Supplementary Material: 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,liujianzhuang}@huawei.com, dymeng@mail.xjtu.edu.cn

Appendix A: Proof of Theorem 1

It is reasonable to assume that each cluster has at least two samples. Let $x_i, x_j \in C$, $x_k \notin C$, x_i, x_j, $x_k \in V$, where $C \subset V$ is some cluster. Then their images after the mapping ϕ are x'_i, x'_j, $x'_k \in V'$, where $x'_i, x'_j \in C'$, $x'_k \notin C'$, and $C' = \phi(C)$.

(i) It can be verified that if $d'(x'_i, x'_j) \geq d_0 \in R^+$, then there exists a partition $C_1 \cup C_2 = C$ such that $d(C_1, C_2) \geq d_0$. Such a partition can be obtained by the following steps:

1) Initialize $H = C$, $m = 1$, $C_1 = \varnothing$, and $C_2 = \varnothing$.

2) Find a path $\mathcal{P}$ including the transitive edge from x_i to x_j in H.

3) Cut the transitive edge on the path $\mathcal{P}$. Let $\mathcal{P}_m(\mathcal{Q}_m)$ be the set consisting of the samples on P that are on the same side with $x_i(x_j)$ after the cutting, except x_i (x_j).

4) $C_1 \leftarrow C_1 \cup \mathcal{P}_m$, $C_2 \leftarrow C_2 \cup \mathcal{Q}_m$, $H \leftarrow H\{\mathcal{P}_m \cup \mathcal{Q}_m\}$, and $m \leftarrow m + 1$.

5) Repeat 2), 3), and 4) until only x_i and x_j are left in H.

6) $P_m \leftarrow \{x_i\}$, $Q_m \leftarrow \{x_j\}$, $C_1 \leftarrow C_1 \cup P_m$, and $C_2 \leftarrow C_2 \cup Q_m$.

In this procedure, from (3) in the paper we can see that $d(\mathcal{P}_s, \mathcal{Q}_t) \geq d'(x'_i, x'_j)$, $1 \leq s,\, t \leq m$. Since $C_1 = \mathcal{P}_1 \cup \mathcal{P}_2 \cup ... \cup \mathcal{P}_m$ and $C_2 = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup ... \cup \mathcal{Q}_m$, we have $d(C_1, C_2) = \min_{1 \leq s,t \leq m}\{d(\mathcal{P}_s, \mathcal{Q}_t)\}$. Thus, $d(C_1, C_2) \geq d'(x'_i, x'_j) \geq d_0$.

(ii) Second, we show that there exist $x_u \in C$ and $x_v \notin C$ such that $d'(x'_i, x'_k) \geq d(x_u, x_v)$. From Definition 1, we have a path $\mathcal{P}$ connecting x_i and x_k including the transitive edge. Then there exists an edge $x_u x_v \in \mathcal{P}$ such that $x_u \in C$ and $x_v \notin C$, and from (3) in the paper, we have $d'(x'_i, x'_k) \geq d(x_u, x_v)$.

(iii) Third, we show that

$$d'(x'_i, x'_j) \leq \min\{d'(x'_i, x'_k), d'(x'_j, x'_k)\}. \quad (1)$$

Assume, to the contrary, that $d'(x'_i, x'_j) > d'(x'_i, x'_k)$. From (i) and (ii), we have a partition $C_1 \cup C_2 = C$, and $x_u \in C$, $x_v \notin C$ such that $d(C_1, C_2) \geq d'(x'_i, x'_j)$ and $d'(x'_i, x'_k) \geq d(x_u, x_v)$. Thus $d(C_1, C_2) \geq d'(x'_i, x'_j) > d'(x'_i, x'_k) \geq d(x_u, x_v) \geq d(C, x_v)$, which contradicts the consistency of V. Therefore, (1) holds.

(iv) Let $C = \{x_{c_1}, ..., x_{c_m}\}$ be a cluster in V, with its image $C' = \phi(C) = \{x'_{c_1}, ..., x'_{c_m}\} \subset V'$. Let $\widetilde{C}'$ be the convex hull of C'. Now we verify that no samples not in C' are in $\widetilde{C}'$. Assume, to the contrary, that there exists a sample $y' \in \widetilde{C}'$, $y \notin C'$. Consider a sample $x' \in C'$. Let P be the hyperplane, each point on which has the same distance to x' and z'. Then there must exist another sample $z' \in C'$ such that y' and z' are in the same side of P, which leads to $d'(x', z') > d'(y', z')$, a contradiction to (1).

In (iv), we have verified that for any cluster $C' \in V'$, no samples from other clusters can be in the convex hull of C'. Thus, the convex hulls of all the clusters in V' are not intersecting each other.

