Generalized Transitive Distance with Minimum Spanning Random Forest

Zhiding Yu[1], **Weiyang Liu**[2], **Wenbo Liu**[1], **Xi Peng**[3], **Zhuo Hui**[1], **B. V. K. Vijaya Kumar**[1]
[1]Dept. of Electrical and Computer Eng., Carnegie Mellon University
[2]School of Electronic and Computer Eng., Peking University, P.R. China
[3]I2R, Agency for Sci., Tech. and Research (A*STAR), Singapore
yzhiding@andrew.cmu.edu, wyliu@pku.edu.cn, pangsaai@gmail.com, kumar@ece.cmu.edu

Abstract

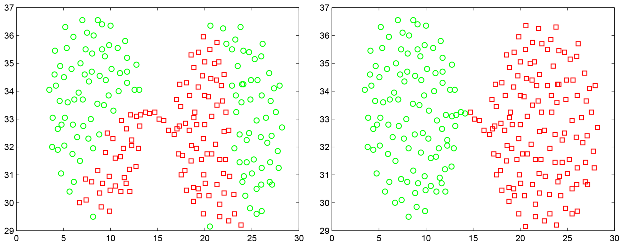
Transitive distance is an ultrametric with elegant properties for clustering. Conventional transitive distance can be found by referring to the minimum spanning tree (MST). We show that such distance metric can be generalized onto a minimum spanning random forest (MSRF) with element-wise max pooling over the set of transitive distance matrices from an MSRF. Our proposed approach is both intuitively reasonable and theoretically attractive. Intuitively, max pooling alleviates undesired short links with single MST when noise is present. Theoretically, one can see that the distance metric obtained max pooling is still an ultrametric, rendering many good clustering properties. Comprehensive experiments on data clustering and image segmentation show that MSRF with max pooling improves the clustering performance over single MST and achieves state of the art performance on the Berkeley Segmentation Dataset.

1 Introduction

Over the past decades, clustering has been and is still one of the most important and fundamental machine learning problem. A number of clustering methods have been proposed, ranging from the famous k-means algorithm and graph-based approaches (such as single linkage algorithm) [Sibson, 1973], to the family of mode seeking [Comaniciu and Meer, 2002], spectral clustering [Ng *et al.*, 2002; Zelnik-Manor and Perona, 2004; Shi and Malik, 2000], and subspace clustering [Elhamifar and Vidal, 2009; Liu *et al.*, 2013; 2013; Peng *et al.*, 2013; 2015]. Despite the large variety of different methods, some general principles are commonly considered when evaluating the performance among different methods. These principles include:

- Ability to discover clusters with arbitrary shape.
- Robustness against noise
- Scalability

The family of spectral clustering methods received much attention and found wide applications for the excellent clustering performance. Given n data points, eigendecomposition is conducted on an $n \times n$ normalized pairwise similarity matrix, followed by k-means to generate clusters. A key reason for spectral clustering's success lies in its ability to discover non-convex latent structures. This comes from the the fact that eigendecomposition projects data in to a kernel space with nicely shaped clusters.



(a)

(b)

Figure 1: (a) Clustering with transitive distance on a single MST. (b) Clustering with the proposed framework.

Spectral clustering is not the only family of methods that can handle clusters with arbitrary shapes. Transitive distance clustering (also known as path based clustering) provides an elegant and intuitive non-eigendecomposition alternative also effective in handling non-convex clusters. Specifically, transitive distance emphasizes the connectivity rather than absolute distance between pairwise data. This is achieved by finding the set of largest hops (edges) along all possible connecting paths and defining the pairwise distance as the minimum hop. [Fischer and Buhmann, 2003b] proposed the concept of transitive distance and an agglomerative bottom-up clustering framework. The idea of connectivity kernel was later proposed in [Fischer *et al.*, 2004]. Other works include the transitive closure [Ding *et al.*, 2006] and transitive affinity [Chang and Yeung, 2005; 2008].

There exist some inherent connections between transitive distance and minimum spanning tree. It was proved that the transitive distance edge for all pairwise data lies on the minimum spanning tree, if the maximum order (number of nodes) of a path is equal to n. This, however, does not necessarily mean that transitive distance clustering is identical to early graph based method such as single linkage algorithm. There are many nice properties associated with transitive distance, one of them being that the transitive distance is an ultrametric and can be embedded into another space with better cluster shapes. This means transitive distance can serve a similar role to eigendecomposition and makes it possible to design a top-down clustering framework with transitive distance [Yu *et al.*, 2014]. Such framework can be regarded as an approximate spectral clustering and behaves much more robust than many MST based bottom-up methods.

