Towards continuous consistency axiom

Mieczysław A. Kłopotek1 · Robert A. Kłopotek2

Accepted: 29 April 2022 / Published online: 30 June 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

It is shown for the first time in this paper, that Kleinberg’s (2002) (self-contradictory) axiomatic system for distance-based clustering fails (that is one of the data transforming axioms, consistency axiom, turns out to be identity transformation) in fixed-dimensional Euclidean space due to the consistency axiom limitations and that its replacement with inner-consistency or outer consistency does not help if continuous data transformations are required. Therefore we formulate a new, sound axiomatic framework for cluster analysis in the fixed dimensional Euclidean space, suitable for $k$-means like algorithms. The system incorporates centric consistency axiom and motion consistency axiom which induce clustering preserving transformations useful e.g. for deriving new labelled sets for testing clustering procedures. It is suitable for continuous data transformations so that labelled data with small perturbations can be derived. Unlike Kleinberg’s consistency, the new axioms do not lead the data outside of Euclidean space nor cause increase in data dimensionality. Our cluster preserving transformations have linear complexity in data transformation and checking. They are in practice less restrictive, less rigid than Kleinberg’s consistency as they do not enforce inter-cluster distance increase and inner cluster distance decrease when performing clustering preserving transformation.

Keywords Cluster analysis · Clustering axioms · Consistency · Continuous consistency · Inner- and outer-consistency · Continuous inner- and outer-consistency · Gravitational consistency · Centric consistency · Motion consistency · $k$-means algorithm

1 Introduction

Development of data mining algorithms, in particular also of clustering algorithms (see Def. 3), requires a considerable body of labeled data. The data may be used for development of new algorithms, fine-tuning of algorithm parameters, testing of algorithm implementations, comparison of various brands of algorithms, investigating of algorithm properties like scaling in sample size, stability under perturbation and other.

There exist publicly available repositories of labeled data1, there exist technologies for obtaining new labeled data sets like crowdsourcing [11], as well as extending existing small bodies of labeled data into bigger ones (semi-supervised learning [20, 38, 45]) as well as various cluster generators [19, 34].

Nonetheless these resources may prove sparse for an extensive development and testing efforts because of risk of overfitting, risk of instability etc. Therefore efforts are made to provide the developers with abundant amount of training data.

The development of axiomatic systems may be exploited for such purposes. Theoretical works on axiomatization of clustering algorithms provide with quite a cheap way to produce labeled data sets from existing ones. These axiomatic systems (Def. 2) include among others definitions of data set transformations under which some properties

---

1 See e.g. https://ifcs.boku.ac.at/repository/, https://archive.ics.uci.edu/ml/datasets.php
of the clustering algorithm will remain unchanged. The most interesting property is the partition of the data. The axiomatic systems serve also other purposes, like deepening the understanding of the concept of clusters, the group similarity, partition and the clustering algorithm itself.

Various axiomatic frameworks have been proposed, e.g. for unsharp partitioning by [47], for graph clustering by [28], for cost function driven algorithms by [7], for linkage algorithms by [2], for hierarchical algorithms by [10, 16, 42], for multi-scale clustering by [9], for settings with increasing sample sizes by [18], for community detection by [49], for pattern clustering by [39], and for distance based clustering [21], [7], see also [3]. Regrettably, the natural settings for many algorithms, like $k$-means, seem to have been ignored, that is (1) the embedding in the Euclidean space (Def.10), (2) partition of not only the sample but of the sample space, and (3) the behavior under continuous transformations (Def.12).

It would also be a useful property if the clustering would be the same under some perturbation of the data within the range of error. Perturbation $p(S)$ is an akin concept to continuous transformation in that for some small $\epsilon > 0$ the distance between $P \in S$ and $p(P)$ is below $\epsilon$.

The importance of the perturbation robustness in clustering has been recognized years ago and was studied by [5, 6, 33] among others.

The candidate axiomatic system for this purpose seems to be the mentioned Kleinberg’s system [21], as there are no formal obstacles to apply the axioms under Euclidean continuous settings. This axiomatic system deserves a very special attention in the Artificial Intelligence community, both from theoretical and practical point of view. It claims namely that it consists of three “natural axioms” for distance based clustering algorithms. But these three axioms are contradictory. What dies it imply in practice? It implies that the entire domain of distance-based clustering methods does not make sense at all and all such algorithms have to be thrown away. Even if the most controversial axiom of richness is restricted to so-called $k$-richness, then still the most popular family of algorithms around $k$-means does not fit the “natural” view of clustering.

Can we accept this point of view? The answer to this question by the authors of this paper is definitely “no”. We demonstrate that Kleinberg’s axioms are not natural at all, in particular the consistency axiom needs a repair and we provide one, the gravitational consistency (Theorem 23) under $k$-richness. But we resolve also the trouble implied by combination of Kleinberg’s consistency and richness, via introducing convergent consistency (Property 25). Furthermore we go beyond the plain repair. We propose axioms that are far less restrictive than Kleinberg’s consistency, that is the centric consistency (Property 29) and motion consistency (Def. 39). Therefore these new axioms can be considered as a break-through in formulating theoretical axioms and practical transformations for $k$-means like algorithms. Though we restrict ourselves to the domain of data embedded into Euclidean space, we have shown formerly [22, 25] a clear relationship between results of $k$-means clustering in Euclidean and non-Euclidean space.

In particular, one of the transformations implied by Kleinberg’s axioms, the consistency axiom transformation ($\Gamma$-transformation) (Property. 6) turns out to be identity transformation in Euclidean space, if continuity is required (Def.12.) Furthermore, its special case of inner-consistency (Def. 8) turns out to be identity transformation in Euclidean space even without continuity requirement (Theorem 16 and Theorem 17). The special case of outer-consistency (Def.9) suffers also from problems under continuous transformation (Theorem 20).

The possibility to perform continuous $\Gamma$-transformations preserving consistency is very important from the point of view of clustering algorithm testing because it is a desirable property that small perturbations of data do not produce different clusterings. In case that a consistency transform reduces to identity transform, the transformation is useless from the point of view of algorithm testing (same data set obtained).

Therefore, the axiom of consistency has to be replaced. We suggest to consider replacements for inner-consistency in terms of the centric consistency (Property.29) and the outer-consistency should be replaced with motion consistency (Def.39). Both special types of consistency need replacements because each of them leads to contradictions, as shown in Section 4.1

We will demonstrate that we repair in this way the basic deficiency of Kleinberg, that is the self-contradiction as well as transfer the axiomatic thinking into continuous domain (Theorem 42). In Section 12, we will verify the validity of our basic cluster-preserving transformations.

Our contribution consists in the following:

- Formulation of an axiomatic framework for cluster analysis in the fixed dimensional Euclidean space, suitable for $k$-means like algorithms and proof that it is free from contradictions. The system incorporates centric consistency axiom and motion consistency axiom.
- In particular introducing two clustering preserving transformations induced by this axiomatic system which are useful e.g. for deriving new labelled sets for testing clustering procedures.
Towards continuous consistency axiom

- The axiomatic framework is suitable for continuous data transformations which means that the reliability of clustering algorithms for small perturbations of the data can be tested when using proposed transformations.
- We show that in an analogous way cluster preserving transformations for single-link-type algorithms and for clustering functions with some special property of the cluster quality function as well as special hierarchical clustering functions can be applied.
- Demonstrating that the previously proposed and heavily cited axiomatic framework of Kleinberg [21] fails in Euclidean space because clustering preserving transformations like inner-Γ-transform reduce to identity transformation and therefore they are not suitable for clustering algorithms testing at all, and other like Γ-transform and outer-Γ-transform reduce to identity transformation in case of continuous transformations and therefore they are not suitable for clustering algorithms testing for small perturbations.
- We demonstrate that Kleinberg’s Γ-transform can be repaired if restricted to gravitational Γ-transform.
- We demonstrate with a series of experiments that Kleinberg’s consistency is much more restrictive than centric consistency and motion consistency so that for practical purposes the corresponding cluster preserving transforms are much more useful in test settings.

Though there are already hundreds of papers citing Kleinberg’s axiomatic framework [21], we are apparently the first ones to point at the self-contradictory idea of his consistency axiom when applied to a fixed dimensional Euclidean space which is home for many clustering algorithms including \(k\)-means. Also we are the first to demonstrate that the remedies like outer- or inner-consistency axioms do not help at all especially if continuous application of transformations is essential (small perturbation analysis). As a consequence, we are the first to introduce real remedies for \(k\)-means family, proposing the concepts of the gravitational consistency and convergent consistency. We achieve, however, the essential breakthrough by introducing the centric consistency and motion consistency concepts as replacements for inner- and outer-consistency.

These two latter concepts are of vital practical and theoretical importance. From practical point of view, they constitute a solid ground for application of cluster-preserving operations for generation of new test data from existent ones in \(\mathbb{R}^m\). Kleinberg’s consistency transform performed on distances my lead easily to higher dimensional spaces than \(m\) or even outside of Euclidean space. Though so-called kernel-\(k\)-means algorithm can operate without an embedding in \(\mathbb{R}^m\), but we have shown in [25] that the results may be ambiguous for non-Euclidean space and only a repair or reparability into Euclidean space makes the kernel-\(k\)-means results meaningful. AS the repair requires distance increases, it violates Kleinberg’s consistency axiom. Last not least it is vital that with these new properties we create a sound axiomatic system for \(k\)-means clustering and provide a guidance to generalizations for other algorithms, in particular by showing that increasing inter-cluster distances is not necessary for cluster preservation. What is more, even increase of inner-cluster distances does not need to violate cluster preservation (see e.g. the concept of bisectional-auto-\(k\)-means in this paper).

From the point of view of performance of cluster-preserving transformation, the motion-consistency transform and centric-consistency transforms are superior to Kleinberg’s Γ-transform. Transformation and eventual checking for each data point requires only inspection of the relationship to cluster center and other cluster centers. Application of Kleinberg’s Γ-transform requires inspection of each pair of data points to check for proper increase/decrease of distances. But we face with Γ-transform a far more serious problem: it may lead to a higher-dimensional Euclidean space or to a non-Euclidean space which requires a not so easy correction to keep Kleinberg’s consistency constraints, thus a task of hardly predictable complexity. Our transforms leave the data points in the same Euclidean spaces as prior to transformation.

The paper is organized as follows: Section 2 recalls the previous work in the area. Section 3 introduces notation used. Section 4 presents our theoretical results. Sections 5-11 present proofs of theorems formulated in Section 4. Section 12 presents results of experimental investigations. Section 13 discusses issues raised by the theoretical and experimental investigations. Section 14 contains concluding remarks.

2 Previous work

Let us recall that an

**Definition 1** Axiom or postulate is a proposition regarded as self-evidently true without proof.

A clustering axiom is understood as a property that has to hold for a clustering algorithm in order for that algorithm to be considered as a clustering algorithm.

**Definition 2** An axiomatic system is a set of axioms that are considered to hold at the same time.
If an axiomatic system holds for an algorithm, then one can reason about further properties of that algorithm without bothering of the details of that algorithm.

Kleinberg in the paper [21, Section 2], cited more than 600 times according to Google Scholar, defines clustering function as:

**Definition 3** A clustering function is a function \( f \) that takes a distance function \( d \) on [set] \( S \) [of size \( n \geq 2 \)] and returns a partition \( \Gamma \) of \( S \). The sets in \( \Gamma \) will be called its clusters.

Given that, Kleinberg [21] formulated axioms for distance-based cluster analysis (Properties 4, 5, 6), which are rather termed properties by other e.g. [3].

**Property 4** Let \( \text{Range}(f) \) denote the set of all partitions \( \Gamma \) such that \( f(d) = \Gamma \) for some distance function \( d \). A function \( f \) has the richness property if \( \text{Range}(f) \) is equal to the set of all partitions of \( S \).

**Property 5** A function \( f \) has the scale-invariance property if for any distance function \( d \) and any \( \alpha > 0 \), we have \( f(d) = f(\alpha \cdot d) \).

**Property 6** Let \( \Gamma \) be a partition of \( S \), and \( d \) and \( d' \) two distance functions on \( S \). We say that \( d' \) is a \( \Gamma \)-transformation of \( d \) (\( (\Gamma(d) = d') \)) if (a) for all \( i, j \in S \) belonging to the same cluster of \( \Gamma \), we have \( d'(i, j) \leq d(i, j) \) and (b) for all \( i, j \in S \) belonging to different clusters of \( \Gamma \), we have \( d'(i, j) \geq d(i, j) \). The clustering function \( f \) has the consistency property if for each distance function \( d \) and its \( \Gamma \)-transform \( d' \) the following holds: if \( f(d) = \Gamma \), then \( f(d') = \Gamma \).

For algorithms like \( k \)-means, Kleinberg defined also the property of \( k \)-richness: in which he requires not generation of “all partitions of \( S \)” but rather of “all partitions of \( S \) into exactly \( k \) non-empty clusters”.

**Property 7** Let \( \text{Range}(f) \) denote the set of all partitions \( \Gamma \) such that \( f(d) = \Gamma \) for some distance function \( d \). A function \( f \) has the \( k \)-richness property if \( \text{Range}(f) \) is equal to the set of all partitions of \( S \) into \( k \) non-empty sets.

Kleinberg demonstrated that his three axioms (Properties of richness 4, scale-invariance 5 and consistency 6) cannot be met all at once. (but only pair-wise), see his Impossibility Theorem [[21] Theorem 2.1]. In order to resolve the conflict, there was proposed a replacement of consistency axiom with order-consistency axiom [48], refinement-consistency ([21]), inner/outer-consistency [3], order-invariance and so on (see [1]). We will be interested here in two concepts:

**Definition 8** Inner-consistency is a special case of consistency in which the distances between members of different clusters do not change.

**Definition 9** Outer-consistency is a special case of consistency in which the distances between members of the same cluster do not change.

Note that efforts were made to extend the axiomatic framework of Kleinberg to soft clustering, see e.g. [8], where also a soft version of \( k \)-means is analysed within the framework of so-called three-way-clustering [44]. Same approach is used in the domain of rough sets to develop “natural” clustering axioms [14]. The issue of cluster preserving is also pursued in [43] in preferential logic.

The interest in cluster-preserving transformations goes beyond generating new labelled data sets. Such transformations are of interest e.g. in privacy preserving clustering [50], in detection of heavy hitters in streams [29], in dimensionality reduction [30].

