Towards Property-Based Classification of Clustering Paradigms: Supplementary material

Anonymous Author(s)
Affiliation
Address
City, State/Province, Postal Code, Country
email

In Appendix B, we include definitions of properties from the literature. In Appendix C some common clustering functions are defined. Appendix D includes proofs that were omitted from the body of the paper.

1 Appendix B: Definitions of properties from the literature

1.1 Isomorphism invariance

The following invariance property, proposed in [2] under the name "representation independence", seems to be an essential part of our understanding of what clustering is. It requires that the output of a clustering function is independent of the labels of the data points.

Definition 1. *A clustering function F is* isomorphism invariant *if whenever $(X, d) \sim (X', d')$, then, for every k, $F(X, d, k)$ and $F(X', d', k)$ are isomorphic clusterings.*

1.2 Scale invariance

Scale invariance, proposed by Kleinberg [8], requires that the output of a clustering be invariant to uniform scaling of the data.

Definition 2. *A clustering function F is* scale invariant *if for any data sets d and d', if there exists a real number $c > 0$ so that for all $x, y \in X$, $d(x, y) = cd'(x, y)$ then for every $1 \leqslant k \leqslant |X|$, $F(X, d, k) = F(X, d', k)$.*

1.3 Order invariance

Order invariance, proposed by Jardine and Sibson[6], describes clustering functions that are based on the ordering of pairwise distances.

A distance function d' of X is an *order invariant modification* of d over X if for all $x_1, x_2, x_3, x_4 \in X$, $d(x_1, x_2) < d(x_3, x_4)$ if and only if $d'(x_1, x_2) < d'(x_3, x_4)$.

Definition 3 (Order invariance)**.** *A clustering function F is* order invariant *if whenever a distance function d' over X is an order invariant modification of d, $F(X, d, k) = F(X, d', k)$ for all k.*

1.4 Consistency

Consistency, proposed by Kleinberg [8], aims to formalize the preference for clusters that are dense and well-separated. This property requires that the output of a clustering function should remain unchanged after shrinking within-cluster distances and stretching between-cluster distances.

Definition 4 (consistency)**.** • *Given a clustering C of some domain (X, d), we say that a distance function d' over X, is* (C, d)-consistent *if*

1. $d'_X(x, y) \leqslant d_X(x, y)$ *whenever* $x \sim_C y$*, and*
2. $d'_X(x, y) \geqslant d_X(x, y)$ *whenever* $x \not\sim_C y$*.*

- *A clustering function F is* consistent *if for every X, d, k, if d' is $(F(X, d, k), d)$-consistent then $F(X, d, k) = F(X, d', k)$.*

While this property may sound desirable and natural, it turns out that many common clustering paradigms fail to satisfy it. In a sense, this property may be viewed as the main weakness of Kleinberg's impossibility result. The following two properties are straightforward relaxations of consistency were proposed in [2]. Most of the common clustering paradigms satisfy at least one of them, and they can thus be used to highlight some inherent differences between clustering methods.

Threshold richness, which we introduce in Appendix A, can be viewed as an even more drastic relaxation that may be even be viewed as a possible axiom for clustering.

1.5 Inner and Outer consistency

Outer consistency represents the preference for well separated clusters, by requiring that the output of a clustering function should not change if clusters are moved away from each other.

Definition 5 (outer-consistency)**.** • *Given a clustering C of some domain (X, d), we say that a distance function d' over X, is* (C, d)-outer consistent *if*

1. $d'_X(x, y) = d_X(x, y)$ *whenever* $x \sim_C y$*, and*
2. $d'_X(x, y) \geqslant d_X(x, y)$ *whenever* $x \not\sim_C y$*.*

- *A clustering function F is* outer consistent *if for every X, d, k, if d' is $(F(X, d, k), d)$-outer consistent then $F(X, d, k) = F(X, d', k)$.*

Inner consistency represents the preference for placing points that are close together within the same cluster, by requiring that the output of a clustering function should not change if elements of the same cluster are moved closer to each other.

