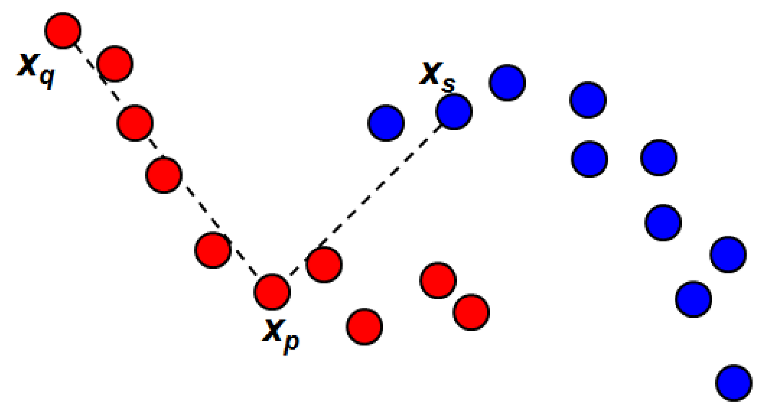
Generalized Transitive Distance with Minimum Spanning Random Forest

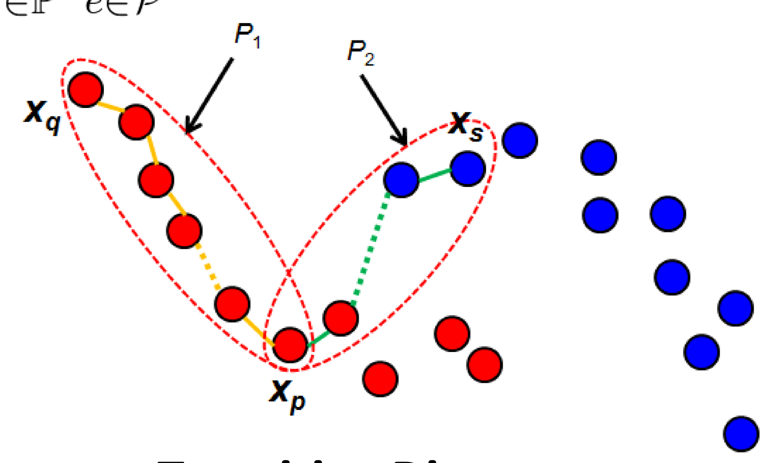
Zhiding Yu, Weiyang Liu, Wenbo Liu, Xi Peng, Zhuo Hui, B. V. K. Vijaya Kumar
yzhiding@andrew.cmu.edu, kumar@ece.cmu.com

Introduction I: Transitive Distance (TD)

- **Math Definition:** $D_T(x_p, x_q) = \min_{\mathcal{P}\in\mathbb{P}} \max_{e\in\mathcal{P}}\{d(e)\}$

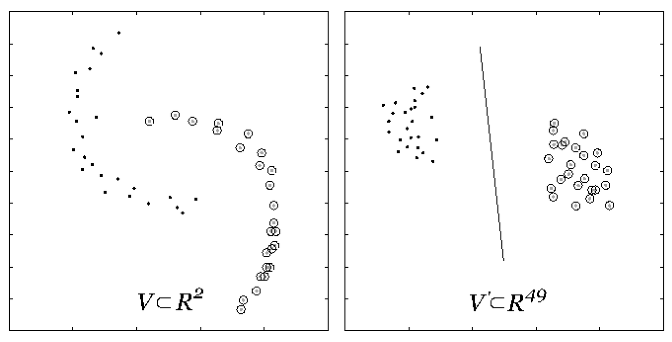


Euclidean Distance



Transitive Distance

- **Proposition 1:** The transitive Distance is an ultra-metric.
- **Proposition 2:** Every finite ultrametric space with n distinct points can be isometrically embedded into an n−1 dim Euclidean space.



