Isoperimetric Cut on a Directed Graph

Mo Chen[1], Ming Liu[1], Jianzhuang Liu[1,2], and Xiaoou Tang[1,2]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
{cm007,mliu6,jzliu,xtang}@ie.cuhk.edu.hk

Abstract

In this paper, we propose a novel probabilistic view of the spectral clustering algorithm. In our framework, the spectral clustering algorithm can be viewed as assigning class labels to samples to minimize the Bayes classification error rate by using a kernel density estimator (KDE). From this perspective, we propose to construct directed graphs using variable bandwidth KDEs. Such a variable bandwidth KDE based directed graph has the advantage that it encodes the local density information of the data in the graph edge weights. In order to cluster vertices of the directed graph, we develop a directed graph partitioning algorithm which optimizes a random walk isoperimetric ratio. The partitioning result can be obtained efficiently by solving a system of linear equations. We have applied our algorithm to several benchmark data sets and obtained promising results.

1. Introduction

Due to the success of spectral clustering methods [13, 16, 19, 20, 11], graph based clustering algorithms are of great interests recently. These methods first compute the pairwise similarities of the data to construct an undirected graph. Then the clustering result of the data is obtained by partitioning the vertices of the graph into disjoint sets. One advantage of these methods is that they do not make strong assumptions about the distribution of the data. Therefore, they can potentially deal with data of irregular shapes.

Despite the success of the graph based clustering methods, there are still some unsolved issues, such as how to compute edge weights to reflect the underlying data distribution and how to select the free parameters. It has been noticed that the fixed bandwidth Gaussian kernel based spectral clustering algorithms cannot achieve satisfactory results on many data sets [15]. As pointed by the authors, these algorithms cannot deal with multi-scale data sets well even with carefully tuned parameters. How to construct the graph remains an art which is still based on heuristics [21].

In this paper, we first present a novel probabilistic view of the spectral clustering algorithm. We show that the Gaussian kernel based spectral clustering algorithm can be seen as assigning data samples to disjoint classes to minimize the kernel density estimator (KDE) based Bayes classification error rate. From this viewpoint, we can see that in order to obtain a good clustering result, one should construct a graph reflecting the underlying density of the data. Therefore, we propose to construct a graph by using a variable bandwidth KDE which naturally results in a directed graph. The digraph effectively explores the local density of the data.

A directed graph partitioning algorithm called random walk isoperimetric cut (RWICut) is proposed to cut a directed graph into two disjoint parts by minimizing a random walk isoperimetric ratio. This random walk isoperimetric constant generalizes the isoperimetric constant of an undirected graph to the state space of the Markov chain. By adopting the random walk view, we can handle both the directed and undirected graph partitioning problems in a unified framework. Finding the exact solution of this combinatorial problem is NP-hard. However, an approximate solution can be efficiently achieved by solving a sparse linear system of equations. Given a data set, the clustering result is then obtained by iteratively cutting the constructed digraph into disconnected subgraphs.

The rest of the paper is organized as follows. In Section 2, we briefly review the spectral clustering algorithm (SC) and KDE. Then we show the connections between KDE, SC and the Bayes classification error rate. Section 3 describes the framework of the proposed KDE digraph based random walk isoperimetric cut approach. Section 4 presents our experimental results. Section 5 concludes this paper.

2. Analysis of Spectral Clustering

In this section, we revisit the isoperimetric constant (for a manifold and an undirected graph), the spectral clustering algorithm, and the kernel density estimator. Then we develop a novel probabilistic view of the spectral clustering algorithm from the nonparametric density estimation perspective.

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

2.1. The isoperimetric constant on manifolds

The isoperimetric constant is originally defined by Cheeger [2] in Riemannian geometry. Let $\mathcal{M}$ be a d-dimensional closed Riemannian manifold, $\mathrm{Vol}(S)$ be the volume of a d-dimensional submanifold S, and $\mathrm{Vol}(\partial S)$ be the volume of the boundary ∂S, which is a $(d-1)$-dimensional submanifold. The Cheeger isoperimetric constant of $\mathcal{M}$ is defined as

$$h(\mathcal{M}) = \inf_S \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(S)}. \tag{1}$$

Intuitively, the Cheeger isoperimetric constant defines the small bottleneck boundary of the manifold which separates the manifold into two parts with large volumes.

In the application of data clustering, we assume that the data are sampled from an underlying distribution with the probability measure $P(x)$ from $\mathcal{M}$. Then we have

$$\mathrm{Vol}(S) = \int_S dP(x) = \int_S p(x)dx, \tag{2}$$

where $P(x)$ is a probability measure on $\mathcal{M}$ satisfying $\int_{\mathcal{M}} dP(x) = 1$, and $p(x)$ is the corresponding probability density function.