Inner consistency is defined in the same way outer-consistency, except that conditions 1 and 2 are modified as follows:

1. $d'_X(x, y) \leqslant d_X(x, y)$ whenever $x \sim_C y$, and
2. $d'_X(x, y) = d_X(x, y)$ whenever $x \not\sim_C y$.

Clearly, consistency implies both outer-consistency and inner-consistency. Note also that if a function is both inner-consistent and outer-consistent then it is consistent.

1.6 Richness properties

The richness property requires that we be able to obtain any partition of the domain by modifying the distances between elements. This property is based on Kleinberg's [8] richness axiom, and appears in [3].

Definition 6 (Richness)**.** *A clustering function F* satisfies richness *if for any sets $X_1, X_2, \ldots, X_k$, there exists a distance function d over $X' = \bigcup_{i=1}^{k} X_i$ so that $F(X', d, k) = \{X_1, X_2, \ldots, X_k\}$.*

We propose two new variants of the richness property: outer richness and inner richness.

1.7 Outer richness

Outer richness, a natural variation on the richness property, was proposed in [2] under the name "extended richness." (we have renames it to contrast this property with "inner richness" that we propose in Appendix A). Given k sets, a clustering function satisfies outer richness if by setting the distances between some data sets, without modifying distances within the sets, we can get F to output each of these data sets as a cluster. This corresponds to the intuition that any groups of points, regardless of within distances, can be made into separate clusters.

Definition 7 (Outer Richness)**.** *For every set of domains, $\{(X_1, d_1), \ldots (X_n, d_k)\}$, there exists a distance function $\hat{d}$ over $\bigcup_{i=1}^{n} X_i$ that extends each of the d_i's (for $i \leqslant k$), such that $F(\bigcup_{i=1}^{k} X_i, \hat{d}, k) = \{X_1, X_2, \ldots, X_k\}$.*

The following properties are more straightforward, explicit characteristics of certain specific clustering methods.

1.8 Hierarchical clustering

The term Hierarchical clustering is widely used to denote clustering algorithms that operate in a "bottom up" or "top down" manner. The following formalization of was proposed in [2].

A clustering C of X is a *refinement* of clustering C' of X if every cluster in C is a subset of some cluster in C', or, equivalently, if every cluster of C' is a union of clusters of C.

The range of hierarchical functions over the same data with different number of clusters is limited to clusterings that are refinements of each other.

Definition 8 (Hierarchical Functions)**.** *A clustering function is* hierarchical *if for every $1 \leqslant k \leqslant k' \leqslant |X|$, $F(X, d, k')$ is a refinement of $F(X, d, k)$.*

1.9 Locality

Intuitively, a clustering function is local if its behavior on a union of a set of clusters depends only on distances between elements of that union, and is independent of the rest of the domain set. Locality was proposed in [2].

Definition 9 (Locality)**.** *A clustering function F is* local *if for any clustering C output by F and every subset of clusters, $C' \subseteq C$,*

$$F(\bigcup C', d, |C'|) = C'.$$

In other words, for every domain (X, d) and number of clusters, k, if X' is the union of k' clusters in $F(X, d, k)$ for some $k' \leqslant k$, then, applying F to (X', d) and asking for a k'-clustering, will yield the same clusters that we started with.

2 Appendix C: Clustering functions

2.1 Linkage-based clustering

Linkage-based clustering algorithms are iterative algorithms that begin by placing each point in a distinct cluster, and then repeatedly merge the closest clusters until a specified number of clusters is formed.

The closest clusters are determined by a linkage function. The linkage functions used by the most common linkage-based algorithms are as follows.

- *Single linkage:* $\min_{a \in A, b \in B} d(a, b)$.

- *Average linkage:* $\frac{\sum_{a\in A, b\in B} d(a,b)}{|A|\cdot|B|}$
- *Complete linkage:* $\max_{a\in A, b\in B} d(a,b)$.

Linkage-based algorithms are sometimes allowed to run until a single cluster is formed, and the dendrogram produced by the algorithm is then pruned to obtain a clustering. We note that the properties in the above taxonomy can be reformulated in this framework, resulting in the same taxonomy of linkage based algorithms.

