Transitive Distance Clustering with K-Means Duality

Zhiding Yu*,[1], Chunjing Xu*,[2], Deyu Meng[3,4], Zhuo Hui[1], Fanyi Xiao[4], Wenbo Liu[1], Jianzhuang Liu[2]
[1] Department of Electrical and Computer Engineering, Carnegie Mellon University
[2] Huawei Technologies Co. Ltd., Shenzhen, China
[3] Inst. for Info. & System Sciences, Faculty of Math. & Stat., Xi'an Jiaotong University
[4] The Robotics Institute, Carnegie Mellon University
yzhiding@andrew.cmu.edu, {xuchunjing,liu.jianzhuang}@huawei.com, dymeng@mail.xjtu.edu.cn

Abstract

We propose a very intuitive and simple approximation for the conventional spectral clustering methods. It effectively alleviates the computational burden of spectral clustering - reducing the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ - while capable of gaining better performance in our experiments. Specifically, by involving a more realistic and effective distance and the "k-means duality" property, our algorithm can handle datasets with complex cluster shapes, multi-scale clusters and noise. We also show its superiority in a series of its real applications on tasks including digit clustering as well as image segmentation.

1. Introduction

Data clustering plays a key role in many applications. Much effort has been devoted to this research [7, 10, 9, 6, 13]. A fundamental principle (assumption) that guides the design of a clustering algorithm is:

Consistency. *Data within the same cluster are close to each other, while data in different clusters are relatively far away.*

According to this principle, the hierarchy approach begins with each sample trivially being a cluster, and iteratively agglomerate the closest pairs of clusters. Such technique completely depends on local data structure of data without global optimization, thus it is prone to errors caused by multi-scale clusters [13]. Besides consistency, early methods such as k-means and EM assume relatively simple distribution shapes with Euclidean / Mahananobis distances. These methods, however, do not perform well on data with manifold or irregularly shaped clusters.

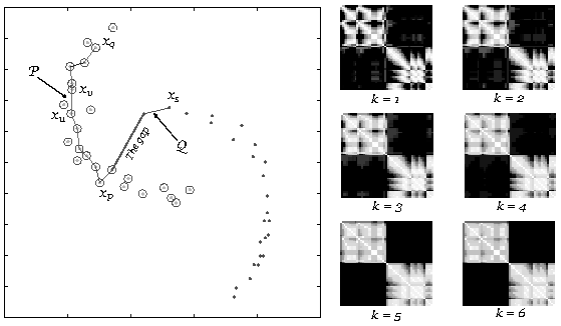
Spectral clustering methods [9, 13, 10] consider that clusters in a dataset can have more complex shapes than compact sample clouds. To overcome problems such as multi-scale clusters in [9], self-tuning spectral clustering [13] further considers local scale of data and the structure of the eigenvectors. Impressive results have been demonstrated and it is regarded as one of the most promising clustering techniques [12]. However spectral clustering often suffers from the scalability problem due to large affinity matrix and $\mathcal{O}(n^3)$ computational complexity with eigendecompostion. Therefore recent trend has been addressing the scalability problem [16, 18, 19, 21].

Alternatively, our work cast a much more intuitive and simpler perspective into this problem. Despite being more effective in finding clusters [18], Eigenproblem needs to be solved in most spectral clustering methods. We show that with a reasonable mapping and the property called "k-means duality", no eigendecomposition is needed and a simple k-means is able to produce results comparable to or even better than many spectral clustering methods. In a sense, our work can also be regarded as an approximate spectral clustering method. It is able to tackle data with clusters of complex shapes and scales. The corresponding complexity of our algorithm is $\mathcal{O}(n^2)$[1], compared with $\mathcal{O}(n^3)$ in many spectral algorithms. The philosophy in this paper share commonality with [3] where spectral clustering is unified to the kernel k-means framework. Yet methods in [3] still need to solve the eigenproblem.

Our key contribution in this paper lies in the formulation which simultaneously offers more efficiency, straightforwardness as well as flexibility. In terms of efficiency, our work avoids the problem of taking eigendecomposition. In terms of straightforwardness, the objective of the proposed method avoids discrete-to-continuous relaxations in many methods which sometimes can lead to undesired consequences, such as over-segmentations in the middle of very smooth regions. In terms of flexibility, the simple formulation allows many convenient extensions. For example, we can easily incorporate the Potts model in the k-means for

*indicates equal contribution.

[1]For the detailed algorithm complexity analysis, the readers please kindly refer to Appendix D in supplementary material.