2.2. Isoperimetric problem on undirected graphs

In the context of an undirected graph $G = (V, E)$, let S be a subset of the vertex set V. The boundary of S is defined as an edge set $\partial S = \{e_{ij} | i \in S, j \in \overline{S}\}$. Then the isoperimetric constant $h(G)$ is [14]

$$h(G) = \min_S \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(S)}, \tag{3}$$

where $\mathrm{Vol}(\partial S) = \sum_{i\in S, j\in\overline{S}} w_{ij}$, $\mathrm{Vol}(S) = \sum_{i\in S, j\in V} w_{ij}$, $\mathrm{Vol}(S) \leq \mathrm{Vol}(V)/2$, and w_{ij} is the weight of edge e_{ij} computed from the sample pair x_i and x_j by $w_{ij} = \exp(-\beta\|x_i - x_j\|^2)$.

The isoperimetric constant of an undirected graph satisfies $h(G) \in [0, 1]$, and is strictly positive iff the graph is connected. In [8, 9], the authors propose an algorithm that minimizes the graph isoperimetric constant to solve an image segmentation problem.

The spectral clustering algorithm proposed in [19, 11] is a graph bi-partitioning algorithm which minimizes the normalized cut (NCut) criterion

$$NCut(S) = \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(S)} + \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(\overline{S})}. \tag{4}$$

As shown in [11, 5], the algorithm also minimizes the upper bound of the graph isoperimetric constant.

2.3. Kernel density estimators

In statistics, a KDE (also called Parzen window) is a non-parametric way of estimating the probability density function of a random variable. It is given by

$$f_h(x) = \frac{1}{nh}\sum_{j=1}^{n} K(\frac{x - x_j}{h}), \tag{5}$$

where K is a kernel function, and h is the bandwidth (smoothing parameter) depending only on the sample size. To assure the convergence of the estimator $f_h(x)$, one imposes the conditions: $h \to 0$ and $nh \to \infty$ when $n \to \infty$. A widely used kernel is the Gaussian kernel

$$K(\frac{x - y}{h}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{\|x - y\|^2}{2h^2}). \tag{6}$$

If the Gaussian kernel is used, the KDE becomes

$$p(x) = \frac{1}{nh\sqrt{2\pi}}\sum_{j=1}^{n} \exp(-\frac{\|x - x_j\|^2}{2h^2}). \tag{7}$$

2.4. A Bayes error view of spectral clustering

Here we analyze the spectral clustering algorithm from a classification perspective, which gives the algorithm a probabilistic interpretation.

Given a set of vectors $\{x_i\}_{i=1}^n$ sampled from an underlying manifold $\mathcal{M}$, let V be the index set of the vectors. Also let S_1 and S_2 be the index sets of two disjoint subsets of the samples with class labels c_1 and c_2 which satisfy $S_1 \bigcup S_2 = V$, and T_1 and T_2 be the disjoint submanifolds enclosing the samples of S_1 and S_2, respectively, which satisfy $T_1 \bigcup T_2 = \mathcal{M}$.

The density function for the component c_l, $l = 1, 2$, is estimated by the KDE as

$$p(x|c_l) = \frac{1}{|S_l|h\sqrt{2\pi}}\sum_{j\in S_l} \exp(-\frac{\|x - x_j\|^2}{2h^2}). \tag{8}$$

Then the Bayes error [7] for the two-class classification problem is given by

$$P(err) = \int_{T_1} p(x|c_2)p(c_2)dx + \int_{T_2} p(x|c_1)p(c_1)dx. \tag{9}$$

By assuming the equal prior $p(c_1) = p(c_2) = 1/2$ and replacing the integrals with empirical summations over the samples, the Bayes error can be approximated by

$$P(err) \approx \frac{1}{2}\left(\frac{\sum_{i\in S_1} p(x_i|c_2)}{\sum_{i=1}^n p(x_i|c_2)} + \frac{\sum_{i\in S_2} p(x_i|c_1)}{\sum_{i=1}^n p(x_i|c_1)}\right). \tag{10}$$

Replacing the conditional density with the KDE in (8) and setting $w_{ij} = \exp(-\frac{\|x_i - x_j\|^2}{2h^2})$, we have

$$P(err) \approx \frac{1}{2}\left(\frac{\sum_{i\in S_1, j\in S_2} w_{ij}}{\sum_{i=1}^{n}\sum_{j\in S_2} w_{ij}} + \frac{\sum_{i\in S_1, j\in S_2} w_{ij}}{\sum_{i=1}^{n}\sum_{j\in S_1} w_{ij}}\right) \propto NCut(S). \tag{11}$$

From (11) we can see that the NCut spectral clustering algorithm can be viewed as finding the optimal partition of the data that minimizes the approximated Bayes classification error rate.

It is worth noting that the above analysis does not depend on the choice of the KDE. However, the analysis does suggest that in order to obtain a good clustering result, one should construct the underlying graph to reflect the data distribution. Therefore, in the next section, we propose to construct the graph using a variable bandwidth KDE, which naturally results in a directed graph.

