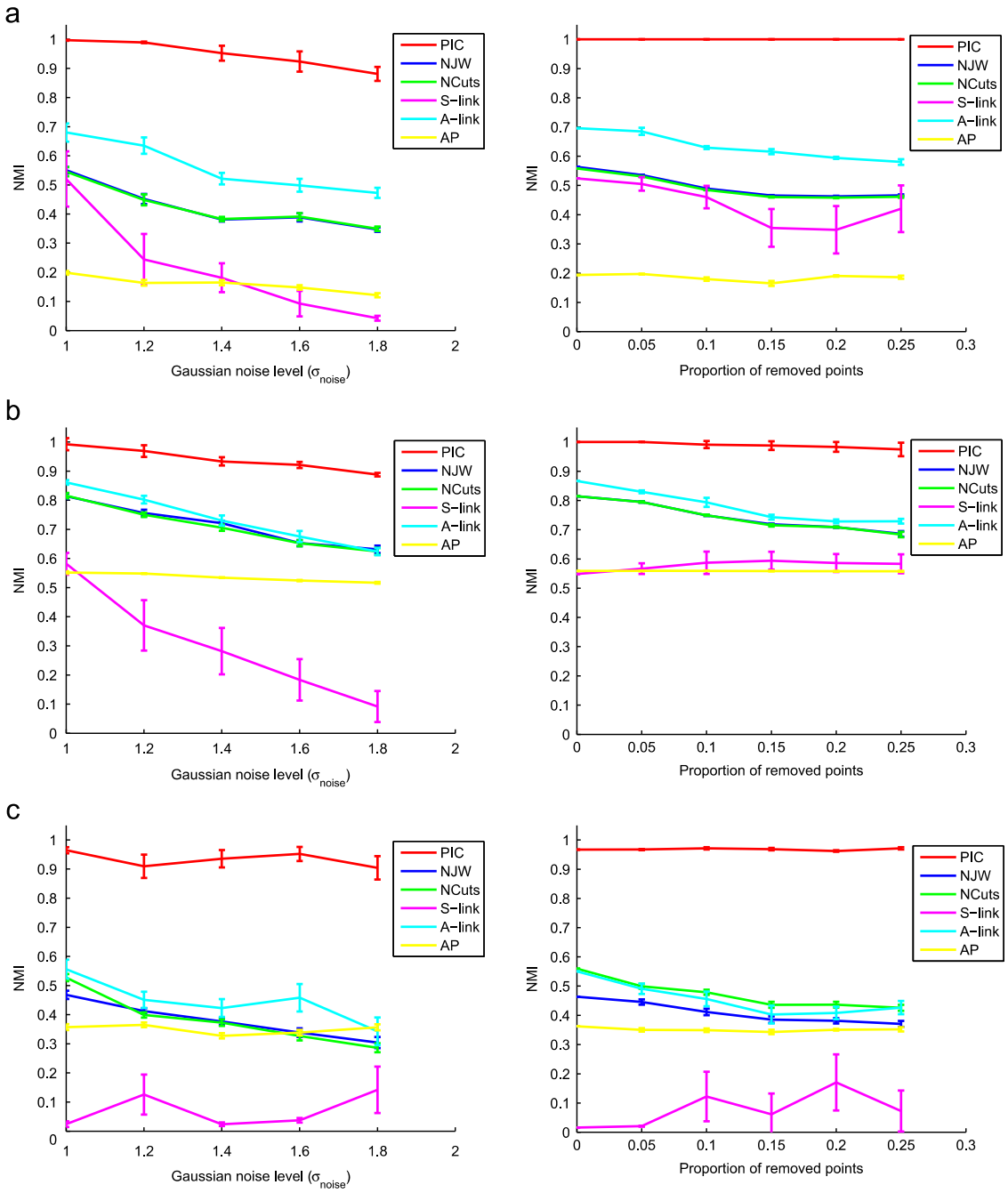

Fig. 4. NMI scores of clustering results on the synthetic datasets I–III after adding different types and different levels of random noise. For each noise level on each dataset, the experiments repeat for 20 times. The curves are the averages of NMI scores and the bars indicate standard deviations. (a1–c1) For Gaussian noise, the horizontal axis indicates that the standard deviations of Gaussian distributions range from σ_{noise} to $1.8\sigma_{\text{noise}}$, where σ_{noise} is the standard deviation of Gaussian noise on the original datasets in Fig. 3. (a2–c2) Different levels of structural noise are obtained by randomly removing different proportions (0–35%) of points from the original datasets in Fig. 3. (a1) Gaussian noise on dataset I, (a2) structural noise on dataset I, (b1) Gaussian noise on dataset II, (b2) structural noise on dataset II, (c1) Gaussian noise on dataset III, (c2) Structural noise on dataset III.