(a) (b)

Figure 1. (a) A two-moon dataset used to demonstrate the transitive distance, where samples of one cluster are denoted by circles and samples of another cluster are denoted by dots. (b) Maps of transitive distance matrices with different orders

image segmentation. Our experiment demonstrates the superior performance of the method.

We believe the proposed formulation can offer scalability as well. Our work is totally compatible with the framework of [16], and could potentially incorporate many scalable schemes on k-means, such as incremental k-means which is both even more time and memory saving.

2. The transitive distance

We first show how to obtain a more reasonable metric from a traditional metric where the triangle inequality holds. In Fig. 1(a), the Euclidean distance between intra-cluster samples x_p and x_q is larger than that between inter-cluster samples x_p and x_s. A more reasonable metric would give a closer relationship between x_p and x_q than x_p and x_s. A common method is to create a non-linear mapping

$$\phi : V \subset R^l \mapsto V' \subset R^s, \tag{1}$$

such that any two clusters in R^s can be split linearly, known as the "kernel trick". This process, however, is often complicated and time-consuming. We want to have a non-eigen method that can achieve similar nonlinear kernel mapping. Intuitively, it is the inter-connecting samples forming a manifold "path" that indicates strong intra-cluster correlation between x_p and x_q. Suppose each such sample is called a "messenger". We can define a distance through k of these messengers. Let $\mathcal{P} = x_{u_1} x_{u_2} ... x_{u_k}$ be a path with k vertices, where $x_{u_1} = x_p$ and $x_{u_k} = x_q$. The way a metric closer than the Euclidean distance between x_p and x_q with $\mathcal{P}$ can be formulated as

$$D_{\mathcal{P}}(x_p, x_q) = \max_{\substack{x_{u_i}, x_{u_{i+1}} \in \mathcal{P} \\ 1 \le i \le k-1}} \{d(x_{u_i}, x_{u_{i+1}})\}. \tag{2}$$

An example is shown in Fig. 1(a), where the path $\mathcal{P}$ connecting x_p, x_q and the path $\mathcal{Q}$ connecting x_p, x_s are given. In this case, $D_{\mathcal{P}}(x_p, x_q) = d(x_u, x_v)$ is smaller than the inter-cluster gap. Given this intuition, the transitive distance between any two samples can be defined as follows:

Definition 1. *Given the Euclidean distance $d(\cdot,\cdot)$, the derived transitive distance between samples $x_p, x_q \in V$ with order k is*

$$D_k(x_p, x_q) = \min_{\mathcal{P} \in \mathbb{P}_k} \max_{e \in \mathcal{P}} \{d(e)\}, \tag{3}$$

where $\mathbb{P}_k$ is the set of paths connecting x_p and x_q, each such path is composed of at most k vertices, $e \stackrel{def}{=} (x_i, x_j)$, and $d(e) \stackrel{def}{=} d(x_i, x_j)$.

The transitive distance matrices for the dataset in Fig. 1(a) with orders from 1 to 6 are shown in Fig. 1(b). As k becomes larger, the discriminative ability increases. For simplicity, we denote D_n with D when $k = n$, where n is the number of samples.

The concept of "transitive distance" is not new. Transitive closure is used [4] for protein interaction module detection with cliques. For the tractability of problem, the method only considers paths up to order 3. This is often not sufficient to model the intrinsic structure as it loses much of the connectivity resolution. The algorithm complexity remains as high as $\mathcal{O}(n^3)$ where n is the total number of samples. The work in [1] adopted transitive distance to perform texture segmentation and edge grouping but with a bottom-up clustering framework. [2] further generalized it to a Mercer kernel, using Kernel PCA and k-means for clustering. Therefore a large scale eigen-analysis is still needed. We revisit this problem to study a way of using its strength to formulate a fast, intuitive yet effective top-down clustering. More importantly, there are a set of nice theories motivating and justifying the study rather than heuristically choosing a simple method.

Our work also shows close relation to minimum spanning tree (MST) based clustering, efficient graph-based image segmentation (EGS) [15], graphical mode seeking [14] and the normalized tree partition (NTP) [17]. We will give more discussion in Section 5.

Notice that although we use $d(\cdot,\cdot)$ to denote the Euclidean distance in the previous discussion, we can replace $d(\cdot,\cdot)$ with any other meaningful distance (metric). Therefore, $d(\cdot,\cdot)$ is used to denote any distance in the following. This further extends our work to scenarios where better similarity metrics can be easily incorporated.

