Clustering via Random Walk Hitting Time on Directed Graphs

Mo Chen[1] **Jianzhuang Liu**[1] **Xiaoou Tang**[1,2]

[1]Dept. of Information Engineering
The Chinese University of Hong Kong
{mochen,jzliu}@ie.cuhk.edu.hk

[2]Microsoft Research Asia
Beijing, China
xitang@microsoft.com

Abstract

In this paper, we present a general data clustering algorithm which is based on the asymmetric pairwise measure of Markov random walk hitting time on directed graphs. Unlike traditional graph based clustering methods, we do not explicitly calculate the pairwise similarities between points. Instead, we form a transition matrix of Markov random walk on a directed graph directly from the data. Our algorithm constructs the probabilistic relations of dependence between local sample pairs by studying the local distributions of the data. Such dependence relations are asymmetric, which is a more general measure of pairwise relations than the similarity measures in traditional undirected graph based methods in that it considers both the local density and geometry of the data. The probabilistic relations of the data naturally result in a transition matrix of Markov random walk. Based on the random walk viewpoint, we compute the expected hitting time for all sample pairs, which explores the global information of the structure of the underlying directed graph. An asymmetric measure based clustering algorithm, called K-destinations, is proposed for partitioning the nodes of the directed graph into disjoint sets. By utilizing the local distribution information of the data and the global structure information of the directed graph, our method is able to conquer some limitations of traditional pairwise similarity based methods. Experimental results are provided to validate the effectiveness of the proposed approach.

Introduction

Recently, pairwise relation based clustering algorithms attract great attention. A successful example is the spectral clustering (Meila & Shi 2001; Ng, Jordan, & Weiss 2001; Shi & Malik 2000; Yu & Shi 2003). These methods have the advantage that they do not make strong assumptions about the distribution of the data. Instead, similarities between sample pairs are first computed to construct an undirected graph of the data and then a global decision is made to partition all data points into disjoint sets according to some criterions. Therefore, these methods can potentially deal with data sets whose clusters are of irregular shapes.

Despite the great success of the graph based methods, there are still open problems: 1) How to construct the pairwise similarities between sample points to reflect the underlying distribution of the data; 2) How to deal with multiscale data; 3) How to handle the data whose clusters are defined by geometry. Moreover, Nadler and Galun recently pointed out that there are fundamental limitations of these graph based approaches (Nadler & Galun 2007). According to their analysis, even with carefully tuned parameters, the spectral clustering algorithms still cannot successfully cluster the multi-scale data sets. They showed examples that the clusters, which can be easily captured by human, cannot be properly identified by the spectral clustering methods.

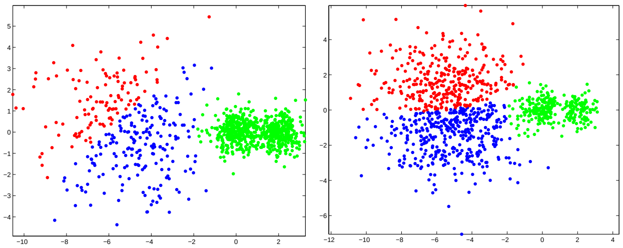
In this paper, we show that the widely used parametric Gaussian kernel based similarities are not informative enough for modeling pairwise relations. As a result, the undirected graph constructed based on the similarities does not necessarily capture the intrinsic structure of the underlying data distribution. Therefore, the natural clusters of the data cannot be obtained by partitioning the graph.

From our analysis, we propose a data clustering algorithm based on a directed graph model. The edge weights of the graph are the probabilistic dependence relations between local sample points, which are constructed by exploring the local distributions of the data. Such relations are asymmetric and more general than the similarities used in traditional undirected graph based methods since they consider both the local density and geometry of the data.

The probabilistic relations between all sample pairs naturally result in a stochastic matrix, which can be considered as the transition matrix of the Markov random walk process on a directed graph. Our new clustering algorithm works on this directed graph, which is based on the random walk model, more specifically the expected hitting time of random walk model.