3. Random Walk Isoperimetric Cut

In this section, we first introduce variable bandwidth KDE based directed graph construction methods. Then we propose the random walk isoperimetric cut algorithm to partition the vertices of the graph into disjoint subsets. We also analyze our algorithm from different viewpoints.

3.1. Local scaling directed graph construction

The first step of spectral graph clustering methods is to construct a graph from a vector data set. The edge weights are usually computed by the Gaussian kernel $\exp(-\|x_i - x_j\|^2/(2\sigma^2))$[1]. However, as indicated in [15], for certain data sets, say, multi-scale data, the Gaussian kernel with a single uniform scaling parameter σ is not informative enough for modeling the pairwise relations. The constructed graph is not able to capture the underlying data distribution. As a result, the intrinsic clusters of the data may not be obtained by partitioning the graph.

Instead of selecting a single scaling parameter, several papers suggest to compute the edge weights by incorporating local information in various ways. The authors of [3] propose to estimate local Gaussian distributions to construct a directed probabilistic graph. The authors of [22] construct the graph using coding length. However, these methods are very time consuming which limits their practical use. The authors of [21] suggest to replace the uniform σ^2 of the Gaussian kernel with a location dependent scale $\sigma(x_i)\sigma(x_j)$, but they do not provide a principal justification why the edges should be constructed this way.

[1]Without ambiguity, in this section, x or x_i is used to denote a sample vector.

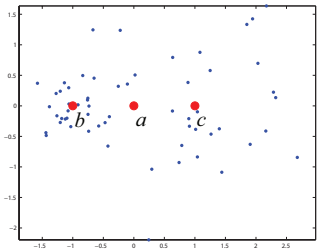

Figure 1. The local density affects the relationships between sample pairs.

Consider the data shown in Figure 1. The Euclidean distances $d(a, b)$ and $d(a, c)$ are equal. Then the similarities computed by the fixed bandwidth Gaussian kernel are the same. However with the context data points around sample a, apparently a is more likely to belong to the same cluster as b than as c. Here the local density of the data distribution is important for modeling the relationships between the sample pairs. This intuitive example motivates us to use a location dependent method to construct the graph.

In this paper, we incorporate the nonparametric density estimation view to use a variable bandwidth KDE to construct the graph from a data set. The variable bandwidth KDE is given by

$$f_b(x) = \sum_{j=1}^{n} \frac{1}{nh_j} K(\frac{x - x_j}{h_j}), \tag{12}$$

where the bandwidth h_j depends on the context information of x_j. It is well known that with a fixed bandwidth, the kernel estimate tends to oversmooth at the main part and undersmooth at the tail part of the distribution. This is the basic motivation for considering a variable bandwidth KDE, which allows the bandwidth to vary from one observation to another. It gives the flexibility of using a smaller bandwidth (hence reduces the bias of the estimate) in regions where there are many observations, and a larger bandwidth (hence reduces the variance of the estimate) in regions where there are relatively few observations.

In this paper, the local bandwidth h_j is set to be the distance between x_j and its kth nearest neighbor. The parameter k is selected by cross validation. The k with which the variable bandwidth KDE (12) has the largest leave-one-out likelihood on the given data set is used in our algorithm. According to the analysis in Section 2.4, by using the variable bandwidth KDE, the edge weight of the constructed graph is $w_{ij} = \frac{1}{h_i} K(\frac{x_i - x_j}{h_i})$. We use the Gaussian kernel for the KDE in this paper. Therefore the edge weight of the constructed graph between x_i and x_j is

$$w_{ij} = \frac{1}{h_i} \exp(-\frac{\|x_i - x_j\|^2}{2h_i^2}). \tag{13}$$

Notice that, in general, w_{ij} is not necessarily equal to w_{ji}. Therefore, the constructed graph is a directed graph.

3.2. Random walk on a directed graph

After having the directed graph, in order to obtain a clustering result, we partition the graph into disjoint subgraphs. Random walk is a powerful tool for dealing with graph structured data. In [13], the authors utilize random walk to analyze the normalized cut algorithm. In [4], the authors propose to use random walk to embed directed graphs into vector spaces. In this paper, we propose a random walk based graph cut algorithm which can deal with undirected and directed graphs in a unified way.

A directed graph $G = (V, E)$ consists of a finite set of vertices $v \in V$ together with a subset of edges $e \in E \subseteq V \times V$. An edge e_{ij} of the directed graph is an ordered pair from vertex i to vertex j associated with the edge weight w_{ij}. The degree of vertex i is $d_i = \sum_j w_{ij}$.

