Knowledge and Information Systems (2020) 62:1961–2009
https://doi.org/10.1007/s10115-019-01412-8

REGULAR PAPER

Machine learning friendly set version of Johnson–Lindenstrauss lemma

Mieczysław A. Kłopotek[1]

Received: 9 November 2018 / Revised: 26 September 2019 / Accepted: 29 September 2019 /
Published online: 14 October 2019

Abstract
The widely discussed and applied Johnson–Lindenstrauss (JL) Lemma has an existential form saying that for each set of data points Q in n-dimensional space, there exists a transformation f into an n'-dimensional space ($n' < n$) such that for each pair $\mathbf{u}, \mathbf{v} \in Q$ $(1-\delta)\|\mathbf{u}-\mathbf{v}\|^2 \leq \|f(\mathbf{u})-f(\mathbf{v})\|^2 \leq (1+\delta)\|\mathbf{u}-\mathbf{v}\|^2$ for a user-defined error parameter δ. Furthermore, it is asserted that with some finite probability the transformation f may be found as a random projection (with scaling) onto the n' dimensional subspace so that after sufficiently many repetitions of random projection, f will be found with user-defined success rate $1-\epsilon$. In this paper, we make a novel use of the JL Lemma. We prove a theorem stating that we can choose the target dimensionality in a random projection-type JL linear transformation in such a way that with probability $1-\epsilon$ all of data points from Q fall into predefined error range δ for any user-predefined failure probability ϵ when performing a *single* random projection. This result is important for applications such as data clustering where we want to have a priori dimensionality reducing transformation instead of attempting a (large) number of them, as with traditional Johnson–Lindenstrauss Lemma. Furthermore, we investigate an important issue whether or not the projection according to JL Lemma is really useful when conducting data processing, that is whether the solutions to the clustering in the projected space apply to the original space. In particular, we take a closer look at the k-means algorithm and prove that a good solution in the projected space is also a good solution in the original space. Furthermore, under proper assumptions local optima in the original space are also ones in the projected space. We investigate also a broader issue of preserving clusterability under JL Lemma projection. We define the conditions for which clusterability property of the original space is transmitted to the projected space, so that a broad class of clustering algorithms for the original space is applicable in the projected space.

Keywords Johnson–Lindenstrauss lemma · Random projection · Sample distortion · Dimensionality reduction · Linear JL transform · k-means algorithm · Clusterability retention

✉ Mieczysław A. Kłopotek
Mieczyslaw.Klopotek@ipipan.waw.pl

[1] Institute of Computer Science of the Polish Academy of Sciences, ul. Jana Kazimierza 5, 01-248 Warsaw, Poland

1 Introduction

Dimensionality reduction plays an important role in many areas of data processing, and especially in machine learning (cluster analysis, classifier learning, model validation, data visualization, etc.).

Usually, it is associated with manifold learning, that is a belief that the data lie in fact in a low-dimensional subspace that needs to be identified and the data projected onto it so that the number of degrees of freedom is reduced and as a consequence also sample sizes can be smaller without loss of reliability. Techniques such as reduced k-means [40], PCA (Principal Component Analysis), Kernel PCA, LLE (Locally Linear Embedding), LEM (Laplacian Eigenmaps), MDS (Metric Multidimensional Scaling), Isomap, SDE (Semidefinite Embedding), just to mention a few, are applied in order to achieve dimensionality reduction.

But there exists still another possibility of approaching the dimensionality reduction problems, in particular when such intrinsic subspace where data is located cannot be identified. The problem of choice of the subspace has been circumvented by several authors by the so-called random projection, applicable in extremely high-dimensional spaces (tens of thousands of dimensions) and correspondingly large data sets (of at least hundreds of points).

1.1 Johnson–Lindenstrauss lemma and dimensionality reduction

The starting point here is the Johnson–Lindenstrauss (JL) Lemma [25]. Roughly speaking, it states that there exists a linear[1] mapping from a higher dimensional space into a sufficiently high-dimensional subspace that will preserve approximately the distances between points, as needed, e.g. by k-means algorithm [5]. In fact, when designing optimal k-means clustering algorithms, the possibility of dimensionality limitation via Johnson–Lindenstrauss Lemma is assumed, see, e.g. [3].

To be more formal, consider a set $\mathfrak{Q}$ of m objects $\mathfrak{Q} = \{1, \ldots, m\}$. An object $i \in \mathfrak{Q}$ may have a representation $\mathbf{x}_i \in \mathbb{R}^n$. Then, the set of these representations will be denoted by Q. An object $i \in \mathfrak{Q}$ may have a representation $\mathbf{x'}_i \in \mathbb{R}^{n'}$, in a different space. Then, the set of these representations will be denoted by Q'.