The random walk hitting time has a nice property: it decreases when the number of paths connecting two nodes increases and the length of any path decreases. Informally speaking, the shorter the paths connect two nodes are, the more related the two nodes are; strongly connected nodes are more related than weakly connected nodes.

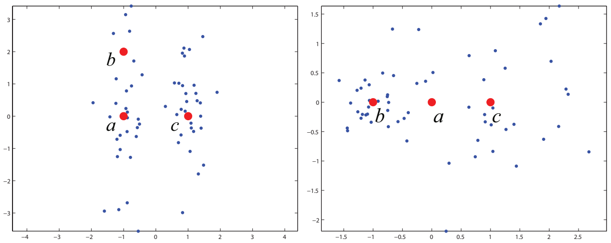
There are some other applications that consider various measures based on random walk models. For example, the paper of (Fouss *et al.* 2007) proposes an embedding method



(a) Three Gaussians (1:1:1)

(b) Three Gaussians (8:1:1)

Figure 1: Clustering results by the NJW algorithm on two multi-scale data sets. Different clusters are denoted by different colors.



(a) Shape

(b) Density

Figure 2: The local density and shape affect the relations between sample pairs.

based on the commute time distance on undirected graphs for collaborative recommendation systems. Another paper of (Brand 2005) proposes to use an angular based quantity for semi-supervised learning problems, which can be seen as a normalized version of the commute time on undirected graphs.

All these approaches are based on undirected graph models, with symmetric measures (similarity) between sample pairs. We will see later that symmetric measures cannot fully capture the relations between sample points. Sometimes it may even hinder the performance of the algorithms.

The rest of the paper is organized as follows. In Section 2, we briefly discuss the limitations of the pairwise similarity based clustering methods. Section 3 describes the framework of the proposed random walk hitting time based digraph clustering algorithm. Section 4 presents our experimental results. Section 5 concludes this paper.

Limitations of Pairwise Similarity Based Methods

The first step of pairwise relation based algorithms is to construct an undirected graph for the vector data. Sample points are connected by undirected edges. The edge weights reflect the similarities between sample pairs. Usually a Gaussian kernel $\exp(-\|x_i - x_j\|^2/2\sigma^2)$ with a manually adjusted parameter σ is used for setting the weights. A problem of this step is how to choose the parameter σ. When it is not properly set, the clustering results can be poor. A more severe problem is that a single σ for all sample pairs implies that if the Euclidean distances between two pairs are the same, the two similarities are the same too. When the input data are with different density and geometry, there may not exist a single value of σ that works well for the whole data set. For certain data set, the intrinsic cluster structure essentially may not be explored by the spectral clustering algorithm, no matter what value of σ is chosen.

Figures 1a and 1b are two multi-scale data sets from (Nadler & Galun 2007), where 1000 sample points are generated by three Gaussians with variances $\sigma_1 = 2$ and $\sigma_2 = \sigma_3 = 0.5$. The point numbers of the Gaussians are 1:1:1 for Figure 1a and 8:1:1 for Figure 1b. The best results that can be achieved by the spectral clustering are shown with different colors denoting the clusters. As indicated in (Nadler & Galun 2007), these multi-scale problems essentially cannot be solved by spectral clustering, mainly because it does not explore the density information inherent in the data. Even though the parameter σ has been carefully tuned, from the figure we can see that the spectral clustering algorithm cannot obtain satisfactory results on these data sets.

When dealing with these kinds of data, exploring the local data distribution is very important. Consider the data shown in Figure 2a. The Euclidean distances $d(a, b)$ and $d(a, c)$ between the sample pairs (a, b) and (a, c) are the same. Then the similarities computed by the 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 geometric shape of the local data distribution is important for modeling the relations between sample pairs. Another example in Figure 2b shows the importance of the density of the local data distribution that affects the relations between sample pairs. Although sample a lies in the middle of b and c, a is more likely to have the same class label as c than as b when considering the density of the context data. Here the local density of the data distribution is important for modeling the relations between the sample pairs.

These two intuitive examples suggest that we should analyze the local data distribution when modeling the pairwise relations between sample points.