3. Kernel trick by the transitive distance

The transitive distance is an ultrametric, as is proved in [4]. We show that such ultrametric distance well reflects the relationship among data samples and a kernel mapping with a promising property can be obtained. First we introduce a lemma from [8] and [5].

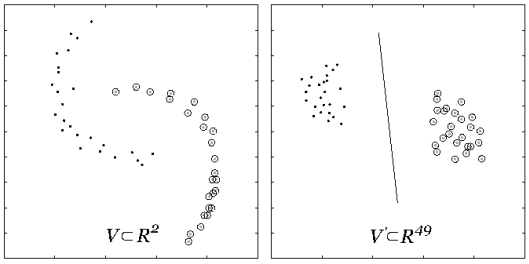


Figure 2. Mapping 50 data samples in $V \subset R^2$ to $V' \subset R^{49}$.

Lemma 1. *Every finite ultrametric space consisting of n distinct points can be isometrically embedded into an $n-1$ dimensional Euclidean space.*

With Lemma 1, we have the mapping[2]

$$\phi : (V \subset R^l, D) \mapsto (V' \subset R^s, d'), \tag{4}$$

where $\phi(x_i) = x'_i \in V'$, $s = n-1$, and n is the number of points in a set V. We also have $d'(\phi(x_i), \phi(x_j)) = D(x_i, x_j)$, where $d'(\cdot,\cdot)$ is the Euclidean distance in R^s.

Definition 2. *A labeling scheme $\{(x_i, l_i)\}$ of a dataset $V = \{x_i | i = 1, 2, ..., n\}$, where l_i is the cluster label of x_i, is called consistent with some distance $d(\cdot,\cdot)$ if the following conditions hold: for any $y \notin C$ and any partition $C = C_1 \cup C_2$, we have $d(C_1, C_2) < d(y, C)$, where $C \subset V$ is some cluster, $y \in V$, $d(C1, C2) \stackrel{def}{=} \min_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$ is the distance between the two sets C_1 and C_2, and $d(y, C) \stackrel{def}{=} \min_{x \in C} d(y, x)$ is the distance between a point y and the set C.*

Theorem 1. *If a labeling scheme of a dataset $V = \{x_i | i = 1, 2, ..., n\}$, is consistent with a distance $d(\cdot,\cdot)$, then given the derived transitive distance D and the embedding $\phi : (V, D) \mapsto (V', d')$, the convex hulls of the images of the clusters in V' do not intersect with each other.*

The proof of Theorem 1 can be found in Appendix A in the supplementary material. An example is illustrated in Fig. 2. A dataset V with 50 points in R^2 is embedded into R^{49}, where the convex hulls of the two clusters do not intersect. We can see the embedding ϕ is a desirable mapping.

The underlying intuition regarding Theorem 1 is that clustering on V' can be much easier than clustering V. While performing k-means on V' seems to be a favorable choice, we only have the distance matrix $E' = [d'_{ij}] = [D_{ij}]$ of V', instead of the absolute coordinates of $x'_i \in V'$. We will show in the following section how we can circumvent this problem using the k-means duality, which leads to a novel clustering algorithm in Section 5.

[2]We use $d(\cdot,\cdot)$ to denote any traditional distance metric in V and $d(\cdot,\cdot)$ the Euclidean distance in V'.

4. The k-means duality

Let $E = [d_{ij}]$ be the distance matrix obtained from a dataset $V = \{x_i | i = 1, 2, ..., n\}$[3]. From E, we can derive a new set $Z = \{z_i | i = 1, 2, ..., n\}$, with $z_i \in R_n$ being the ith row of E. We have the following property, called the duality of the k-means algorithm.

Property 1. *(K-Means Duality): The clustering result obtained by the k-means algorithm on Z is very similar to that obtained on V if the clusters in V are hyperellipsoid-shaped.*

4.1. A matrix perturbation interpretation

The matrix perturbation theory [11] can be used to explain this observation. We begin with the following distance matrix

$$\hat{E} = \begin{pmatrix} E_1 & \cdots & \cdots & \cdots \\ \cdots & E_2 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & E_k \end{pmatrix} \begin{matrix} \} & n_1 \ rows \\ \} & n_2 \ rows \\ & \\ \} & n_k \ rows \end{matrix} \tag{5}$$

where data within the same cluster are consecutively indexed. In the ideal case, $E_i = \mathbf{0}, 1 \le i \le k$, represents the distance matrix within the ith cluster, $n_1 + n_2 + ... + n_k = n$, and k denotes the number of clusters. Let $\hat{Z} = \{\hat{z}_i | i = 1, 2, ..., n\}$ with $\hat{z}_i$ being the ith row of $\hat{E}$. $\hat{z}_i \approx \hat{z}_j$ if i, j belong to the same cluster. And $d(\hat{z}_i, \hat{z}_k) \gg d(\hat{z}_i, \hat{z}_j)$ if k, i belong to different clusters. The distance relationship in the original dataset is preserved completely in $\hat{Z}$ and a properly initialized k-means algorithm on the original dataset will give the same result as that on $\hat{Z}$.