With this notation, let us state:

Theorem 1.1 (Johnson–Lindenstrauss) *Let $\delta \in (0, \frac{1}{2})$. Let $\mathfrak{Q}$ be a set of m objects and Q—a set of points representing them in $\mathbb{R}^n$, and let $n' \geq \frac{C \ln m}{\delta^2}$, where C is a sufficiently large constant. There exists a Lipschitz mapping $f : \mathbb{R}^n \to \mathbb{R}^{n'}$ such that for all $\mathbf{u}, \mathbf{v} \in Q$*

$$(1-\delta)\|\mathbf{u}-\mathbf{v}\|^2 \leq \|f(\mathbf{u}) - f(\mathbf{v})\|^2 \leq (1+\delta)\|\mathbf{u}-\mathbf{v}\|^2 \tag{1}$$

The above formulation is cited after [31]. Larsen and Nelson [28] demonstrated that if the function f has to be linear, then the lower bound on n' is optimal for δ and m.

Other papers propose slightly different formulas for n', for example Gupta and Dasgupta [20]: prove the bound $n' \geq 4\frac{\ln m}{\delta^2 - \delta^3}$. with a different denominator. But as δ is limited from above by $\frac{1}{2}$, it is easily seen that the formula of [20] implies $n' \geq 8\frac{\ln m}{\delta^2}$, that is the original formulation with $C = 8$. Still another formulation by Frankl and Maehara [23] implies C is around 9. Empirical studies [42] suggest that $C = 2$ is sufficient.[2]

[1] JL Lemma speaks about a general transformation, but many researchers look just for linear ones.

[2] Other recommendations for C include value 20, see lecture on Random Projections by Sham Kakade and Greg Shakhnarovich http://ttic.uchicago.edu/~gregory/courses/LargeScaleLearning/lectures/jl.pdf.

The researchers are interested also in extensions to original JL Lemma. For example, Matousek [31] considers other spaces than Euclidean, e.g. with ℓ_1 norm or sparse spaces. Achlioptas [1] studies random projections with sparse matrices. Baraniuk et al. [13] extend the JL Lemma to manifolds. Magen [30] explored the preservation of n-dimensional angles when projecting to lower dimensional spaces so that volumes can be preserved also.

The JL Lemma has diverse applications. It has been used for rounding, embedding into low-dimensional spaces, neural network-based machine learning, information retrieval, compressed sensing, data stream analysis, approximate nearest neighbour search, just to mention a few. Image analysis, in particular motion analysis, is a quite typical domain giving rise to high-dimensional data that can be reduced via JL Lemma (see e.g. [35]).

In this paper, we are particularly interested in applications to cluster analysis. Already, Schulman [36] was interested in this topic, in particular in optimizing the intracluster sum of weights in graph clustering. He uses JL Lemma to reduce the computational time. Tremblay et al. [41] exploit the JL Lemma also for the task of graph clustering in a kind of graph sampling approach (in the spectral space).

A number of proofs and applications of Theorem 1.1 have been proposed which in fact do not prove Theorem 1.1 as such but rather create a probabilistic version of it, e.g. [1,4,17, 20,24,29]. For an overview of Johnson–Lindenstrauss Lemma variants, see, e.g. [31].

Essentially, the idea behind these probabilistic proofs is as follows: Assume that it is proven that the probability of reconstructing the length of a random vector from a projection onto a subspace within a reasonable error bounds is high.

One then inverts the thinking and states that the probability of reconstructing the length of a given vector from a projection onto a (uniformly selected) random subspace within a reasonable error bounds is high.

But uniform sampling of high-dimensional subspaces is a hard task. So n' vectors with random coordinates are sampled instead from the original n-dimensional space and one uses them as a coordinate system in the n'-dimensional subspace which is a much simpler process. One hopes that the sampled vectors will be orthogonal (and hence the coordinate system will be orthogonal) which in case of vectors with thousands of coordinates is reasonable. That means we create a matrix M of n' rows and n columns as follows: for each row i we sample n numbers from $\mathcal{N}(0, 1)$ forming a row vector $\mathbf{a}_i^T$. We normalize it obtaining the row vector $\mathbf{b}_i^T = \mathbf{a}_i^T \cdot (\mathbf{a}_i^T \mathbf{a}_i)^{-1/2}$. This becomes the ith row of the matrix M. Then, for any data point $\mathbf{x}$ in the original space its random projection is obtained as $\mathbf{x}' = M\mathbf{x}$.