Random Walk Hitting Time Based Digraph Clustering

Based on the analysis above, we propose to study the local context of the data to setup the relations between local sample points. The relations between sample pairs are not necessarily symmetric. We adopt a probabilistic framework to model the local pairwise relations to form a directed graph, and then compute the random walk hitting time between all sample pairs which explores the global information of the structure of the underlying directed graph. An iterative algorithm called K-destinations is proposed to cluster the data based on the hitting time measure.

Local Gaussian based Bayesian inference

Let $x_i \in \mathbb{R}^d,\ i = 1, 2, \cdots, n$, be points that we wish to assign to K clusters $\Gamma = \{V_l\}_l,\ l \in \{1, 2, \cdots, K\}$. The good performance of a clustering method indicates that the label of a data point can be well estimated based on its neighbors.

The data of a local neighborhood can be deemed as a single Gaussian. Denote $N(i)$ as the index set of x_i's neighbors. In this paper, the k nearest neighbors of x_i are used to compose the neighborhood $N(i)$. Each sample $x_j,\ j \in N(i)$, can be thought to lie in a local Gaussian centered at x_i, i.e., $x_j \sim \mathcal{N}(x_i, C_i),\ j \in N(i)$, where x_i and C_i are the mean and covariance of this Gaussian. The covariance C_i of the Gaussian can be estimated using the data of the neighborhood $N(i)$ by the maximal likelihood estimation (MLE). Let $X_i = [x_j]_j,\ j \in N(i)$, be a matrix with each column being a neighbor of x_i. A regularized covariance matrix C_i of the local Gaussian distribution $\mathcal{N}(x_i, C_i)$ is

$$C_i = \frac{1}{|N(i)|}(X_i - x_i e^T)(X_i - x_i e^T)^T + \alpha I, \qquad (1)$$

where $|N(i)|$ is the cardinality of the set $N(i)$, α is the regularization factor, e is a vector with all entries being 1, and I is the identity matrix (Hastie, Tibshirani, & Friedman 2001).

Let $\widehat{C}_i = (X_i - x_i e^T)(X_i - x_i e^T)^T/|N(i)|$. In this paper, we use a modified version of the regularized covariance matrix proposed in (Srivastava & Gupta 2006):

$$C_i = \widehat{C}_i + \frac{tr(\widehat{C}_i)}{d} I. \qquad (2)$$

We write $\mathcal{N}_i$ as the abbreviation of $\mathcal{N}(x_i, C_i)$. Then for a sample point x_j, the multivariate Gaussian density, with which x_j is generated by the Gaussian $\mathcal{N}(x_i, C_i)$, is given by

$$p(x_j|\mathcal{N}_i) = \frac{\exp\left(-\frac{1}{2}(x_j - x_i)^T C_i^{-1}(x_j - x_i)\right)}{\sqrt{(2\pi)^d |C_i|}}. \qquad (3)$$

As shown in Figure 3a, given the neighbor Gaussians, the probability that x_j is generated by the Gaussian $\mathcal{N}(x_i, C_i)$ can be computed by the Bayesian rule:

$$P(\mathcal{N}_i|x_j) = \frac{p(x_j|\mathcal{N}_i)P(\mathcal{N}_i)}{\sum_{i \in N(j)} p(x_j|\mathcal{N}_i)P(\mathcal{N}_i)}. \qquad (4)$$

For simplicity, the prior probabilities are set equal.

$P(\mathcal{N}_i|x_j)$ can be thought as the local dependence of x_j on x_i given the context of local data distributions when determining the cluster membership of each sample, as shown in Figure 3a. It represents the dependence of x_j on x_i. Also denote $p_{ji} = P(\mathcal{N}_i|x_j)$, then $\sum_i p_{ji} = 1$. Notice that the probabilistic relations between points i and j are not asymmetric, i.e., in general, p_{ji} is not necessarily equal to p_{ij}. Then all samples and the asymmetric relations between sample pairs naturally result in a directed graph.

