Locality Constrained Transitive Distance Clustering on Speech Data

Wenbo Liu[*,1,3], *Zhiding Yu*[*,3], *Bhiksha Raj*[4], *Ming Li*[1,2]

[1]SYSU-CMU Joint Institute of Engineering, Sun Yat-Sen University
[2]SYSU-CMU Joint Research Institute, Sun Yat-Sen University
[3]Department of Electrical and Computer Engineering, Carnegie Mellon University
[4]Language Technologies Institute, Carnegie Mellon University
{wenboliu,yzhiding}@andrew.cmu.edu, bhiksha@cs.cmu.edu, liming46@mail.sysu.edu.cn

Abstract

The idea of developing unsupervised learning methods has received significant attention in recent years. An important application is whether one can train a high quality speaker verification model given large quantities of unlabeled speech data. Unsupervised learning methods such as data clustering often play a central role since they are able to analyze the underlying latent patterns without any supervision information. In this paper, we focus on developing an effective clustering method for speech data. We propose the locality constrained transitive distance, a distance measure which better models speech data with arbitrarily shaped clusters. We also propose a robust top-down clustering framework on top of the distance measure to generate accurate cluster labels. Experimental results show the good performance of the proposed method.

Index Terms: clustering, speaker verification, unsupervised learning, i-vector, transitive distance

1. Introduction

With the fast growth of digital media and data storage, obtaining data becomes relatively easy, giving rise to the so-called "data deluge". In reality however, large quantities of data are often not sufficiently labeled and the labor cost of labeling can be high. Today the growing speed of annotation capability can hardly match the explosive increase of media contents, leaving large amounts of unlabeled data. An important problem, therefore, is whether one develop machine learning techniques that lead to less human annotation efforts. Such problem has recently received significant attention. Learning methods such as domain adaptation and semi-supervised learning were proposed to partially address this problem. In domain adaptation, models originally built from the labeled out-of-domain data are adapted to the target domain with unlabeled in-domain data [1] [2]. On the other hand, semi-supervised learning can be applied on database with small amounts of labeled data and large amounts of unlabeled data. Despite the reduced label requirement, certain amounts of labels are still needed for both methods.

Recently the NIST i-vector Machine Learning Challenge 2014 [3] organized competitions to design unsupervised speaker verification systems with fully unlabeled i-vector development dataset. Since outside data are not allowed and no development labels are provided, both domain adaptation and semi-supervised learning can not be directly applied under such setting. Unsupervised learning methods, particularly clustering techniques were widely adopted by many competition teams as a key step in their proposed systems. [4, 5, 6, 7, 8]. More specifically, clustering in general plays the role of feeding the backend with estimated data labels or analyzed latent structures. Therefore, designing appropriate clustering heuristics can directly benefit the performance of backend in a unsupervisedly trained speaker verification system.

Clustering is one of the most fundamental and important machine learning problems. A number of clustering methods were proposed, ranging from the well known k-means and graph-based approaches [9], to mode seeking [10, 11, 12, 13], spectral clustering [14, 15, 16], and subspace clustering [17]. Spectral clustering methods were widely used for their excellent performance, one of the key reasons being the ability to discover non-convex latent structures. However spectral clustering is not the only family of methods that can handle clusters with arbitrary shapes. Transitive distance clustering provides an elegant non-eigendecomposition alternative to handling arbitrarily shaped clusters. Specifically, transitive distance emphasizes on the connectivity rather than absolute distance between pairwise data samples. This is achieved by solving a minimax problem in which one first finds the largest hops (edges) along all possible connecting paths and then defines the pairwise distance as the minimum largest hop among all paths. [18] proposed the concept of transitive distance and an agglomerative bottom-up approximation for pairwise distance clustering. Other important works include the connectivity kernel [19], transitive closure [20], transitive affinity [21, 22] and the top-down transitive distance clustering framework with "k-means duality" [23].