Despite the fact that top-down clustering benefits the algorithm with noise robustness. Transitive distance may still suffer from noisy short links due to its bottom up nature. An example is illustrated in Fig. 1(a). Resampling based bagging [Fischer and Buhmann, 2003a] was shown to be effective against short links. However, when the average cluster size becomes small, resampling turns out to lose considerable discriminative cluster information and label permutation becomes computationally expensive. In addition, resampling requires a totally connected graph while many applications such as image segmentation often works better with sparsely connected adjacency graphs.

In this paper, we extend the concept of transitive distance from a single MST to a minimum spanning random forest. Our contribution in this paper lies in several aspects: (1) The proposed framework is a generalization of the conventional transitive distance framework; (2) The proposed framework presents an alternative strategy that naturally generates distance metric more robust to noise (See Fig. 1(b)); (3) With max pooling, the proposed framework preserves many nice properties for clustering. We conduct comprehensive experiments ranging from challenging toy examples to large scale speech data clustering (up to tens of thousands of samples and thousands of clusters) and image segmentation. Our method shows excellent performance, including the state of the art results on Berkeley Segmentation Dataset.

2 The Generalized Transitive Distance (GTD)

The goal of spectral clustering is to cluster data that is connected but not necessarily compact or clustered within convex boundaries. A common way to handle data with arbitrary cluster shapes is to use kernel method to create a nonlinear mapping

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

such that the clusters in $\mathbf{R}^s$ has a more compact cluster shape. In spectral clustering, such nonlinear mapping is often obtained by eigendecomposition on the normalized variants of its affinity matrix (Such as the Laplacian matrix).

2.1 Definition of The GTD

Transitive distance seeks to implicitly build a non-linear mapping similar to spectral clustering without eigendecomposition. The pairwise transitive distance for any pairwise data is defined as follows:

Definition 1. *Given certain pairwise distance $d(\cdot,\cdot)$, the transitive distance is defined as:*

$$D_T(x_p, x_q) = \min_{\mathcal{P}\in\mathbb{P}} \max_{e\in\mathcal{P}}\{d(e)\}, \tag{2}$$

where $\mathbb{P}$ is the set of paths connecting x_p and x_q with at most n nodes. In addition:

$$\max_{e\in\mathcal{P}}\{d(e)\} = \max_{(x_u,x_v)\in\mathcal{P}}\{d(x_u, x_v)\}. \tag{3}$$

Intuitively, transitive distance looks into the connectivity between pairwise data by searching for the minimum gap among all possible paths. Even if the Euclidean distance is far away, two data samples are close if there is at least one path that strongly correlates them.

To further allow more robustness, a natural and reasonable generalization is to extend such definition to the case of considering multiple paths instead of a single path. Here we propose the following generalized pairwise transitive distance:

Definition 2. *Given certain pairwise distance $d(\cdot,\cdot)$, the generalized transitive distance is defined as:*

$$D_G(x_p, x_q) = \max_t \underset{\substack{\mathcal{P}_t\in\mathbb{P}_t,\\ \forall t\in\{1,\ldots,T\}}}{\mathrm{gmin}} \max_{e\in\mathcal{P}_t}\{d(e)\}, \tag{4}$$

where gmin *denotes the generalized* min *returning a set of minimum values from multiple sets instead of just one minimum value from one set. $\mathbb{P}_t$ denote the sets of all candidate paths respectively from multiple diversified graphs G_t.*

The idea of bagging is a widely used strategy to increase robustness. Similar ideas can be found in many other machine learning problems, such as extending a decision tree classifier to random forest classifier.

2.2 Kernel Trick with The GTD

Definition 3. *A distance metric D is called an ultrametric if it satisfies non-negativity, symmetry, identity of indiscernibles and the following strong triangularity:*

$$D(x_i, x_j) \le \max\{D(x_i, x_k), D(x_k, x_j)\}, \forall\{i, j, k\}. \tag{5}$$

Lemma 1. *The transitive distance is an ultrametric.*

The corresponding proof is very easy and can be found in [Fischer *et al.*, 2004] and [Ding *et al.*, 2006].