### 3 Preliminaries

In this paper, we depart from the assumptions of previous papers on Kleinberg’s axiomatic system and its variants in that we introduce the assumption of continuity in the aspects: (1) the embedding of data in a fixed-dimensional space is assumed (Def. 11, 10), (2) the \( \Gamma \)-transformation has to be performed in a continuous way (Def.12, Def.13) and (3) the data set \( S \) is considered as a sample from a probability density function over the continuous space, like discussed already by Pollard [36] in 1981 and by MacQueen [32] in 1967 when considering the in the limit behavior of \( k \)-means target function, and by Klopotek [23] for \( k \)-means++. We assume that this probability density function is continuous. In this paper, also, we will go beyond the classical convex-shaped structure of \( k \)-means clusters.

Let us first explain our understanding of embedding of the data points in a fixed-dimensional Euclidean space, with distance between data points implied by the embedding and continuous versions of data transforms will be considered.

**Definition 10** An embedding of a data set \( S \) into the \( m \)-dimensional space \( \mathbb{R}^m \) is a function \( \mathcal{E} : S \rightarrow \mathbb{R}^m \) inducing a distance function \( d_\mathcal{E}(i, j) \) between these data points being the Euclidean distance between \( \mathcal{E}(i) \) and \( \mathcal{E}(j) \).
We consider here finite dimensional Euclidean spaces $\mathbb{R}^m$. The continuity in this space is understood as follows:

**Definition 11** A space / set $S \subseteq \mathbb{R}^m$ is continuous iff within a distance of $\epsilon > 0$ from any point $P \in S$ there exist infinitely many points $Q \in S$.

Same applies to clusters: within a distance of $\epsilon > 0$ from any point $P$ of a cluster $C$ there exist infinitely many points $Q$ of the same cluster $C$. That is while a clustering algorithm like $k$-means operates on a finite sample, it actually splits the entire space into clusters, that is it assigns class membership under continuous changes of positions in space.

**Definition 12** By a continuous transformation of a (finite or infinite) data set $S$ in such a space ($S \subseteq \mathbb{R}^m$) we shall understand a function $t(t(S),P)$, where $t \in [0,1]$ for any $P \in S$, t(0, P) = P and for each $t \in [0,1]$ and each $\epsilon > 0$ there exist $\delta_{t,\epsilon} > 0$ such that for any $0 < \delta < \delta_{t,\epsilon}$ and any $t \in [0,1]$ the point $Q = t(t + \delta, P)$ lies within the distance of $\epsilon$ from $t(t, P)$ for each $P$.

$k$-means, kernel-$k$-means, their variants and many other clustering functions rely on the embedding into Euclidean space which raises the natural question on the behavior under continuous changes of positions in space.

Let $\mathcal{E}_m$ be the set of all possible embeddings of the data set $S$ in $\mathbb{R}^m$. Define:

**Definition 13** We speak about continuous consistency transformation or continuous $\Gamma$-transformation $t$ of a clustering $\Gamma$ if $t$ is a continuous transformation as defined in Def.12 and $d(t(t(S), P)), t(t(Q))) \geq d(t(t(S), P)), t(t(Q)))$ for any $0 \leq t < t' \leq 1$ if $P, Q$ belong to the same cluster $C \in \Gamma$ and $d(t(t(S), P), t(t(Q))) \leq d(t(t(S), P), t(t(Q)))$ for any $0 \leq t < t' \leq 1$ if $P, Q$ do not belong to the same cluster.

The transform for the probability density function is understood in a natural way as a transformation of the probability density estimated from the sampling. The clustering function $f$ is also understood as partitioning the space in such a way that with increasing sample size the sample-based partition of the space approximates the partition of the probability density, as discussed in [26]. Continuous inner-$\Gamma$-transformation/consistency and continuous outer-$\Gamma$-transformation/consistency are understood in analogy to inner/outer-$\Gamma$-transformation/consistency.

Subsequently, whenever we talk about distance, we mean Euclidean distance.

**Property 14** A clustering function $f$ has the continuous-consistency property, if for each dataset $S$ and the induced clustering $\Gamma = f(S)$ each continuous $\Gamma$-transformation $t$ has the property that for each $0 \leq t \leq 1$ the clustering of $t(t(S))$ will result in the same clustering $\Gamma$.

The property of continuous consistency is a very important notion from the point of view of clustering in the Euclidean space, especially in the light of research on perturbation robustness [5, 6, 33].

Note that a $\Gamma$-transformation cannot be in general replaced by a finite sequence of inner-consistency and outer $\Gamma$-transformations. But if we have to do with a continuous $\Gamma$-transformation then it can be replaced by a continuum of inner-consistency and outer $\Gamma$-transformations.

Let us make the remark, that the inner-$\Gamma$-transformation and the outer-$\Gamma$-transformation combination cannot replace the $\Gamma$-transformation, even if we use a finite sequence of such transformations. However we can think of a sequence of sequences of $\Gamma$-transformations approximating a continuous $\Gamma$-transformation. By a sequence $t_{i,n}$ of $\Gamma$-transformations of a data set $S$ we understand the following sequence: $t_{0,n}(S) = S$ and for each $i = 1, \ldots, n$, $t_{i,n}(S)$ is a $\Gamma$-transformation of $t_{i-1,n}(S)$. In particular, for a continuous $\Gamma$-transformation $t$, the sequence $t(i/n, S)$ for $i = 0, \ldots, n$ would be such a sequence. We say that a sequence $t_{i,n}$ approximates the the continuous transformation $t$ with precision $\pi$, if for each $P \in S$ the distance between $t_{i,n}(P)$ and $t(t, P)$, where $t \in [i/n, (i + 1)/n]$ does not exceed $\pi$.

**Theorem 15** If $t$ is a continuous-$\Gamma$-transformation, then there exists an $n$ for which the approximating sequence of $\Gamma$-transformations $t_{i,n}$ can be found approximating $t$ up to $\pi$. What is more, for possibly a larger $n$, a sequence exists consisting of a mixture of only inner and outer $\Gamma$-transformations.

**Proof** It can be shown by appropriate playing with the $\pi$s, $e$s at various scales of $\delta$.

The Kleingberg’s axioms reflect three important features of a clustering (1) compactness of clusters, (2) separation of clusters and (3) the balance between compactness and separation. In particular, the inner-consistency property says that the increase of compactness of a cluster should preserve the partition of data. The outer consistency property says that the increase of separation of clusters should preserve the partition of data. The scale-invariance states that preserving the balance between compactness and separation should
preserve partition. The richness means that variation of balances between compactness and separation should lead to different partitions.

We will demonstrate however a number of impossibility results for continuous inner and outer-consistency (Section 4.1). Essentially we show that in a number of interesting cases these operations are reduced to identity operations.

Therefore we will subsequently (Sections 4.2 and 4.3) seek replacements for continuous consistency, that will relax their rigidity, by proposing centric consistency introduced in [26] as a replacement for inner-consistency, and motion-consistency, introduced in [27] as a replacement for outer-consistency. We will demonstrate that under this replacement, the (nearly) richness axiom may be fulfilled in combination with centric consistency and scale-invariance.

As we will refer frequently to the $k$-means algorithm, let us recall that it is aimed at minimizing the cost function $Q$ (reflecting its quality in that the lower $Q$ the higher the quality) of the form:

$$Q(\Gamma) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \|x_i - \mu_j\|^2 = \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x_l \in C_j} \|x_l - \mu_j\|^2$$

for a dataset $X$ under some partition $\Gamma$ into the predefined number $k$ of clusters, where $u_{ij}$ is an indicator of the membership of data point $x_i$ in the cluster $C_j$ having the cluster center at $\mu_j$ (which is the cluster’s gravity center). Note that [36] modified this definition by dividing the right hand side by $m$ in order to make comparable values for samples and the population, but we will only handle samples of a fixed size so this definition is sufficient for our purposes. A $k$-means algorithm finding exactly the clustering optimizing $Q$ shall be referred to as $k$-means-ideal. Realistic implementations start from a randomized initial partition and then improve the $Q$ iteratively.

An algorithm where the initial partition is obtained by random choice of cluster centers shall be called random-seed $k$-means, with $R$ implementation being an example of such an algorithm. For various versions of $k$-means algorithm see e.g. [46].

4 Main results

4.1 Internal contradictions of consistency axioms

In this section we demonstrate that inner consistency axiom alone induces contradictions (Theorem 16 and Theorem 17). The continuous versions of the respective consistency transformations amplify the problems. Theorem 18 shows that if concave clusters are produced, continuous consistency transform may not be applicable. In general settings, continuous outer consistency transform (Theorem 20) may not be applicable to convex clusters either. These inner contradictions can be cured by imposing additional constraints on continuous consistency transform in form of gravitational (Def.22), or homothetic (Theorem 24) or convergent (Property 25) transforms.

We claim first of all that inner-consistency cannot be satisfied non-trivially for quite natural clustering tasks, see Theorem 16 and Theorem 17.

\begin{theorem}
In $m$-dimensional Euclidean space for a data set with no cohyperplanarity of any $m + 1$ data points (that is when the points are in general position) the inner-$\Gamma$-transform is not applicable non-trivially, if we have a partition into more than $m + 1$ clusters.
\end{theorem}

See proof on page 17. By non-trivial we mean the case of $\Gamma$-transformation different from isometric transformation. Note that here we do not even assume continuous inner-consistency. Non-existence of non-trivial inner-consistency makes continuous one impossible.

As the data sets with properties mentioned in the premise of Theorem 16 are always possible, no algorithm producing the $m + 1$ or more clusters in $\mathbb{R}^m$ has non-trivially the inner-consistency property. This theorem sheds a different light on claim of [3] that $k$-means does not possess the property of inner-consistency. No algorithm producing that many clusters has this property if we consider embeddings into a fixed dimensional Euclidean space. Inner-consistency is a special case of Kleinberg’s consistency.

The above Theorem can be slightly generalized:

\begin{theorem}
In a fixed-dimensional space $\mathbb{R}^m$ under Euclidean distance if there are at least $m + 1$ clusters such that we can pick from each cluster a data point so that $m + 1$ points are not co-hyperplanar (that is they are “in general position”), and at least two clusters contain at least two data points each, then inner-$\Gamma$-transform is not applicable (in non-trivial way).
\end{theorem}

See proof on page 17. Again continuous inner-consistency was not assumed.

\[\text{ Springer}^3\]
Towards continuous consistency axiom

**Theorem 18** If a clustering function is able to produce concave clusters such that there are mutual intersections of one cluster with convex hull of a part of the other cluster, then it does not have the continuous consistency property if identity transformation is excluded.

See proof on page 17. The Theorem 18 referred to clustering algorithms producing concave clusters. But what when we restrict ourselves to partitions into convex clusters? Let us consider the continuous outer-consistency.

**Definition 19** Consider three clusters $C_1, C_2, C_3$. If the points $S_i$, the separating hyperplanes $h_{ij}$ and vectors $v_{ij}$ with properties mentioned below exist, then we say that the three clusters follow the three-cluster-consistency principle. The requested properties are: (1) There exist points $S_1, S_2, S_3$ in convex hulls of $C_1, C_2, C_3$ resp. such that $C_i$ is separated from $S_j$ by a hyperplane $h_{ij}$ such that the line segment $S_iS_j$ is orthogonal to $h_{ij}$ for $i, j = 1, 2, 3$. (2) Let $v_{ij}$ be the speed vector of cluster $C_i$ relatively to $C_j$, $v_{12}$ for $i, j = 1, 2, 3$ such that $v_{12} \perp h_{12}, v_{13} \perp h_{13}$, and $v_{23} \perp h_{23}$, and $\|v_{12}\|/|S_1S_2| = \|v_{13}\|/|S_1S_3|$. 

Note that condition (2) implies that $v_{23} = v_{12} - v_{13}$ and $\|v_{12}\|/|S_1S_2| = \|v_{23}\|/|S_2S_3|$. 

**Theorem 20** If a clustering function may produce partitioning such that there exists a sequence of cluster indices (first and last being identical) so that one is unable to find hyperplanes separating the clusters not violating for any subsequence of three clusters the three-cluster-consistency principle and without a mismatch between the first and the last $S$ point, then this function does not have continuous outer-consistency property if we discard the isometric transformation.

See proof on page 20. 

Note that $k$-means clustering function does not suffer from the deficiency described by the precondition of Theorem 20 because it implies a Voronoi tessellation: if we choose always the gravity centers of the clusters as the $S$ points, then $h$ hyperplanes can always be chosen as hyperplanes orthogonal to $S_jS_i$ lying half-way between $S_i$ and $S_j$, Hence

**Theorem 21** Given $k$-means clustering (in form of a Voronoi tessellation), there exists a non-trivial continuous outer-$\Gamma$-transform preserving the outer-consistency property that we fix the position of one cluster $C$ and move each other cluster $C_j$ with a speed vector proportional to the vector connecting the cluster center $C$ with cluster center $C_j$.

So, by irony, the $k$-means clustering algorithm for which Kleinberg demonstrated that it does not have the consistency property, is the candidate for a non-trivial continuous outer consistency variant. Let us elaborate now on this variant.

**Definition 22** A $\Gamma$-transform has the property of gravitational consistency, if within each cluster the distances between gravity centers of any two disjoint subsets of cluster elements do not increase.

**Theorem 23** Under gravitational $\Gamma$-transformation, the $k$-means possesses the property of Kleinberg’s consistency and continuous consistency.

See proof on page 22. 

The gravitational consistency property is not easy to verify (the number of comparisons is exponential in the size of the cluster). However, some transformations have this property, for example:

**Theorem 24** The homothetic transformation of cluster with a ratio in the range $(0,1)$ in the whole space or any subspace has the property that the distances between gravity centers of any two disjoint subsets of cluster elements do not increase.

See proof on page 23. 

The gravitational consistency may be deemed as a property that is too $k$-means specific.

Therefore, we considered, in the paper [26], the problem of Kleinberg’s consistency in its generality. We showed there that the problem of Kleinberg’s contradictions lies in the very concept of consistency because his $\Gamma$-transformation creates new clusters within existent one (departs from cluster homogeneity) and clusters existent ones into bigger ones (balance between compactness and separation is lost). This effect can be avoided probably only via requiring the convergent consistency, as explained below.