Lemma 2. *If the labeling scheme of a dataset is consistent with the transitive distance, all the samples with the same cluster label are locally connected[4] by the constructed MST of the whole dataset.*

Lemma 3. *If the labeling scheme of a dataset is consistent, the transitive distances between any sample in the same cluster and a sample from another cluster are the same.*

The proofs of Lemma 2 and 3 are omitted here and can be found in Appendix C. Lemma 3 is very useful. Intuitively, this lemma states when we consider clustering the rows of Z, the difference between any two co-cluster rows only comes from E_i, since all elements in other columns will be exactly the same. The difference from E_i by definition is also tiny. This from one perspective indicates why the property of k-means duality widely exists.

[3]With a slight abuse of notation, the dataset V in this section does not necessarily represent the original space solely. It can also represent the embedded space where the pair-wise sample Euclidean distance is characterized by the distance matrix Z.

[4]By "locally connected" we mean the path connecting the two samples on the MST only consists of other samples that are also from the same cluster

Theorem 2. *The optimal k-means clustering on Z is the same as that on V, if any intra-cluster transitive distance is less than half of any inter-cluster transitive distance.*

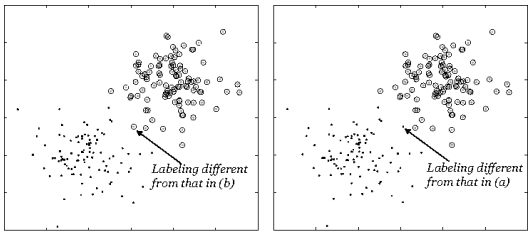
We can prove under such situations the following inequality holds: $d(z_i, z_k) > d(z_i, z_j)$ where i, j belong to the same cluster and k, i different clusters. And such inequality leads to the theorem. The assumption here seems to be strong yet we have not considered two other factors: 1. The discriminative information (difference) from inter-cluster transitive distances[5] for z_i and z_k. 2. The hyperellipsoid-shaped distribution. In general cases, this problem is relaxed to the situation where a perturbation P is added, i.e., $E = \hat{E} + P$, with all diagonal elements of P bring zero. The matrix perturbation theory [11] indicates that the k-means clustering result on the dataset Z that is derived from E is similar to that on $\hat{Z}$ if P is not dominant over $\hat{E}$. Given Lemma 3 and the mentioned two factors, the above property often holds when the intra-cluster transitive distances are larger than half of the inter-cluster transitive distances, or even when the labeling comes inconsistent. The k-means clustering strategy also makes our method considerably more robust than some other MST clustering methods through weak edge cutting. For more details the readers are recommended to refer to section 5.3 where the consistency assumption is clearly violated.

4.2. Experimental verification

We conduct a large number of experiments on different data sets to verify the above property. Most data sets were randomly generated with multi-Gaussian distributions. From more than 100 data sets where each set contains 200 samples, we compared the results obtained by the k-means algorithms on original data sets V and their corresponding sets Z. As a whole, the sample labeling difference is only 0.7%. One example is shown in Fig. 3, in which only one sample is labeled differently by the two clustering methods.

We are now able to give a solution to the problem mentioned at the end of Section 3. From Theorem 1, we know that a dataset V can be mapped to $V' \subset R^{n-1}$ where the clustering is easier if the clusters with the original distance are consistent in V. The problem we need to handle is that in R^{n-1} we only have the distance matrix instead of the coordinates of the samples in V'. Using the k-means duality in this section, we can perform the clustering based on the distance matrix by the k-means algorithm. Therefore, the main ingredients for a new clustering algorithm are already available.

[5]The intuition here is that the configuration of inter-cluster transitive distances of z_i and z_k can be quite different, as opposed to the completely identical configuration of z_i and z_j. (See Lemma 2)



(a) (b)

Figure 3. (a) Clustering result obtained by the k-means algorithm on the dataset V. (b) Clustering result obtained by the k-means algorithm on Z derived from the distance matrix of V. Only one sample has different labelings from the two results.