An example of TD embedding

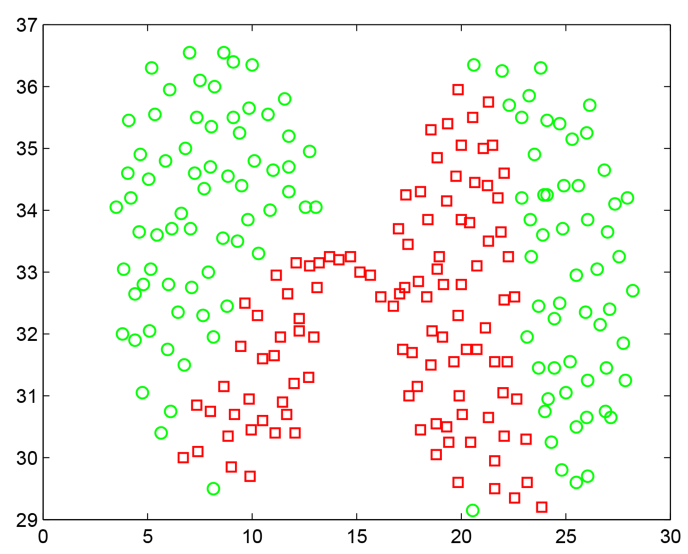
- **Proposition 3:** Given a weighted graph with edge weights, each transitive edge lies on the minimum spanning tree (MST).

Introduction II: TD Clustering

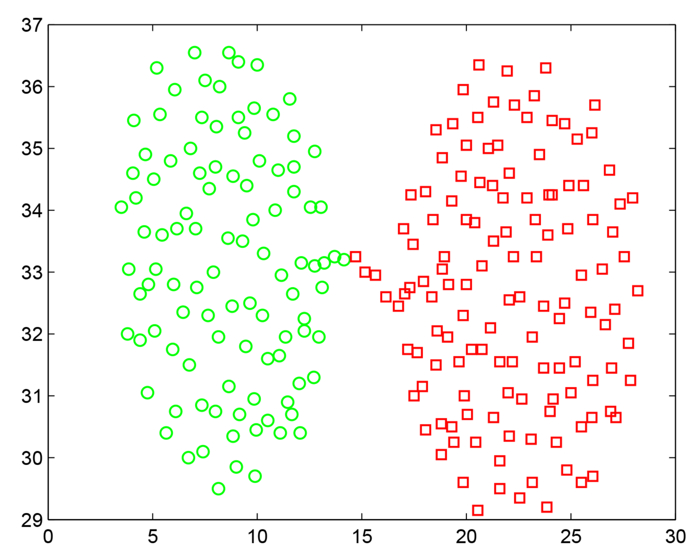
- Under TD embedding, data from the same cluster becomes compact. It is therefore desirable to perform clustering in the embedded space.
- Intuitively, TD clustering can be regarded as an approximate spectral clustering (SC) where TD embedding is similar to eigen decomposition.
- Clustering with **K-means duality**: Treat each row of TD matrix as embedded data and apply k-means. This produces similar clustering results as directly performing k-means in the embedded space.

Generalized Transitive Distance: Bagging TD with Random Forest

- Math Definition: $D_G(x_p, x_q) = \max_t \underset{\substack{\mathcal{P}_t\in\mathbb{P}_t,\\ \forall t\in\{1,\ldots,T\}}}{\text{gmin}} \max_{e\in\mathcal{P}_t}\{d(e)\}$
- Here, "gmin" denotes the generalized min returning a set of minimum values from multiple sets. denote the sets of all candidate paths respectively from multiple diversified graphs.
- TD is sensitive to short links (see following left figure). Bagging (right) introduces more robustnes $\mathbb{P}_t$



Clustering with TD



Clustering with GTD

- **Theorem 1:** The GTD is also an ultrametric and can be embedded.
- **Theorem 2:** Given the sets of candidate paths, the transitive distance edge lies on the minimum spanning random forest formed by MSTs extracted from the perturbated, diversified graphs (for bagging).

Random Forest Extraction

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.