**Property 25** Define the convergent-$\Gamma$-transform as $\Gamma$-transform from Property 6 in which if $d(i, j) \leq d(k, l)$, then also $d'(i, j) \leq d'(k, l)$ and $d(i, j)/d(k, l) \leq d'(i, j)/d'(k, l)$. The clustering function $f$ has the convergent consistency property if for each distance function $d$ and its convergent $\Gamma$-transformation, with $f(d) = d'$ the following holds: if $f(d) = \Gamma$, then $f(d') = \Gamma$.
Theorem 26 The axioms of richness, scale-invariance and convergent-consistency are not contradictory.

See proof in paper [26].

When we actually cluster data sampled from a continuous distribution, we would expect the “in the limit” condition to hold, that is that the clustering obtained via a sample converges to the clustering of the sample space.

Theorem 27 If the in-the-limit condition has to hold then the convergent-\(\Gamma\)-transform must be an isomorphism.

See proof in paper [26].

For this reason, under conditions of continuity, we need to weaken the constraints imposed by Kleinberg on clustering functions via his consistency axiom. As stated in Theorem 15, the \(\Gamma\)-transformation of Kleinberg under continuous conditions can be replaced by a limit on a sequence of inner and outer \(\Gamma\)-transforms. Deficiencies of both have been just listed.

Therefore we propose here two relaxations of the two properties: centric consistency (Section 4.2) as a replacement of inner-consistency and motion consistency (Section 4.3) as a replacement of outer-consistency.

4.2 Centric consistency

Let us start with the centric consistency as a remedy for contradictions with inner-consistency. It is inspired by the result of Theorem 24.

Definition 28 Let \(\mathcal{E}\) be an embedding of the dataset \(S\) with distance function \(d\) (induced by this embedding). Let \(\Gamma\) be a partition of this dataset. Let \(C \in \Gamma\) and let \(\mu_C\) be the gravity center of the cluster \(C\) in \(\mathcal{E}\). We say that we execute the \(\Gamma^\ast\) transformation (or a centric transformation, \(\Gamma(d; \lambda) = d^\prime\)) if for some \(0 < \lambda \leq 1\) a new embedding \(\mathcal{E}^\prime\) is created in which each element \(x\) of \(C\), with coordinates \(x\) in \(\mathcal{E}\), has coordinates \(x^\prime\) in \(\mathcal{E}\) such that \(x^\prime = \mu_C + \lambda(x - \mu_C)\), all coordinates of all other data points are identical, and \(d^\prime\) is induced by \(\mathcal{E}^\prime\). \(C\) is then said to be subject of the centric transform.

Note that the set of possible centric \(\Gamma\)-transformations for a given partition is neither a subset nor superset of the set of possible Kleinberg’s \(\Gamma\)-transformation in general. However, if we restrict ourselves to the fixed-dimensional space and to the fact that the inner-consistency property means in this space isometry, as shown in Theorems 16 and 17, then obviously the centric consistency property can be considered as much more general, in spite of the fact that it is a \(k\)-means clustering model-specific adaptation of the general idea of shrinking the cluster.

Property 29 A clustering method has the property of centric consistency if after a \(\Gamma^\ast\) transform it returns the same partition.

Theorem 30 \(k\)-means algorithm satisfies centric consistency property in the following way: if the partition \(\Gamma\) of the set \(S\) with distances \(d\) is a global minimum of \(k\)-means, and \(k = 2\), and the partition \(\Gamma^\prime\) has been subject to centric \(\Gamma^\ast\)-transformation yielding distances \(d^\prime\), then \(\Gamma^\prime\) is also a global minimum of \(k\)-means under distances \(d^\prime\).

See proof in paper [26].

For the sake of brevity, let us use the following convention subsequently. By saying that a \(\Gamma\) transformation produced the clustering \(\Gamma^\prime\) out of clustering \(\Gamma\) we mean that the original embedding was changed in such a way that any data point \(x\) with original coordinates \(x\) is located now at \(X\). If we say that the transformation turned elements of sets \(P, Q\) into \(P^\prime, Q^\prime\), then we mean that elements of \(P\) did not change coordinates while the elements of \(Q\) changed coordinates under new embedding.

Theorem 31 Let a partition \(\{T, Z\}\) be an optimal partition under \(2\)-means algorithm. Let a subset \(P\) of \(T\) be subjected to centric transform yielding \(P^\prime\) (that is all points of \(P\) were moved to the gravity center of \(P\) by a factor \(\lambda\)), and \(T^\prime = (T - P) \cup P^\prime\). Then partition \(\{T^\prime, Z\}\) is an optimal partition of \(T^\prime \cup Z\) under \(2\)-means.

See the proof on page 33.

Let us discuss a variant of bisectional-\(k\)-means by [40]. The idea behind bisectional versions of \(k\)-means is that we apply \(2\)-means to the entire data set and then recursively \(2\)-means is applied to some cluster obtained in the previous step, e.g. the cluster with the largest cardinality. The algorithm is terminated e.g. upon reaching the desired number of clusters, \(k\). Note that in this case the \(k\)-means-ideal quality function is not optimized, and even the cluster borders do not constitute a Voronoi diagram. However, we will discuss, for the purposes of this paper, a version of \(k\)-means such that the number of clusters is selected by the algorithm itself. We introduce a stopping criterion that a cluster is not partitioned, if for that cluster the decrease of \(Q\) (Def. 1) for the bisection relatively to the original cluster \(Q\) does not reach some threshold. An additional stopping criterion will be used, that the number of clusters cannot exceed \(k\).

It is easily seen that such an algorithm is scale-invariant (because the stopping criterion is a relative one) and it is also rich “to a large extent”, that is we exclude partitions with more than \(k\) clusters only.

If a clustering algorithm can return any clustering except one with the number of cluster over \(k\), we shall call this
property \( k \downarrow\text{-near-richness} \). We shall call it \textit{bisectional-auto –} \( k \)-means algorithm.

Theorem 31 implies that

**Theorem 32** Bisectional-auto – \( k \)-means algorithm is centric-consistent, (scale)-invariant and \( k \downarrow\)-nearly-rich.

See proof on page 36. Note that the Kleinberg’s impossibility theorem [21] could be worked around so far only if we assumed clustering into exactly \( k \) clusters. But one ran into contradictions already when the number of clusters could have a range of more than two values, because the so-called anti-chain property of the set of possible partitions does not hold any more (see [21] for the impact of missing antichain property on the contradictions of Kleinberg axioms). So this theorem shows the superiority of the concept of centric consistency. Though centric consistency relaxes only inner-consistency, nonetheless Kleinberg’s axiomatic system with inner-consistency fails also.

Let us now demonstrate theoretically, that \( k \)-means algorithm really fits \textit{in the limit} the centric-consistency axiom.

**Theorem 33** \( k \)-means algorithm satisfies centric consistency in the following way: if the partition \( \Gamma \) is a local minimum of \( k \)-means, and the partition \( \Gamma \) has been subject to centric consistency yielding \( \Gamma' \), then \( \Gamma' \) is also a local minimum of \( k \)-means.

See proof on page 26. However, it is possible to demonstrate that the newly defined transform preserves also the global optimum of \( k \)-means.

**Theorem 34** \( k \)-means algorithm satisfies centric consistency in the following way: if the partition \( \Gamma \) is a global minimum of \( k \)-means, and the partition \( \Gamma \) has been subject to centric consistency yielding \( \Gamma' \), then \( \Gamma' \) is also a global minimum of \( k \)-means.

See the proof on page 27.

Hence it is obvious that

**Theorem 35** \( k \)-means algorithm satisfies Scale-invariance, \( k \)-Richness, and centric Consistency.

Note that \( k \)-means does not have the property of inner-consistency. This means that the centric consistency truly relaxes inner-consistency.

Under some restrictions, the concept of “centric consistency” may be broadened to non-convex clustering methods. Consider the \( k \)-single link algorithm product \( \Gamma \). Each cluster \( C \) can be viewed as a tree \( T_C \) if we consider the links in the cluster as graph edges. Define the area \( A_C \) of a cluster \( C \) as the union of balls centered at each node and with a radius equal to the longest edge incident with the node. A cluster may be deemed \textit{link-ball-separated} if for each node \( N \notin C \) the distance of \( N \) to \( A_C \) is bigger than any edge incident with \( N \). In such a case let move a leaf node \( L \) of \( T_C \) along the edge connecting it to \( T_C \) by factor \( 1 > \lambda_L > 0 \). For non-leaf nodes \( M \) if a branch starting at \( M \) lies entirely within original \( A_C \), let move all nodes of the branch can be moved by the same factor \( \lambda_M \) towards \( M \). Both operations be called semi-centric transformation.

**Theorem 36**

(a) If the cluster \( C \) is link-ball-separated, then application of semi-centric transformation yield a new data set that has the same cluster structure as \( \Gamma \), and \( C \) remains link-ball-separated.

(b) If the cluster \( C_1 \) is link-ball-separated from other clusters and the cluster \( C_2 \) is link-ball-separated from other clusters and a just-mentioned transformation is performed on \( C_1 \), then both remain link-ball-separated

The proof relies on trivial geometric observations. Part (a) is valid because distances will decrease within a cluster and no node from outside will get closer to \( N' \) than to nodes from its own cluster, though now the tree \( T_C' \) of \( C \) may connect other nodes than \( T_C \). The cluster area \( A_{C'} \) will be a subarea of \( A_C \) hence \( C \) is still \textit{link-ball-separated} from the other clusters. This transformation may violate Kleinberg’s outer as well as inner consistency constraints, but will still yield new data sets with same clustering properties. Part (b) holds due to smaller decrease of between-cluster distances compared to the \( \lambda \).

Note that such “area” restrictions are not needed in case of \( k \)-means centric consistency.

**4.3 Motion consistency**

Centric consistency replaces only one side of Kleinberg’s consistency, the so-called inner-consistency. But we need also a replacement for the outer-consistency. Let us therefore introduce the motion-consistency (see [27]).

**Definition 37** Cluster area is any solid body containing all cluster data points.

*Gap between two clusters* is the minimum distance between the cluster areas, i.e., Euclidean distance between the closest points of both areas.

**Definition 38** Given a clustered data set embedded in a fixed dimensional Euclidean space, the motion-transformation is any continuous transformation of the data
set in the fixed dimensional space that (1) preserves the cluster areas (the areas may only be subject of isomorphic transformations) and (2) keeps the minimum required gaps between clusters (the minimum gaps being fixed prior to transformation). By continuous we mean: there exists a continuous trajectory for each data point such that the conditions (1) and (2) are kept all the way.

**Definition 39** A clustering method has the property of motion-consistency, if it returns the same clustering after motion-transformation.

Note that motion-consistency is a relaxation of outer-consistency because it does not impose restrictions on increasing distances between points in different clusters but rather it requires keeping a distance between cluster gravity centers.

**Theorem 40** If random-seed k-means has a local minimum in a ball form that is such that the clusters are enclosed into equal radius balls centered at the respective cluster centers, and gaps are fixed at zero, then the motion-transform preserves this local minimum.

See proof in paper [27].

However, we are not so interested in keeping the local minimum of k-means, but rather the global one (as required by motion-consistency property definition).

In [27], we have discussed some special cases of motion-consistency. Here, however, we are interested in the general case.

**Theorem 41** Let the partition \( \Gamma_0 \) be the optimal clustering for k-means. Let \( R \) the radius such that, for each cluster \( C \in \Gamma_0 \), the ball centered at gravity center of \( C \) and with radius \( R \) contains all data points of \( C \). Let us perform a transformation according to Theorem 21 yielding a clustering \( \Gamma \) so that the distances between the cluster centres is equal to or greater than the distance between them under \( \Gamma_0 \) plus \( 4R \). (This transformation is by the way motion-consistent.) Then the clustering is motion consistent under any motion transformation keeping cluster center distances implied by \( \Gamma \).

See proof on page 37.

So we can state that

**Theorem 42** The axioms of k-richness, scale-invariance, centric-consistency and motion-consistency are not contradictory.

The proof is straight forward: k-means algorithm has all these properties.

Note that the theorem 41 can be extended to a broader range of clustering functions. Let \( \mathcal{F} \) be the class of distance-based clustering functions embedded into Euclidean space such that the clustering quality does not decrease with the decrease of distances between cluster elements and changes of distances between elements from distinct clusters do not impact the quality function.³

**Theorem 43** If we replace in Theorem 41 the phrase “k-means” with “a function from the class \( \mathcal{F} \)”, then the Theorem is still valid.

See proof on page 38.

But what about the richness, scale-invariance, centric-consistency and motion-consistency? As already mentioned, we will not consider the full richness but rather a narrower set of possible clusterings that still does not have the property of anti-chain.

Consider the Auto-means algorithm introduced in [26]. It differs from the just introduced bisectional-auto -k-means in that two parameters: \( p, g \) are introduced such that at each step both created clusters \( A, B \) have cardinalities such that \( p \leq |A|/|B| \leq 1/p \) for some parameter \( 0 < p < 1 \), and \( |A| \geq m + 1, |B| \geq m + 1 \), where \( m \) is the dimensionality of the data set, and both \( A \) and \( B \) can be enclosed in a ball centered at each gravity center with a common radius \( R \) such that the distance between gravity centers is not smaller than \( (2 + g)R \), where \( g > 0 \) is a relative gap parameter.

Let us introduce the concept of radius-R-bound motion transform of two clusters \( A, B \) as motion transform preserving the gravity center of the set \( A \cup B \) and keeping all data points within the radius \( R \) around the gravity center of \( A \cup B \). A hierarchical motion transform for auto-means should be understood as follows: If clusters \( A, B \) are at the top of the hierarchy, then the radius-R-bound motion transform is performed with \( R = \infty \). If clusters \( C, D \) are subclusters of a cluster \( A \) from the next higher hierarchy level where \( A \) was included into a ball of radius \( R' \), then \( R' \)-bound motion transform is applied to the clusters \( C, D \). If the hierarchical motion transform does not change clustering, then we speak about hierarchical motion consistency.

**Theorem 44** The axioms of k \( \downarrow \)-nearly-richness, scale-invariance, centric-consistency and hierarchical motion-consistency are not contradictory.

The proof is straight forward: Auto-means algorithm has all these properties.