Then, to measure the affinity between $\mathcal{C}_a$ and $\mathcal{C}_b$, we have Eq. (3). The first part ($\mathcal{S}_{\mathcal{C}_a|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_a}$) measures the separability of $\mathcal{C}_b$ from $\mathcal{C}_a\cup\mathcal{C}_b$, by the increasing amount of absorbing probability of random walk started in $\mathcal{C}_a$, if we select the states in $\mathcal{C}_b$ from $\mathcal{C}_a\cup\mathcal{C}_b$ and set them to be absorbing states. Apparently, if $\mathcal{C}_a$ and $\mathcal{C}_b$ come from the same cluster, the increasing absorbing probability should be large. Similar analysis can be applied for the second part ($\mathcal{S}_{\mathcal{C}_b|\mathcal{C}_a\cup\mathcal{C}_b}-\mathcal{S}_{\mathcal{C}_b}$). This explains why Eq. (3) is a good affinity measure between clusters.

5. Experiments

We conduct experiments on toy data and multiple benchmark imagery datasets to evaluate the proposed Path Integral based Clustering (PIC) algorithm. Eleven representative algorithms are taken into comparison, i.e., k-medoids (k-med) [2], Average linkage (A-link), Single linkage (S-link), Complete linkage (C-link) [2], Affinity Propagation (AP) [9], Normalized Cuts (NCuts) [3],

Table 1
Statistics of imagery data benchmarks.

Dataset	USPS	MNIST	FRGC-T	PubFig	Caltech-256
No. of samples	11 000	5139	12 776	5803	600
No. of clusters	10	5	222	60	6
Min. cluster size	1100	980	36	62	100
Max. cluster size	1100	1135	64	100	100
Dimensionality	256	784	2891	2891	4200

NJW algorithm [5], Commute Time based clustering (CT) [32], Zeta function based clustering (Zell) [34], connectivity kernel based clustering (C-kernel) [31], and diffusion kernel based clustering (D-kernel). Here we use k-medoids instead of k-means because it can handle the case where distances between points are not measured by Euclidean distances. Although diffusion kernels [28,29,19,20] have been used to compute similarities between samples, we have not found any papers of directly using them for clustering. In D-kernel, we first use the von Neumann kernel [19] to compute the similarities of samples and then use the average linkage algorithm to cluster samples based on the similarities. For fair comparison, we run A-link, S-link, C-link, NCuts, NJW, CT, Zell, D-kernel, C-kernel and our algorithm on the graphs built by the same parameters, which are set as $z=0.01$, $a=0.95$ and $K=20$.

We adopt the widely used Normalized Mutual Information (NMI) [37] and Clustering Error (CE) [38] to quantitatively evaluate the performance of clustering algorithms. The NMI quantifies the normalized statistical information shared between two distributions. A larger NMI value indicates a better clustering result. The CE is defined as the minimum overall error rate among all possible permutation mappings between true class labels and clusters. A smaller CE value indicates a better clustering result.

5.1. On synthetic data

We first evaluate the algorithms on three synthetic datasets and the results are visualized in Fig. 3.[4] All the algorithms use the ground-truth cluster numbers as input. The two datasets in the top rows cannot be clustered in a meaningful way by methods that assume compact shapes of clusters, like k-medoids, AP, and C-link. A-link and S-link perform better, but suffer from noisy distances caused by perturbations. For the multi-scale dataset in the bottom row, S-link fails. NCuts and NJW do not work well for such multi-scale data either, even if the scale parameters K and a are exhaustively explored and the results with the best NMI are reported. PIC works very well on all these datasets, even simply using the default parameters. Note that we do not use any advanced graph construction techniques [25,26], such as using variable bandwidth. Surprisingly, PIC is not sensitive to the parameters for graph. When we vary K in the set $10\times\{1,2,\ldots,5\}$, and σ in the set $\sigma=\sigma_0\times 2^r$, $r\in\{-2.5,-2,\ldots,2,2.5\}$, where σ_0 corresponds to $a=0.95$, the clustering results are almost the same.