2.2 Objective-based clustering

Many clustering algorithms aim to find clusterings with low loss with respect to a specific objective function. An example of such an objective function is Min-Sum, the sum of within-cluster distances, $\sum_{x\sim_C y} d(x,y)$. Every objective function $\mathcal{O}$ has a corresponding clustering function which outputs a clustering that *optimizes* $\mathcal{O}$. Such clustering functions differ from other (often more computationally efficient) algorithms that aim to find clusterings with low loss with respect to a specific objective function, but often do not output an optimal solution. We now present centroid, k-means, and spectral clustering objective functions.

2.2.1 Centroid

Following Kleinberg's [8] definition, (k,g)-centroid clustering functions find a set of k "centroids" $\{c_1,\ldots,c_k\} \subseteq X$ so that $\sum_{x\in X} \min_i g(d(x,c_i))$ is minimized. Then k-median is obtained by setting g to the identity. In this setting, Kleinberg defines k-means by setting $g(x) = x^2$, we refer to this clustering function as Centroid k-means. The most common definition of k-means is below.

2.2.2 k-means

The k-means objective is to find a set of k elements $\{c_1, c_2, \ldots, c_k\}$ in the *underlying space*, so that $\sum_{x\in X} \min_i d(x,c_i)$ is minimized. This formalization assumes that the distance between any element of the domain and point in the underlying space is defined. k-means is often applied in Euclidean space, where the k-means objective is equivalent to $\sum_{C_i\in C} \frac{1}{|C_i|} \sum_{x,y\in C_i} d(x,y)^2$.

3 Appendix D: Proofs

3.1 Property Relationships

Theorem 1. *If a clustering function F satisfies richness and consistency, then it satisfies threshold-richness.*

Proof. Let $X = \cup_{i=1}^k X_i$ be some data set. Since F is rich, there exists a distance function d over X such that $F(X,d,k) = \{X_1,\ldots,X_k\} = C$. Let a' by the minimum within cluster distance. By using outer consistency, we construct distance function d' from d by making the minimum between cluster distance larger than the maximum within cluster distance, and let this value be b'. This is an outer consistent change, and so $F(X,d',k) = F(X,d,k)$. Take any distance function d^* such that $d^*(x,y) \leqslant a' \leqslant d'(x,y)$ if $x \sim_C y$, and $d^*(x,y) \geqslant b \geqslant d'(x,y)$ if $x \not\sim_C y$. By definition, d^* is a C, d'-consistent variant and therefore $F(X,d^*,k) = C$. □

3.2 Taxonomy proofs

Theorem 2. *Every (k,g)-centroid clustering function is local.*

Proof. For clustering $C = \{C_1, C_2, \ldots, C_k\}$, let $T = \{c_1, c_2, \ldots, c_k\}$ for some $c_i \in C_i$. We refer to elements of T as *centers*. Let $sum_{x\in X} \min_{t\in T} g(d(x,t))$ denote the loss of a clustering C. F returns the k-clustering C with minimal loss.

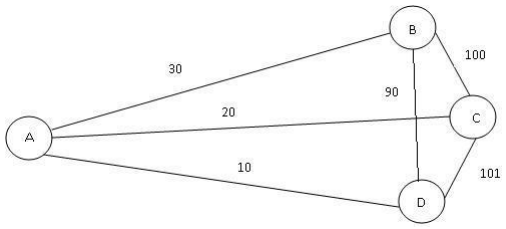

Figure 1: A data set used to illustrate that Ratio-Cut does not satisfy locality.

Let k'-clustering C' be a subset of $F(X, d, k)$. Let $S = X/C'$. Since g is non-decreasing, C' is induced by centers in $S \cap T$. Assume by way of contradiction that there exists a clustering C'' of S with lower loss than C'.