An interesting aspect is that there exist some inherent connections between transitive distance and the single linkage algorithm, while the latter is widely used in the speech community [4, 6]. It was proved that if the maximum possible path order (number of nodes forming a path) is equal to N which is the number of data, the transitive distance edges always lie on the minimum spanning tree (MST). Despite such resemblance, transitive distance clustering is not necessarily identical to single linkage algorithm. Single linkage algorithm seeks to form clusters by cutting the maximum edges on an MST. Such bottom-up greedy heuristic not always makes a stable strategy to correctly generate cluster configurations, especially under noise. On the other hand, transitive distance aims at embedding the original data into another kernel space with better cluster shapes, which resembles eigendecomposition in spectral clustering. With a top-down clustering framework [23], transitive

* indicates equal contribution.

This research is supported in part by the National Natural Science Foundation of China (61401524), Natural Science Foundation of Guangdong Province (2014A030313123), SYSU-CMU Shunde International Joint Research Institute and CMU-SYSU Collaborative Innovation Research Center.

distance clustering can be regarded as an approximate spectral clustering, behaving much more robust than many bottom-up methods. Readers who are interested in algorithm details and properties may kindly refer to [23].

Basically, conventional transitive distance can be regarded as an opposite extreme to Euclidean distance and cosine distance in cluster shape flexibility. The former prefers data with low intrinsic dimensions (having elongated or manifold-shaped clusters) while the latter two in general works best on data with convex-like clusters. Although top-down clustering can render more robustness under noise, conventional transitive distance loses considerable amount of discriminative information for data points that are "really far away". We will discuss later why this is the case in speech data clustering. In this paper, we propose the locality constrained transitive distance (LCTD), a distance measure which inherits the cluster-shape flexibility of transitive distance at local scale, while simultaneously incorporating discriminative information at global scale. We also propose a more robust top-down clustering framework which further improves the clustering performance. Experimental results show that LCTD works better than both transitive distance and cosine distance, and that the proposed clustering method achieves excellent performance on all experiment dataset with considerable improvement over baselines.

2. Top-down Transitive Distance Clustering

To handle data with arbitrary cluster shapes, a common way is to use kernel method to create the following 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 more compact cluster shapes. Spectral clustering obtains such nonlinear mapping through eigendecomposition on the normalized variants of its affinity matrix (Such as a Laplacian matrix). On the other hand, transitive distance implicitly builds a non-linear mapping similar to spectral clustering without eigendecomposition. The transitive distance for any pairwise data is defined as follows [23]:

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:

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

According to the definition, transitive distance explores the connectivity between pairwise data by searching for a path that connects them with the minimum gap[1]. The way how transitive distance handles arbitrarily-shaped clusters is to reduce intra-cluster distances: Even if the Euclidean distance is far away, two data samples are close if there is at least one path that strongly correlates them.

Now given any pair of data samples, the next step is to locate the so-called "transitive edge": the very gap that determine the pairwise transitive distance according to definition. This seems to be almost a complicated searching problem as there are almost infinitely many combinations of paths. The following lemma, however, indicates that there is a computationally feasible way to locate the transitive edges:

[1]Gap refers to the largest hop along a path.

Lemma 1. *Given a weighted graph, the transitive edge lies on the MST of the graph.*

The proof of Lemma 1 can be found in [19, 23]. The lemma basically states that to determine the pairwise transitive distance, one only needs to construct an MST and look for the largest edge. We can use the Kruskal's algorithm to compute the transitive distance matrix:

Algorithm 1 Computing the transitive distance matrix

1: Construct a total graph $G = (V, E)$ from data where edge weights in E are the Euclidean distances.
2: Sort E based on edge weights. Initialize $G_{MST} = (V, E')$ where $E' = \varnothing$.
3: Take an edge $e_{i,j}$ from E.
4: If G_{MST} and $e_{i,j}$ form a circle, discard $e_{i,j}$.
5: Otherwise, add $e_{i,j}$ to E'. Find sets of nodes V_i and V_j currently connected to edge nodes i and j respectively in G_{MST}. Update the pairwise distances of all possible combinations with $|e_{i,j}|$.
6: Repeat 3-5 sequentially for all edges in E.