For a given weighted directed graph, there is a natural random walk on the graph with the one step transition probability from i to its adjacent j defined as $p_{ij} = w_{ij}/d_i$. For all sample pairs, we have the stochastic matrix $P = [p_{ij}]_{ij},\ i, j = 1, \cdots, |V|$ satisfying $P\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a vector with all entries being 1. The unique stationary distribution $\pi = [\pi_i]_i,\ i = 1, \cdots, |V|$, of the Markov chain is guaranteed if P is irreducible or equivalently the graph associated with P is strongly connected and aperiodic [1]. The stationary distribution vector, also called Pagerank vector in information retrieval literatures, satisfies $\pi^T P = \pi^T$. Then it can be obtained by solving the linear system subject to the normalization $\pi^T \mathbf{1} = 1$.

The Perron-Frobenius Theorem [10] states that there exists an unique left eigenvector ϕ with all entries positive such that $\phi^T P = \rho\phi^T$ where ρ is the spectral radius of P . When P is irreducible, the spectral radius of P is 1. In this case, ϕ simply equals the stationary distribution π up to a constant factor. Therefore, the stationary distribution can also be computed by an eigen-decomposition algorithm or a power algorithm [12]. In the following parts of this section, we assume that P is irreducible. How to enforce the irreducibility will be discussed in Section 3.6.

3.3. Random walk isoperimetric constant

In this paper, we generalize the isoperimetric problem into the scenario of random walk process. As a result, we can treat the isoperimetric problem of undirected and directed graphs in a unified way. The isoperimetric problem of an undirected graph can be considered as a special case of the random walk isoperimetric problem.

For a finite state irreducible Markov chain on a graph with the transition probability matrix P, define the volume of the boundary of the vertex (state) set S as the sum of the weighted transition probabilities: $\mathrm{Vol}(\partial S) = \sum_{i\in S, j\in\overline{S}} \pi_i p_{ij}$. $\mathrm{Vol}(\partial S)$ is also the probability with which a random walker jumps from S to its complement set $\overline{S}$. It can be shown that the volume of the boundary is symmetric.

Theorem 1. $\mathrm{Vol}(\partial S) = \mathrm{Vol}(\partial\overline{S})$.

Proof. It is easy to see that $\sum_{i=1}^n \pi_i p_{ij} = \pi_j = \sum_{i=1}^n \pi_j p_{ji}$. Then we have

$$\begin{aligned}\mathrm{Vol}(\partial\overline{S}) &= \sum_{i\in\overline{S}, j\in S} \pi_i p_{ij} = \sum_{j\in S}\sum_{i=1}^{n} \pi_i p_{ij} - \sum_{j\in S}\sum_{i\in S} \pi_i p_{ij} \\ &= \sum_{j\in S}\sum_{i=1}^{n} \pi_j p_{ji} - \sum_{j\in S}\sum_{i\in S} \pi_j p_{ji} = \sum_{j\in S, i\in\overline{S}} \pi_j p_{ji} \\ &= \mathrm{Vol}(\partial S),\end{aligned} \tag{14}$$

which completes the proof. □

Next, define the volume of S as $\mathrm{Vol}(S) = \sum_{i\in S} \pi_i$. $\mathrm{Vol}(S)$ is the probability with which the random walker occupies a vertex in S. Then the isoperimetric constant of the random walk is defined as

$$h(G) = \inf_S \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(S)} = \min_S \frac{\sum_{i\in S, j\in\overline{S}} \pi_i p_{ij}}{\sum_{i\in S} \pi_i}. \tag{15}$$

The constant $h(G)$ is the minimal probability of the random walker jumping from the vertex set S to its complement set $\overline{S}$ in one step if the current state is in S, i.e., $\min_S Pr(S \to \overline{S}|S)$. It represents the probability bottleneck on the state space of the random walk process.

Define an indicator vector $y \in \{0, 1\}^n$, where

$$y_i = \begin{cases} 1 & i \in \overline{S} \\ 0 & i \in S. \end{cases} \tag{16}$$

Then, from Theorem 2 the volume of the boundary becomes

$$\mathrm{Vol}(\partial S) = 2y^T \Pi(I - P)y, \tag{17}$$

and the volume of the vertex set S becomes

$$\mathrm{Vol}(S) = y^T \pi = y^T \Pi \mathbf{1}, \tag{18}$$

where Π is a diagonal matrix of the elements of the stationary distribution vector, i.e., $\Pi = \mathrm{diag}(\pi)$.

Theorem 2. $\mathrm{Vol}(\partial S) = 2y^T \Pi(I - P)y$.