5. A new clustering algorithm

In this section, we give a solution to the problem at the end of Section 3, describe the corresponding algorithm and give detailed analysis.

5.1. The proposed algorithm

Given a dataset $V = \{x_i | i = 1, 2, ..., n\}$, our clustering algorithm is described as follows.

Algorithm 1 Clustering Based on the Transitive Distance and the k-means Duality

1: Construct a complete graph $G = (V, E)$ where $E = [d_{ij}]_{n \times n}$ is the distance matrix containing weights of all edges with d_{ij} being the distance between x_i, x_j.
2: Compute the transitive distance matrix $E' = [d'_{ij}] = [D_{ij}]$ based on G, where D_{ij} is the order n transitive distance between samples x_i and x_j.
3: Perform clustering on the dataset $Z' = \{z'_i | i = 1, 2, ..., n\}$ with z'_i being the ith row of E' by the k-means algorithm and then assign the cluster label of z'_i to x_i, $i = 1, 2, ..., n$.

In step 2, we need to compute the transitive distance with order n between any two samples in V, or equivalently, to find the transitive edge, which is defined below.

Definition 3. *For a weighted complete graph $G = (V, E)$ and any two vertices $x_p, x_q \in V$, the transitive edge for the pair x_p and x_q is an edge $e = x_u, x_v$, such that e lies on a path connecting x_p and x_q and $D_{pq} = D(x_p, x_q) = d(x_u, x_v)$.*

Because the number of paths between two samples is exponential in the total number of samples, the brutal searching for the transitive distance between two samples is infeasible. It is necessary to design a faster algorithm to carry out this task. Theorem 3 is proved for this purpose.

Theorem 3. *Given a weighted complete graph $G = (V, E)$ with distinct weights, each transitive edge lies on the minimum spanning tree $\tilde{G} = (V, \tilde{E})$ of G.*

For the proof of Theorem 3, please refer to Appendix B in the supplementary material. The theorem suggests an efficient algorithm to compute the transitive matrix $E' = [d'_{ij}]_{n\times n}$ which is shown in Algorithm 2.

Algorithm 2 Computing the transitive distance matrix $E' = [d'_{ij}]_{n\times n}$

1: Build the minimum spanning tree $\tilde{G} = (V, \tilde{E})$ from $G = (V, E)$.
2: Initialize a forest $F \leftarrow \tilde{G}$.
3: **Repeat**
4: **For** each tree $T \in F$ **do**
5: Cut the edge with the largest weight w_T and partition T into T_1 and T_2.
6: **For** each pair (x_i, x_j), $x_i \in T_1$, $x_j \in T_2$ **do**
7: $d'_{ij} \leftarrow w_T$
8: **End for**
9: **End for**
10: **Until** each tree in F has only one vertex.

5.2. Relation to hierarchical clustering and EGS

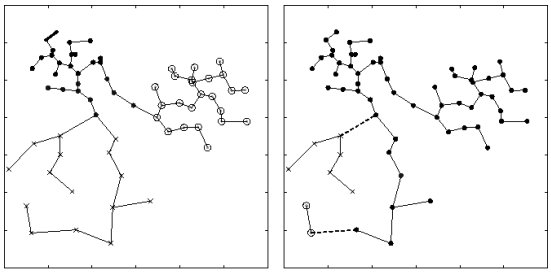
Although MST is used in both methods, the motivations are quite different. Our purpose is to generate a non-linear embedding with which the k-means algorithm provides a top-down optimization, whereas in hierarchical clustering, each iteration only focuses on local structure. This leads to significant differences. We carry out the proposed method and hierarchical clustering on the same test data. Fig. 4 shows the clustered results by the two approaches.

EGS is essentially a method where the cuts on MST are smartly chosen. Despite the fact that EGS is proved to be "neither too coarse nor too fine", the method still suffers from over-merging at weak boundaries like other bottom-up contour finding methods. Alternatively, our method focuses on modeling the intra-cluster similarity with top-down information included, suffering less from this problem.

5.3. Relation to graphical mode seeking and NTP

Our method also shows close relationship to the graphical mode seeking [14] and the normalized tree partition [17]. Paper [14] uses an MST to model the intrinsic structures. Different from this work, [14] did not incorporate the transitive distance information. Intuitively, such clustering formulation is prone to over splitting in situations where a cluster is diffusively distributed while the minimum inter-cluster gaps between this cluster and other clusters are not significant. Our method on the other hand share similar characteristics with spectral clustering methods, which are good at handling this situation. [13].