Appendix B: Proof of Theorem 3

For any two distinct vertices x_1 and x_2 in G, let $\mathcal{P} = x_{k_1} x_{k_2} ... x_{k_s}$ be the path connecting them including the transitive edge $x_{k_i} x_{k_{i+1}}$, where $k_1 = 1$ and $k_s = 2$. Then from Definition 1, we have

$$d(x_{k_m}, x_{k_{m+1}}) < d(x_{k_i}, x_{k_{i+1}}), m = 1, 2, ..., i{-}1, i{+}1, ..., s. \quad (2)$$

[*] indicates equal contribution.

Next we verify that the edge $x_{k_i}x_{k_{i+1}}$ is in $\widetilde{G}$. Let $\widetilde{G}_{\mathcal{P}} = \widetilde{G} \cup \mathcal{P}$. Assume, to the contrary, that $x_{k_i}x_{k_{i+1}} \notin \widetilde{G}$. Then the edge $x_{k_i}x_{k_{i+1}}$ must be on a loop $\mathcal{O} \subsetneqq \widetilde{G}_{\mathcal{P}}$. Consider the following two cases:

(i) For any edge $x_u x_v \in \widetilde{G} \cap \mathcal{O}$, $d(x_u, x_v) < d(x_{k_i}, x_{k_{i+1}})$.

(ii) There exists an edge $x_{l_j}x_{l_{j+1}} \in \widetilde{G} \cap \mathcal{O}$ such that $d(x_{l_j}, x_{l_{j+1}}) > d(x_{k_i}, x_{k_{i+1}})$.

Suppose that case (i) is true. Then for any edge on the path $(P \cup \mathcal{O}) \backslash \{x_{k_i}x_{k_{i+1}}\}$ that also connects x_1 and x_2, we have its length smaller than the transitive edge for x_1 and x_2. Thus case (i) cannot be true.

Suppose that case (ii) is true. Since $\widetilde{G}^* = (\widetilde{G} \cup \{x_{k_i}x_{k_{i+1}}\}) \backslash \{x_{l_j}x_{l_{j+1}}\}$ is a spanning tree of G, and the sum of the edge weights in $\widetilde{G}^*$ is smaller than that in G, we have a contradiction to the fact that $\widetilde{G}$ is the minimum spanning tree. Thus case (ii) cannot be true either, which completes the proof.

Appendix C: Proof of Lemma 2 and Lemma 3

Proof of Lemma 2: The readers actually need to first refer to Theorem 3 in Section 5. Theorem 3 states that the transitive distance between pair-wise samples can be found on the path defined by the constructed minimum spanning tree. If any two samples sharing the same labels are not locally connected, then at least one additional sample from other cluster is included in the path. In this case the intra-cluster transitive distance between the two samples is at least as large as the inter-cluster transitive distance, which violates our assumption.

Proof of Lemma 3: Based on Lemma 2 the samples from the same cluster are locally connected and there exist a unique path outside the cluster that connect them to the other sample. It is very easy to infer that the transitive edges will always exist outside the cluster on this unique path. Therefore the transitive distance remains the same for all co-cluster samples.

Appendix D: Algorithm complexity analysis

Building the minimum spanning tree from a complete graph G needs time very close to $\mathcal{O}(n^2)$ by the algorithm in [1][1]. When Algorithm 2 stops, total n non-trivial tree[2] have been generated. The number of the edges in each non-trivial tree is not larger than n. Therefore, the total time taken by searching for the edge with the largest weight on each tree (step 5) in the algorithm is bounded by $\mathcal{O}(n^2)$. Steps 6-8 are for finding the values for the elements of E'. Since each element of E' is visited only once, the total time consumed by steps 6-8 is $\mathcal{O}(n^2)$. Thus the computational complexity of Algorithm 2 is about $\mathcal{O}(n^2)$.

Considering the time $\mathcal{O}(n^2)$ for building the distance matrix E, and the fact that the complexity of the k-means algorithm[3] is close to $\mathcal{O}(n^2)$, we conclude that the computational complexity of Algorithm 1 is about $\mathcal{O}(n2)$.

References

[1] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *J. ACM*, 2000.

2

[1] The fastest algorithm [1] to obtain a minimum spanning tree needs $\mathcal{O}(e\alpha(e, n))$ time, where e is the number of edges and $\alpha(e, n)$ is the inverse of the Ackermann function. The function α increases extremely slowly with e and n, and therefore in practical applications it can be considered as a constant not larger than 4. In our case, $e = \mathcal{O}(n^2)$ for a complete graph, so the complexity for building a minimum spanning tree is about $\mathcal{O}(n^2)$.

[2] A non-trivial tree is a tree with at least one edge.

[3] The time complexity of the k-means algorithm is $\mathcal{O}(npq)$, where p and q are the number of iterations and the dimension of the data samples, respectively. The data set Z' in Algorithm 1 is in R^n and thus $q = n$. In practical applications, p can be considered as smaller than a fixed positive number.