Proof. By the definition of P and π, we have $\sum_j p_{ij} = 1$ and $\pi_j = \sum_i \pi_i p_{ij}$. Then we have

$$\begin{aligned}\mathrm{Vol}(\partial S) &= \sum_{i,j} \pi_i p_{ij} (y_i - y_j)^2 \\ &= \sum_i \pi_i y_i^2 \sum_j p_{ij} + \sum_j y_j^2 \sum_i \pi_i p_{ij} - 2\sum_{i,j} \pi_i p_{ij} y_i y_j \\ &= \sum_i \pi_i y_i^2 + \sum_j \pi_j y_j^2 - 2\sum_{i,j} \pi_i p_{ij} y_i y_j \\ &= 2(y^T \Pi y - y^T \Pi P y) = 2y^T \Pi(I - P)y,\end{aligned}$$

which completes the proof. □

Finally the isoperimetric constant of the random walk can be rewritten as

$$h(G) = \inf_S \frac{\mathrm{Vol}(\partial S)}{\mathrm{Vol}(S)} = \min_y \frac{2y^T \Pi(I-P)y}{y^T \Pi \mathbf{1}}. \quad (19)$$

This definition of the isoperimetric constant in terms of the random walk is consistent with the definition (3) for an undirected graph. Given an undirected graph with the adjacent matrix W, we have the natural random walk with the transition probability $P = D^{-1}W$, where D is a diagonal matrix with each entry on the diagonal being the degree of each vertex, i.e., $D = \mathrm{diag}(W\mathbf{1})$. This Markov chain is reversible. The stationary probability of the random walk is proportional to the degree of each vertex. Substituting $P = D^{-1}W$ and $\Pi = D/\mathrm{trace}(D)$ to (19), we recover the isoperimetric problem described in Section 2.2.

3.4. Random walk isoperimetric cut

Our goal is to design a graph partitioning algorithm to minimize the isoperimetric constant. In fact, directly minimizing the isoperimetric constant is infeasible. The exact solution to this discrete optimization problem is NP-hard [6, 9].

In order to solve the partitioning problem, we relax the binary definition of y so that it can take nonnegative real values. Then the problem is transformed to

$$\begin{aligned} \min_y \quad & y^T \Pi(I-P)y \\ \text{s.t.} \quad & y^T \Pi \mathbf{1} = I. \end{aligned} \quad (20)$$

By introducing a Lagrange multiplier λ, we turn (20) into a constraint free optimization problem

$$Q(y) = y^T \Pi(I-P)y - \lambda y^T \Pi \mathbf{1}. \quad (21)$$

Taking the derivative of $Q(y)$ w.r.t. y, and setting it equal to 0, we have

$$2\Pi(I-P)y = \Pi\mathbf{1}. \quad (22)$$

Therefore, the problem of finding the solution y that minimizes $Q(y)$ reduces to solving a linear system

$$(I-P)y = \mathbf{1}, \quad (23)$$

where the scalar parts are dropped since only relative values are useful.

The matrix $L = I - P$ is singular since $L\mathbf{1} = 0$. Therefore, the linear system (23) is ill posed. To achieve an unique solution of (23), we need extra constraints.

As we assume the transition matrix P is irreducible, the directed graph associated with P is strongly connected. We can designate an arbitrary vertex g to be included in S, i.e., $y_g = 0$ (g is called the ground vertex in the rest of this paper), which is equivalent to removing the gth row and column of L (the remaining matrix is denoted by L_0), and the gth row of y (the remaining vector is denoted by y_0) in (23). Then the linear system

$$L_0 y_0 = \mathbf{1} \quad (24)$$

is well posed, which can be efficiently solved by the conjugate gradient method. The solution y_0 is a nonnegative real-valued vector. The bi-partitioning result can be obtained by thresholding y_0. Vertices with a y_i below the threshold are placed in S. We use y to collectively refer to y_0 and the designated value of $y_g = 0$. Several thresholding strategy can be applied. For example, the jump cut which chooses a threshold that separates vertices on either side of the largest jump in a sorted y, and the criterion cut which chooses the threshold that gives the lowest value of the isoperimetric ratio.

To achieve a multi-class clustering result, the algorithm is recursively applied to the subgraphs with the smallest isoperimetric constants, until the number of subgraphs reaches a predefined value.

There are several ways to choose the vertex g, such as randomly picking a vertex. In this paper, we choose the vertex with the maximal stationary probability. This strategy is based on the heuristic that a vertex with a high stationary probability is the one with high probability that a random walker jumps to it. Such a vertex is likely in the interior of a cluster but not on the boundary. Empirically, we have found that, as long as g is not along the ideal boundary, a reasonable partitioning with a small isoperimetric ratio can be produced.

3.5. A random walk hitting time view

The expected hitting time $m(j|i)$ is defined as the expected number of steps that a random walker, starting from the vertex (state) $i \neq j$, will take to reach the vertex (state) j for the first time [1]. It can be easily verified that the expected hitting time satisfies the following recurrence relations

$$\begin{cases} m(i|i) = 0, \\ m(j|i) = 1 + \sum_{k=1}^n p_{ik} m(j|k), \quad i \neq j. \end{cases} \quad (25)$$

Let m_0 be the vector with each entry being the expected hitting time $m(g|i)$ from any vertex $i \neq g$, to the ground vertex g and P_0 be the matrix obtained from the transition matrix P by removing the gth row and column. Then we can write (25) in a matrix form as $m_0 = \mathbf{1} + P_0 m_0$, which is equivalent to (24). We can see that the approximate solution of the isoperimetric cut problem given by the linear system (24) is the expected hitting times $m(g|i)$ from vertices $i \in V$ to the ground vertex g. From this expected hitting time viewpoint, we have the following insights into the isoperimetric cut algorithm.