Algorithm 2 Random Perturbation Algorithm

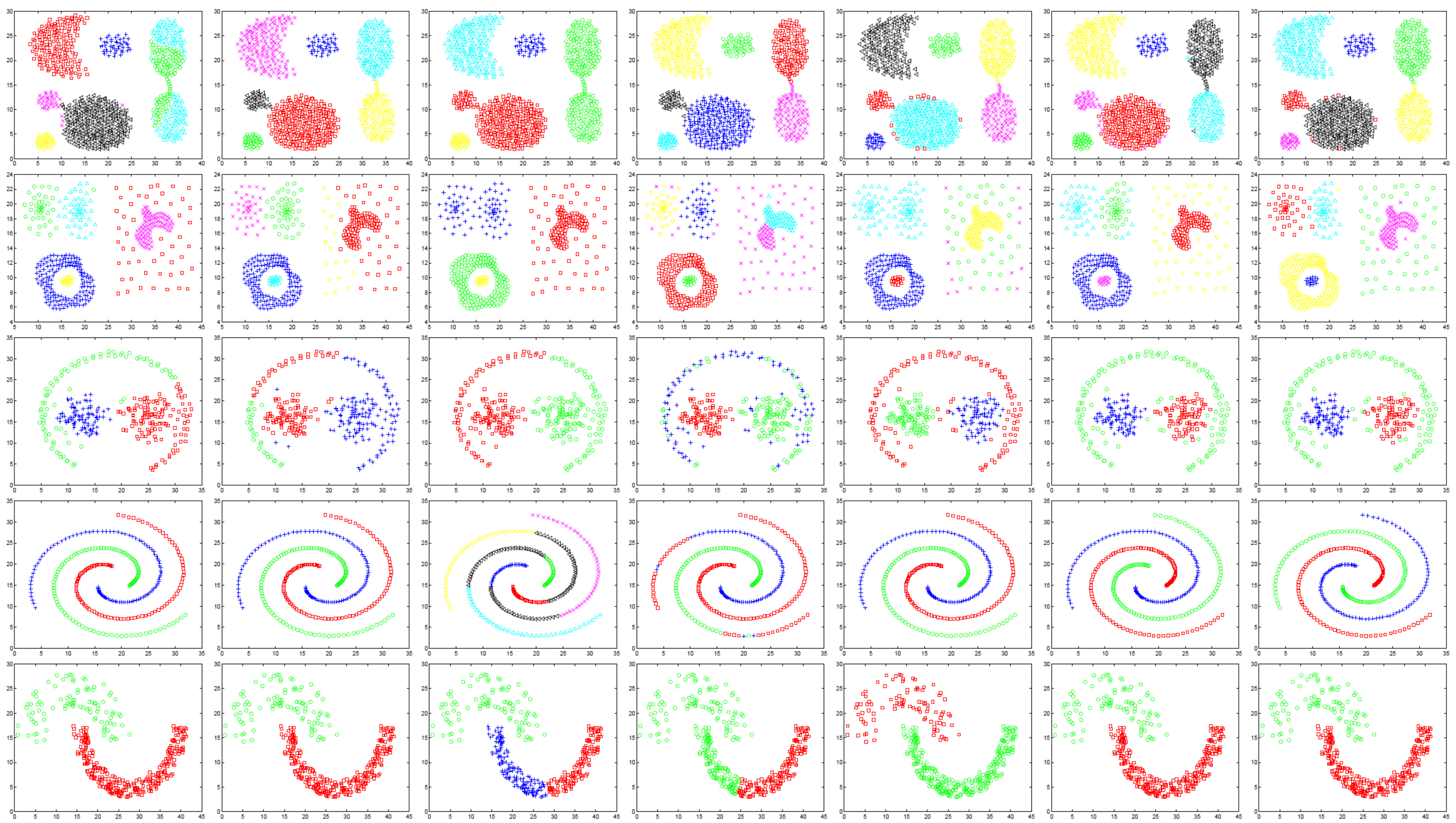
1: Initialize $G_1 = G = (V, E)$, where G is a weighted graph and E is the set of available edges.
2: If $t \neq 1$, obtain G_t by randomly perturbate the edge length of G with a random number $\epsilon * rand(1)$.
3: Extract MST from G_t using the Kruskal's algorithm and return the $n \times n$ pairwise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

Improved Top-Down Clustering

- Given a computed GTD matrix D, perform SVD: $D = U\Sigma V^*$
- Treat each row from the top several columns with largest eigenvalues as data samples, and perform k-means.

Experimental Results

Clustering on Toy Example Datasets



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.

Qualitative Segmentation Results on BSDS-300

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.

Quantitative Results on BSDS-300

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**

Speech Data Clustering

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**

Conclusion

- We have proposed the framework of generalized transitive distance, which generalizes the conventional transitive distance.
- GTD possesses many nice theoretical properties.
- GTD obtained by minimum spanning random forest is more robust.
- The framework is open to many other bagging strategies that we so far have not yet fully investigated.