³This property holds clearly for k-means, if quality is measured by inverted \( Q \) function, but also we can measure cluster quality of k-single-link with the inverted longest link in any cluster and then the property holds.
Towards continuous consistency axiom

The disadvantage of using the $k$-means clustering is that it produces convex (in particular ball-shaped) clusters. There are, however, efforts to overcome this limitation via constructing clusters with multiple centers by modifying the $k$-means-cost function \[35\], by a combined $k$-means clustering and agglomerative clustering based on data projections onto lines connecting cluster centers \[31\], or by combining $k$-means-clustering with single-link algorithm \[12\]. Let us follow the spirit of the latter paper and introduce the concave-$k$-means algorithm in the following way: First perform the $k$-means clustering. Then construct a minimum weight spanning tree, where the edge weight is the distance between cluster centres. Then in such a way that the weight of an edge connecting two clusters is the quotient of the actual distance between cluster centres. Then remove $l = 1$ longest edges from the tree, producing $l$ clusters. The algorithm shall be called $k$-means-$l$-MST algorithm. As visible in the examples in Section 11 if we perform centric transformation for such a clustering on constituent $k$-means clusters, the clustering is preserved.

Theorem 45 If the areas of clusters could be moved in such a way that the distances between the closest points of these clusters would not be lower than some quantity $m_{\text{d}}$, then the data can be clustered using $k$-means-$l$-MST algorithm in for $k$ bigger than some $k_{\text{d}}$ in such a way that following some centric $\Gamma$-transform on $k$-means clusters a motion-$\Gamma$-transformation can be applied preserving the clustering (motion consistency property).

See the proof on page 38.

In this way

- We have shown that the concept of consistency introduced by Kleinberg is not suitable for clustering algorithms operating in continuous space as a clustering preserving $\Gamma$-transform reduces generally to identity transform.
- Therefore we proposed, for use with $k$-means, property of centric consistency as a replacement for inner-consistency and the property of motion consistency as a replacement of outer-consistency, that are free from the shortcomings of Kleinberg’s $\Gamma$-transform.
- The newly proposed transforms can be used as a method for generating new labeled data for testing of $k$-means like clustering algorithms.
- In case of continuous axiomatisation, one can overcome the Kleinberg’s impossibility result on clustering by slightly strengthening his richness axiom ($k$ ↓-nearly-richness which is much weaker than $k$- richness because it allows for non-anti-chain partition sets) and by relaxing axioms of inner-consistency to centric-consistency and outer-consistency to hierarchical-motion-consistency and keeping the scale-invariance.

5 Problems with inner-consistency property - proofs

Proof Theorem 16. Let $E_1$ and $E_2$ be two embeddings such that the second is an inner-$\Gamma$-transform of the first one. In $\mathbb{R}^m$, the position of a point is uniquely defined by distances from $m + 1$ distinct non-cohyperplanar points with fixed positions. Assume that the inner-$\Gamma$-transform moves closer points in cluster $C_0$, when switching from embedding $E_1$ to $E_2$. So pick $m + 1$ point embeddings $p_1, \ldots, p_{m+1}$ from any $m + 1$ other different clusters $C_1, \ldots, C_{m+1}$. The distances between these $m + 1$ points are fixed. So let their positions, without any decrease in generality, be the same under both $E_1, E_2$. Now the distances of any point $p_Z$ in the cluster $C_0$ to any of these selected points cannot be changed under the transform. Hence the positions of points of (the embedding of) the first cluster are fixed, no non-trivial inner-$\Gamma$-transformation applies.

Proof of Theorem 17 The $m + 1$ data points from different clusters have to be rigid under inner-$\Gamma$-transformation. Any point from outside of this set is uniquely determined in space given the distances to these points. So for any two points from clusters not belonging to the selected $m + 1$ clusters no distance change is possible. So assume that these two clusters, containing at least two points each, are among those $m + 1$ selected. Then a point different from the selected points shall become closer to the selected point after inner-$\Gamma$-transformation, then it has to be on the other side of a hyperplane formed by $m$ points than the $m + 1$st point of the same cluster before transformation and on the same side after the transformation. But such an effect is possible for one hyperplane only and not for two because the distances between the other points will be violated.

6 Problems with continuous outer consistency property - proofs

But what when we want to move several clusters?

Proof of Theorem 18. 2D case proof

Consider two clusters, calling them a green one and a blue one. By definition, one of them must have points outside of the convex hull of a part of the other. For an illustrative example see Fig. 1.
always possible). For this $\omega$ define the point $M$ such that $M = \omega X + (1 - \omega) Y$. Obviously it will belong to the interior of the line segment $XY$. The elementary geometry tells us that $\cos(\angle YXB) = \frac{|YX|^2 + |XB|^2 - |YB|^2}{2|YX||XB|}$, and at the same time $|MB|^2 = |XM|^2 + |BX|^2 - 2\cos(\angle YXB)|XM||XB|$ because $\angle YXB = \angle MXB$. Hence

$$|MB|^2 = |XM|^2 + |BX|^2 - 2|XM||XB|\frac{|YX|^2 + |XB|^2 - |YB|^2}{2|YX||XB|} = \omega^2|YX|^2 + |XB|^2 - 2\omega^2|XY||XB|\frac{|YX|^2 + |XB|^2 - |YB|^2}{2|YX||XB|} = (1 - \omega)|XB|^2 + \omega|YB|^2 - (1 - \omega)\omega|XY|^2$$

This result makes it clear that under Kleinberg’s $\Gamma$-transformation $MB$ will increase or stay the same, as $|XB|$ and $|YB|$ increase or remain the same while $|XY|$ decreases or is unchanged. By definition of Kleinberg’s consistency: $|BC| \geq |B'C'|$, $|XY| \geq |X'Y'|$, $|BX| \leq |B'X'|$, $|CX| \leq |C'X'|$, $|BY| \leq |B'Y'|$, $|CY| \leq |C'Y'|$.

Now recall that $|BC| \leq |BM| + |MC|$, $|B'C'| = |B'M'| + |MC'|$. We have already shown that if $|XY| > |X'Y'|$ or $|BX| < |B'X'|$ or $|BY| < |B'Y'|$, then $|BM| < |B'M'|$, and if $|XY| > |X'Y'|$ or $|CY| < |C'Y'|$, then $|MC| < |M'C'|$ and hence $|BC| > |B'C'|$. This would be, however, a contradiction which implies that $|BC| = |B'C'|$, $|XY| = |X'Y'|$, $|BX| = |B'X'|$, $|CX| = |C'X'|$, $|BY| = |B'Y'|$, $|CY| = |C'Y'|$.

Now for any green datapoint $A$ such that there exists a blue point $Z$ (not necessarily different from $Y$) and the line segment $XZ$ intersects in the interior with $BA$, we can repeat the reasoning and state that the distances to $B$, $C$, $X$, $Y$ will remain unchanged. Obviously, for all points $A$ from outside of the blue convex hull it would be the case (a blue line segment must separate $B$ from any green point outside of the convex hull). Any point inside it will either lie on the other side of the straight line $XY$ than $B$ or $C$. If the respective segments intersect, we are done. If not, the green line segment passes through blue area either “behind” $X$ or $Y$. So there exists still another blue line segment intersecting with it. So the green data set is rigid. By the same argument the blue data set is also rigid - and the distances between them are also rigid, so that all data points are rigid under Kleinberg’s $\Gamma$-transformation.

This may be easily generalized to $n$-dimensional space. $\square$

**Proof** of Theorem 20.

It is necessary that the relative motion direction is orthogonal to a face separating two clusters (see Section 7). If no such face exists, then there exists such a face orthogonal to the relative speed vector that members of both clusters are on each side of this face. Hence a motion causes

---

**Fig. 1** Impossible continuous $\Gamma$-transform (top figure) and impossible continuous outer-$\Gamma$- transform and impossible inner-$\Gamma$- transform (bottom figure)
decrease of distance between some elements of different clusters.

Consider now a loop \( i_1, i_2, \ldots, i_n \) of cluster indices that is an index sequence such that \( i_1 = i_n \). We want that continuous outer-\( \Gamma \)-transform is applicable to them. So we need to determine points \( S_{i_1}, S_{i_2}, \ldots, S_{i_3} \) following the above-mentioned principle. Then, when dealing with clusters \( C_{i_1}, C_{i_2}, \ldots, C_{i_3} \), the points \( S_{i_1}, S_{i_2}, \ldots, S_{i_3} \) are predefined, and the point \( S_{i_4} \) will be automatically defined, if it exists. If it turns out that \( S_{i_1} \neq S_{i_2} \), then we have a problem because the vector of relative speed of \( C_{i_1} \) with respect to \( C_{i_2} \) is determined to be \( v_{i_1} = v_{i_1,i_2} + v_{i_2,i_3} + \ldots + v_{i_n,i_1} \) would not be zero which is a contradiction - a cluster has zero relative speed to itself.

The proof follows directly from the indicated problem of non-zero relative speed of the first cluster with respect to itself.

For an example in 2D look at the Fig. 1 to the right for an explanation, why each cluster has to have its own speed. Assume that we want to move only one cluster, that is \( FACH \) with speed orthogonal to the edge \( AC \). Imagine a point \( X \) in the cluster \( FABG \) close to \( F \) and a point \( Y \) in cluster \( FACH \) close to \( A \). When moving the cluster \( FACH \) away from \( ABC \), all the points of \( FACH \) will increase their distances to all points of \( ABC \), but the points \( X \) and \( Y \) will get closer, because the relative speed of \( FACH \) with respect to \( FABG \) is not orthogonal to the edge \( FA \). So, in a plane, the differences between the speed vectors of clusters sharing an edge must be orthogonal to that edge (or approximately orthogonal, depending on the size of the gap). The Fig. 1 to the right illustrates the problem. In this figure the direction angles of the lines \( AF, BG, CH \) were deliberately chosen in such a way that it is impossible. (For the sin values of the respective angles see below). The choice was as follows: Assume the speed of cluster \( ACHF \) with respect to cluster \( ABC \) is fixed to 1. Then the speed of \( ABGF \) has to be 0.26794 because \( \sin(F,A,C)=0.20048 \) and \( \sin(F,A,B)=0.74820 \). Therefore the speed of \( BCHG \) has to be 0.26794 because \( \sin(G,B,A)=0.5 \) and \( \sin(G,B,C)=0.5 \). Therefore the speed of \( ACHF \) has to be 0.15311 because \( \sin(H,C,B)=0.35921 \) and \( \sin(H,C,A)=0.62861 \). But this is a contradiction because the speed of \( ACHF \) was assumed to be 1 which differs from 0.15311.

7 Experimental illustration of outer consistency impossibility Theorem 20

In order to illustrate the need to move clusters perpendicularly to cluster border between clusters in case of continuous outer-\( \Gamma \)-transformation, compare e.g. Theorem 20, we generated randomly uniformly on a circle small clusters of about 100 data points each in 2D. In the first experiment 2 clusters and in the second 5 clusters were considered, as shown in Figs. 2 and 3 left. In the first experiment one of the clusters was moved in different directions from its original position. The more the motion deviated from the perpendicular direction, the move violations of the \( \Gamma \)-transform requirements were observed, as visible in Table 1. In the second experiment one of the clusters was moved orthogonally to one of its orders, but necessarily not orthogonal to other borders. The number of distance violations was the largest for the smallest motions as visible in Table 2. As soon as the data points of the cluster got out of the convex hall of other clusters, the number of violations dropped to zero.
8 The concept of gravitational consistency property - Proofs

The problem with \(k\)-means in Kleinberg’s counter example on consistency relies on the fact that \(k\)-means is a centric algorithm and decrease in distances between data points may cause the distance to the cluster center to be increased. It is easily shown that this is not a problem between clusters - upon consistency transformation cluster centers become more distant. The problem is within a cluster (Figs. 4 and 5). Therefore we propose to add an additional constraint (gravity center constraint) on consistency that is that it is forbidden to increase the distance not only between data points but also between gravity centers of any disjoint subsets of a cluster.

\[\mu(c) = \frac{1}{|C|} \sum_{x_i \in C} x_i.\]

Obviously, for a partition \(\mathcal{C}\), \(\sum_{C \in \mathcal{C}} CSS(C)\) is the \(k\)-means criterion function. There exists, however, a different way to express it. Let \(\mathcal{M}(C)\) be an operator replacing each data point in \(C\) with \(\mu(C)\) (forming a multiset). Let \(\mathcal{E}(C)\) be a partition of \(c\) into disjoint sets. Then \(CSS(C) = \sum_{C_i \in \mathcal{C}(C)} CSS(C_i) + CSS(\cup_{C_j \in \mathcal{E}(C)} \mathcal{M}(C_j)).\) Note that the latter is just the weighted sum of distances between centers of clusters from \(\mathcal{E}(C)\). Assume now we have a partition \(\mathcal{C}_0\) being the optimal one under \(k\)-means and a competing one (but not optimal) \(\mathcal{E}_1\). Construct a new partition (possibly into much more than \(k\) clusters) \(\mathcal{C}_{10} = \{C; C = C_i \cap C_j, C_i \in \mathcal{E}_0, C_j \in \mathcal{E}_j\}\) The cost function of both \(\mathcal{C}_0, \mathcal{E}_1\) can be expressed as a sum of \(CSS\) of all elements of \(\mathcal{E}_{10}\) plus connections between gravity centers of \(\mathcal{C}_{10}\) lying inside of a \(\mathcal{E}_0\) cluster or outside. By the virtue of the property of gravity center constraint the latter connections will decrease under consistency operation, and due to properties of the third form of \(CSS\) the distances between clusters of \(\mathcal{C}_{10}\) contained in different clusters of \(\mathcal{E}_0\) will increase. Just look at the difference \(CSS(A \cup B) - CSS(A) - CSS(B)\) - this difference, computed as sum of distances between data points will increase so the distance between gravity centers

| Table 1 | Number of data points from different clusters that the distance was reduced (“bad distances” row) during moving one cluster (hence violating \(\Gamma\)-transform condition) |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| rotation        | -90°  | -60°  | -30°  | 0°    | 30°    | 60°    | 90°    |
| bad distances   | 8064  | 3166  | 480   | 0     | 296    | 2956   | 7920   |
| % of bad distances | 23.6% | 9.3%  | 1.4%  | 0%    | 0.9%   | 8.6%   | 23.1%  |

The move was by 0.1 of the distance between the cluster centers in the direction indicated by the rotation
Table 2  Number of data points from different clusters that the distance was reduced ("bad dst" row) during moving one cluster (hence violating $r'$-transform condition)

| shift $s$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1  |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| bad dst   | 392 | 280 | 158 | 62  | 26  | 6   | 0   | 0   | 0   | 0  |
| % bad dst | 0.2%| 0.1%| 0.1%| 0%  | 0%  | 0%  | 0%  | 0%  | 0%  | 0% |

The top cluster was moved away from the central cluster on the distance of $s$ times the distance between cluster centers.