Then, the mapping we seek is the projection multiplied by a suitable factor.

It is claimed afterwards that this mapping is distance-preserving not only for a single vector, but also for large sets of points with positive but usually very small probability, as Dasgupta and Gupta [20] maintain. Via applying the above process, many times one can finally get the mapping f that is needed. That is, each time we sample a subspace from the space of subspaces and check whether condition expressed by inequality (1) holds for all the points. If it does not, we sample again, while we have the reasonable hope that we will get the subspace with required properties after a finite number of steps with probability that we assume.[3]

1.2 JL dimensionality reduction and *k*-means clustering problem

In this paper, we explore the following deficiency of the discussed approach: If we want to apply, for example, a k-means clustering algorithm, we are in fact not interested in resampling

[3] Sivakumar [39] proposes an approach to de-randomization of the process of seeking the f function.

the subspaces in order to find a convenient one so that the distances are sufficiently preserved. Computation over and over again of $m^2/2$ distances between the points in the projected space may turn out to be much more expensive than computing $O(mk)$ distances during k-means clustering (if $m \gg k$) in the original space. In fact, we are primarily interested in clustering data. But we do not have any criterion for the k-means algorithm that would say that this particular subspace is the right one via, e.g. minimization of k-means criterion (and in fact for any other clustering algorithm).

Therefore, we rather seek a scheme that will allow us to say that by a certain random sampling we have already found the suitable subspace that we sought with a sufficiently high probability. As far as we know, this is the first time such a problem has been posed.

But from the point of view of clustering (and more generally, other data mining tasks), the fact of keeping the projection errors of pairs of points in a certain range is not by itself sufficient from practical point of view. We want also to know whether or not the projection will distort the solution to the problem in the original space.

In particular, we take a closer look at the k-means algorithm and prove that a good solution in the projected space is also a good solution in the original space (see Theorem 2.3). Furthermore, under proper assumptions local optima in the original space are also ones in the projected space and vice versa (Theorems 2.4, 2.5). Finally, we show that a perfect k-means algorithm (an ideal algorithm that returns global optimum solution to the k-means clustering problem) in the projected space provides with a constant factor approximation of the global optimum in the original space (Theorem 2.6). In this way, we state conditions under which it is worthwhile performing k-means in the projected spaces with guarantees that the solutions will tell us something about the original problem in the original space.

To formulate claims concerning k-means, we need to introduce additional notation. Let us denote with $\mathfrak{C}$ a partition of $\mathfrak{Q}$ into k clusters $\{C_1, \ldots, C_k\}$. For any $i \in \mathfrak{Q}$, let $\mathfrak{C}(i)$ denote the cluster C_j to which i belongs. Let $Q = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be the set of representations of these objects in some Euclidean space, and let $Q' = \{\mathbf{x'}_1, \ldots, \mathbf{x'}_m\}$ be the set of representations of these objects in another Euclidean space (after a random projection). For any set of objects C_j, let $\boldsymbol{\mu}(C_j) = \frac{1}{|C_j|}\sum_{i \in C_j} \mathbf{x}_i$ and $\boldsymbol{\mu}'(C_j) = \frac{1}{|C_j|}\sum_{i \in C_j} \mathbf{x'}_i$.

Under this notation, the k-means cost function may be written as

$$\mathfrak{J}(Q, \mathfrak{C}) = \sum_{i \in \mathfrak{Q}} \|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \tag{2}$$

$$\mathfrak{J}(Q', \mathfrak{C}) = \sum_{i \in \mathfrak{Q}} \|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \tag{3}$$

for the sets Q, Q'.

1.3 JL dimensionality reduction and clusterability problem

We investigate also the broader issue of preserving clusterability when JL Lemma projection is applied. We define the conditions for which clusterability property of the original space is transformed to the projected space, so that a broad class of clustering algorithms for the original space is applicable in the projected space. The property of clusterability is informally speaking understood as a special property of data that makes finding the good clustering of data an easy task. In the literature, five brands of clusterability are usually distinguished (see [14]): Perturbation Robustness, σ-Separatedness, c, σ-Approximation-Stability, β-Centre-Stability and the Weak Deletion stability. Perturbation Robustness refers to the property of the data that under slight distortions of distances between data points the optimal clustering will be