The advantage of using the local Gaussian based Bayesian inference to model the dependence relations between sample pairs can be seen from Figure 3. When solely using the Euclidean distances to model the pairwise relations, the clustering boundary may not be reasonable as shown in Figure 3b. By considering the local distribution, we can obtain a satisfactory boundary as shown in Figure 3c. Using this approach, we avoid setting the bandwidth parameter σ of the Gaussian kernel in the spectral clustering methods. Moreover, this estimation of local relations considers both the local density and geometry of the data, thus the constructed graph reflects the underlying distribution of the data set.

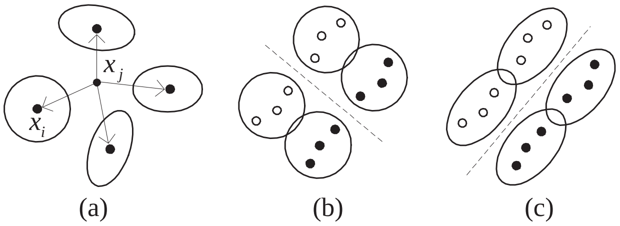


Figure 3: Advantage of using local Gaussian based Bayesian inference to construct neighbor relations. (a) Local Gaussian based Bayesian inference. (b) Boundary found by modeling pairwise relations with the isotropic Gaussian kernel. (c) Boundary found by modeling pairwise relations with local Gaussian estimation.

Random walk

For all sample pairs, we have the matrix $P = [p_{ij}]_{ij}$, which has the property that $Pe = e$, i.e., P is stochastic. After obtaining the stochastic matrix P, we can naturally define the Markov random walk on the directed graph associated with P. The random walk is defined with the single-step transition probability p_{ij} of jumping from any node (state) i to one of its adjacent nodes j where $p_{ij} = P[i \rightarrow j]$ is the probability of one step discrete time random walk transition from i to j. Then P can be regarded as the transition probability matrix of the random walk process.

The transition probabilities depend only on the current states (first-order Markov chain). If the directed graph associated with the matrix P is strongly connected, the Markov chain is irreducible, that is, every state can be reached from any other states. If this is not the case, the Markov chain can be decomposed into closed subsets of states which are independent (there is no communication between them), each closed subset is irreducible, and the procedure can be applied independently to these closed subsets.

The unique stationary distribution of the Markov chain is guaranteed if P is irreducible (Aldous & Fill 2001). Let $\pi = [\pi_i]_i,\ i = 1, \cdots, n$, be the vector of the stationary distribution of the Markov random walk. The stationary distribution vector, also called Pagerank vector in the information retrieval literature, can be obtained by solving a linear system $\pi^T P = \pi^T$ subject to a normalized equation $\pi^T e = 1$. Either the power algorithm or the eigen-decomposition algorithm can be used to compute the stationary distribution (Langville & Meyer 2005).

The expected hitting time $h(j|i)$ of the random walk is the expected number of steps before node j is visited starting from node i. It can be easily verified that the hitting time satisfies the following recurrence relations

$$\begin{cases} h(i|i) = 0 \\ h(j|i) = 1 + \sum_{k=1}^{n} p_{ik} h(j|k) \quad i \neq j. \end{cases} \tag{5}$$

The recurrence relations can be used in order to iteratively compute the expected hitting time. The meaning of these formulae is quite obvious: in order to jump from node i to node j, one has to go to any adjacent state k of node i and proceeds from there.

The closed form of the hitting time in terms of transition matrix exists (Aldous & Fill 2001). By introducing a matrix

$$Z = (I - (P - e\pi^T))^{-1}, \tag{6}$$

the matrix H with its entry $H_{ij} = h(j|i)$ can be computed by

$$H_{ij} = (Z_{jj} - Z_{ij})/\pi_j, \tag{7}$$

where Z_{ij} is the entry of Z. More efficient way of computing the hitting time also can be found in (Brand 2005).

As mentioned before, the hitting time from node i to node j has the property of decreasing when the number of paths from i to j increases and the lengths of the paths decrease (Doyle & Snell 1984). This is a desirable property for representing the dependence of the label of one point on another for the clustering task when the global distribution of the data is taken into account.