First, we can easily see that if the ground vertex g is selected such that for any other vertex $i \neq g$, there exists a path from i to g, then the linear system (24) is well posed, even if the graph is not strongly connected.

Second, we can examine the connectivity properties of the partitions obtained by thresholding y_0 obtained from solving (24). We will prove that the partition containing the ground vertex (i.e., the set S) must be connected, regardless of how a threshold (i.e., cut) is chosen. The strategy for establishing this is that every vertex has a path to the ground vertex with a monotonically decreasing expected hitting time. Note that the partition not containing the ground vertex may or may not be connected. This result extends the result for undirected graphs in [8] to general state spaces of Markov random walk with a weaker assumption.

Lemma 3. *For every vertex i, there exists a path $(i, v^1, v^2, \cdots, g)$ to the ground vertex g, such that $y_i \geq y_{v^1} \geq y_{v^2} \geq \cdots \geq 0$, when $L_0 y_0 = \mathbf{1}$.*

Proof. By the definition of the expected hitting time, each non-grounded vertex has a value

$$y_i = 1 + \sum_{e_{ij} \in E} p_{ij} y_j. \tag{26}$$

For a vertex set $S \subseteq V$, let the boundary vertex set of S be $S_b \subset V$, such that $S_b = \{j | e_{ij} \in E, \exists i \in S, j \notin S\}$. Then for any vertex i, we can explicitly construct a path to the ground vertex g with nonincreasing expected hitting time by the following procedure:
1) Start with $S = \{i\}$.
2) Repeat adding $j \in S_b$ to S such that $y_j \leq \min y_k, \forall k \in S$ by (26), until $g \in S_b$.
Step 2) is feasible, because for every vertex $k \in S$, there exists a path from k to g. □

Proposition 4. *If the set of vertices V is strongly connected, for any c, the subgraph with vertex set $M \subseteq V$ defined by $M = \{i \in V | y_i < c\}$ is connected when y_0 satisfies $L_0 y_0 = \mathbf{1}$.*

Proof. Since V is strongly connected, for any $g \in V, \forall i \in V, i \neq g$, there exists a path from i to g. Then from Lemma 3, all $j \in M$ are connected to g. Therefore the subgraph M is connected. □

The relationship between the expected hitting time and the isoperimetric problem also explains why the expected hitting time, as a proximity measure, performs very well in the ranking and retrieval tasks [18]. The small expected hitting time between the query and a sample in the database implies that they are likely of the same class in the clustering sense.

3.6. Forcing irreducibility

One problem that might occur in practice is that the assumption of irreducibility of P is not satisfied. In such a case, the stationary distribution is not guaranteed.

However, we can always construct an auxiliary graph G_α by adding a dummy vertex to the original graph G which has an out-link to every other vertex and an in-link from every other vertex. G_α is apparently strongly connected, making the Markov chain irreducible. The transition matrix corresponding to the auxiliary graph is

$$P_\alpha = \begin{pmatrix} (1-\alpha) D^{-1} W & \alpha e \\ e^T / n & 0 \end{pmatrix}, \tag{27}$$

where $\alpha \in (0, 1)$ is a small perturbation factor (say, $\alpha = 10^{-6}$). The dummy vertex is a teleport state of the chain. At any other vertex, a random walker has a small probability α of transitioning to the teleport state, from which it teleports to one of the n original vertices with the probability $1/n$.

When α is small, the auxiliary graph G_α with the transition matrix P_α has approximately the same isoperimetric constant as the original graph G. Assume that the true stationary distribution and the evaluated π on the auxiliary graph G_α are approximately the same. From (15), if the vertex set S is fixed, the absolute difference of the isoperimetric ratios between G and G_α is

$$|h(G) - h(G_\alpha)| \approx \alpha (1 - h(G)). \tag{28}$$

Therefore, if α is small, adding the teleport vertex dose not change the cluster structure of the graph.

4. Experiments

In this section, we conduct experiments on a number of benchmark data sets to evaluate the proposed random walk isoperimetric cut (RWICut). Five related algorithms are compared to show the effectiveness of our algorithm, including Kmeans, iterative normalized cut (NCut) [19], NJW [16], self-tuning graph construction based normalized cut (StNCut) [21], and self-tuning graph construction based NJW (StNJW). The parameters in these algorithms are all tuned to ensure the best results in terms of the normalized mutual information evaluation. Furthermore, we analyze the computational efficiency of our algorithm compared with eigen-decomposition based approaches such as NCut, where the execution times of the algorithms with different numbers of samples and different numbers of neighborhoods are examined.