Then we can obtain a k-clustering of X with lower loss than $F(X, d, k)$ by clustering $X \cap S$ using C'' instead of C'. Since $F(X, d, k)$ has minimal loss over all k-clusterings of X, this is a contradiction. □

Centroid-based clustering functions are also outer-consistent.

Theorem 3. *Every (k,g)-centroid clustering function is outer-consistent.*

Proof. Assume by way of contradiction that some (k,g)-centroid clustering function F is not outer-consistent. Then there exists a data set (X, d), $k \in Z^+$ and d' an $(F(X, d, k), d)$-outer-consistent variant, so that $F(X, d, k) \neq F(X, d', k)$. Let $C = F(X, d, k)$ and $C' = F(X, d', k)$. Since d' is a (C, d)-outer-consistent variant, C has the same loss on (X, d) and (X, d'). $F(X, d', k) \neq C$, C' has lower loss than C on (X, d'). Now consider clustering (C, d) with C'. Since d' is a (C, d)-outer-consistent variant, for all $x, y \in X$, $d(x, y) \leqslant d'(x, y)$. Since g is non-decreasing, the loss of C' on (X, d) is at most the loss of C' on (X, d'). However, since $C = F(X, d)$, the minimal loss clustering on (X, d) is C, contradiction. □

It can be shown that Min-Sum is local and outer-consistent using similar arguments to Theorem 2 and Theorem 3.

Kleinberg showed that centroid-based clustering functions are not consistent (Theorem 4.1, [8]). Indeed, his proof shows that centroid-based clustering functions are not inner-consistent.

Theorem 4. *Ratio-Cut is not local.*

Proof. Figure 1 illustrates a data set (with the similarity indicated on the arrows) where the optimal ratio-cut 3-clustering is $\{\{A\}, \{B, C\}, \{D\}\}$. However, on data set $\{B, C, D\}$ (with the same pairwise similarities as in Figure 1), the clustering $\{\{B\}, \{C, D\}\}$ has lower ratio-cut than $\{\{B, C\}, \{D\}\}$. □

Let $vol(A_i) = \sum_{a,b \in A_i} s(a, b)$.

$$NormalizedCut(A_1, A_2, \ldots, A_k) = \sum_{1}^{k} \frac{cut(A_i, \bar{A}_i)}{vol(A_i)}.$$

We now show that normalized-cut is not local.

Theorem 5. *Normalized-Cut is not local.*

Proof. Figure 2 illustrates a data set with the similarities indicated on the arrows - a missing arrow indicates a similarity of 0. The optimal normalized-cut 3-clustering is

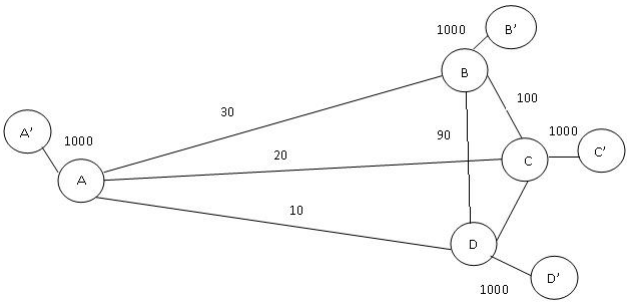

Figure 2: A data set used to illustrate that Ratio-Cut does not satisfy locality.

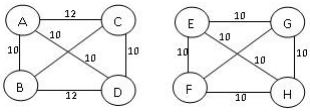

Figure 3: A data set used to illustrate that normalized cut does not satisfy inner-consistency. The similarities not marked are set to 0.

$\{\{A, A'\}, \{B, B', C, C'\}, \{D, D'\}\}$. However, on data set $\{B, B', C, C', D, D'\}$ (with the same pairwise similarity as in Figure 2), the clustering $\{\{B, B'\}, \{C, C', D, D'\}\}$ has lower normalized-cut than
$\{\{B, B', C, C'\}, \{D, D'\}\}$. □

We now prove that inner consistency distinguished between ratio cut and normalized cut.