We also evaluate the performance of clustering algorithms under different types and different levels of noise. Fig. 4 shows the NMI scores of clustering results after adding Gaussian noise or structural noise to the synthetic datasets I–III in Fig. 3. For each noise level on each dataset, the experiments repeat for 20 times. The curves show the average NMI scores and bars show the standard deviations. In each original dataset in Fig. 3, their multiple structures, and data points belonging to each structure are perturbed with a Gaussian distribution. Different structures have different Gaussian noises. In Fig. 4, we increase the standard deviations of the original Gaussian noise by up to 1.8 times. The experimental results show that the performance of other clustering methods drops significantly when Gaussian noise increases and their standard deviations also increase. Our approach is much more stable under different levels of Gaussian noise. The structural noise is added by randomly removing a certain proportion of data points from the original datasets in Fig. 3, such that some cluster structures may be destroyed. Experimental results show that our method is much more stable with the existence of structural noise. The performance of other methods in comparison either decreases or shows large standard deviations, when a significant portion of points is randomly removed.

Table 2
Quantitative clustering results in NMI on imagery data. The best values are in bold.

Dataset	USPS	MNIST	FRGC-T	PubFig	Caltech-256
k-med	0.310	0.483	0.540	0.363	0.593
AP	0.313	0.451	0.600	0.398	0.509
A-link	0.688	0.845	0.727	0.573	0.721
S-link	0.013	0.012	0.292	0.067	0.045
C-link	0.029	0.022	0.241	0.135	0.069
NCuts	0.628	0.792	0.709	0.537	0.722
NJW	0.619	0.795	0.723	0.553	0.722
CT	0.646	0.831	0.726	0.555	0.732
Zell	0.772	0.865	0.670	0.429	0.710
C-kernel	0.661	0.916	0.727	0.570	0.735
D-kernel	0.623	0.804	0.716	0.562	0.704
PIC	**0.825**	**0.940**	**0.747**	**0.602**	**0.761**

Table 3
Quantitative clustering results in CE on imagery data. The best values are in bold.

Dataset	USPS	MNIST	FRGC-T	PubFig	Caltech-256
k-med	0.661	0.324	0.712	0.723	0.258
AP	0.623	0.382	0.649	0.680	0.392
A-link	0.594	0.205	0.648	0.548	0.173
S-link	0.900	0.779	0.904	0.976	0.828
C-link	0.899	0.778	0.963	0.967	0.827
NCuts	0.433	0.122	0.565	0.538	0.163
NJW	0.419	0.120	0.585	0.538	0.163
CT	0.407	0.101	0.578	0.558	0.157
Zell	0.412	0.206	0.565	0.762	0.277
C-kernel	0.414	0.025	0.596	0.528	0.158
D-kernel	0.444	0.281	0.640	0.558	0.558
PIC	**0.246**	**0.016**	**0.560**	**0.504**	**0.153**

5.2. On imagery data

We carry out experiments on five real image datasets: handwritten digit images from MNIST and USPS databases,[5] cropped facial images from FRGC ver2.0 [39] and PubFig databases [40], and object images from Caltech-256 database [41]. For MNIST, we select all the images of digits from 0 to 4 in the testing set. For FRGC ver2.0, we use all the facial images in the training set of experiment 4. For PubFig, we use all the people in the development set. We collect the first 100 images of each person, if the person has more than 100 images. Otherwise, we collect all the images of the person. For Caltech-256, we use six categories (hibiscus, ketch, leopards, motorbikes, airplanes, faces-easy), and select the first 100 images in each category for experiments. For the other datasets, we use all the images. For digits, we use the intensities of pixels as features and Euclidean distance. For facial

[4] Because of space limit, only some algorithms are selected.

[5] Both are downloaded from http://www.cs.nyu.edu/~roweis/data.html.