Lemma 2. *The transitive distance is an ultrametric and can be embedded into an $n-1$ dimensional vector space.*

The corresponding proof and details of **Lemma 2** can be found in [23]. This lemma conveys the following important information: with the generalized transitive distance, we have an implicit nonlinear mapping:

$$\phi : (V \subset \mathbf{R}^l, D) \mapsto (V' \subset \mathbf{R}^{n-1}, d'), \tag{4}$$

where $d'(\cdot,\cdot)$ is the Euclidean distance in $\mathbf{R}^{n-1}$ and $D(\cdot,\cdot)$ is the transitive distance in $\mathbf{R}^l$, $d'(\phi(x_i), \phi(x_j)) = D(x_i, x_j)$. 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 only the pairwise Euclidean distance. [23] also detailedly discussed why the projected data in the kernel space form nicely shaped clusters.

Given the transitive distance obtained from **Algorithm 1**, [23] proposed to simply perform a k-means clustering on the rows of the distance matrix[2]. The obtained clustering labels are regarded as the final output labels.

3. The Proposed LCTD Clustering

Basically, an ideal distance matrix should be block diagonal[3]. Pairwise distances should be 0 if two data samples belong to the same true cluster while relatively large if they do not. In such case, a simple k-means clustering on the rows of the distance matrix can easily recover the correct cluster labels. In general cases however, what we often obtain is a distance matrix perturbated by noise: $\mathbf{D} = \mathbf{D}_{block} + \mathbf{E}$. The task of both top-down transitive distance clustering [23] and many spectral clustering methods can be regarded as recovering $\mathbf{D}_{block}$. One can intuitively treat the inter/intra cluster distance discrimination as signal strength, and the perturbation strength as noise strength. According to the matrix perturbation theory, such "signal to noise ratio" directly influences the accuracy of correct recovery.

[2]Treat each row of the distance matrix just like one data sample

[3]Here we assume data with the same ground truth labels are arranged to be consecutive.

For data with low intrinsic dimensions where intra-cluster variation dominates, reducing the intra-cluster variation certainly benefits the "signal to noise ratio". For speech data however, such low intrinsic dimension assumption may no longer hold at global scale. One major reason is that dimensionality reduction methods such as factor analysis are widely used in many frontend methods. An example is the i-vector framework which extracts features at the frontend for low-dimensional representation of speech utterances. In this framework, factor analysis is conducted on the supervectors to generate a low dimensional total variability space. The very distance representation strategy in transitive distance that benefits clustering on manifolded data is now losing discriminative information, since samples that used to be very far away are dragged much closer in transitive distance. On the other hand, some conventional distances make more reasonable measurements under such cases.

Does this mean cosine and Euclidean distances are the optimal ones? Probably not. At local scale, it may still be reasonable to assume that certain clusters do have non-convex shapes. A possibly better design of distance measure is to combine advantages from both sides. Therefore, we propose the locality constrained transitive distance where discrimination between far away points are reinforced. We will show that LCTD works better than using Euclidean distance, cosine distance and transitive distance alone.

3.1. The Locality Constrained Transitive Distance

A very natural reinforcement of discriminative information for far away samples is to weight the pairwise transitive distance with cosine distance. The reason why we call it "locality constrained" is because the influence of transitive distance is mostly concentrated on neighboring pairwise data and decades fast for far away ones. More formally, we have the following definition:

$$D_{lctd1}(i,j) = D_{td}(i,j) * D_{eu}(i,j), \tag{5}$$

where D_{td} is the transitive distance that be obtained via **Algorithm 1**, and D_{eu} is the Euclidean distance. We will denote the locality constrained transitive distance defined by (5) as LCTD-1 in the rest of the paper.

Note that LCTD-1 is no longer an ultrametric. In fact it is not even a metric since the metric triangle inequality does not hold. Such pairwise distances violate metricity and, therefore, cannot be naturally embedded in a vector space [24]. On the other hand, designing a distance metric guarantees many nicer theoretical properties. Therefore we propose another variation of locality constrained transitive distance with linear combination to impose locality constraints:

$$D_{lctd1}(i,j) = \alpha D_{td}(i,j) + (1-\alpha) D_{eu}(i,j), \tag{6}$$

where $\alpha \in [0,1]$ is a weight parameter to balance locality. We denote the locality constrained transitive distance defined by (6) as LCTD-2. It turns out that empirically setting α to 0.5 already gives very good performance. We will use this parameter setting for all the following experiments.