Theorem 6. *Ratio-cut is inner-consistent.*

Proof. Assume by way of contradiction that ratio-cut is not inner-consistent. Then there exist some (X, s), k, and d' an $(F(X, d, k), s)$-outer inner-consistent variant so that $F(X, s', k) \neq F(X, s, k)$. Let $C = F(X, s, k)$ and $C' = F(X, d, k)$. Then $RatioCut(C', X, d') < RatioCut(C, X, d)$. Now consider clustering C' on (X, s). The ratio-cut of C' on (X, d) is at most the ratio-cut of C' on (X, s') since going from s' to s can only decrease similarities, which can only decrease the ratio-cut. That is, $RatioCut(C', X, s) \leqslant RatioCut(C', X, s')$. Therefore, $RatioCut(C', X, s) \leqslant RatioCut(C', X, s') < RatioCut(C, X, s)$, which contradicts that $F(X, s, k) = C \neq C'$. □

Theorem 7. *Normalized-cut is not inner-consistent.*

Proof. Consider the data set (X, d) in Figure 3. For $k = 3$, $NormalizedNut(X, d, 3) = \{\{A, C\}, \{B, D\}, \{E, F, G, H\}\}$. Define a distance function d' over X so that $d'(E, F) = d'(G, H) = 100$, and $d'(x, y) = d(x, y)$ for all $\{x, y\} \neq \{E, F\}$, $\{x, y\} \neq \{G, H\}$. Then d' is a $(F(X, d, 3), d), d)$-inner consistent change, however, $NormalizedCut(X, d', 3) = \{\{A, B, C, D\}\}$. But then $NormalizedCut(X, d', 3) \neq NormalizedCut(X, d, 3)$, violating inner-consistency. □

Lemma 1. *Ratio cut satisfies inner-richness.*

Proof. Consider any data set (X, s) and partitioning $\{X_1, X_2, \ldots, X_n\}$ of s. Let $m = \max_{i \neq j, a \in X_i, b \in X_j} s(a, b)$. Construct s' as follows: for all $i \neq j, a \in X_i, b \in X_j$, set $s'(a, b) = s(a, b)$. Otherwise, set $s(a, b) = m|X|^3 + 1$. The ratio cut loss of $\{X_1, X_2, \ldots, X_n\}$ on (X, s') is less than $m|X|^2$, and any other n-clustering of (X, s') has loss greater than $m|X|^2$. □

Lemma 2. *Normalized cut satisfies inner-richness.*

Proof. We can modify the within edges to make the normalized cut of the clustering $\{X_1, X_2, \ldots, X_k\}$ arbitrarily close to 0, making all within edges equal. The cost of any other clustering would have an edges (x, y) so that $x, y \in X_i$ for some i, and so the cost of any such clustering is arbitrarily greater than the cost of $\{X_1, X_2, \ldots, X_k\}$ (in particular, great than $1/m$ where m is the number of edges). □

Lemma 3. *Average linkage and complete linkage are not inner consistent.*

Proof. We present here a counter example for both. Let $X = \{A, B, C, D\}$ and define distance d over X as follows: $d(A, B) = 1 + \epsilon, d(A, C) = 1 - 3.5\epsilon, d(A, D) = 1, d(B, C) = 1 - 4\epsilon, d(B, D) = 1 - \epsilon$ and $d(C, D) = 1 - 2\epsilon$.

For sufficiently small epsilon, all individual lengths are approximately 1, but the sum of any path between two points in X is approximately 2 or more. For both average and complete linkage, B and C are merged first, followed by (B, C) and D. If we make an inner consistent change, and set $d(B, D) = 1 - 5\epsilon$, then B and D are merged first, followed by A and C. □

Lemma 4. *Min-sum is inner consistent.*

Proof. Given a data set (X, d), minsum yields a clustering C^* of X. Assume, by means of contradiction, that shrinking some within cluster edges yields a different clustering as the output to minsum, and denote this clustering by C'. Let the sum of all differences over the edges we shrunk be denoted by α, and the new distance function be denoted by d'. Define $cost(C, d) = \sum_{x \sim_C y} d(x, y)$. The difference between $cost(C', d')$ and $cost(C', d)$ is at most α. So, $cost(C', d') >= cost(C', d) - \alpha > cost(C^*, d) - alpha = cost(C^*, d')$, since C^* had the minimum cost with distance function d. □