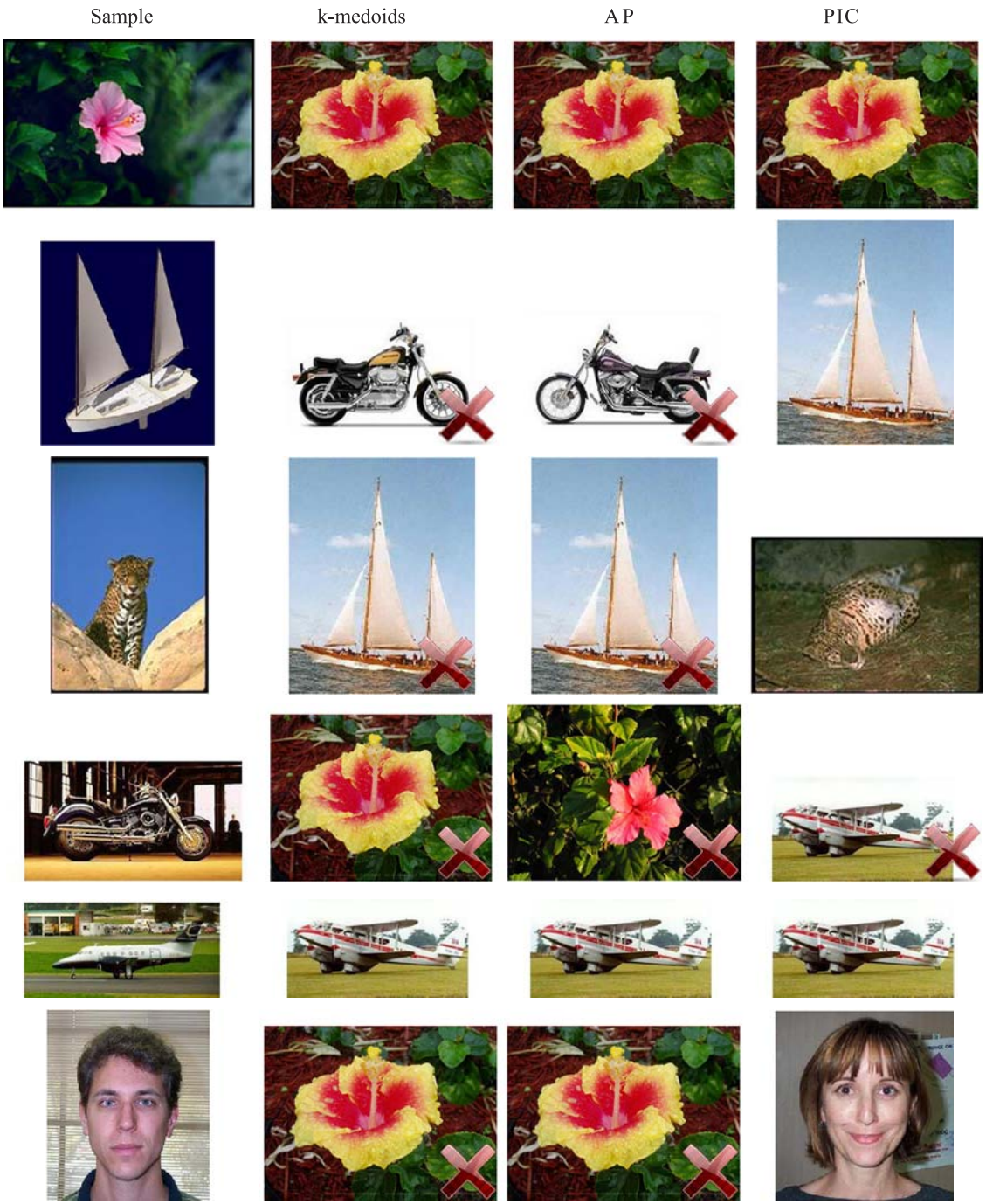

Fig. 5. The comparisons of detected exemplars on the Caltech-256 set. The first column is the most difficult sample for clustering in each category, and the other columns are exemplars of the sample's clusters given by *k*-medoids, AP and PIC. Incorrect exemplars are marked with a cross.

images, we use the local binary patterns as features [42] and χ^2 distance. For object images, we use the spatial pyramid features [43] and χ^2 distance. The statistics of the datasets used in our experiments are summarized in Table 1. The last three sets are extremely challenging for the clustering task. The faces in the FRGC-T set have large lighting and expression variations, and some faces are blurred. The PubFig dataset consists of uncontrolled real-world faces collected from the internet. The images in the Caltech-256 set have large intra-category variations.