A closely related quantity, the commute time $c(i, j)$, is defined as the expected number of steps that a random walker, starting from node $i \neq j$, takes to meet node j for the first time and goes back to i. That is, $c(i, j) = h(j|i) + h(i|j)$. The commute time distance is also known as the resistance distance in the electrical literature (Klein & Randić 1993). The commute time distance on undirected graphs is widely used in many applications (Brand 2005; Fouss *et al.* 2007). However we argue that it is not suitable for our case. In the case of a directed graph, if the hitting time from node i to node j is small, which means node i and node j are tightly related, but the hitting time from node j to node i is not necessarily small. Such cases often happen on the points that lie on the boundaries of clusters, which have short hitting times to the central points in the same clusters, but often have very large hitting times from the central points to points close to the boundaries. So in this paper, we use the hitting time instead of commute time as the measure of pairwise relations.

K-destinations algorithm

After obtaining the asymmetric hitting times between all sample pairs, we are ready to apply a clustering algorithm to categorize the data into disjoint classes. Since traditional pairwise relation based clustering algorithms often require the pairwise relations be symmetric and the similarity functions be semi-definite, such as the spectral clustering (Ng, Jordan, & Weiss 2001; Shi & Malik 2000; Yu & Shi 2003), they are not suitable for clustering using the hitting time measure.

In this work, we develop an iterative clustering algorithm based on the asymmetric hitting time measure, which is similar to the K-means algorithm, called the K-destinations that directly works on the pairwise hitting time matrix H.

Table 1: Random walk hitting time based digraph clustering algorithm

Input: The data matrix X, the number of nearest neighbors k, and the number of clusters K.

1. For each i, $i = 1, 2, \cdots, n$
   (a) Form $X_i = [x_j]_j$, $j \in N(i)$, by the k nearest neighbors of x_i.
   (b) Compute the m left singular vectors $u_{i_1}, u_{i_2}, \cdots, u_{i_m}$ corresponding to the non-zero singular values of $\widehat{X}_i = X_i - x_i e^T$ to form $\widehat{U}_i = [u_{i_1}, u_{i_2}, \cdots, u_{i_m}]$.
   (c) For those j with $i \in N(j)$, compute $p(x_j|\mathcal{N}_i)$ as in (12).
2. Compute $P = [p_{ij}]_{ij}$ with each entry $p_{ij} = p(\mathcal{N}_j|x_i)/\sum_{j\in N(i)} p(x_i|\mathcal{N}_j)$.
3. Compute $Z = (I - P - e\pi^T)^{-1}$ and compute H where $H_{ij} = (Z_{jj} - Z_{ij})/\pi_j$.
4. Perform the hitting time based K-destinations algorithm.

Each cluster V_l, $l = 1, \cdots, K$ is represented by an exemplar v_l, which is called a destination node in our algorithm. The destination node is selected from the samples. Intuitively, we want to choose the destinations that save the walkers' (hitting) time. Therefore, we propose to cluster the data by minimizing the sum of the hitting times from the samples to the destination node in each cluster:

$$J = \sum_{l=1}^{K} \sum_{i \in V_l} h(v_l|i). \tag{8}$$

Finding the global optimum of this criteria is a hard problem. Instead, we optimize the function in a greedy manner. Similar to the K-means algorithm, we iteratively minimize J by two steps:

- First, we fix the destination nodes and assign each sample to the cluster that has minimal hitting time from it to the destination node corresponding to the cluster.
- Then, in each cluster, we update the destination node from the samples that minimize the sum of the hitting times from all samples in the cluster to the destination node.

The clustering algorithm repeats the two steps until the cluster membership of each sample does not change. It can be seen that the algorithm monotonously decreases the value of J in each iteration, so the convergence of the algorithm is guaranteed.