Lemma 5. *Normalized cut and ratio cut are not outer consistent.*

Proof. We present a simple counter example. Let $X = \{a, b, c, d\}$ and define similarity function d over X as follows: $d(a, b) = 1, d(a, d) = 0.999, d(b, c) = 1.0015, d(c, d) = 1.001, d(a, c) = 0$ and $d(b, d) = 0$. With this arrangement, using ratio cut we arrive at the 3-clustering $a, d, \{b, c\}$. If we change the similarity between a and b to 0.997, which is an outer consistent change because we are dealing with similarities, then we arrive at the 3-clustering $a, b, \{c, d\}$. Therefore, ratio cut is not outer consistent. The same example works for normalized cut, except that we create points x_a, x_b, x_c, x_d such that $d(x_i, i) = 100$ and the similarity between x_i and every other point is 0. □

We briefly sketch the ideas for the remaining proofs in the above taxonomy. The linkage algorithms are outer-consistent since the distance between two cluster can only increase by increasing between cluster edges. Single linkage is inner-consistent since by Kleinberg's Theorem 2.2(a) single-linkage is consistent. The linkage-based algorithms are hierarchical by definition. It comes as no surprise that the remaining algorithms are not hierarchical, which can demonstrated by specific examples. Locality of the three linkage based algorithms is a result of the fact that the distance between two clusters depends only on the distances between those clusters.

It can be shown that single linkage and complete linkage are order invariant since the algorithm makes use only of relative distances according to the less-than relation. All other clustering functions that we classify make use of the exact values in the distance function, and it can be shown that those functions are not order invariant by demonstrating data sets with order invariant modifications of those data sets on which the output of the clustering functions differ.

Outer consistency of the spectral clustering algorithm is achieved by setting between cluster similarity to 0. Both outer and inner richness for the remaining clustering functions in the above taxonomy can be demonstrated by making the ratio of the maximum between edges and minimum within edge sufficiently large.

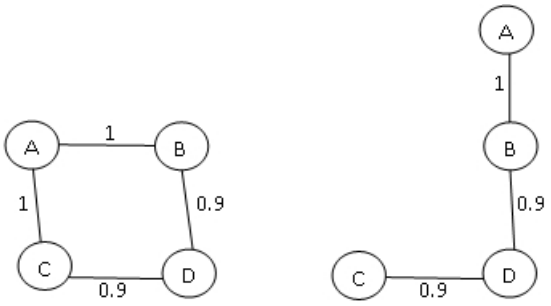

Figure 4: A data set used to illustrate that Furthest Centroids is not outer consistent.

It is easy to see that all algorithms listed in our taxonomy satisfy scale invariance and isomorphism invariance. It can also be easily verified that they are all threshold rich, and thus must also satisfy richness.

3.3 k-means proofs

We now present the proofs for the results displayed in our analysis of heuristics meant to approximate k-means. First, we note that all of the heuristics are obviously exemplar-based since they use the Lloyd method. Further, both heuristics are clearly scale invariant and isomorphism invariant.

Lemma 6. *Furthest Centroids satisfies outer richness.*

Proof. Given data sets $(X_1, d_1), \ldots, (X_n, d_n)$, place the data sets sufficiently far apart so that the n points selected for the initial centers belong to distinct X_is and $\{X_1, X_2, \ldots, X_k\}$ is a local minimum. □

Since Furthest Centroids satisfies outer richness, it also satisfies richness.