The quantitative results, measured in NMI and CE, are given in Tables 2 and 3, respectively. As *k*-medoids is sensitive to initialization, we select the result with the smallest intra-cluster variations among 1000 random runs, and thus its performance is comparable with AP's.[6] S-link and C-link do not perform well on most real datasets, while A-link performs better. This is due to large intra-cluster variations and the complex cluster structures in real datasets. NCuts, NJW, CT, Zell, and C-kernel have good performance on most datasets. Our PIC performs the best among all the algorithms.

To visually compare the algorithms, we use the exemplar of a cluster to show the clustering result of a given sample. If a sample is in

[6] The input parameter of AP is a preference value. Therefore, we search for an appropriate preference value, so that the number of output clusters is equal to the number of ground-truth clusters.

an incorrect cluster, the exemplar should not be correct. The results of k-medoids, AP and PIC on the Caltech-256 set are shown in Fig. 5. Our PIC algorithm can detect the exemplar of each cluster as introduced in Section 3.4, while NCuts, NJW, CT, and Zell cannot. The samples are selected as follows: for each category, we search the sample with the largest average distance to the other samples in the same category (i.e., the most difficult sample for clustering in each category).

6. Conclusion

In this paper, we propose a novel graph-structural agglomerative clustering approach using path integral as the structural descriptor of clusters and incremental path integral as the affinity measurement of clusters. The incremental path integral measures the structural change of clusters after the merging and its closed-form exact solution can be efficiently computed in a linear time complexity. A probabilistic view of our algorithm from absorbing random walk is provided. Extensive experimental comparisons show that the new algorithm outperforms the state-of-the-art clustering methods. The success of this new graph-structural agglomerative framework inspires us to find more effective cluster descriptors in the future work.

Conflict of interest statement

None declared.

Acknowledgments

This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project No. CUHK417110 and CUHK417011) and National Natural Science Foundation of China (Project no. 61005057).

Appendix A. Proof of Theorem 3.1

Proof. By matrix computations, the (i,j)-element of $\mathbf{P}_{\mathcal{C}}^k$ is

$$[\mathbf{P}_{\mathcal{C}}^k]_{ij} = \sum_{\gamma \in \Gamma_{ij}^{(k)}} \prod_{s=1}^{k} p_{u_{s-1},u_s}.$$

By Definition 3.3, we have

$$s_{ij} = \delta_{ij} + \sum_{k=1}^{\infty} z^k \sum_{\gamma \in \Gamma_{ij}^{(k)}} \prod_{s=1}^{k} p_{u_{s-1},u_s} = \delta_{ij} + \sum_{k=1}^{\infty} z^k [\mathbf{P}_{\mathcal{C}}^k]_{ij}$$

$$= \left[\mathbf{I} + \sum_{k=1}^{\infty} z^k \mathbf{P}_{\mathcal{C}}^k\right]_{ij} = [(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}]_{ij}$$

By Gershgorin disk theorem [24], the spectral radius of $\mathbf{P}_{\mathcal{C}}$ has an upper limit

$$\rho(\mathbf{P}_{\mathcal{C}}) \le \max_{i\in\mathcal{C}} \sum_{j\in\mathcal{C}} |p_{ij}| \le 1.$$

So, $\rho(z\mathbf{P}_{\mathcal{C}}) < 1$, which guarantees that the series $\mathbf{I} + \sum_{k=1}^{\infty} z^k \mathbf{P}_{\mathcal{C}}^k$ converges. □

Appendix B. Proof of Proposition 3.2

Proof. $\forall i \in \mathcal{C}$, we have $\sum_{j\in\mathcal{C}} |p_{ij}| \le 1$. Since $z < 1$, $\sum_{j\in\mathcal{C}} |zp_{ij}| < 1$ and thus $|1-zp_{ii}| > \sum_{j\ne i, j\in\mathcal{C}} |zp_{ij}|$, i.e., $(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})$ is a strictly diagonally dominant matrix.