Theorem 1. *The proposed generalized transitive distance is an ultrametric.*

Proof: According to Lemma 1, the pairwise transitive distance defined over every set of G_t satisfies:

$$D_T^t(x_i, x_j) \le \max\{D_T^t(x_i, x_k), D_T^t(x_k, x_j)\}, \forall t. \tag{6}$$

Therefore, we have:

$$\begin{aligned} &\max_t D_T^t(x_i, x_j) \\ \le &\max_t \max\{D_T^t(x_i, x_k), D_T^t(x_k, x_j)\} \\ = &\max\{\max_t D_T^t(x_i, x_k), \max_t D_T^t(x_k, x_j)\} \end{aligned} \tag{7}$$

Proposition 1. *The generalized transitive distance metric can be embedded into an $n-1$ dimensional vector space.*

Proposition 1 directly comes from the lemma that every finite ultrametric space with n discernible points can be embedded into an $n-1$ dimensional vector space. More details of this lemma can be found in [Lemin, 1985; Fiedler, 1998]. The proposition conveys the following important information: with the generalized transitive distance, we have an implicit nonlinear mapping:

$$\phi : (V \subset \mathbf{R}^l, D) \mapsto (V' \subset \mathbf{R}^{n-1}, d'), \tag{8}$$

where $d'(\phi(x_i), \phi(x_j)) = D(x_i, x_j)$ and $d'(\cdot, \cdot)$ is the Euclidean distance in $\mathbf{R}^{n-1}$. Such mapping plays a similar role to the kernel trick in spectral clustering except that it is an implicit mapping where one does not have the mapped feature in the kernel space but the pairwise Euclidean distance.

We now analyze conditions and properties associated with cluster distributions in the nonlinearly projected space. Similar to spectral clustering, one would ideally hope that the data projected by the transitive distance form well separated clusters in the embedded space. [Yu *et al.*, 2014] shows that if a labeling scheme of a dataset is consistent[1] with certain distance (say Euclidean), then the convex hulls of clusters in the projected transitive distance space do not intersect with each other. However, in the case of generalized transitive distance, the same condition no longer holds and whether the projected clusters will intersect now dependent on the specific strategy one chooses to generate sets of candidate paths.

To analyze the clustering properties, we cut into this problem from a even more direct perspective which does not depend on any specific strategy one chooses. We first need to redefine the concept of consistency:

Definition 4. *A distance metric is called strictly consistent with the labeling scheme of a dataset, if for any intra-cluster pair x_i, x_j and any x_k with a different cluster label, the following relation holds:*

$$D(x_i, x_j) < \min(D(x_i, x_k), D(x_j, x_k)), \forall\{i, j, k\} \quad (9)$$

Note that the consistency term redefined here differs from the conventional definition of "consistency" in [Yu *et al.*, 2014]. Its definition is no longer related to the method that constructed such distance. An underlying relation is that the "consistency" (conventional) in the original data space is a sufficient condition for the consistency (redefined) in the transitive distance space returned by a single MST. But such relation no longer holds under GTD.

Theorem 2. *If a distance metric satisfies the strict consistency with the labeling scheme and there exist an Euclidean embedding, the convex hulls of the images of clusters in the embedded space do not intersect with each other.*

Proof: We prove this theorem using contradiction. Suppose (x_i, x_j) are the pair of samples from cluster C in the embedded space returning the maximum possible pairwise intra-cluster distance. Also assume there exist a point y with a different label such that y lies in the convex hull of C. Since y also lies in this convex hull, by definition there must exist a linear combination such that $y = \sum_{k=1}^{|C|} \alpha_k x_k$, which leads to $(y - x_j) = \sum_{k=1}^{|C|} \alpha_k (x_k - x_j)$. In this case, it is very easy to prove that $||y - x_j||_2$ is upper bounded by $||x_i - x_j||_2$, which contradicts with the fact that $||y - x_j||_2 > ||x_i - x_j||_2$.

Proposition 2. *For the GTD to be consistent with the label, the following inequality should be satisfied:*

$$\begin{aligned} &\max_t D_t(x_i, x_j) \\ &< \min(\max_t D_t(x_i, x_k), \max_t D_t(x_j, x_k)), \forall\{i, j, k\} \end{aligned}. \quad (10)$$

This is a direct conclusion from definition of GTD and (9). (10) intuitively shows why the proposed framework can be robust against noise. In the extreme case, even if none of the G_t return a transitive distance consistent to the label, there is a chance that (10) can still be satisfied. Considering the fact that a point inside a cluster often has a much larger degree of nearby points than that on a margin (due to the nature of density difference), The chance of recovery can further increase considerably with more tree numbers.

2.3 Locating Transitive Edges

We now generalize transitive edge localization under our framework. The following lemma efficiently locates the transitive edge for any pairwise data:

Lemma 2. *Given a weighted graph, the transitive edge lies on the minimum spanning tree of the graph.*

The proof can be found in [Fischer *et al.*, 2004; Yu *et al.*, 2014]. The lemma basically states that there is an computationally feasible way to find the transitive distance given any pairwise data. To efficiently find the pairwise generalized transitive distance, we have the following proposition:

Proposition 3. *Given the sets of candidate paths, the transitive distance edge lies on the MSRF formed by MSTs extracted from each set of candidate paths.*

This is a direct extension of Lemma 2 and the proof is omitted here. The proposition shows a general process to obtain the generalized pairwise transitive distance. Detailed algorithm for obtaining the transitive distance with an MST can be found in [Fischer *et al.*, 2004].