Lemma 7. *Furthest Centroids is not outer-consistent.*

Proof. Consider the data set (X, d) embedded in R^2 illustrated on the left hand size of Figure 4. For this example, all distances are as implied by the embedding. Set $k = 2$. Then KKZ selects the centers A and D, and outputs the clustering $C = \{\{A\}, \{B, C, D\}\}$. Consider the (C, d)-outer consistent change illustrated on the right hand side of Figure 4. A and C are selected as centers, and the algorithm outputs the clustering $\{\{A, B\}, \{C, D\}\}$. □

Lemma 8. *Furthest Centroids is not local.*

Proof. Furthest Centroids is not local because the selection of the initial center is a global decision. For example, consider the data set in Figure 5. Furthest Centroids for $k = 3$ selects A, B and E as the original centers. It then creates the clusters $\{A, C\}, \{B, D\}$ and $\{E\}$, which is a local optimum. However, if we restrict the data set to $\{A, B, C, D\}$ and set $k = 2$, the original centers are B and C. This leads to the clustering $\{\{A, C, D\}, \{B\}\}$, and A, C and D are closer to the center of mass of $\{A, C, D\}$ than to D, thus it is a local optimum. □

Finally, it is easy to see that furthest centroids, because the seeding is deterministic, is threshold rich. We simply ensure that b is large enough relative to a so that the furthest centres are always from separate clusters.

Lemma 9. *Random Centroid does not satisfy locality.*

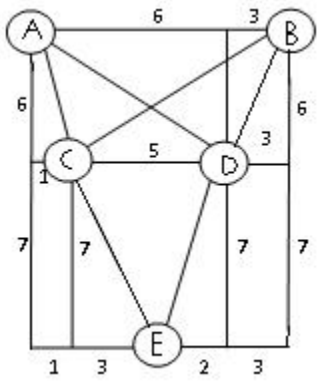

Figure 5: A data set embedding in R^2 illustrating that Furthest Centroids is not local. The pairwise distances are as shown in the embedding.

Proof. Consider the following data set on the real line. On the left there is a dense group of many points all at distance at most ϵ from each other, label this group of points X_1. Then 10 units to the right there is a point a, 20 units to the right of point a there is point b. Finally, 10 units to the right of b there is another group of points at distance at most ϵ from each other, call this group of points X_2. The groups X_1 and X_2 have the same number of points. We consider the case where we get two randomized centroids in X_1, and two in X_2. Because both X_1 and X_2 lie on a line, one point will be closest to the points labeled a and b. Further, due to the distances, all points in X_1 are closer to a than all points in X_2, and vice-versa. Thus, we will end with the 4-clustering $\{X_1, X_2, a, b\}$. If we consider now the subset of points $X_1 \cup a \cup b$ and ask for its 3-clustering, we can expect that all randomized centroids will be within X_1, and only one centre will consume both of a and b. Thus, we will end with a clustering that splits X_1 and has a and b together. In fact, the probability of having a and b in separate clusters approaches 0 as we increase the number of points in X_1. □

Lemma 10. *Random centroid satisfies richness.*

Here we give a sketch of the proof. For fixed k we consider the k-dimensional simplex. We position the data points at the vertices of the simplex, and form curves leaving the vertex such that each curve moves slightly closer to one other vertex in the simplex, while moving further away from every other vertex. We can do this for each vertex v by examining the $k-1$ k-dimensional hyperspheres centred at every other vertex v' with radius $d(v, v')$. To move closer to vertex v' from v while moving further away from every other vertex, we follow the curvature of the hypersphere centred at v', leading away from v. We simply do not move towards any intersection with any other hypersphere. We then move each successive point closer to v' as it moves away from v by some ϵ small enough so that we still do not intersect any other hypersphere. We distribute the points in each cluster evenly along each curve leading away from v to another vertex on the simplex. By making the distance between vertices large enough relative to within cluster distances, any point in a cluster will be far closer to every other point in the cluster than a point from a different cluster.

The last detail is to move away clusters that do not contain enough points to have at least one point per curve leading from vertex v to every other vertex in the simplex. For these clusters, we need to move them further away from every other vertex, and move *all* points onto one curve leading to some other vertex. This must be done for each cluster with too few points, but for each one all points must be moved onto a curve leading to a vertex that has not been used by another cluster with too few points. Thus, each cluster with too few points has all points in the cluster on a particular curve, leading to a single vertex in the simplex, and no other cluster with too few points has points on a curve leading to the same vertex.