The induced ∞-norm of $(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})$ is given by

$$\|(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})\|_\infty = \max_{i\in\mathcal{C}} \sum_{j\in\mathcal{C}} |\delta_{ij} - zp_{ij}| \le \max_{i\in\mathcal{C}} \sum_{j\in\mathcal{C}} \delta_{ij} + zp_{ij} \le 1 + z.$$

The ∞-norm of $(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}$ is given by

$$\|(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}\|_\infty = \|(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}\mathbf{1}\|_\infty = \left\|\left(\sum_{k=0}^{\infty} z^k \mathbf{P}_{\mathcal{C}}^k\right)\mathbf{1}\right\|_\infty$$

$$\le \sum_{k=0}^{\infty} z^k \|\mathbf{P}_{\mathcal{C}}^k \mathbf{1}\|_\infty \le \sum_{k=0}^{\infty} z^k = \frac{1}{1-z}.$$

So, we have the condition number

$$\kappa(\mathbf{I}-z\mathbf{P}_{\mathcal{C}}) = \|(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})\|_\infty \|(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}\|_\infty \le \frac{1+z}{1-z}. \quad \square$$

Appendix C. Proof of Theorem 4.1

Proof. Without loss of generality, the index of the vertices are permuted so that the transition probability matrix is partitioned as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{C}} & \mathbf{P}_{\mathcal{C},\overline{\mathcal{C}}} \\ \mathbf{P}_{\overline{\mathcal{C}},\mathcal{C}} & \mathbf{P}_{\overline{\mathcal{C}}} \end{bmatrix},$$

where $\mathbf{P}_{\mathcal{C},\overline{\mathcal{C}}}$ is the transition probabilities from the vertices in $\mathcal{C}$ to $\overline{\mathcal{C}}$.

The absorbing random walk has the transition probability matrix

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P}_{\mathcal{C}} & \mathbf{P}_{\mathcal{C},\overline{\mathcal{C}}} \\ \mathbf{0} & \mathbf{P}_{\overline{\mathcal{C}}} \end{bmatrix}$$

From (9), we can see that

$$\mathcal{S}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|^2}\mathbf{1}^T(\mathbf{I}-z\mathbf{P}_{\mathcal{C}})^{-1}\mathbf{1} = \frac{1}{|\mathcal{C}|^2}\mathbf{1}^T\left(\sum_{k=0}^{\infty} z^k\mathbf{P}_{\mathcal{C}}^k\right)\mathbf{1} = \frac{1}{|\mathcal{C}|^2}\mathbf{1}_{\mathcal{C}}^T\left(\sum_{k=0}^{\infty} z^k(\mathbf{P}')^k\right)\mathbf{1}_{\mathcal{C}}$$

$$= \frac{1}{|\mathcal{C}|}\sum_{k=0}^{\infty} z^k \sum_{i\in\mathcal{C}, j\in\mathcal{C}} \Pr(Y_0 = i)\Pr(Y_k = j \,|\, Y_0 = i)$$

$$= \frac{1}{|\mathcal{C}|}\sum_{k=0}^{\infty} z^k \Pr(Y_k \in \mathcal{C} \,|\, Y_0 \in \mathcal{C}). \quad \square$$