4.1. Evaluation measures

To evaluate the performances of the clustering algorithms, we compute the following two performance measures from the clustering results: normalized mutual information (NMI) and minimal clustering error (Error). NMI is

Figure 2. Example images in the Scene data set [17].

Table 1. Descriptions of the image data sets used in the experiments.

Dataset	clusters	dimensions	objects
Scene	8	512	2688
UMist-all	20	10304	575
UMist-10	10	10304	265
UMist-5	5	91	140
USPS-all	10	256	5000
USPS-5	5	256	2500

defined as

$$\text{NMI}(x, y) = \frac{I(x, y)}{\sqrt{H(x)H(y)}}, \tag{29}$$

where $I(x, y)$ is the mutual information between x and y, and $H(x)$ and $H(y)$ are the entropies of x and y respectively. Note that $0 \leq \text{NMI}(x, y) \leq 1$ and $\text{NMI}(x, y) = 1$ when $x = y$. The larger the value of NMI, the better a clustering result.

The clustering error is defined as the minimal classification error among all possible permutation mappings defined as:

$$\text{Error} = \min(1 - \frac{1}{n}\sum_{i=1}^{n} \delta(y_i, c_i)), \tag{30}$$

where y_i and c_i are the true class label and the obtained clustering result of x_i, respectively, $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and 0 otherwise.

4.2. Experimental results on image data sets

To validate our algorithm on real image data set and demonstrate the superiority of the proposed algorithm compared with the state-of-the-art related ones, we carry out experiments on three data sets, UMist, USPS, and a scene category data set (Scene) [17]. UMist consists of 575 multi-view face images of 20 different persons with varied poses from profiles to frontal views. USPS consists of 5000 images of 10 handwritten digits (0-9). To further exploit the databases, we randomly select 10 and 5 classes from UMist to construct two data sets UMist-10 and UMist-5, and use 5 digital numbers $(2, 3, 5, 6, 8)$ from USPS as another data set USPS-5 for the experiments. For UMist-5, the dimensions of the images are reduced by PCA while maintaining 99% of the total energy. The Scene data set was collected by Oliva and Torralba [17], containing 8 categories of natural scenes. We use the feature called Spatial Envelope [17] to represent each scene image, although other choices can be used. The feature is a 512-dimensional vector, capturing the dominant spatial structure of the scene. The description of the data sets used in our experiments are summarized in Table 1.

Table 2. NMI comparison results on the ten real data sets. The best values are bold.

Dataset	Kmeans	NCut	NJW	StNCut	StNJW	RWICut
Iris	0.7582	0.7571	0.7661	0.6524	0.7857	**0.8449**
Wine	0.4288	**0.4624**	0.4351	0.3665	0.4199	0.4496
WDBC	0.4672	0.5754	0.5358	0.4679	0.4845	**0.5868**
Satimage	0.6138	0.6749	0.6373	0.6336	0.6307	**0.6932**
Segment	0.6124	0.6465	0.6629	0.5852	0.6801	**0.7440**
UMist-all	0.6726	0.6157	0.8009	0.5364	0.6512	**0.8785**
UMist-10	0.6161	0.5769	0.8214	0.4918	0.5850	**0.8634**
UMist-5	0.7065	0.8903	0.8655	0.6384	0.6371	**1**
USPS-all	0.4038	0.4517	0.5180	0.1894	0.3606	**0.6880**
USPS-5	0.4469	0.5789	0.4247	0.2536	0.3197	**0.6910**
Scene	0.3951	0.4100	0.4471	0.3605	0.4204	**0.4695**

Table 3. Error comparison results on the ten real data sets. The best values are bold.

Dataset	Kmeans	NCut	NJW	StNCut	StNJW	RWICut
Iris	0.1067	0.0933	0.1000	0.4867	0.0933	**0.0533**
Wine	0.2978	0.2697	0.2921	0.2921	0.2865	**0.2472**
WDBC	0.1459	0.0879	0.109	0.1388	0.1248	**0.0796**
Satimage	0.3310	0.2544	0.2457	0.2810	0.2737	**0.2197**
Segment	0.3342	0.4004	**0.2740**	0.5165	0.3407	0.2922
UMist-all	0.5339	0.5791	0.3948	0.6348	0.5739	**0.2661**
UMist-10	0.5509	0.5208	0.3057	0.5849	0.5547	**0.2604**
UMist-5	0.2214	0.0857	0.1214	0.3786	0.3071	**0**
USPS-all	0.6008	0.6404	0.4882	0.8396	0.6388	**0.3398**
USPS-5	0.3468	0.4140	0.4256	0.6224	0.4572	**0.2232**
Scene	0.5056	0.4835	0.4014	0.5443	0.4725	**0.3857**

The clustering results by the six algorithms, Kmeans, NCut, NJW, StNCut, StNJW, and RWICut, are shown in Table 2 and Table 3, from which we can see that RWICut performs best in all the data consistently.