![Fig. 4](image)

**Fig. 4** Getting some data points closer to the cluster center does not ensure cluster stability for $k$-means, because the cluster center can move. Left picture - data partition before moving data closer to the cluster center. Right picture - data partition thereafter.

![Fig. 5](image)

**Fig. 5** Getting all data points closer to the cluster center without changing the position of cluster center, if we do not ensure that they move along the line connecting each with the center. Left picture - data partition before moving data closer to the cluster center. Right picture - data partition thereafter.
will increase too, by the equivalence of first two methods of computing CSS.

Proof of Theorem 24. The proof follows from elementary geometry.

When homothetic transformations were applied to various clusters, the distances between them need to be increased by the biggest distance decrease in order for the transformation to be a \( \Gamma \)-transformation in the spirit of Kleinberg.

9 k-means and the centric-consistency axiom

The first differentiating feature of the centric consistency is that no new structures are introduced in the cluster at any scale. The second important feature is that the requirement of keeping the minimum distance to elements of other clusters is dropped and only cluster centers do not get closer to one another.

Note also that the centric consistency does not suffer from the impossibility of transformation for clusters that turn out to be internal (Figs. 6 and 7).

Proof of Theorem 33

\( V(C_j) \) be the sum of squares of distances of all objects of the cluster \( C_j \) from its gravity center and \( Q \) be from (1). Hence \( Q(\Gamma) = \sum_{j=1}^{k} \frac{1}{n_j} V(C_j) \). Consider moving a data point \( x^* \) from the cluster \( C_{j0} \) to cluster \( C_{ji} \). As demonstrated by [15], \( V(C_{j0} - (x^*)) = V(C_{j0}) - \frac{n_{j0}}{n_{j0} + 1} ||x^* - \mu_{j0}||^2 \) and \( V(C_{ji} \cup \{x^*\}) = V(C_{ji}) + \frac{n_{ji}}{n_{ji} + 1} ||x^* - \mu_{ji}||^2 \) So it pays to move the point \( x^* \) from one cluster to another if \( \frac{n_{ji}}{n_{ji} + 1} ||x^* - \mu_{ji}||^2 > \frac{n_{j0}}{n_{j0} + 1} ||x^* - \mu_{j0}||^2 \). If we assume local optimality of \( \Gamma \), this obviously did not pay off.

Now transform this data set to \( X' \) in that we transform elements of cluster \( C_{ji} \) in such a way that it has now elements \( x'_j = x_i + \lambda(x_i - \mu_{j0}) \) for some \( 0 < \lambda < 1 \), see Fig. 8. Consider a partition \( \Gamma' \) of \( X' \). All clusters are the same as in \( \Gamma \) except for the transformed elements that form now a cluster \( C'_{j0} \).

The question is: does it pay off to move a data point \( x'^* \in C'_{j0} \) between the clusters? Consider the plane containing \( x^* \), \( \mu_{j0} \), and \( \mu_{ji} \). Project orthogonally the point \( x^* \) onto the line \( \mu_{j0} - \mu_{ji} \), giving a point \( p \). Either \( p \) lies between \( \mu_{j0} \) and \( \mu_{ji} \), or \( \mu_{j0} \) lies between \( p \) and \( \mu_{ji} \). Properties of k-means exclude other possibilities. Denote distances \( y = ||x^* - p|| \), \( x = ||\mu_{j0} - p|| \), \( d = ||\mu_{j0} - \mu_{ji}|| \). In the second case the condition that moving the point does not pay off means:

\[
\frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) \leq \frac{n_{j0}}{n_{j0} - 1}((d - x)^2 + y^2)
\]

If we multiply both sides with \( \lambda^2 \), we have:

\[
\lambda^2 \frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) = \frac{n_{j0}}{n_{j0} - 1}((\lambda x)^2 + (\lambda y)^2)
\]

\[
\lambda^2 \frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) \leq \frac{n_{ji}}{n_{ji} + 1}((d + \lambda x)^2 + (\lambda y)^2)
\]

which means that it does not pay off to move the point \( x'^* \) between clusters either. Consider now the first case and assume that it pays off to move \( x'^* \). So we would have

\[
\frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) \leq \frac{n_{ji}}{n_{ji} + 1}((d - x)^2 + y^2)
\]

and at the same time

\[
\lambda^2 \frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) > \frac{n_{ji}}{n_{ji} + 1}((d - \lambda x)^2 + (\lambda y)^2)
\]

Subtract now both sides:

\[
\lambda^2 \frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) > \frac{n_{ji}}{n_{ji} + 1}((d - \lambda x)^2 + (\lambda y)^2)
\]

\[
\lambda^2 \frac{n_{j0}}{n_{j0} - 1}(d^2 + y^2) < \frac{n_{ji}}{n_{ji} + 1}((d - \lambda x)^2 + (\lambda y)^2)
\]

This implies

\[
\frac{n_{j0}}{n_{j0} - 1}(d - \lambda x)^2 + (\lambda y)^2 < \frac{n_{ji}}{n_{ji} + 1}((d - \lambda x)^2 + (\lambda y)^2)
\]
Towards continuous consistency axiom

If the data have one stable optimum only like in case of well separated normally distributed $k$ real clusters, then both turn to global optima.

**Proof** of Theorem 34. Let us consider first the simple case of two clusters only (2-means). Let the optimal clustering for a given set of objects $X$ consist of two clusters: $T$ and $Z$. The subset $T$ shall have its gravity center at the origin of the coordinate system. The quality of this partition $Q((T, Z)) = n_T Var(T) + n_Z Var(Z)$, where $n_T, n_Z$ denote the cardinalities of $T, Z$ and $Var(T), Var(Z)$ their variances (averaged squared distances to gravity center).

We will prove by contradiction that by applying our $\Gamma$ transform we get partition that will be still optimal for the transformed data points. We shall assume the contrary that is we can transform the set $T$ by some $1 > \lambda > 0$ to $T'$ in such a way that optimum of 2-means clustering is not the partition $\{T', Z\}$ but another one, say $\{A' \cup D, B' \cup C\}$ where $Z = C \cup D, A'$ and $B'$ are transforms of sets $A, B$ for which in turn $A \cup B = T$. It may be easily verified that

$$Q(A \cup B, C \cup D) = n_A Var(A) + n_B Var(B) + n_C Var(C) + n_D Var(D)$$

while

$$Q((A \cup C, B \cup D)) = n_A Var(A) + n_B Var(B) + n_A Var(A) + n_B Var(B) + n_C Var(C) + n_D Var(D)$$

and

$$Q((A' \cup B', C \cup D)) = n_A \lambda^2 Var(A) + n_B \lambda^2 Var(B) + n_C Var(C) + n_D Var(D)$$

while

$$Q((A' \cup B', C' \cup D')) = n_A \lambda^2 Var(A) + n_B \lambda^2 Var(B) + n_C Var(C) + n_D Var(D)$$

The following must hold:

$$Q((A' \cup B', C' \cup D')) > Q((A' \cup D, B' \cup C))$$  
(3)

and

$$Q((A \cup B, C \cup D)) < Q((A \cup D, B' \cup C))$$  
(4)

Additionally also

$$Q((A \cup B, C \cup D)) < Q((A \cup B \cup C, D))$$  
(5)

and

$$Q((A \cup B, C \cup D)) < Q((A \cup B \cup D, C))$$  
(6)

\[ k \text{-means quality function is known to exhibit local minima at which the k-means algorithm may get stuck at. This claim means that after the centric } \Gamma\text{-transformation a partition will still be a local optimum. If the quality function has a unique local optimum then of course it is a global optimum and after the transform the partition yielding this global optimum will remain the global optimum.} \]
\[
\frac{nCnD}{nC + nD} (vC - vD)^2 < \frac{(nA + nB)nC}{(nA + nB) + nC} vC^2
\]
and
\[
\frac{nCnD}{nC + nD} (vC - vD)^2 < \frac{(nA + nB)nD}{(nA + nB) + nD} vD^2
\]

Consider now an extreme contraction (\( \lambda = 0 \)) yielding sets \( A'', B'' \) out of \( A, B \). Then we have

\[
Q((A'' \cup B'', C \cup D)) - Q((A'' \cup C, B'' \cup D)) = \frac{nCnD}{nC + nD} (vC - vD)^2 - \frac{nA}{nA + nD} (vA - vD)^2 - \frac{nB}{nB + nC} (vB - vC)^2
\]

Furthermore
\[
h(1) = Q((A \cup B, C \cup D)) - Q((A \cup C, B \cup D)) < 0,
\]
\[
h(\lambda) = Q((A' \cup B', C \cup D)) - Q((A' \cup C, B' \cup D)) > 0,
\]
\[
h(0) = Q((A'' \cup B'', C \cup D)) - Q((A'' \cup C, B'' \cup D)) < 0.
\]

But no quadratic polynomial with a positive coefficient at \( x^2 \) can be negative at the ends of an interval and positive in the middle. So we have the contradiction. This proves

the thesis that the (globally) optimal 2-means clustering remains (globally) optimal after transformation.

Let us turn to the general case of \( k \)-means. Let the optimal clustering for a given set of objects \( X \) consist of \( k \) clusters: \( T \) and \( Z_1, \ldots, Z_{k-1} \). The subset \( T \) shall have its gravity center at the origin of the coordinate system. The quality of this partition \( Q((T, Z_1, \ldots, Z_{k-1})) = ntVar(T) + \sum_{j=1}^{k-1} nZ_jVar(Z_j) \), where \( nZ_j \) is the cardinality of the cluster \( Z_j \). We will prove by contradiction that by applying our \( \Gamma \) transform we get partition that will be still optimal for the transformed data points. We shall assume the contrary that is that we can transform the set \( T \) by some \( 1 > \lambda > 0 \) to \( T' \) in such a way that optimum of \( k \)-means clustering is not the partition \( \{T', Z_1, \ldots, Z_{k-1}\} \) but another one, say \( \{T'_1 \cup Z_{k-1,1}, \ldots \cup Z_{k-1,k-1}, T'_2 \cup Z_{1,2} \cup \cdots \cup Z_{k-1,2}, \ldots, T'_k \cup Z_{1,k} \cup \cdots \cup Z_{k-1,k}\} \) where \( Z_j = \bigcup_{i=1}^{k-1} Z_{i,j} \) (where \( Z_{i,j} \) are pairwise disjoint), \( T'_1, \ldots, T'_k \) are transforms of disjoint sets \( T_1, \ldots, T_k \) for which in turn \( \bigcup_{j=1}^{k-1} T'_j = T \). It may be easily verified that

\[
Q((T, Z_1, \ldots, Z_{k-1})) = \sum_{j=1}^{k} nT_jVar(T_j) + \sum_{j=1}^{k} nT_j v_{T,j}^2 + \sum_{i=1}^{k-1} nZ_i Var(Z_i)
\]

while (denoting \( Z_{*,j} = \bigcup_{i=1}^{k-1} (i - 1)Z_{i,j} \))

\[
Q((T_1 \cup Z_{*,1}, \ldots, T_k \cup Z_{*,k})) =
\]

\[
= \sum_{j=1}^{k} \left( nT_jVar(T_j) + nZ_{*,j} Var(Z_{*,j}) + \frac{nZ_{*,j}}{nT_j + nZ_{*,j}} (vT_j - vZ_{*,j})^2 \right)
\]

whereas

\[
((T', Z_1, \ldots, Z_{k-1})) = \sum_{j=1}^{k} nT_j Var(T_j) + \sum_{j=1}^{k} nT_j \lambda^2 v_{T,j}^2
\]

\[
+ \sum_{i=1}^{k-1} nZ_i Var(Z_i)
\]

while

\[
Q((T'_1 \cup Z_{*,1}, \ldots, T'_k \cup Z_{*,k})) =
\]

\[
= \sum_{j=1}^{k} \left( nT_j \lambda^2 Var(T_j) + nZ_{*,j} Var(Z_{*,j}) + \frac{nZ_{*,j}}{nT_j + nZ_{*,j}} (\lambda vT_j - vZ_{*,j})^2 \right)
\]

The following must hold:

\[
Q((T', Z_1, \ldots, Z_{k-1})) > Q((T'_1 \cup Z_{*,1}, \ldots, T'_k \cup Z_{*,k})) \tag{7}
\]

and

\[
Q((T, Z_1, \ldots, Z_{k-1})) < Q((T_1 \cup Z_{*,1}, \ldots, T_k \cup Z_{*,k})) \tag{8}
\]
Towards continuous consistency axiom

Additionally also

\[ Q((T, Z_1, \ldots, Z_{k-1})) < Q((T \cup Z_{s,1}, Z_{s,2}, \ldots, Z_{s,k})) \]  

(9)

and

\[ Q((T, Z_1, \ldots, Z_{k-1})) < Q((T \cup Z_{s,k}, Z_{s,1}, \ldots, Z_{s,k-1})) \]  

(10)

and ... and

\[ Q((T, Z_1, \ldots, Z_{k-1})) < Q((T \cup Z_{s,k'}, Z_{s,1}, \ldots, Z_{s,k-1})) \]  

(11)

These latter \( k \) inequalities imply that for \( l = 1, \ldots, k \):

\[
Q((T, Z_1, \ldots, Z_{k-1})) = n_T \text{Var}(T) + \sum_{j=1}^{k} n_{Tj} \text{Var}(T_j) + \sum_{j=1}^{k} n_{Tj} v_{Tj}^2
\]

\[
+ \sum_{i=1}^{k-1} n_{T} \text{Var}(Z_i) <
\]

\[
Q((T \cup Z_{s,l}, Z_{s,1}, \ldots, Z_{s,l-1}, Z_{s,l+1}, \ldots, Z_{s,k})) =
\]

\[
= n_T \text{Var}(T) + \sum_{j=1}^{k} n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2
\]

\[
+ \sum_{i=1}^{k-1} n_{T} \text{Var}(Z_i) <
\]

\[
\leq \sum_{j=1}^{k} n_{z_{s,j}} \text{Var}(Z_{s,j}) + \sum_{i=1}^{k-1} n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2
\]