Implementation

In typical applications of our algorithm such as image clustering, the dimension d of the data can be very high. Therefore the computation of the Gaussian density in (3) is time consuming, where the inverse of $d \times d$ matrices is involved. Usually in the neighborhood, although the covariance matrix $C = \widehat{C} + \alpha I$ is of rank d, the rank of $\widehat{C}$ is very low[1]. Denoting the rank of $\widehat{C}$ as m, we have $m \leq k \ll d$, where k is the number of neighbors in the neighborhood. Let $U = [u_i]_i,\ i = 1, \cdots, d$, and $\Lambda = diag([\lambda_i]_i),\ i = 1, \cdots, d$, where u_i and λ_i are the eigenvectors and eigenvalues of $\widehat{C}$ respectively, then

$$\begin{aligned} C^{-1} &= U(\Lambda + \alpha I)^{-1} U^T \\ &= U(A + \frac{1}{\alpha} I) U^T, \end{aligned} \tag{9}$$

where A is a diagonal matrix with entries

$$A_{ii} = -\frac{\lambda_i}{\alpha(\lambda_i + \alpha)}. \tag{10}$$

Let $\widehat{U} = [u_i]_i,\ i = 1, \cdots, m$, be a matrix formed by the eigenvectors corresponding to non-zero eigenvalues of $\widehat{C}$, which can be efficiently computed by applying singular value decomposition on a $d \times k$ matrix (see Table 1). Then the Mahalanobis term in (3) is

$$\begin{aligned} d_C(x_j, x) =& (x_j - x)^T C^{-1} (x_j - x) \\ =& \|\widehat{A}^{1/2} \widehat{U}^T (x_j - x)\|^2 + \|x_j - x\|^2 / \alpha, \end{aligned} \tag{11}$$

where $\widehat{A} = diag([A_{ii}]_i),\ i = 1, \cdots, m$, is only a $m \times m$ matrix with $m << d$. Then the density with which x_j is generated by $\mathcal{N}(x, C)$ can be computed as

$$p(x_j|\mathcal{N}) = \exp\left(-\frac{1}{2}(d_C(x_j, x) + \sum_{i=1}^{m} \ln \lambda_i + d \ln(2\pi))\right). \tag{12}$$

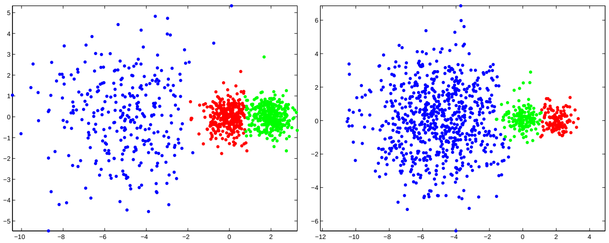
Thus we avoid storing the $d \times d$ matrix C and explicitly computing the inverse of it, which brings us time/space efficiency and numerical stability. The complete algorithm is summarized in Table 1.

Experiments

In this section, we present the clustering results obtained by the proposed random walk hitting time based digraph clustering (HDC) algorithm on a number of synthetic and real data sets. We also compare our HDC algorithm with the K-means and the NJW (Ng, Jordan, & Weiss 2001) algorithms.

Synthetic data

In order to show the advantage of our HDC algorithm, we apply it to the data sets shown in Figure 1, on which the NJW algorithm fails. Our clustering results are given in Figure 4. From the figure we can see that, by exploring both the local distribution information of the data and the global structure information of the graph, the HDC algorithm can work well on the multi-scale data sets. The clustering results satisfactorily capture the natural cluster structures of the data.



(a) Three Gaussians (1:1:1) (b) Three Gaussians (8:1:1)

Figure 4: Clustering results by HDC on the two data sets. Different clusters are denoted by different colors.

Real data

In this section, we conduct experiments on the following real data sets:

- Iris: The data are from the UCI repository comprising 3 classes of 50 instances each, where each class refers to a type of iris plant.
- Wine: The data are from the UCI repository comprising 3 different wines. This data set is used for chemical analysis to determine the origin of wines.
- Satimage: The data are the 10% sampling of the UCI repository Landsat Satellite, which consists of the multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image.
- Ionosphere: The data are from the UCI repository referring to the radar returns from the ionosphere that is a binary clustering to detect “Good” radar.
- Segmentation: The data are from the UCI repository. The instances are drawn randomly from a database of seven outdoor images.
- WDBC: The data are from the UCI repository that is used for diagnostic Wisconsin Breast Cancer.
- UMist-5: The data are from UMist database containing 5 classes, which are randomly selected from all the face images of 20 different persons in UMist database. The dimension of the data are reduced by principle component analysis (PCA) while maintaining 99% of the total energy.