Lemma 11. *Randomized centroid Lloyd method is not outer consistent.*

Proof. Consider a data set where the vast majority of points are in a unit ball centred at $(0, 0)$ in two dimensional euclidean space, such that no two points have the same x or y coordinate. Consider two other points, and make them far from the ball relative to its radius. Now, increase the distance between these two points along one dimension, so that they are both at the same y coordinate b, with $b > 0$, say, with one point being at $(-a, b)$ and the other being at (a, b). If we increase a large enough, and it doesn't have to be that large, the 3-clustering returned will consist of one cluster being the unit ball, and the other two clusters being the extra points we spaced far apart.

Now, maintaining the distance from the closest point in the unit ball, rotate both extra points up so that they lie on the y-axis at coordinate $\sqrt{a^2 + b^2}$, placing one of the points sufficiently far from the unit ball so that the distance between a and b is at least what it was in the previous arrangement. If we increase the y coordinate of these two points by a large enough factor, we can be certain that the random centre chosen from our unit ball with highest y-coordinate will be the one and only centre to be assigned both extra points. Also, by moving the two extra points further away, we can guarantee that the centre of mass will then shift outside of the unit ball, and the Lloyd method will stabilize at the following clustering: the unit ball will be split into two clusters, and the extra points will become a single cluster. □

Lemma 12. *Randomized centroid Lloyd method is not threshold rich.*

Proof. This follows trivially by looking at three cluster example. Given any $a < b$, we can make each cluster a ball with all points lying on surface of the ball. Then, regardless of how we arrange the balls, we can pick a ball that contains a point that is closer to two balls than two other points within the same ball. Thus, there is some probability that we will not get the clustering we desire that depends only on the number of points, and not on the arrangement.

□

Inner richness fails in Euclidean space, and so we have that any Lloyd method heuristic, regardless of initialization criteria, will not satisfy inner richness.

Lemma 13. *The Lloyd method with any initialization method does not satisfy inner richness.*

Proof. Consider the four points A, B, C and D, with distances between all points as on the real line embedding of these four points in a row, with A followed by B at distance 1, B followed by C at distance 7.9, then C followed by D at distance 1.1 so that the distance between A and D is 10. In order to apply the Lloyd method, the underlying space needs to satisfy the triangle inequality. Thus, if only the distance between C and D can be modified, it must remain at least 7.9. We argue that this restriction means that it is impossible to set the distance between C and D in such a way that the algorithm will outputs $\{\{A\}, \{B, C\}, \{D\}\}$. For any initial centers, the algorithm output either $\{A, B\}$ or $\{C, D\}$ as a cluster. □

References

[1] M. Ackerman and S. Ben-David. Measures of Clustering Quality: A Working Set of Axioms for Clustering. NIPS, 2008.

[2] Margareta Ackerman, Shai Ben-David, and David Loker. Characterization of Linkage-based Clustering. COLT, 2010.

[3] Reza Bosagh Zadeh and Shai Ben-David. "A Uniqueness Theorem for Clustering." The 25th Annual Conference on Uncertainty in Artificial Intelligence UAI, 2009.

[4] E. Forgy. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. In WNAR meetings, Univ of Calif Riverside, number 768, 1965.

[5] He, J., Lan, M., Tan, C.-L., Sung, S. -Y., and Low, H.-B. (2004). Initialization of cluster refinement algorithms: A review and comparative study. In Proc. IEEE Int. Joint Conf. Neural Networks (pp. 297-302).

[6] N. Jardine, R. Sibson, Mathematical Taxonomy Wiley, 1971.

[7] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang. A new initialization technique for generalized Lloyd iteration. IEEE Signal Processing Letters, 1(10):144146, 1994.

[8] Jon Kleinberg. "An Impossibility Theorem for Clustering." Advances in Neural Information Processing Systems (NIPS) 15, 2002.

[9] U. von Luxburg. A Tutorial on Spectral Clustering. Statistics and Computing 17(4): 395-416, 2007