References

[1] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[2] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning: data mining, inference, and prediction, 2nd ed., Springer Verlag, 2009.
[3] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[4] M. Meila, J. Shi, A random walks view of spectral segmentation, in: AISTATS, 2001.
[5] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2001.
[6] S. Yu, J. Shi, Multiclass spectral clustering, in: Proceedings of International Conference on Computer Vision, 2003.
[7] F. Bach, M. Jordan, Learning spectral clustering, in: Advances in Neural Information Processing Systems, 2004.
[8] U. Von Luxburg, M. Belkin, O. Bousquet, Consistency of spectral clustering, The Annals of Statistics 36 (2) (2008) 555–586.
[9] B. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[10] Y. Ma, H. Derksen, W. Hong, J. Wright, Segmentation of multivariate mixed data via lossy data coding and compression, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (9) (2007) 1546–1562.
[11] K. Heller, Z. Ghahramani, Bayesian hierarchical clustering, in: International Conference on Machine Learning, 2005.
[12] L. Grady, E. Schwartz, Isoperimetric graph partitioning for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (3) (2006) 469–475.
[13] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 167–172.
[14] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: International Conference on Machine Learning, 2003.
[15] D. Zhou, O. Bousquet, T. Lai, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, 2004.
[16] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[17] F. Bavaud, G. Guex, Interpolating between random walks and shortest paths: a path functional approach, arXiv:1207.1253.
[18] M. Saerens, Y. Achbany, F. Fouss, L. Yen, Randomized shortest-path problems: two related models, Neural Computation 21 (2009) 2363–2404.
[19] R. Kondor, J. Lafferty, Diffusion kernels on graphs and other discrete input spaces, in: International Conference on Machine Learning, 2002.
[20] J. Kandola, J. Shawe-taylor, N. Cristianini, Learning semantic similarity, in: Advances in Neural Information Processing Systems, 2003.
[21] R. P. Feynman, Space-time approach to non-relativistic quantum mechanics, Review of Modern Physics 20 (367–387) 1948.
[22] H. Kleinert, Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets, 3rd ed., World Scientific, 2004.
[23] J. Rudnick, G. Gaspari, Elements of the Random Walk: An Introduction for Advanced Students and Researchers, Cambridge University Press, 2004.
[24] R. Horn, C. Johnson, Matrix Analysis, Cambridge University Press, 2005.
[25] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in Neural Information Processing Systems, 2005.
[26] B. Nadler, M. Galun, Fundamental limitations of spectral clustering, in: Advances in Neural Information Processing Systems, 2007.
[27] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Pub. Co., 1996.
[28] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[29] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM 46 (1999) 604–632.
[30] B. Fischer, J. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (4) (2003) 513–518.
[31] B. Fischer, V. Roth, J. Buhmann, Clustering with the connectivity kernel, in: Advances in Neural Information Processing Systems, 2004.
[32] H. Qiu, E. Hancock, Clustering and embedding using commute times, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (11) (2007) 1873–1890.
[33] M. Chen, J. Liu, X. Tang, Clustering via random walk hitting time on directed graphs, in: Proceedings of 23rd AAAI Conference on Artificial Intelligence, 2008.
[34] D. Zhao, X. Tang, Cyclizing clusters via zeta function of a graph, in: Advances in Neural Information Processing Systems, 2008.
[35] A. Fred, J. Leitao, A new cluster isolation criterion based on dissimilarity increments, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 944–958.
[36] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical Review E 74.
[37] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2003) 583–617.
[38] M. Wu, B. Schölkopf, A local learning approach for clustering, in: Advances in Neural Information Processing Systems, 2007.
[39] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[40] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: Proceedings of International Conference on Computer Vision, 2009.
[41] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Technical Report 7694, Caltech, 2007.
[42] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 2037–2041.
[43] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2006.

Wei Zhang received the B.E. degree in electronic engineering from the Tsinghua University, Beijing, in 2007, and the M.Phil. degree and Ph.D. degree in information engineering in 2009 and 2012, respectively, from the Chinese University of Hong Kong. His research interests include machine learning, computer vision, and image processing.

Deli Zhao received the M.S. degree in electronic information and electrical engineering from the Shanghai Jiao Tong University. After graduation, he worked in Microsoft research Asia for one and a half year. He is currently a research assistant in the Department of Information Engineering, Chinese University of Hong Kong. His research interests included dimensionality reduction, face recognition, pattern clustering, data mining, and complex network.

Xiaogang Wang (**S'03-M'10**) received the B.S. degree from the Special Class for Gifted Young at University of Science and Technology of China in Electrical Engineering and Information Science in 2001, and the M.Phil. degree from Chinese University of Hong Kong in Information Engineering in 2004. He received the Ph.D. degree in Computer Science from the Massachusetts Institute of Technology. He is currently an assistant professor in the Department of Electronic Engineering at the Chinese University of Hong Kong. He was the area chair of IEEE International Conference on Computer Vision (ICCV) 2011. He is the associate editor of the Image and Visual Computing Journal. His research interests include computer vision and machine learning.