Consider now an extreme contraction \((\lambda = 0)\) yielding sets \(T_j''\) out of \(T_j\). Then we have

\[
Q((T'', Z_1, \ldots, Z_{k-1})) - Q((T'' \cup Z_{s,1}, \ldots, T'' \cup Z_{s,k}))
\]

\[
= \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i) - \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2
\]

\[
= \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i) - \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i)
\]

\[
- \sum_{j=1}^{k} n_{Tj} \text{Var}(Z_{s,j}) + \sum_{j=1}^{k} n_{Tj} \text{Var}(Z_{s,j})
\]

\[
- \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2
\]

\[
\leq \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i) - \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2 < 0
\]

because the linear combination of numbers that are bigger than a third yields another number bigger than this. Let us define a function

\[
g(x) = \sum_{j=1}^{k} n_{Tj} x^2 v_{Tj}^2 + \sum_{i=1}^{k-1} n_{Z_{s,i}} \text{Var}(Z_i)
\]

\[
- \sum_{j=1}^{k} \left( n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}}(v_T - v_{Z_{s,j}})^2 \right)
\]

It can be easily verified that \( g(x) \) is a quadratic polynomial with a positive coefficient at \( x^2 \). Furthermore \( g(1) = Q((T, Z_1, \ldots, Z_{k-1}) - Q((T \cup Z_{s,1}, \ldots, T \cup Z_{s,k})) < 0 \), \( g(\lambda) = Q((T', Z_1, \ldots, Z_{k-1}) - Q((T' \cup Z_{s,1}, \ldots, T' \cup Z_{s,k})) > 0 \), \( g(0) = Q((T'', Z_1, \ldots, Z_{k-1}) - Q((T'' \cup Z_{s,1}, \ldots, T'' \cup Z_{s,k})) < 0 \). But no quadratic polynomial with a positive coefficient at \( x^2 \) can be negative at the ends of an interval and positive in the middle. So we have the contradiction. This proves the thesis that the (globally) optimal \( k \)-means clustering remains (globally) optimal after transformation.

\[ \square \]

So summarizing the new \( \Gamma \) transformation preserves local and global optima of \( k \)-means for a fixed \( k \). Therefore \( k \)-means algorithm is consistent under this transformation.

Note that \((P^*\text{ based})\) centric-consistency is not a specialization of Kleinberg’s consistency as the requirement of increased distance between all elements of different clusters is not required in \( P^* \) based Consistency.

10 Bisectonal-auto-X-means

Proof of Theorem 31 (Outline) Let the optimal clustering for a given set of objects \( X \) consist of two clusters: \( T \) and \( Z \). Let \( T \) consist of two disjoint subsets \( P, Y, T = P \cup Y \) and let us ask the question whether or not centric transformation of the set \( P \) will affect the optimality of clustering. Let \( T'(\lambda) = P'(\lambda) \cup Y \) with \( P'(\lambda) \) being an image of \( P \) under centric transformation. The cluster centre of \( T'(\lambda) \) will be the same as that of \( T \). We ask if \( T'(\lambda), Z \) is the globally optimal clustering of \( T'(\lambda) \cup Z \). Assume the contrary, that is that there exists a clustering into sets \( K'(\lambda) = A'(\lambda) \cup B \cup C, L'(\lambda) = D'(\lambda) \cup E \cup F \), where \( P = A \cup D \), and \( A'(\lambda) \) are the points obtained from \( A \) when \( P \) is subjected to centric transformation, and \( D'(\lambda) \) is defined analogously, hence \( P'(\lambda) = A'(\lambda) \cup D'(\lambda), Y = B \cup E, Z = C \cup F \), that, for some \( \lambda = \lambda^* \in (0,1) \) has lower clustering quality function value \( Q((K'(\lambda^*), L'(\lambda^*))) \). Define also the function \( h(\lambda) = Q((T'(\lambda), Z)) - Q((K'(\lambda), L'(\lambda))) \). Due to optimality assumption, \( h(1) \leq 0 \). Let us discuss now the centric transform with \( \lambda = 0 \). In this case all points from \( A'(0) \) and \( D'(0) \) collapse to a single point. This point can be closer to either \( \mu(K'(0)) \) or \( \mu(L'(0)) \). Assume they are closer to \( \mu(K'(0)) \). In this case \( Q((K'(\lambda), L'(\lambda))) \leq Q((K'(\lambda), L'(\lambda))) \), where \( K'(\lambda) = P'(\lambda) \cup B \cup C, L'(\lambda) = \)}
where \( c_0 \) is a constant (independent of \( \lambda \) and all the \( \mu \)s that depend on \( \lambda \), are linearly dependent on it (by definition of centric consistency). Recall that \( \mu(A'(\lambda)) - \mu(P) = 1/|A| \sum_{x \in A} \lambda(x - \mu(P)) = \lambda \mathbf{v}_A \), where \( \mathbf{v}_A \) is a vector.
Towards continuous consistency axiom

Towards continuous consistency axiom

$h(\lambda)$ is the case by our assumption of an alternative clustering. Therefore we can rewrite the $h(\lambda)$ follows. At a given stage of the algorithm, when a bisection is restored. $Q$ than 9-fold decrease of $Q$ on a line segment. If we set the stopping criterion as lower partition consists of clusters of predefined cardinalities. $\mu(A'(\lambda)) - \mu(C) = \lambda^2 v_A^T v_A + \lambda c_{ACP} + c_{CP}$ with $c_{ACP}, c_{CP}$ being constants independent of $\lambda$, and so on. Therefore we can rewrite the $h(\lambda)$ as

$$h(\lambda) = |A| \lambda^2 v_A^T v_A + |D| \lambda^2 v_D^T v_D$$

$$- \frac{1}{|A| + |B| + |C|} \left( [A][B](\lambda^2 v_A^T v_A + \lambda c_{ABP} + c_{BP}) + |A||C|(\lambda^2 v_A^T v_A + \lambda c_{ACP} + c_{CP}) \right)$$

$$- \frac{1}{|D| + |E| + |F|} \left( |D||E| \lambda^2 v_D^T v_D + \lambda c_{DEP} + c_{EP}) \right)$$

$$+ |D||F| \left( \lambda^2 v_D^T v_D + \lambda c_{DEP} + c_{EP}) \right) + c_h$$

So the coefficient at $\lambda^2$ amounts to:

$$\left( [A] - \frac{|A||(B) + |C|}{|A| + |B| + |C|} \right) v_A^T v_A + \left( [D] - \frac{|D||E| + |F|}{|D| + |E| + |F|} \right) v_D^T v_D$$

which is bigger than 0 if only $|A| > 0$ and $|D| > 0$, which is true by our assumption of an alternative clustering. Therefore, for $\lambda$ large enough, $h(\lambda) > 0$.

As $h(\lambda)$ is a quadratic function in $\lambda$, and $h(0) \leq 0$ and $h(1) \leq 0$, then also $h(\lambda) \leq 0$ for any value of $\lambda$ between 0 and 1. This completes the proof.

Theorem 32. The scale-invariance is implied by the properties of the $k$-means algorithm that serves as the subroutine and the fact that the stopping criterion is the relative decrease of $Q$, so that the stopping criterion is also not affected.

The $2 \uparrow$-nearly-richness can be achieved as follows. Each partition consists of clusters of predefined cardinalities. Within each partition the data points uniformly on a line segment. If we set the stopping criterion as lower than 9-fold decrease of $Q$, then by proper manipulation of distances between groups of clusters when combining them to the cluster hierarchy will ensure that the targeted partition is restored.

The centric consistency is implied by Theorem 31 as follows. At a given stage of the algorithm, when a bisection is to be performed, Theorem 31 ensures that the optimal bisection is the one that was used in the original partitioning process. The question is only about the stopping criteria: whether or not centric consistency would imply continuing the partitioning. Consider the notation from the previous proof. The $Q$ improvement in the original clustering would amount to:

$$\frac{1}{\sum_{x \in X} \|x - \mu(X)\|^2} \left( \left( \sum_{x \in P} \|x - \mu(P)\|^2 \right) + \left( \sum_{x \in Y} \|x - \mu(Y)\|^2 \right) \right)$$

$$+ \|\mu(P) - \mu(Y)\|^2 \cdot \frac{1}{1/|P| + 1/|Y|} + \left( \sum_{x \in Z} \|x - \mu(Z)\|^2 \right)$$

and afterwards

$$\frac{1}{\sum_{x \in X} \|x - \mu(X)\|^2 - \left( \sum_{x \in P} \|x - \mu(P)\|^2 \right) - \left( \sum_{x \in P} \|x - \mu(P)\|^2 \right) \left( \sum_{x \in Y} \|x - \mu(Y)\|^2 \right) + \left( \sum_{x \in Z} \|x - \mu(Z)\|^2 \right)}$$

This expression is as if $(\sum_{x \in P} \|x - \mu(P)\|^2) - \left( \sum_{x \in P} \|x - \mu(P)\|^2 \right)$ (positive and smaller than the denominator) were subtracted from the numerator and the denominator of the first quotient. That is the second quotient is smaller, so if the first did not stop the partitioning, so the second would not either.

Kleinberg’s three axioms/properties of consistency, scale-invariance and richness are contradictory. We obtained here three similar axioms that are not contradictory. If we would replace Kleinberg’s richness axiom only with 2 $\uparrow$-nearly-richness, this would not help to resolve the contradiction. The real driving force behind the conflict resolution is the centric consistency. Centric consistency may appear as more rigid than Kleinberg’s consistency, but on the other hand it is broader than Kleinberg’s, as already mentioned (not all distances between elements of distinct clusters need to increase).

So at this point we know that a reasonable version of Kleinberg’s $\Gamma$-transform is no more general than centric $\Gamma$-transform. Let us briefly demonstrate that centric consistency raises claims of equivalent clustering also in those cases when Kleinberg’s consistency does not.

Imagine points $A[-11], B[-9], C[-2], D[2], E[9], F[11]$ in 1d, constituting three clusters $\{A, B\}, \{C, D\}, \{E, F\}$. Imagine centric-$\Gamma$-transform of the cluster $\{C, D\}$ into $\{C'[-1], D'[1]\}$. Note that hereby the quotient $\frac{|CB|}{|AB|}$ increases while $\frac{|CE|}{|EF|}$ decreases, and at the same time $\frac{|DB|}{|AB|}$ decreases while $\frac{|CD|}{|EF|}$ increases. No sequence of Kleinberg’s $\Gamma$-transforms and invariance transforms on these data can achieve such a transformation because all the
mentioned quotients are non-decreasing upon consistency and invariance transforms.

This implies that convergent-\(\Gamma\)-transform (the reasonable version of Kleinberg’s \(\Gamma\)-transform) is a special case of centric-\(\Gamma\)-transform.

11 Motion consistency proofs

Proof of Theorem 41. Given an optimal solution \(\Gamma_0\) to \(k\)-means problem, where each cluster is enclosed in a (hyper)ball with radius \(R\) centered at its gravity center, if we rotate the clusters around their gravity centers, then the position of any data point can change by at most \(2R\). That is, the distances between points from different clusters can decrease by at most \(4R\). So if we move away the clusters by the distance of \(4R\), then (1) \(\Gamma\) clustering gives the same value of \(k\)-means cost function as \(\Gamma_0\) (as the distances within clusters do not change) and (2) upon rotation in the new position no clustering different from \(\Gamma\) can have a lower cost function value than any competing clustering in the initial position of data points. Hence the optimum clustering is preserved upon any rotating transformation, and when moving the clusters keeping the center distances as prescribed, the optimal clustering is preserved also. \(\Box\)

Proof of Theorem 45. The assumption of the Theorem 45 implies that when applying the \(k\)-means-l-MST algorithm, the \(k\)-means cluster centers from different l-MST clusters will lie at distance of at least \(md\). Therefore we can increase \(k\) to such an extent that the distances between \(k\)-means clusters in different l-MST clusters are smaller than \(md\). This means that the clustering will be preserved upon centric \(\Gamma\)-transform applied to \(k\)-means clusters. Let \(R_M\) be the maximum radius of a hyperball enclosing any \(k\)-means cluster. Let \(d_M\) be the maximum distance between \(k\)-means clusters from different l-MST clusters. So apply centric \(\Gamma\)-transform to \(k\)-means clusters with \(\lambda < \frac{md}{2R_M + d_M}\). If we move now the clusters, the motion consistency will be preserved, as implied by Theorem 41 (Fig. 9). \(\Box\)

Proof of Theorem 43. The proof follows the general idea behind the case of \(k\)-means. If the solution \(\Gamma_n\) is optimal, then the operation performed will induce no change in the distances within the cluster and the distances between elements from different clusters will increase and this will cause an increase in the quality of the clustering function. For any alternative clustering, that is a non-optimal one \(\Gamma_m\), we can divide the distances between data points in the following four categories: \(K_1\) - intra-cluster distances both under \(\Gamma_o\) and \(\Gamma_n\), \(K_2\) - intra-cluster distances under \(\Gamma_o\) and inter-cluster distances under \(\Gamma_n\), \(K_3\) - inter-cluster distances under \(\Gamma_o\) and intra-cluster distances under \(\Gamma_n\), \(K_4\) - inter-cluster distances both under \(\Gamma_o\) and \(\Gamma_n\). Only \(K_3\) is interesting for our considerations, as neither \(K_1\) nor \(K_2\) nor \(K_4\) change the quality function. Under \(K_2\), the quality of \(\Gamma_n\) does not change, while that of \(\Gamma_o\) may decrease. This completes the proof. \(\Box\)

12 Experiments

In order to validate the clustering preservation by the consistency transformations and motion transformations proposed in this paper in the context of theorems 34 and 41, we investigated their applicability to the datasets described in Table 3 with respect to \(k\)-means clustering algorithm. Note that both transformations are not conform with Kleinberg’s consistency axiom requirements. Therefore we show also that these transformations are valid clustering preserving transformations in spite of violation of Kleinberg’s consistency constraints.

In Table 3, the column \textit{recs} contains the number of data records, \textit{cols} - the number of columns used in the experiments (numeric columns only), \textit{cls} - the number of clusters into which the data was clustered in the experiments, \textit{nst} the number of restarts used for random-seed \(k\)-means algorithm (\(R\) implementation), \textit{time.sec} is the execution time of the experiment (encompassing not only clustering, but also time for computing various quality measures).