In [17], the authors also use tree structures to represent the inherent data structure and the tree is partitioned k-way by checking normalized cut scores. The spectral clustering formulation between our work and theirs, however, is quite different. In a sense, [17] used a tree to significantly reduce the space of possible cut configurations, while our work does not have this limit.



(a) (b)

Figure 4. (a) The minimum spanning tree and the clustering result by our algorithm. (b) The minimum spanning tree and the clustering result by the hierarchical clustering. The dashed lines are the cutting edges. The number of clusters is 3.

6. Experiments

We apply our algorithm to a number of clustering problems. The results are compared with those by the k-means algorithm, the spectral clustering algorithm (NJW) [9] and the self-tuning spectral clustering algorithm [13]. For each dataset, the NJW algorithm needs manually tuning of the scale and the self-tuning algorithm needs to set the number of nearest neighbors. We show the best clustering results that are obtain by adjusting the parameters. The numbers of clusters are all assumed to be known.

6.1. Synthetic datasets

Eight synthetic data sets are used in the experiments. Bounded in a region $(0, 1) \times (0, 1)$, these data sets are with complex cluster shapes, multi-scale clusters, and noise. The clustering results are shown in Fig. 5. Note that the results obtained by k-means are not given because it is obvious that it cannot deal with these data sets.

In Fig. 5(a)-(c), all the three algorithms obtain the same results. Fig. 5(d)-(f) and (g)-(i) show three data sets on which the self-tuning algorithm gives different results from the other two algorithms. The self-tuning algorithm fails to cluster the data sets no matter how we tune its parameter. Fig. 5(j) and (k) show two clustering results where the dataset is with multi-scale clusters. The former is produced by the NJW algorithm and the latter by the self-tuning and our algorithms. To cluster the dataset in Fig. 5(l)-(n) is a challenging task, where two relatively tightly connected clusters are surrounded by uniformly distributed noise samples (the third cluster). Our algorithm obtains the more reasonable result (Fig. 5(l)) than the results by another two algorithms (Figs. 5(m) and (n)). On the synthetic dataset, we can see that our algorithm performs similar to or even better than the NJW and self-tuning spectral clustering.