3 Random Forest Generation

The remaining issue is how to obtain different sets of candidate paths that generates diversified trees. Here we propose two exemplar methods that gives diversified sets of trees. Note that many other alternative strategies can also be incorporated into the proposed framework.

3.1 Generating Mutually Exclusive Trees

A very intuitive and effective way to generate diversified random forest is to force the minimum spanning trees to be completely non-overlapping. In other words, the edges that has previously been used to construct a minimum spanning tree can not be used for subsequent ones. The sets of candidate paths and the minimum random spanning forest can be generated with the extended sequential Kruskal's algorithm:

Algorithm 1 Extended Sequential Kruskal's Algorithm

```
1: Initialize $G_1 = G = (V, E)$, where $G$ is a weighted
   graph and $E$ is the set of available edges.
2: Extract MST from $G_t$ using the Kruskal's algorithm and
   return the $n \times n$ pairwise transitive distance matrix.
3: Remove the set of MST edges $P_t$ from $G_t$ and update:
   $G_{t+1} = (V, E_t - P_t)$.
4: Repeat 2 to 4 for $T$ times.
5: Perform element wise max pooling over the stack of
   transitive distance matrices.
```

[1] The readers please kindly refer to the original paper for detailed conventional definition of the term "consistency".

Theorem 3. *Without loss of generality, let $G = (V, E)$ be the complete graph, $V_c = \{x_i | i = 1, ..., n_c\} \subset V$ be the set of nodes from cluster c, G_c and $\widetilde{G}_c$ respectively be the intra-cluster graph and the inter-cluster graph of c. Also let the constructed MSRF containing T trees from G with Algorithm1. A sufficient condition for the GTD to be consistent with the cluster label is that $\forall c$, the following inequality holds:*

$$\max_{(x_i, x_j) \subset V_c} D_G(x_i, x_j) < \mathrm{sort}(\widetilde{G}_c, T), \tag{11}$$

where $\mathrm{sort}(\widetilde{G}_c, T)$ refers to returning the Tth largest edge from $\widetilde{G}_c$.

Proof: Let y denote any point with a different label. It can be verified that $\min(D_G(x_i, y), D_G(x_j, y))$ is lower bounded by $\mathrm{sort}(\widetilde{G}_c, T)$, for $\widetilde{G}_c$ defines the gap between c and other clusters. Since we construct a minimum spanning random forest with T non-overlapping trees and adopt max pooling, the minimum possible inter-cluster distance therefore is defined by the Tth largest edge in $\widetilde{G}_c$.

In a sense, Theorem 3 can be regarded as a random forest generalization of the "consistency" in [Yu *et al.*, 2014].

3.2 Generating Perturbated Trees

Many clustering applications in reality have small cluster sizes, or sparse edge connections on the graph. An example with small cluster sizes is the speech dataset [Greenberg *et al.*, 2014] which has thousands of clusters (speaker identities), each sometimes containing as few as three or four samples. Image segmentation is another case where sparsely connected graph such as the region adjacency graph is often preferred. There are reasons for such preference. One is that segmentation is not a pure clustering problem, but a perceptual grouping problem also emphasizing spatial continuity besides cluster compactness. Another reason being that many state of the art boundary features such as gPb[Arbelaez *et al.*, 2011] and structured random forest edge detection[Zelnik-Manor and Perona, 2004] are edge-oriented and by nature only work with neighboring superpixels.

Forcing to select non-overlapping MSTs under such cases can reduce the discriminative cluster information. A possibly better solution is not to diversify the trees so aggressively, and allow overlap of the trees. We propose the following alternative algorithm to generate randomized trees:

Algorithm 2 Random Perturbation Algorithm

```
1: Initialize $G_1 = G = (V, E)$, where $G$ is a weighted graph and $E$ is the set of available edges.
2: If $t \neq 1$, obtain $G_t$ by randomly perturbate the edge length of $G$ with a random number $\epsilon * rand(1)$.
3: Extract MST from $G_t$ using the Kruskal's algorithm and return the $n \times n$ pairwise transitive distance matrix.
4: Repeat 2 to 4 for $T$ times.
5: Perform element wise max pooling over the stack of transitive distance matrices.
```

It is also very interesting to intuitively look into the reason why such perturbation strategy works. One of the key reasons again being that the degree of closely neighboring on points inside a cluster can be much larger than marginal ones. There is a higher chance that edges on undesired short cut links gets magnified compared with intra cluster edges, since a point inside a cluster still gets considerably many choices for short paths due to dense neighboring samples.