The datasets were downloaded from (or from links from) the Web page \url{http://cs.joensuu.fi/sipu/datasets/}.

As \(k\)-means-ideal implementation is not available (at reasonable costs), we use a random-seed \(k\)-means. It may miss the global minimum of quality function (true clustering) and for this reason our transformations may fail to preserve clustering because they are valid for true clustering. So this experimental section gives also some hints about random-seed \(k\)-means weaknesses.

12.1 Experiments related to theorem 34

The Theorem 34 introduced a new cluster preserving transformation: centric consistency transformation (\(\Gamma^*\) transformation).

The experiments were performed in the following manner: For each dataset, the \(k\)-means with \(k = \text{cls}\) was performed with the number of restarts equal to \textit{nst} column, as mentioned in Table 3. The result served as a “golden standard” that is the “true” clustering via \(k\)-means.

Then the cluster was selected that was lying “most centrally” among all the clusters, so that the violation of Kleinberg’s consistency upon centric consistency transformation would have the biggest effect. Thereafter four experiments...
Fig. 9 Illustration of continuous motion consistency for the $k$-means-2-MST clustering algorithm. The presented sample consists of 11200 data points, $k=1500$. (a) original data, (b) clustering using $k$-means-2-MST, spanning trees shown (c) clustering from (b) subjected to centric $\Gamma$-transform with $\lambda = 0.7$, reclustered, (d) clustering from (b) subjected to centric $\Gamma$-transform with $\lambda = 0.052$, enabling motion with motion consistency, reclustered, (e) clustering from (d) subjected to motion (perpendicular vector with length 2, reclustered after motion, (f) clustering from (d) subjected to motion (perpendicular vector with length 4, reclustered after motion
Table 3  Data sets information

| Dataset name         | recs  | cols | cls | nst  | time.sec |
|----------------------|-------|------|-----|------|----------|
| MopsiLocations2012-Joensuu.txt | 6014  | 2    | 5   | 1000 | 53       |
| MopsiLocationsUntil2012-Finland.txt | 13467 | 2    | 5   | 1000 | 80       |
| t4.8k ConfLong Demo.txt | 8000  | 2    | 5   | 1000 | 68       |
| dim032.txt           | 1024  | 32   | 16  | 1000 | 243      |
| ConfLongDemo_JS1_164860.txt | 164860 | 3    | 5   | 1000 | 2088     |
| dim064.txt           | 1024  | 64   | 16  | 977  | 472      |
| KDDCUP04Bio.txt      | 145751| 74   | 8   | 54   | 6119     |
| kddcup99_csv.csv     | 494020| 5    | 10  | 640  | 3488     |

As visible, many cases on the dataset of centric consistency transformation on that cluster were run with \( \lambda = 0.8, 0.6, 0.4, 0.2 \) resp. Then the resulting set was re-clustered using the same \( k \)-means parameters. The number of disagreements in cluster membership is reported in Table 4 (the number of points that failed to be classified into their original cluster). As visible, only KDDCUP04Bio.txt exhibits problems, due to the fact that the number of restarts was kept low in order to handle the large number of records.

Note that there were few violations of cluster membership (in 3 out of 32 clusters) in case of the dim032.txt dataset. This fact exposes the weakness of random-seed \( k \)-means that this algorithm poorly recovers from bad initialization in high dimensional space. Note that with diminishing \( \lambda \) the distances between cluster borders increase but this does not help random-seed \( k \)-means.

In order to demonstrate that the centric consistency allows for cluster preserving transformation in spite of Kleinberg consistency condition violations, the following statistics was performed: 100 data points were randomly picked from the “central” cluster (if there were fewer of them, all were taken) and 100 data points from other clusters were randomly picked. Then the distances prior and after the centric consistency transformations were computed (for each aforementioned \( \lambda \)). The percentage of cases, when the Kleinberg consistency condition was violated, was computed (for each \( \lambda \)). The results are presented in Table 5.

As visible, as many as 70% of distances may violate Kleinberg’s condition. The seriousness of the problem depends both on the dataset and the \( \lambda \) considered. This means that Kleinberg’s consistency transformation (\( I^* \) transformation) is much more rigid than centric consistency transformation (\( I^* \) transformation).

The conclusion is that the centric consistency transformation provides with derived labelled datasets for investigation of \( k \)-means like algorithms even in cases when Kleinberg axiomatic system is violated.

12.2 Experiments related to theorem 41

Theorem 41 is related to motion consistency cluster preserving transformation.

The experiments were performed in the following manner: For each dataset, the \( k \)-means with \( k = \text{cls} \) was performed with the number of restarts equal to \( nst \), as mentioned in Table 3. As previously, the result served as a “golden standard” that is the “true” clustering via \( k \)-means.

Then the motion experiment was initiated. First a centric consistency transformation was performed on each cluster so that the radii of all balls enclosing clusters are equal (the the radius of the smallest ball). Then centric consistency transformation was applied to each cluster so that the conditions of the Theorem 41 hold. After these two transformations, the correctness of clustering

Table 4  Clustering errors upon centric consistency transformation - see text for details

| Dataset name         | \( \lambda = 0.8 \) | \( \lambda = 0.6 \) | \( \lambda = 0.4 \) | \( \lambda = 0.2 \) |
|----------------------|--------------------|--------------------|--------------------|--------------------|
| MopsiLocations2012-Joensuu.txt | 0                  | 0                  | 0                  | 0                  |
| MopsiLocationsUntil2012-Finland.txt | 0                  | 0                  | 0                  | 0                  |
| t4.8k ConfLong Demo.txt | 0                  | 0                  | 0                  | 0                  |
| dim032.txt           | 2                  | 2                  | 3                  | 1                  |
| ConfLongDemo_JS1_164860.txt | 0                  | 0                  | 0                  | 0                  |
| dim064.txt           | 0                  | 0                  | 1                  | 0                  |
| KDDCUP04Bio.txt      | 610                | 610                | 0                  | 610                |
| kddcup99_csv.csv     | 0                  | 0                  | 0                  | 0                  |
and the percentage of Kleinberg consistency violations was computed following the guidelines of the previous subsection. The results are reported in columns \( \text{cen.cl} \) and \( \text{cen.ds} \) of Table 6 resp. The column \( \text{cen.cl} \) says for how many clusters clustering errors have occurred after these two transformations. The column \( \text{cen.ds} \) presents the percentage of distances violating Kleinberg’s consistency axiom: that is the number of distances that are consistent with centric consistency but are not consistent with Kleinberg’s consistency, divided by the number of tested distances.

   Afterwards twenty times motion by a small step (randomly picking the length of the step) in random direction was performed. It was checked if conditions of Theorem 41 hold. If so, the correctness of clustering and the percentage of Kleinberg consistency violations was computed following the guidelines of the previous subsection. The maximal values of both are reported in columns \( \text{mot.cl} \) and \( \text{mot.ds} \) of Table 6 resp. The column \( \text{mot.cl} \) says for how many clusters clustering errors have occurred after the motion transformations. The column \( \text{mot.ds} \) presents the percentage of distances violating Kleinberg’s consistency axiom.

   The columns columns \( \text{st.cl} \) and \( \text{st.ds} \) of Table 6 report the same but not with respect to the final position before the motion but rather with respect to the position in the previous motion step.

We see that clustering errors occur only for \( \text{dim032.txt}, \text{dim064.txt}, \text{KDDCUP04Bio.txt} \) datasets. These are the data sets with a large number of clusters. Again, the dataset \( \text{KDDCUP04Bio.txt} \) proves to be most problematic, due to the low number of restarts with \( k \)-means clustering, even 16,000 (about 11%) disagreements in cluster membership were observed. Note however that this dataset is huge compared to the other ones 145,751 datapoints in over 70 dimensions. Such data are problematic for the random-seed \( k \)-means itself (difficulties in proper seeding and ten recovery from erroneous seeding). The problems with with the artificial benchmark datasets \( \text{dim064.txt} \) and \( \text{dim032.txt} \) are minor. Otherwise the even non-optimal random-seed \( k \)-means algorithm approximated the true optimum sufficiently well to show that the motion consistency transformation really preserves the clustering.

   Note that the Theorem 41 makes claims about \( k \)-means-ideal. This example illustrates therefore how much and when the random-seed \( k \)-means deviates from the ideal one. Studies exploiting this theorem along this guideline may be useful when elaborating better versions of \( k \)-means and for pinpointing at weaknesses of them.

   At the same time it is visible (columns \( \text{cen.ds}, \text{mot.ds}, \text{st.ds} \)) that the motion consistency transformation provides with derived labelled datasets for investigation of \( k \)-means like algorithms even in cases when Kleinberg axiomatic consistency violations occur.

### Table 5  Kleinberg consistency violation percentage

| \( \lambda \) | 0.8 | 0.6 | 0.4 | 0.2 |
|---------------|-----|-----|-----|-----|
| MopsiLocations2012-Joensuu.txt | 1.32 | 3.16 | 54.6 | 65.85 |
| MopsiLocationsUntil2012-Finland.txt | 18.75 | 1.965 | 0 | 0 |
| t4.8k ConfLong Demo.txt | 17.59 | 57.18 | 56.67 | 53.66 |
| dim032.txt | 0 | 0 | 0.49 | 0 |
| ConfLongDemo_JSI_164860.txt | 72.55 | 68.98 | 19.22 | 8.73 |
| dim064.txt | 6 | 4 | 1 | 6 |
| KDDCUP04Bio.txt | 72.55 | 68.98 | 19.22 | 8.73 |
| kddcup99.csv.csv | 3.62 | 3.02 | 3.12 | 60.74 |

### Table 6  Kleinberg consistency distance errors percentage upon motion

|       | cen.ds | cen.cl | mot.ds | mot.cl | st.ds | st.cl | time  |
|-------|--------|--------|--------|--------|-------|-------|-------|
| Mopsi Locations 2012-Joensuu.txt | 65.92 | 0 | 74.23 | 0 | 98.08 | 0 | 384.0906 |
| Mopsi Locations Until 2012-Finland.txt | 60.59 | 0 | 60.86 | 0 | 97 | 0 | 392 |
| t4.8k ConfLong Demo.txt | 61.07 | 0 | 64.46 | 0 | 100 | 0 | 408 |
| dim032.txt | 64.12 | 4 | 68.15 | 5 | 92 | 6 | 2215.708 |
| Conf LongDemo_JSI_164860.txt | 57.22 | 0 | 72.05 | 0 | 100 | 0 | 498.239 |
| dim064.txt | 58.89 | 2 | 64.46 | 5 | 100 | 8 | 4473.421 |
| KDDCUP04Bio.txt | 68.41 | 155 | 80.93 | 5393 | 98 | 16679 | 6737.905 |
| kddcup99.csv.csv | 52.87 | 0 | 65.78 | 0 | 95 | 0 | 1084.221 |

The columns columns \( \text{cen.ds}, \text{mot.ds}, \text{st.ds} \) that the motion consistency transformation provides with derived labelled datasets for investigation of \( k \)-means like algorithms even in cases when Kleinberg axiomatic consistency violations occur.
system is seriously violated. It becomes obvious that Kleinberg’s consistency is quite rigid and not well suited for deriving new test samples for clustering. Motion consistency transforms and centric consistency transforms discussed in this paper are better framework for this purpose.

13 Discussion

The attempts to axiomatize the domain of clustering have a rich history. Van Laarhoven and Marchiori [28] and Ben-David and Ackerman [7] the clustering axiomatic frameworks address either: (1) required properties of clustering functions, or (2) required properties of the values of a clustering quality function, or (3) required properties of the relation between qualities of different partitions. The work of Kleinberg [21], which we discussed extensively, fits into the first category.

Van Laarhoven and Marchiori [28] developed an axiomatic system for graphs, fitting the same category. Though it is possible to view the datapoints in $\mathbb{R}^m$ as nodes of a graph connected by edges with appropriate weights, their axioms essentially reflect the axioms of Kleinberg with all the problems for $k$-means like algorithms.

To overcome the problems with the consistency property, Ackerman et al. [3] propose splitting of it into the concept of outer-consistency and of inner-consistency. However, the $k$-means algorithm is not inner-consistent, see [3, Section 3.1]. We show that the problem with inner-consistency is much deeper, as it may be not applicable at all (Theorem 16 and Theorem 17).

The $k$-means-ideal algorithm is said to be in this sense outer-consistent [3, Section 5.2]. But we have demonstrated in this paper that outer-consistency constitutes a problem in the Euclidean space, if continuous transformations are involved, see Theorem 20. They claim also that in the Euclidean space, if continuous transformations are permitted, outer-consistency will yield exactly $k$ clusters has the property of locality. The property of locality, required, see Theorem 20. They claim also that in the Euclidean space, if continuous transformations are permitted, outer-consistency constitutes a problem in the Euclidean space, if continuous transformations are involved, see Theorem 20. They claim also that in the Euclidean space, if continuous transformations are permitted, outer-consistency will yield exactly $k$ clusters has the property of locality.

Ackerman and Ben-David propose among others the following axioms:

Property 47 (CQM-scale-invariance) A quality measure $J$ satisfies scale invariance if for every clustering $\Gamma$ of $(X, d)$ and every positive $\beta$, $J(X, \Gamma, d) = J(X, \Gamma, \beta d)$.

Property 48 (CQM-consistency) A quality measure $J$ satisfies consistency if for every clustering $\mathcal{C}$ over $(X, d)$, whenever $d'$ is consistency-transformation of $d$, then $J(X, \Gamma, d') \geq J(X, \Gamma, d)$.

If we define a clustering function $f$ to maximize the quality function, then the clustering function is (scale)-invariant, but the clustering function does not need to be consistent in Kleinberg’s sense. E.g. $k$-means is CQM-consistent, not being consistent in Kleinberg’s sense.

The basic problem with this axiomatic set is that the CQM-consistency does not tell anything about the (optimal) partition being the result of the consistency-transform, while Kleinberg’s axioms make a definitive statement: the partition before and after consistency-transform has to be the same. So $k$-means could be in particular rendered to become CQM-consistent, CQM-scale-invariant, and CQM-rich, if one applies a bi-sectional version (bisectional-auto-means).

A number of further characterizations of clustering functions has been proposed e.g. Ackerman’s et al. [2] for linkage algorithms, Carlsson’s [9] for multiscale clustering. None of the transforms proposed there seems however to fit the purpose of test set derivation for $k$-means testing by a continuous derivation.