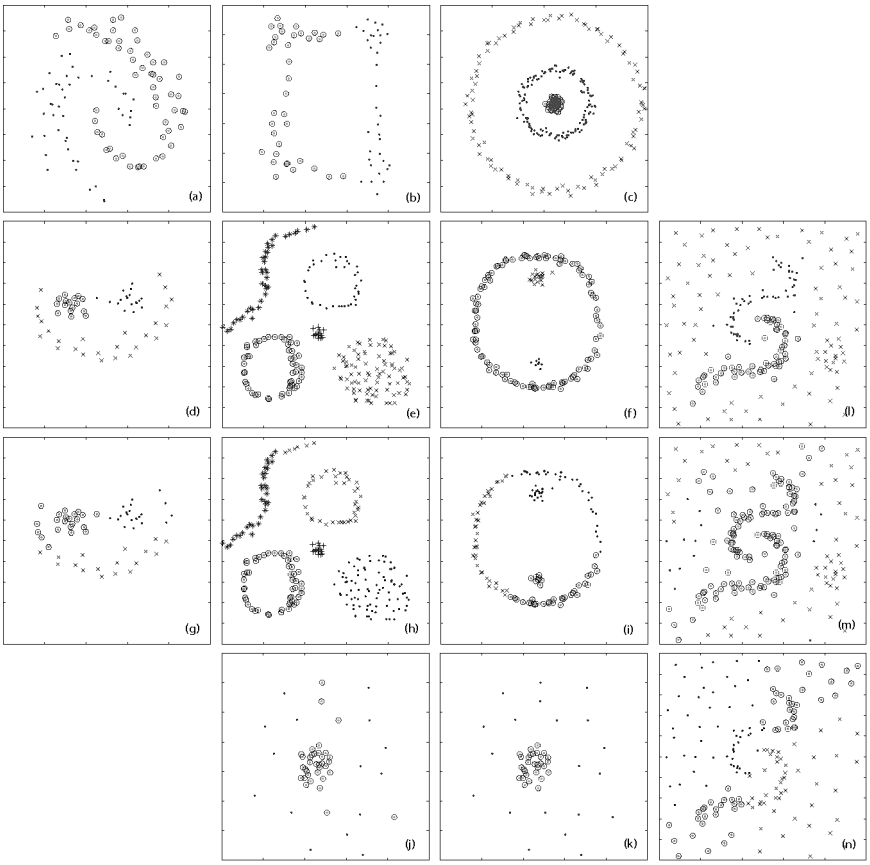


Figure 5. Clustering results by our algorithm and the two spectral algorithms. (a)(b)(c) Results by the three algorithms. (d)(e)(f) Results by the NJW algorithm and ours. (g)(h)(i) Results by the self-tuning algorithm. (j) Result by the NJW algorithm. (k) Result by the self-tuning algorithm and ours. (l)(m)(n) Results by our algorithm, the NJW algorithm, and the self-tuning algorithm, respectively.

6.2. Datasets from the USPS database

There are 9298 handwriting digit images of size 16×16 from "0" to "9" in the USPS database, from which we construct ten data sets from this database. Each set has 1000 images selected randomly with two, three, or four clusters. Each image has a 256-dimensional feature. Fig. 6 shows the error rates of the four algorithms on these sets. The parameters for the NJW and self-tuning algorithms are tuned to obtain the smallest error rates. These results show that as a whole, our algorithm achieves the best performance.

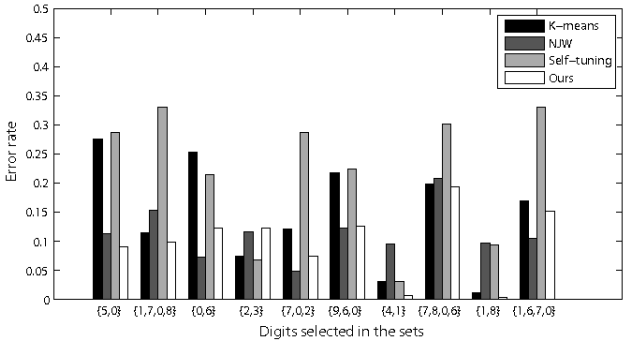


Figure 6. The error rates of four algorithms on ten data sets constructed from the USPS database.

6.3. Iris and Ionosphere datasets

We also test the algorithms on two commonly-used data sets, Iris and Ionosphere, from UCI machine learning database. In Table 2 we show the error rates of our clustering algorithm compared with k-means (KMS), NJW and self-tuning. For the NJW and self-tuning algorithms, we have to adjust their parameters (δ and N)[6] to obtain the smallest error rates, which are shown in the table. Our algorithm results in the smallest error rates.

[6]We tried different δ from 0.01 to 0.1 with step 0.001 and 0.1 to 4 with step 0.1, and different N from 2 to 30 with step 1.

6.4. Image segmentation

We conduct segmentation experiments on the Berkeley segmentation dataset. Segmentation is conducted on top of superpixels obtained by the method in [22], with texton histograms extracted from each superpixel and the χ^2 distance

Table 1. Error rates of the four algorithms on Iris and Ionosphere data sets

	KMS	NJW	Self-Tuning	Ours
Iris	0.11	$0.09(\delta = 0.4)$	$0.15(N = 5)$	**0.07**
Iono	0.29	$0.27(\delta = 0.2)$	$0.30(N = 6)$	**0.15**

modeling dissimilarity. We compare the results with normalized cuts [10], self-tuning spectral clustering, EGS [14] under the same superpixelization and dissimilarity setting. Sparse affinity matrices were constructed on top of the dissimilarity measure for the two spectral methods. We also recommend readers compare these results with [23], where the testing sequence and settings are very similar.

Other than qualitative evaluation, we conduct quantitative evaluations. We compare our results to NTP [17], multiscale graph decomposition (MGD) [20] and probablistic rand index label fusion (PRIF) [24] in terms of 1) Probabilistic Rand Index (PRI), 2) Variation of Information (VoI), 3) Global Consistency Error (GCE), and 4) Boundary Displacement Error (BDE). From the results we can see our method generates comparable or even better results compared with other major clustering methods. Note that [24] is a method that is specifically optimized over PRI. The performance gap between two methods is marginal. The result has not yet included techniques that are actually very helpful to boost performance. A simple pre-clustering would further increase every benchmark score over all baselines.

Having an intuitive clustering process endows much more convenience on many direct extensions than eigendecomposition based methods. For example, we can extend our method to the MRF k-means where the label configuration is further penalized by the Potts model.