4 Top-Down Clustering

It is intuitively very attractive to directly performing k-means in the projected transitive distance space. Unfortunately, with the explicit nonlinear mapping missing, finding an optimal cluster partitioning with a pairwise distance matrix is difficult [Fischer and Buhmann, 2003b]. [Yu *et al.*, 2014] proposed an approximation which directly treats each row of the distance matrix as a single data and performs k-means over the rows. Since the clusters after projection become much more compact, the transitive distance matrix can be approximately regarded as an ideal block matrix plus additional noise:

$$D = D_{block} + E \tag{12}$$

Performing k-means on the rows of D can be regarded as certain low rank approximation to recover D_{block}, which inherently is closely related to directly performing k-means in the original data space. In addition, top-down methods in general shows more robustness against noise compared to bottom-up methods. Therefore, such top-down form also benefits transitive distance clustering, making it considerably different from MST based methods. Therefore, the top-down approximation is more favorable for this paper.

We also propose an optional alternative which may further improve the top-down clustering performance. When the dimensionality of D is huge, a considerable number of columns in D contains noise information. Instead of directly performing k-means on the rows the full matrix, one can perform singular value decomposition on $D \approx U\Sigma V^*$ for low rank approximation, followed by k-means on the first several column of U. The following property states the inherent relationship between the original top-down strategy and the new one:

Property 1. *The matrix U approximately equals to the normalized columns of D_{block} if E is small*

In the experiment, we will show results from both top-down clustering strategies.

5 Experiments

5.1 Toy Example Datasets

In this section, we conduct experiment on a set of very challenging toy examples to test the algorithm performance. We compare our results with several popular spectral clustering methods including spectral clustering[Ng *et al.*, 2002], self-tuning spectral clustering[Zelnik-Manor and Perona, 2004] and normalized cuts[Shi and Malik, 2000]. Figure 2 shows a set of toy example clustering results. Overall, We have carefully tuned the scale parameters for both spectral clustering and normalized cuts on each dataset. The number of trees is 3 for GTD (Seq. Kruskal), and 20 for GTD (Perturb.). The perturbation strength ϵ is set to 2. One could see that GTD + SVD performs the best, getting almost all correct on every