4.3. Experimental results on UCI data sets

Five data sets (Iris, Wine, WDBC, Satimage, and Segment) from UCI Machine Learning Repository are used in this experiment, which are widely used to evaluate clustering algorithms. The five data sets origin from the problems in different domains. More details of them are summarized in Table 4.

The comparison results are also listed in Table 2 and Table 3. Among all the 22 comparisons, the RWICut algorithm obtains the best results in 20 cases, and the second best results in another 2 cases. These comparisons demonstrate that RWICut can achieve excellent performances consistently on real world applications with various numbers of clusters, samples, and dimensionalities.

Table 4. Descriptions of the UCI data sets used in the experiments.

Dataset	clusters	dimensions	objects
Iris	3	4	150
Wine	3	13	178
WDBC	2	30	569
Satimage	6	36	6435
Segment	7	19	2310

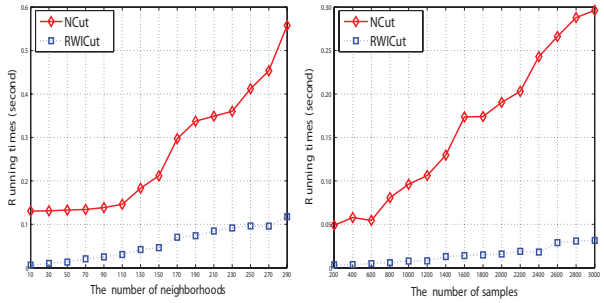

Figure 3. Running time comparisons between RWICut and NCut.

4.4. Computational efficiency analysis

In addition to the excellent performance of our algorithm in accuracy, its computational efficiency is also an advantage over the related ones except Kmeans. Figure 3 shows the running times of RWICut and NCut, with respect to different numbers of samples and neighborhoods. NCut is a representative approach for eigen-decomposition based algorithms, whose computational complexity is similar to NJW, StNCut, and StNJW. From these results, we can see that our algorithm is much faster than the other four algorithms, with much smaller time increasing than that with NCut as the numbers of samples and neighborhoods grow. The algorithms are implemented in Matlab, running on a 2.8 GHz Pentium IV PC with 4GB RAM.

5. Conclusions

In this paper, we propose a kernel density estimator based directed graph clustering algorithm. A variable bandwidth kernel density estimator method is used to construct the directed graph. This method effectively utilizes the local distribution information of the data. An efficient directed graph partitioning algorithm is also developed which optimizes the random walk isoperimetric ratio by solving a linear system. Experimental results show that the proposed method is superior to several popular methods on many benchmark data sets.

Viewing spectral clustering from the density estimation prospective opens a door to solving the graph construction problem. Many nonparametric techniques can be utilized to boost the performance of spectral clustering algorithms. In our future work, we will explore other nonparametric density estimation models such as Dirichlet process. We will also try to derive a multiclass formulation and its algorithm for the random walk isopermetric cut.

Acknowledgement

This work was supported by grants from Natural Science Foundation of China (No. 60975029), Shenzhen Bureau of Science Technology & Information, China (No. JC200903180635A), and the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 415408).

References

[1] D. Aldous and J. Fill. Reversible Markov Chains and Random Walks on Graphs. *Book in preparation*, 2001.

[2] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.

[3] M. Chen, J. Liu, and X. Tang. Clustering via random walk hitting time on directed graphs. *AAAI*, 2008.

[4] M. Chen, Q. Yang, and X. Tang. Directed graph embedding. *IJCAI*, 2007.

[5] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[6] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, pages 36–61, 1991.

[7] P. Duda, D. Stork, and J. Wiley. Pattern Classification. 2001.

[8] L. Grady and E. Schwartz. Isoperimetric Graph Partitioning for Image Segmentation. *PAMI*, 2006.

[9] L. Grady and E. Schwartz. Isoperimetric Partitioning: A New Algorithm for Graph Partitioning. *SIAM Journal on Scientific Computing*, 27:1844, 2006.

[10] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[11] R. Kannan and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[12] A. Langville and C. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2005.

[13] M. Meila and J. Shi. A random walks view of spectral segmentation. *AISTAT*, 2001.

[14] B. Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory Series B*, 47(3):274–291, 1989.

[15] B. Nadler and M. Galun. Fundamental limitations of spectral clustering. *NIPS*, 2007.

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

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *ICCV*, 2001.

[18] P. Sarkar, A. W. Moore, and A. Prakash. Fast incremental proximity search in large graphs. *ICML*, 2008.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[20] S. Yu and J. Shi. Multiclass spectral clustering. *ICCV*, 2003.

[21] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS*, 2005.

[22] D. Zhao, Z. Lin, and X. Tang. Contextual distance for data perception. *ICCV*, 2007.