More details of the data sets are summarized in Table 2.

Table 2: Descriptions of the data sets used in the experiments.

Dataset	K	d	n
Iris	3	4	150
Wine	3	13	178
Satimage	6	36	644
Ionosphere	2	34	351
Segmentation	7	19	2310
WDBC	2	30	569
UMist-5	5	91	140

[1] Without ambiguity, here we ignore the subscript i in C_i in this subsection.

Table 3: Error and NMI comparison results on seven data sets. The best values are bold.

Dataset	Error			NMI		
	K-means	NJW	HDC	K-means	NJW	HDC
Iris	0.1067	0.1000	**0.0267**	0.7582	0.7661	**0.8981**
Wine	0.2978	0.2921	**0.2865**	0.4288	0.4351	**0.4544**
Satimage	0.3323	0.2888	**0.2298**	0.6182	0.6693	**0.7039**
Ionosphere	0.2877	0.1510	**0.1266**	0.1349	0.4621	**0.5609**
Segmentation	0.3342	0.2740	**0.2521**	0.6124	0.6629	**0.7039**
WDBC	0.1459	0.1090	**0.1072**	0.4672	**0.5358**	0.5035
UMist-5	0.2214	0.1214	**0.0643**	0.7065	0.8655	**0.8930**

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). The NMI is defined as

$$\text{NMI}(x,y) = \frac{I(x,y)}{\sqrt{H(x)H(y)}}, \qquad (13)$$

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 is, the better a clustering result is.

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

$$\text{Error} = \min(1 - \frac{1}{n}\sum_{i=1}^{n} \delta(y_i, \text{perm}(c_i))), \qquad (14)$$

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.

The clustering results by the three algorithms, K-means, NJW, and HDC, are summarized in Table 3. The HDC algorithm obtains the smallest errors in all the cases, and produces the largest NMI values on all the data sets except one. These results demonstrate that the HDC can achieve good performances consistently on various real world applications.

Conclusions

We have proposed a random walk hitting time based digraph clustering algorithm for general data clustering. The pairwise relations of probabilistic dependence of the data are obtained by local distribution estimation. A directed graph is constructed based on the asymmetric relations. Then the hitting time measure is computed based the Markov random walk model on the directed graph, which explores the global graph structure. An iterative algorithm is also proposed to work with the asymmetric hitting time measure to cluster the data. Our algorithm is able to conquer some limitations of traditional pairwise similarity based methods. Extensive experiments have shown that convincing results are achieved in both synthetic and real world data by our algorithm.

Acknowledgments

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306).

References

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

Brand, M. 2005. A random walks perspective on maximizing satisfaction and profit. *Proceedings of the 2005 SIAM International Conference on Data Mining*.

Doyle, P., and Snell, J. 1984. Random walks and electric networks.

Fouss, F.; Pirotte, A.; Renders, J.; and Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering* 19(3):355–369.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Klein, D., and Randić, M. 1993. Resistance distance. *Journal of Mathematical Chemistry* 12(1):81–95.

Langville, A., and Meyer, C. 2005. Deeper inside PageRank. *Internet Mathematics*.

Meila, M., and Shi, J. 2001. A random walks view of spectral segmentation. *AI and Statistics (AISTATS)*.

Nadler, B., and Galun, M. 2007. Fundamental limitations of spectral clustering. *Advances in Neural Information Processing Systems*.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*.

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

Srivastava, S., and Gupta, M. 2006. Distribution-based Bayesian minimum expected risk for discriminant analysis. *IEEE International Symposium on Information Theory* 2294–2298.

Yu, S., and Shi, J. 2003. Multiclass spectral clustering. *Proceedings of Ninth IEEE International Conference on Computer Vision* 313–319.