Theorem 1. *The proposed locality constrained transitive distance in (6) is a metric.*

Proof: Lemma 2 states that the transitive distance is an ultrametric. Therefore, the transitive distance follows the following strong triangle inequality:

$$\begin{aligned} D_{td}(i,j) &\leq \max(D_{td}(i,k), D_{td}(k,j)) \\ &< D_{td}(i,k) + D_{td}(k,j), \forall \{i,j,k\}. \end{aligned} \tag{7}$$

Let $\beta \triangleq (1-\alpha)$. we have:

$$\begin{aligned} & D_{lctd2}(i,j) \\ =& \alpha D_{td}(i,j) + \beta D_{eu}(i,j) \\ <& \alpha D_{td}(i,k) + \beta D_{eu}(i,k) + \alpha D_{td}(k,j) + \beta D_{eu}(k,j) \\ =& D_{lctd2}(i,k) + D_{lctd2}(k,j) \end{aligned} \tag{8}$$

We have therefore proved the triangle inequality for LCTD-2. Other properties including non-negativity, symmetry and identity of indiscernibles are easy to prove and is omitted here.

3.2. A New Top-Down Clustering Framework

The LCTD framework possesses many nice properties, one of them being that the underlying distance embedding implicitly renders compact cluster representations. Intuitively, one would want to directly perform k-means in the distance embedded space. However, without explicit representation of the projected data, finding an optimal cluster partitioning with a pairwise distance is an $\mathcal{NP}$-hard combinatorial optimization problem [18].

This by no means indicates one can not find good cluster configurations with approximation under pairwise distance. With compact clusters projection, the LCTD matrix can be approximately regarded as a block diagonal matrix perturbated by additional noise:

$$\mathbf{D}_{lctd} = \mathbf{D}_{block} + \mathbf{E} \tag{9}$$

A top-down clustering approximation is to k-means over the rows [23]. K-means can be regarded as certain low rank approximation to recover D_{block}. In addition, top-down methods in general shows more robustness against noise compared to bottom-up methods. Therefore, such top-down strategy benefits LCTD clustering, making it considerably different from single linkage algorithm.

We propose an alternative algorithm which further improves the top-down clustering performance. Instead of directly performing k-means on $\mathbf{D}_{lctd}$, We consider the rotated and normalized distance matrix[4]:

$$\mathbf{D}' \triangleq \mathbf{D}_{lctd}\mathbf{V}\mathbf{\Lambda}^{-1} = \mathbf{U}, \tag{10}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues by taking SVD on $\mathbf{D}_{lctd}$ and $\mathbf{V}$ is the initial rotation basis:

$$\mathbf{D}_{lctd} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^*, \tag{11}$$

When we perform cosine distance k-means on the rows of $\mathbf{D}'_{lctd}$, we are essentially concerned about the following pairwise distance:

$$\mathbf{D}'\mathbf{D}'^{\top} = \mathbf{U}\mathbf{U}^{\top}. \tag{12}$$

The advantage of using $\mathbf{D}'$ over $\mathbf{D}_{lctd}$ is that the former better separates noise out of useful cluster information. One can minimize the influence from noise by picking the top K columns from $\mathbf{D}'$: $\mathbf{D}'_{lctd} \triangleq \mathbf{D}'(:, 1:K)$ and performing k-means on rows of $\mathbf{D}'_{lctd}$. Even if $\mathbf{D}_{lctd}$ is not full rank, $\mathbf{D}'_{lctd}$ can be obtained from top columns of $\mathbf{U}$.

In addition, the following proposition states the inherent relationship between the original strategy and the proposed one:

Proposition 1. *When $\mathbf{D}_{lctd}$ is ideally block diagonal, performing k-means on $\mathbf{D}'_{lctd}$ exactly recovers the same label as k-means on $\mathbf{D}_{lctd}$.*

The proof is omitted here as it is very straight forward.

[4]Here we assume $\mathbf{D}_{lctd}$ is full rank.