An interesting axiomatic system for hierarchical clustering was developed by Thomann et al [42]. It is defined not for datapoints but for probability distributions over some
Towards continuous consistency axiom

support(s). It includes a scale invariance property, but provides nothing that can be considered as corresponding to Kleinberg’s consistency. This clearly limits the usability for test set generation for (hierarchical) clustering algorithms. Nonetheless this approach is worth noting and worth pursuing for non-hierarchical clustering functions like $k$-means for two reasons: (1) $k$-means not only splits the data themselves but also the Euclidean space as such, (2) $k$-means is probabilistic in nature and rather than exact re-clustering of data one should pay attention to recovery of appropriately defined probability distributions over the space (3) the concepts of centric consistency and that of motion consistency are applicable to probability distributions over finite support. So it may constitute a research path worthy further exploration.

Puzicza et al. [37] propose an axiomatic system in which the data can be transformed by shifting the entire data set (and not the individual clusters). In our approach, individual clusters may be subject to shifts.

Strazzeri et al. [41] introduced the axiom of monotonic consistency to replace Kleinberg’s consistency axiom. He requires that, for an expansion function (monotonically increasing) $v : [0; \infty) \rightarrow [0; \infty)$, each intercluster distance $d$ is changed to $d' = v(d)$ and each intracluster distance $d$ is changed to $d' = v^{-1}(d)$. This transform, though it is non-linear on the one hand, is more rigid than our centric consistency and motion consistency transforms, as it has to apply globally, and not locally to a cluster, and on the other hand, it is designed and applicable for graphs, but not for $\mathbb{R}^n$ except for special cases.

Addad [13] proposed the following modification of the Kleinberg’s consistency axiom. Let $OPT(k)$ be the optimal value of a cost function when clustering into $k$ parts. Let $k^*$ the $k$ maximizing quotient $OPT(k)/OPT(k-1)$ Then their refined consistency requires that the Kleinberg’s consistency transform yields the same clustering only if $k^*$ prior and after the transform agree. They prove that single link and $k$-means fulfill the refined consistency axiom (the latter under some “balance” requirements). While the refined consistency transform (identical to Kleinberg’s consistency transform) is much more flexible in generating derived test sets, it is in practice hard to test whether or not a given transform produces identically clusterable datasets.

Ackerman et al. [4] consider a different way of deriving new samples from old ones, re-weighting of datapoints. It turns out that some algorithms are sensitive to re-weighting, other are robust. Regrettably, $k$-means is sensitive so that the method cannot be applied to derive new clustering test datasets from existent ones.

A discussion and critics of various approaches to axiomatic systems for clustering can be found in [17].

This overview of axiomatic literature demonstrates that cluster preserving transformations proposed in this paper seem to be best-suited for $k$-means like algorithms as other axiomatic proposals are not applicable to $\mathbb{R}^n$, or are not suitable for $k$-means, are too rigid, or do not contain clustering preserving transformations at all.

14 Conclusions

We have formulated in this paper an axiomatic framework for cluster analysis in the fixed dimensional Euclidean space, suitable for $k$-means like algorithms and proved that it is free from contradictions. Kleinberg’s axiomatic system [21] failed in this respect as he created a contradictory system and the popular $k$-means algorithm did not fulfil two of his three axioms. Though Ben David and Ackerman [7] proposed a contradiction-free axiomatic system, but at the expense of losing the ability to provide a cluster-preserving transformation. Nor is it granted by Campagner and Ciucci [8]. The advantage of our axiomatic system is that a cluster-preserving transformation is given with the centric consistency axiom and motion consistency axiom. They may be used for deriving new labelled sets for testing clustering procedures. Furthermore, our axiomatic framework is suitable for continuous data transformations which means that the reliability of clustering algorithms for small perturbations of the data can be tested when using proposed transformations. Kleinberg’s axiomatic system and those derived from in [7, 8, 21], [3] miss clearly this target because in fixed dimensional spaces the corresponding continuous data transformations are identity transformations.

Even in discontinuous case, our centric consistency and our motion consistency transformations are less rigid than the Kleinberg’s consistency transform and its derivatives. It is less rigid also than the method proposed in [37]. We do not have the limitations of monotonic consistency of [41] because our transforms are applicable locally. We do not impose unrealistic optimality requirements like those of [13]. Therefore our approach is much more useful in practice. Note also that we have already proposed a generalization of the centric consistency transform in [24].

Our approach is applicable in an analogous way to single-link-type algorithms and for clustering functions with some special property of the cluster quality function as well as special hierarchical clustering functions.

Both Kleinberg’s and our transformations assume that we have to do with true clusterings when transformations are applied. For our experimental purposes, however, true clusterings were not available but rather they were derived from a practical implementation (random-seed) $k$-means algorithm that may fail to reach the optimum of the target function. Hence a further interesting research topic emerges of creating transformations deriving true clustering
from imperfect one (e.g. by discovering sufficient gaps between clusters) and then applying cluster preserving transformations to get new test data sets.

References

1. Ackerman M (2012) Towards theoretical foundations of clustering, university of Waterloo. PhD Thesis
2. Ackerman M, Ben-david S, Loker D (2010) Characterization of linkage-based clustering. In: COLT. pp 270–291
3. Ackerman M, Ben-David S, Loker D (2010) Towards property-based classification of clustering paradigms. In: Adv neural information proc Sys, 23, pp 10–18, Curran Associates, Inc.
4. Ackerman M, Ben-David S, Brânzei S, Loker D (2011) Weighted clustering: towards solving the user’s dilemma. Pattern Recogn 120:108152. https://doi.org/10.1016/j.patcog.2021.108152
5. Awasthi P, Blum A, Sheffet O (2012) Center-based clustering under perturbation stability. Inf Process Lett 112(1-2):49–54
6. Balcan M, Liang Y (2016) Clustering under perturbation resilience. SIAM J Comput 45(1):102–155
7. Ben-David S, Ackerman M (2009) Measures of clustering quality: a working set of axioms for clustering. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in neural information processing systems 21. Curran Associates, Inc., pp 121–128
8. Campagner A, Ciucci D (2020) A formal learning theory for three-way clustering. In: Davis J, Tabia K (eds) Scalable uncertainty processing systems 21. Curran Associates, Inc., pp 128–140, https://doi.org/10.1007/978-3-030-58449-8_9
9. Carlsson G, Mémoli F (2008) Persistent clustering and a theorem of j. kleinberg. arXiv:0808.2241
10. Carlsson G, Mémoli F (2010) Characterization, stability and convergence of hierarchical clustering methods. J Mach Learn Res 11:1425–1470
11. Chang JC, Amershi S, Kamar E (2017) Revolt: collaborative crowdsourcing for labeling machine learning datasets. In: Mark G, Fussell SR, Lampe C, Schraefel M, Hourcade JP, Appert C, Wigdor D (eds) Proceedings of the 2017 CHI conference on human factors in computing systems, Denver, CO, USA, 06-11 May 2017, pp 2334–2346. ACM, https://doi.org/10.1145/3025453.3026044
12. Lin C-R, chen M-S (2005) Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging. IEEE Trans Knowl Data Eng 17(2):145–159
13. Cohen-Addad V, Kanade V, Mallmann-Trenn F (2018) Clustering redemption beyond the impossibility of kleinberg’s axioms. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in neural information processing systems. Curran Associates, Inc., vol 31
14. Cui X, Yao J, Yao Y (2020) Modeling use-oriented attribute importance with the three-way decision theory. Rough Sets 12179:122–136. https://doi.org/10.1007/978-3-030-52705-1_9
15. Duda R, Hart P, Stork G (2000) Pattern Classification. Wiley, New York 2nd edn
16. Gower JC (1990) Clustering axioms. Classification society of North America newsletter. pp 2–3
17. Hennig C (2015) What are the true clusters? Pattern Recogn Lett 64(15):53–62
18. Hopcroft J, Kannan R (2012) Computer science theory for the information age. chapter 8.13.2. a satisfiable set of axioms, p 272ff
19. Iglesias F, Zseby T, Ferreira D (2019) Mckegen: multidimensional dataset generator for clustering. J Classif 36:599–618. https://doi.org/10.1007/s00357-019-9312-3
20. Ke Z, Wang D, Yan Q, Ren J, Lau RW (2019) Dual student: breaking the limits of the teacher in semi-supervised learning. In: Proc. IEEE CVF international conference on computer vision. pp 6728–6736
21. Kleinberg J (2002) An impossibility theorem for clustering. In: Proc. NIPS 2002. pp 446–453. http://books.nips.cc/papers/files/nips15/LT17.pdf
22. Klopotek M (2017) On the existence of kernel function for kernel-trick of k-means. In: Kryszkiewicz M, Appice A, ślézak D, Rybiński H, Skowron A, Raś Z (eds) Foundations of intelligent systems. ISMIS, Lecture notes in computer science, Springer, Cham. vol 10352
23. Klopotek MA (2020) On the consistency of k-means++ algorithm. Fundam Inform 172(4):361–377
24. Klopotek MA (2022) A clustering preserving transformation for k-means algorithm output
25. Klopotek R, Klopotek M, Wierzchosi S (2020) A feasible k-means kernel trick under non-euclidean feature space. Int J Appl Math Comput Sci 30(4):703–715. https://doi.org/10.34768/amcs-2020-0052
26. Klopotek MA, Klopotek R (2020) In-the-limit clustering axioms. In: To appear in proc. ICAISC2020
27. Klopotek MA, Wierzchosi ST, Klopotek R (2020) k-means cluster shape implications. In: To appear in proc. AIAI
28. van Laarhoven T, Marchiori E (2014) Axioms for graph clustering quality functions. J Mach Learn Res 15:193–215
29. Larsen KG, Nelson J, Nyugundefined HL, Thorup M (2019) Heavy hitters via cluster-preserving clustering. Commun ACM 62(8):95–100. https://doi.org/10.1145/3339185
30. Li W, Hannig J, Mukherjee S (2021) Subspace clustering through sub-clusters. J Mach Learn Res 22:1–37
31. Liu M, Jiang X, Kot AC (2009) A multi-prototype clustering algorithm. Pattern Recognition 42(5):689–698. https://doi.org/10.1016/j.patcog.2008.09.015. http://www.sciencedirect.com/science/article/pii/S0031320308003798
32. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proc. Fifth berkeley symp. on math. statist. and prob. vol 1, pp 281–297, Univ. of calif. Press
33. Moore J, Ackerman M (2016) Foundations of perturbation robust clustering. In: IEEE ICDM. pp 1089–1094, https://doi.org/10.1109/ICDM.2016.0141
34. Mount DM (2005) Knnlocal: A testbed for k-means clustering algorithms. https://www.cs.umd.edu/mount/Projects/KMeans/knnlocal-doc.pdf
35. Nie F, Wang CL, Li X (2019) A multiple-means clustering method with specified k clusters. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 959–967, KDD ’19, association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3292500.3330846
36. Pollard D (1981) Strong consistency of k-means clustering. Ann Statist 9(1):135–140
37. Puzicha J, Hofmann T, Buhmann J (2000) A theory of proximity based clustering: structure detection by optimization. Pattern Recogn 33(4):617–634
38. Qin Y, Ding S, Wang L (2019) Research progress on semi-supervised clustering. Cogn Comput 11:599–612. https://doi.org/10.1007/s12559-019-09664-w
39. Shekar B (1988) A knowledge-based approach to pattern clustering. Ph.D. thesis Indian Institute of Science
40. Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: Proceedings of KDD
workshop on text mining, proceedings of the 6th international conference on knowledge discovery and data mining, Boston, MA

41. Strazzieri F, Sánchez-García RJ (2021) Possibility results for graph clustering: a novel consistency axiom. arXiv:1806.06142

42. Thomann P, Steinwart I, Schmid N (2015) Towards an axiomatic approach to hierarchical clustering of measures

43. Vidal A, Esteva F, Godo L (2020) Axiomatizing logics of fuzzy preferences using graded modalities. Fuzzy Sets and Systems 401:163–188. https://doi.org/10.1016/j.fss.2020.01.002. https://www.sciencedirect.com/science/article/pii/S0165011419303203.

44. Wang P, Yang X (2021) Three-way clustering method based on stability theory. IEEE Access 9:33944–33953. https://doi.org/10.1109/ACCESS.2021.3057405

45. Wang X, Kihara D, Luo J, Qi GJ (2021) Enact: a self-trained framework for semi-supervised and supervised learning with ensemble transformations. IEEE Trans Image Process 30:1639–1647

46. Wierzchoń S, Klopotek M (2018) Modern Clustering Algorithms. Studies in Big Data 34. Springer

47. Wright W (1973) A formalization of cluster analysis. Pattern Rec 5(3):273–282

48. Zadeh RB, Ben-David S (2009) A uniqueness theorem for clustering. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp 639–646, UAI ’09, AUAI Press, Arlington, Virginia, United States

49. Zeng G, Wang Y, Pu J, Liu X, Sun X, Zhang J (2016) Communities in preference networks: refined axioms and beyond. In: ICDM. pp 599–608

50. Zhao Y, Tarus SK, Yang LT, Sun J, Ge Y, Wang J (2020) Privacy-preserving clustering for big data in cyber-physical-social systems: survey and perspectives. Information Sciences 515:132–155. https://doi.org/10.1016/j.ins.2019.10.019. https://www.sciencedirect.com/science/article/pii/S0020025519309764

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mieczysław A. Klopotek received a M.Sc.Eng. and Ph.D. degrees in computer science from Dresden University of Technology. He worked at the Semiconductor Research and Production Center in Warsaw. Thereafter he joined the Institute of Computer Science of Polish Academy of Sciences where he habilitated in 1999. In 2009 he was granted the title of professor by the President of Poland. He worked at IBM Warsaw on parallel statistical software for Netezza appliance, where he co-authored five patents. He led the research group developing the first large-scale Polish semantic search engine. His current research encompasses data, text and web mining and machine learning.

Robert A. Klopotek Ph.D., is a lecturer at the Faculty of Mathematics and Natural Sciences at Cardinal Stefan Wyszyński University in Warsaw, and he is deputy director of the Institute of Computer Science at the University. He also works in an international company as a data scientist. His professional interests are social network analysis, data mining, data analysis, application of GPGPU in selected machine learning algorithms, and theoretical basis for machine learning algorithms.