Table 2. Quantitative segmentation evaluation

	PRI	VoI	GCE	BDE
MGD	0.7559	2.4701	0.1925	15.10
NTP	0.7521	2.4954	0.2373	16.30
Ncut	0.7853	2.1031	0.1947	**12.9703**
PRIF	**0.8006**	–	–	–
Ours	0.7926	**2.0871**	**0.1835**	13.1707

7. Conclusion

In this paper, we have built a connection between the transitive distance and the kernel technique for data clustering. By using the transitive distance, we show that if the consistency conditions is satisfied, the clusters of arbitrary shapes can be mapped to a new space where the clusters are easier to be separated. Based on the observed k-means duality, we have developed an efficient algorithm that does not need to solve the traditional eigen-decomposition problem. Compared with the k-means algorithm, both our algorithm and the spectral algorithms can better handle challenging clustering problems where the data sets are with complex shapes, multi-scale clusters, and noise. The image segmentation experiments also show the superiority and practical application value of our proposed method.

There are associated drawbacks as well. MST sometimes is an over-simplified and non-regularized representation of the underlying structure, and it may cause clustering errors around cluster margins. "Short cutting" is another problem. This, however, can be alleviated in many ways, such as joint clustering with transitive distances from different MSTs, sampling and local density estimation. Further improvements will be included in our future work.

Acknowledgement

This research was supported by 973 Program of China with No. 3202013CB329404 and the NSFC projects with No. 61005011, 61373114, 11131006, 6107505.

References

[1] B. Fischer and J. M. Buhmann. Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation. *IEEE Trans. PAMI*, 2003. 2

[2] B. Fischer, V. Roth and J. M. Buhmann. Clustering with the connectivity kernel. In *NIPS*, 2004. 2

[3] I.S. Dhillon, Y. Guan and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *ACM SIGKDD*, 2004. 1

[4] C. Ding, X. He, H. Xiong and H. Peng. Transitive closure and metric inequality of weighted graphs: detecting protein interaction modules using cliques. *Int. J. of Data Mining and Bioinformatics*, 2006. 2

[5] M. Fiedler. Ultrametric sets in euclidean point spaces. *Electronic J. of Linear Algebra*, 1998. 2

[6] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Networks*, 2002. 1

[7] A.K. Jain, M.N. Murty and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 1999. 1

[8] A.Y. Lemin, Isometric imbedding of isosceles (non-Archimedean) spaces into Euclidean ones. *Dokl. Akad. Nauk SSSR* 285:558-562, 1985. 2

[9] A.Y. Ng, M. Jordan and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001. 1, 5

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 2000. 1, 7

[11] G.W. Stewart and J. Sun. Matrix Perturbation Theory. *Computer Science and Scientific Computing*, 1990. 3, 4

Figure 7. Qualitative segmentation result comparison. From top to bottom: Normalized Cuts, Self-Tuning, EGS and Our method.

[12] D. Verma and M. Meila. A comparison of spectral clustering algorithms. *Technical report 03-05-01*, Dept of CSE, Univ. of Washington, 2003. 1

[13] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004. 1, 5

[14] Z. Yu, et al.. Nonparametric Density Estimation on A Graph: Learning Framework, Fast Approximation and Application in Image Segmentation. In *CVPR*, 2011. 2, 5, 7

[15] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 2004. 2

[16] D. Yan, L. Huang, M.I. Jordan. Fast Approximate Spectral Clustering. *ACM SIGKDD*, 2009. 1, 2

[17] J. Wang, et al.. Normalized Tree Partitioning for Image Segmentation. *CVPR*, 2008. 2, 5, 7

[18] W. Chen, Y. Song, H. Bai, C. Lin and E. Chang. Parallel Spectral Clustering in Distributed Systems. *PAMI*, 2011. 1

[19] C. Fowlkes, S. Belongie, F. Chung and J. Malik. Spectral Grouping Using the Nystrom Method. *PAMI*, 2004. 1

[20] T. Cour, F. Benezit and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. *CVPR*, 2005. 7

[21] M. Li, X. Lian, J. Kwok and B. Lu. Time and Space Efficient Spectral Clustering via Column Sampling. *CVPR*, 2011. 1

[22] G. Mori. Guiding Model Search Using Segmentation. *ICCV*, 2005. 6

[23] Z. Yu, A. Li, O.C. Au and C. Xu. Bag of Textons for Image Segmentation via Soft Clustering and Convex Shift. *CVPR*, 2012. 7

[24] M. Mignotte. A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation. *IEEE Trans. on Image Proc.*, 2010. 7