4. Experimental Results

We conduct comprehensive experiments on multiple datasets. In all experiments, we input the groundtruth number of clusters for all method and measure the cluster purity (accuracy).

4.1. Large scale clustering results

We first conduct large scale clustering experiment on three datasets. The first one is the NIST i-vector Machine Learning Challenge [3] development dataset (denoted as "I-Vector") which consists of 36572 600-dimensional pre-extracted i-vectors with 4958 identities. We also form two additional datasets by extracting the i-vectors under the framework of [25][5]. I-vectors from Switchboard form the "Switchboard" dataset containing 11587 500-dimensional i-vectors and 1052 identities. The rest from NIST SRE form the "NIST" dataset containing 21704 i-vectors and 1738 identities.

We test the proposed methods and compare with other baselines. Table 1 shows the results of the proposed methods and baselines[6]. One could see that the proposed two methods significantly outperform many other baselines. Results also indicate that both LCTD and SVD can benefit performance. Note that single linkage completely fails on the I-Vector dataset for it erroneously group large amount of data into the same cluster.

Table 1: Quantitative evaluation of large scale clustering

Method	NIST	I-Vector	Switchboard
Normalized Cuts	0.4883	0.3654	0.5340
Single Linkage	0.4544	0.156	0.4754
Spectral Clust	0.6841	0.4898	0.8926
Transitive [23]	0.6915	0.498	0.7276
Transitive + SVD	0.7152	0.5226	0.7766
Cosine + SVD	0.8019	0.7145	0.9144
K-Means (Cosine)	0.7897	0.7174	0.9080
LCTD-1 (No SVD)	0.7707	0.6912	0.8503
LCTD-1 + SVD	**0.8228**	0.7178	0.9109
LCTD-2 + SVD	0.8196	**0.7193**	**0.9154**

4.2. Influence of eigenvector number

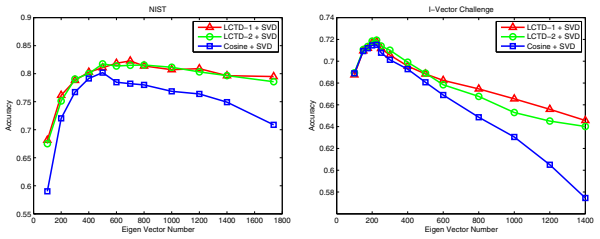
We also conduct detailed analysis on the number of eigenvectors and its influence on the clustering result. Figure 1 shows the accuracies of the proposed two methods and the most relevant baseline **Cosine + SVD**. LCTD not only helps to boost the best possible performance, but also better stabilizes the system performance with different number of eigenvectors.

4.3. Medium scale clustering results

We also test the proposed method and baselines on the NIST subsets (NIST-04, NIST-05, NIST-06 and NIST-08). The numbers of samples are respectively 4305, 5549, 7957, 3893 and the number of identities are 186, 307, 720, 846. The reason we conduct this experiment is that different subsets have varying number of speaker identities and average cluster sizes. Again, quantitative results in Table 2 indicate that the proposed methods consistently achieve the best performance.

[5]The i-vectors are trained on Switchboard II part1 to part3 and NIST SRE 04, 05, 06, 08 corpora on the telephone channel.

[6]**LCTD-1 (No SVD)** denotes directly performing k-means on the matrix rows. The eigenvector numbers of all methods with SVD are tuned for best performance.



(a) (b)

Figure 1: Accuracies versus different eigenvector numbers. (a) Results on the NIST dataset. (b) Results on the I-Vector dataset.

Table 2: Quantitative evaluation of medium scale clustering

Method	NIST04	NIST05	NIST06	NIST08
Transitive	0.5487	0.5909	0.7357	0.6948
Trans+SVD	0.6736	0.6578	0.7958	0.7349
Kms(Cos)	0.8149	0.7708	0.8643	0.7858
Euc(NoSVD)	0.6913	0.7012	0.8124	0.7177
Cos(NoSVD)	0.6897	0.6879	0.8254	0.7131
Euc+SVD	0.8397	0.8009	0.8716	0.8329
Cos+SVD	0.8223	0.7987	0.8713	0.8325
LCTD1+SVD	**0.8462**	0.8018	**0.8986**	**0.8330**
LCTD2+SVD	0.8455	**0.8036**	0.8939	0.8327