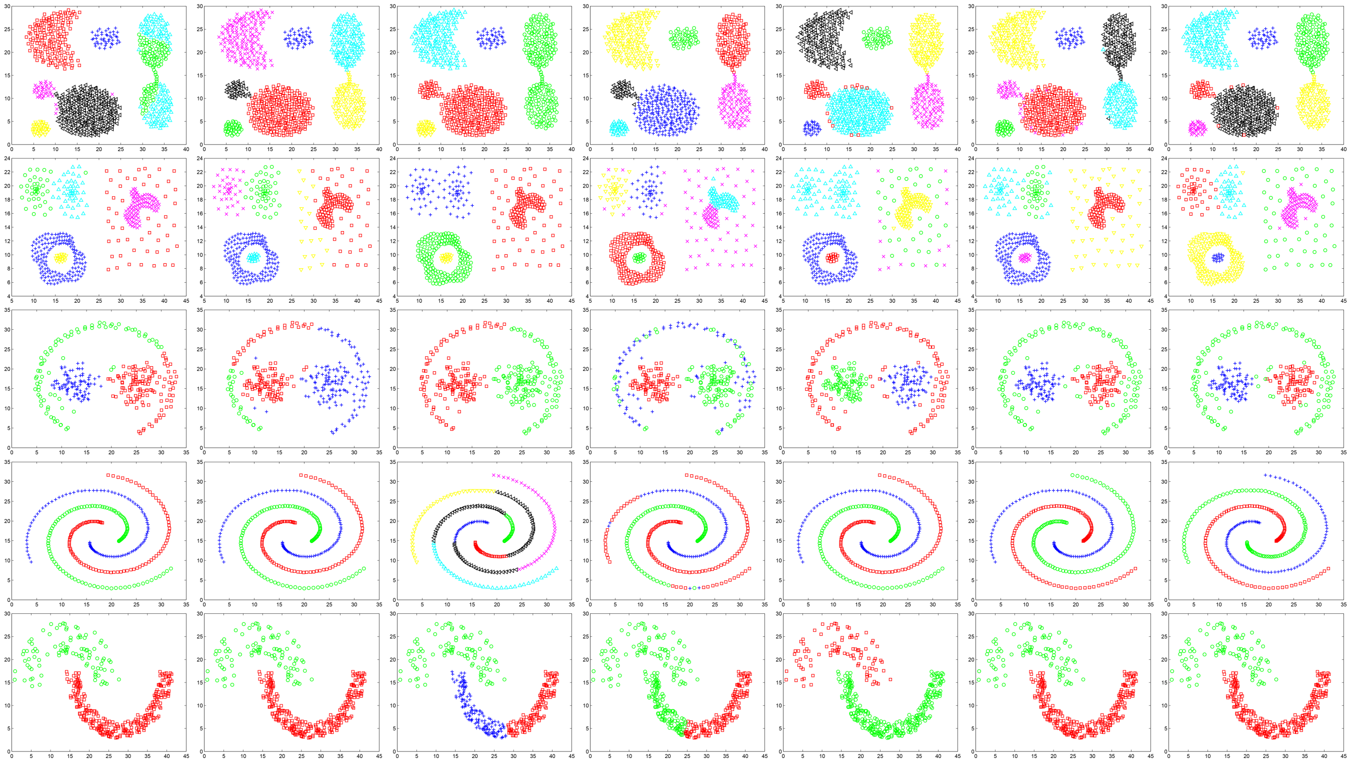


Figure 2: Column 1: Transitive + SVD. Column 2: Spectral clustering. Column 3: Self-tuning spectral clustering (auto scale + cluster num). Column 4: Normalized cuts. Column 5: GTD (Seq. Kruskal). Column 6: GTD (Perturb.). Column 7: GTD (Seq. Kruskal) + SVD. Note that since GTD (Perturb.) + SVD also obtains similar correct results, the figures are omitted.

toy example. Both transitive distance and spectral clustering showed very strong flexibility on non-convex clusters. Results on the second toy example, however, indicates stronger ability of transitive distance in handling multi-scale clusters. In addition, the first and third toy example clearly show that both GTD bagging strategies to some extent reinforced robustness against short links.

5.2 Large Scale Speech Data Clustering

We also consider the application to large scale unsupervised learning of speech samples. A recent hot topic in the speech community is whether one can train a high quality speaker verification model given large quantities of unlabeled speech data. The NIST i-Vector Machine Learning Challenge 2014 (i-Vector) [Greenberg *et al.*, 2014] organized competitions to design unsupervised speaker verification systems with fully unlabeled i-vector development dataset, in which clustering and unsupervised learning methods were heavily emphasized. The i-Vector dataset consists of 36572 600-dimensional pre-extracted i-vectors with 4958 identities. In addition to the i-Vector dataset we also form another large scale dataset (NIST) with the NIST SRE 2004, 2005, 2006 and 2008 corpora on the telephone channel. A total of 21704 500-dimensional i-vectors with 1738 identities were extracted under the framework of [Li and Narayanan, 2014].

With such large scale dataset size and cluster number, the problem becomes very challenging. We input the groundtruth number of clusters for all method and measure the cluster purity (accuracy). Since the cluster sizes are very small, we choose the prosed GTD (perturb.) and GTD (perturb.) + SVD and compare it with other baseline methods. T which is the number of trees are set to 30 for both the NIST dataset and the i-Vector dataset. The perturbation strength ϵ are respectively set to 0.015 and 0.03. Table 1 shows the quantitative results of the proposed methods and other compared methods. In particular, the proposed GTD (Perturb.) + SVD achieves the best performance. It is worth noticing that the single linkage algorithm completely failed on the i-Vector dataset primarily due to serious short link caused by bottom-up clustering. This partially reveals the significant difference between bottom-up and top-down strategies, despite similarity input of two methods are strongly related.

5.3 Image Segmentation

We conduct image segmentation experiments on the BSDS300 dataset. The images are first superpixelized with the code from [Dollár and Zitnick, 2014] and the edge probability maps are extracted using structured random forest. We also consider the texton similarity between pairwise superpixels. The input for GTD and transitive distance is a region adjacency graph weighted by both the χ^2 distance between neighboring superpixels and the average edge response along their boundaries. For normalized cut, the input is a sparse affinity graph where only neighboring superpixels