4.4. Application to unsupervised speaker verification

We finally apply the proposed method to unsupervisedly training a speaker verification system. We use the clustering algorithm to generate estimated data labels and feed them into Gaussian PLDA [26]. The learned PLDA model then gives a similarity score between target and test samples. We conduct this experiment following the NIST i-vector Challenge with multiple-enrollment scoring strategy [27]. We adopt K-means as a baseline for it was widely used for unsupervised label generation. Table 3 shows the evaluation costs on both the evaluation set as well as the progressive set. Results indicate that our method helps to boost the verification performance.

Table 3: Speaker verification result evaluation on I-Vector

Method	K-means	LCTD-1+SVD	LCTD-2+SVD
EER Prog	0.0906	**0.0866**	0.0878
EER Eval	0.0867	**0.0825**	0.0854
DCF Prog	0.3683	0.3500	**0.3482**
DCF Eval	0.3691	0.3459	**0.3455**

5. Conclusion

In this paper, we proposed two versions of locality constrained transitive distances which better model pairwise speech data similarity under the i-vector framework. In addition, we proposed a top-down clustering framework based on SVD on top of LCTD. Experimental results verified the excellence performance of the proposed method. We believe that the result in this paper on robust clustering methods will significantly benefit a wide variety of speech processing methods.

6. References

[1] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. ICASSP*. IEEE, 2014, pp. 4075–4079.

[2] P. Rajan, A. Afanasyev, V. Hautamäki, and T. Kinnunen, "From single to multiple enrollment i-vectors: Practical plda scoring variants for speaker verification," *Digital Signal Processing*, 2014.

[3] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds, "The nist 2014 speaker recognition i-vector machine learning challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[4] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the nist i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, no. EPFL-CONF-198439, 2014.

[5] B. Vesnicer, J. Zganec-Gros, S. Dobrisek, and V. Struc, "Incorporating duration information into i-vector-based speaker-recognition systems," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 241–248.

[6] S. Novoselov, T. Pekhovsky, and K. Simonchik, "Stc speaker recognition system for the nist i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 231–240.

[7] A. Khosravani and M. Homayounpour, "Linearly constrained minimum variance for robust i-vector based speaker recognition," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 249–253.

[8] D. Garcia-Romero, A. McCree, S. Shum, N. Brümmer, and C. Vaquero, "unsupervised domain adaptation for i-vector speaker recognition," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 260–264.

[9] R. Sibson, "Slink: an optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.

[10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[11] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoidshifts," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[12] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 705–718.

[13] Z. Yu, A. Li, O. C. Au, and C. Xu, "Bag of textons for image segmentation via soft clustering and convex shift," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 781–788.

[14] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[15] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2004, pp. 1601–1608.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797.

[18] B. Fischer and J. M. Buhmann, "Path-based clustering for grouping of smooth curves and texture segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 4, pp. 513–518, 2003.

[19] B. Fischer, V. Roth, and J. M. Buhmann, "Clustering with the connectivity kernel," *Advances in Neural Information Processing Systems*, vol. 16, pp. 89–96, 2004.

[20] C. Ding, X. He, H. Xiong, and H. Peng, "Transitive closure and metric inequality of weighted graphs: detecting protein interaction modules using cliques," *International journal of data mining and bioinformatics*, vol. 1, no. 2, pp. 162–177, 2006.

[21] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering with application to image segmentation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 278–285.

[22] ——, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[23] Z. Yu, C. Xu, D. Meng, Z. Hui, F. Xiao, W. Liu, and J. Liu, "Transitive distance clustering with k-means duality," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 987–994.

[24] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1540–1551, 2003.

[25] M. Li and S. Narayanan, "Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification," *Computer Speech & Language*, vol. 28, no. 4, pp. 940–958, 2014.

[26] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Proc. INTERSPEECH*, 2011, pp. 249–252.

[27] W. Liu, Z. Yu, and M. Li, "An iterative framework for unsupervised learning in the plda based speaker verification," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 78–82.