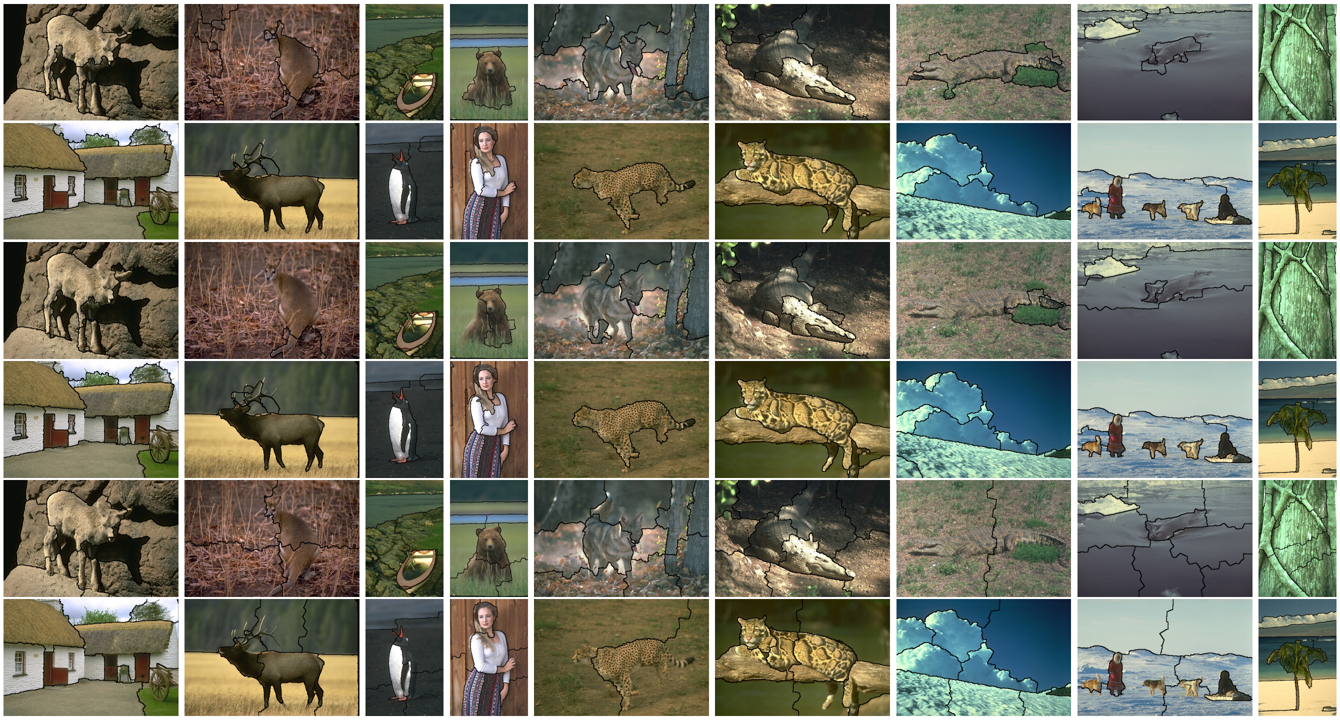


Figure 3: Examples of segmentation results. Row 1-2: Results from GTD clustering. Row 3-4: Results from transitive distance clustering. Row 5-6: Results from normalized cuts.

Table 1: Quantitative Speech Clustering evaluation

Method	NIST	Ivector
Normalized Cuts	0.4883	0.3654
Single Linkage	0.4544	0.156
Spectral Clustering	0.6841	0.4898
[Fischer and Buhmann, 2003a]	0.6713	0.4539
Transitive	0.6915	0.498
Transitive + SVD	0.7152	0.5226
GTD (Perturb.)	0.7016	0.5013
GTD (Perturb.) + SVD	**0.7255**	**0.5297**

Table 2: Quantitative segmentation evaluation

Method	PRI	VoI	GCE	BDE
[Cour *et al.*, 2005]	0.7559	2.47	0.1925	15.10
[Wang *et al.*, 2008]	0.7521	2.495	0.2373	16.30
[Mignotte, 2010]	0.8006	–	–	–
[Li *et al.*, 2011]	0.8205	1.952	0.1998	12.09
[Kim *et al.*, 2013]	0.8146	1.855	0.1809	12.21
[Li *et al.*, 2012]	0.8319	1.685	0.1779	11.29
[Arbelaez *et al.*, 2011]	0.81	1.65	–	–
[Yu *et al.*, 2014]	0.7926	2.087	0.1835	13.171
[Wang *et al.*, 2014]	0.8039	2.021	0.2066	13.77
Baseline: Ncut	0.7607	2.108	0.2217	14.608
Baseline: Transitive	0.8295	1.645	0.1688	10.568
GTD (Perturb.)	**0.8331**	**1.639**	**0.1655**	**10.372**

have nonzero affinity values, computed from the same dissimilarity with a Gaussian kernel. We directly perform k-means on the matrix rows without SVD, and use mean shift to precluster the rows of the GTD matrix to roughly initialize the cluster centers. A lower bound of 2 and an upper bound of 12 is set on the final cluster number. Finally, singular small regions are eliminated and merged with neighboring ones with a fixed threshold.

Figure 3 shows the qualitative results of the proposed method and the baselines. In segmentation, inter-cluster short links usually present in the form of weak boundaries and is the major hindrance against correct segmentation. One could see conventional transitive distance is prone to over-merging while the proposed method benefits from the bagging and generates better results with closed contours. Our method is also compared with other state of the art works on several popular segmentation benchmarks. Results listed in Table 2 show the excellent performance of our method.

6 Conclusion

In this paper, we have proposed the framework of generalized transitive distance, which generalizes the conventional work on transitive distance and possesses many nice theoretical properties. It is shown that the GTD obtained by minimum spanning random forest can be more robust. More importantly, the framework is open to many other diversification strategies that we so far have not yet fully investigated. Our future research will continue to improve the current work.

Acknowledgments

This work was supported by National Nature Science Foundation of China under grant No.61432012 and No.61172180.

References

[Arbelaez *et al.*, 2011] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[Chang and Yeung, 2005] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering with application to image segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 278–285. IEEE, 2005.

[Chang and Yeung, 2008] Hong Chang and Dit-Yan Yeung. Robust path-based spectral clustering. *Pattern Recognition*, 41(1):191–203, 2008.

[Comaniciu and Meer, 2002] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[Cour *et al.*, 2005] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131. IEEE, 2005.

[Ding *et al.*, 2006] Chris Ding, Xiaofeng He, Hui Xiong, and Hanchuan Peng. Transitive closure and metric inequality of weighted graphs: detecting protein interaction modules using cliques. *International journal of data mining and bioinformatics*, 1(2):162–177, 2006.

[Dollár and Zitnick, 2014] Piotr Dollár and C Zitnick. Fast edge detection using structured forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014.

[Elhamifar and Vidal, 2009] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[Fiedler, 1998] Miroslav Fiedler. Ultrametric sets in euclidean point spaces. *Elec. J. Lin. Alg*, 3:23–30, 1998.

[Fischer and Buhmann, 2003a] Bernd Fischer and Joachim M Buhmann. Bagging for path-based clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(11):1411–1415, 2003.

[Fischer and Buhmann, 2003b] Bernd Fischer and Joachim M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(4):513–518, 2003.

[Fischer *et al.*, 2004] Bernd Fischer, Volker Roth, and Joachim M Buhmann. Clustering with the connectivity kernel. *Advances in Neural Information Processing Systems*, 16:89–96, 2004.

[Greenberg *et al.*, 2014] Craig S Greenberg, Désiré Bansé, George R Doddington, Daniel Garcia-Romero, John J Godfrey, Tomi Kinnunen, Alvin F Martin, Alan McCree, Mark Przybocki, and Douglas A Reynolds. The nist 2014 speaker recognition i-vector machine learning challenge. In *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[Kim *et al.*, 2013] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Learning full pairwise affinities for spectral segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7):1690–1703, 2013.

[Lemin, 1985] A.Y. Lemin. Isometric imbedding of isosceles (non-archimedean) spaces into euclidean ones. *Doklady Akademii Nauk SSS*, 285:558–562, 1985.

[Li and Narayanan, 2014] Ming Li and Shrikanth Narayanan. Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification. *Computer Speech & Language*, 28(4):940–958, 2014.

[Li *et al.*, 2011] Mu Li, Xiao-Chen Lian, James T Kwok, and Bao-Liang Lu. Time and space efficient spectral clustering via column sampling. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2297–2304. IEEE, 2011.

[Li *et al.*, 2012] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 789–796. IEEE, 2012.

[Liu *et al.*, 2013] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):171–184, 2013.

[Mignotte, 2010] Max Mignotte. A label field fusion bayesian model and its penalized maximum rand estimator for image segmentation. *Image Processing, IEEE Transactions on*, 19(6):1610–1624, 2010.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Peng *et al.*, 2013] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 430–437. IEEE, 2013.

[Peng *et al.*, 2015] Xi Peng, Zhang Yi, and Huajin Tang. Robust subspace clustering via thresholding ridge regression. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3827–3833. AAAI, 2015.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[Sibson, 1973] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[Wang *et al.*, 2008] Jingdong Wang, Yangqing Jia, Xian-Sheng Hua, Changshui Zhang, and Long Quan. Normalized tree partitioning for image segmentation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[Wang *et al.*, 2014] Jingdong Wang, Huaizu Jiang, Yangqing Jia, Xian-Sheng Hua, Changshui Zhang, and Long Quan. Regularized tree partitioning and its application to unsupervised image segmentation. *Image Processing, IEEE Transactions on*, 23(4):1909–1922, 2014.

[Yu *et al.*, 2014] Zhiding Yu, Chunjing Xu, Deyu Meng, Zhuo Hui, Fanyi Xiao, Wenbo Liu, and Jianzhuang Liu. Transitive distance clustering with k-means duality. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 987–994. IEEE, 2014.

[Zelnik-Manor and Perona, 2004] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2004.