safeguarded. The projection under JL Lemma may obviously lead to violation of this property, because the projection already distorts the distances. Therefore in Theorem 3.6 we establish conditions under which the Perturbation Robustness will be preserved. σ-Separatedness refers to the property that the cost of optimal clustering into k clusters should be by some factor lower than clustering into $k-1$ clusters. As the JL projection changes the distances between points, the cost function value proportions will also change in a way that cannot be determined in advance (decrease or increase). In Theorem 3.1, we establish conditions under which this property is preserved. c, σ-Approximation-Stability property refers to closeness of costs between alternative partitions. This closeness may be obviously distorted by JL projection, so that partitions close in the original space will not be sufficiently close in the projected space and vice versa. Theorem 3.2 establishes conditions, under which the property may be nonetheless preserved under projection. The β-Centre-Stability property requires that the distance of a point to its own cluster centre is by a factor smaller than to other cluster centres. Under the JL projection, a violation of this property may take place as a point may be moved further from its own cluster centre and closer to some other. Theorem 3.7 gives conditions under which the property is retained. Finally, the Weak Deletion stability requires that removal of one cluster centre will impose some minimal increase in the cost function of the clustering. Changes in the distances may clearly lead to violation of this property, and it cannot then be exploited in the projected space. Theorem 3.8 shows conditions under which the projection will uphold the property.

1.4 Our contribution

Our contribution is as follows:

- We formulate and prove a set version of JL Lemma—see Theorem 2.1.
- Based on it, we demonstrate that a good solution to k-means problem in the projected space is also a good one in the original space—see Theorem 2.3.
- We show that there is 1–1 correspondence between local k-means minima in the original and the projected spaces under proper conditions—see Theorems 2.4, 2.5.
- We demonstrate that a perfect k-means algorithm in the projected space is a constant factor approximation of the global optimum in the original space—see Theorem 2.6
- We prove that the projection preserves several clusterability properties such as Multiplicative Perturbation Robustness, (Theorem 3.6), σ-Separatedness (Theorem 3.1), c, σ-Approximation-Stability (Theorem 3.2), β-Centre-Stability (Theorem 3.7) and the Weak Deletion stability (Theorem 3.8).

The structure of the paper is as follows. In Sect. 2, we introduce the set-friendly version of Johnson–Lindenstrauss Lemma together with theorems investigating properties of results of k-means algorithm in the original and the projected spaces (local and global optima). In Sect. A.1, we prove the set-friendly version of JL Lemma, whereas the proofs of the remaining theorems introduced in Sect. 2, that is, the ones relating k-means clustering results in the original and the projected spaces, are deferred to Appendix B. In Sect. 3, we demonstrate an additional advantage of our version of JL Lemma consisting in preservation of various data clusterability criteria. In Sect. 4, we illustrate the advantage of Theorem 2.1 by some numerical simulation results, showing at the same time the impact of various parameters of our version of Johnson–Lindenstrauss Lemma on the dimensionality of the projected space. We show that k-means clustering behaves as expected under JL Lemma projection. We verify also via simulations the clusterability preserving properties of the JL Lemma related

projection. In Sect. 5, we recall the corresponding results of other authors. Section 6 contains some concluding remarks.

2 Relationship between *k*-means solutions in the projected and original space

While it is a desirable property if we can reduce the dimensionality of space in which the data is embedded (e.g. the computational burden is reduced), we should be aware of the fact that we distort the data and in this way we run at risk that operating on the projected data we miss the solution of our original problem. This is the primary concern about the original JL Lemma as expressed in Theorem 1.1. This is in particular worthwhile considering risk in the domain of k-centre data clustering. Distortion of distances may lead to an undesirable situation that some data may switch between clusters and in an extreme case the optimal clusters in the original and in the projected domains are barely overlapping.

2.1 Reformulated JL lemma and the impact on cost function value in projected space

We begin this section with presentation of our version of the JL Lemma, expressed in Theorem 2.1. While the original JL Lemma is concerned with the existence of a solution to the projection problem (a projection fitting error bounds on pairwise distances), we ensure under what circumstances the projection almost surely fits the pairwise distance error bounds.

This theorem constitutes a significant step forward towards applicability of the projection. However, it is not enough. Only keeping error bounds is ensured, but we need more: assurance that the solutions to the clustering problem in the projected space faithfully represent the solutions in the original space.

Therefore, we present subsequently a series of our claims about the relationship between the k-means clustering algorithm results in the projected and the original space. A close relationship is needed if the random projection is to be used as a step preceding application of the k-means clustering algorithm to the data. We express the properties of the original clustering problem required so that the dimensionality reduction makes real sense.

We restrict in this way our attention to a particular class/family of clustering algorithms. Such a restriction is on the one hand necessary in order to get precise results, and on the other hand the popularity of k-means family is so widespread that the results can still be of vital interest for a broad audience.

This limitation, as shown in Theorem 2.3, allows us to exploit the particular form of JL Lemma to limit directly the solution distortions in the projected space compared to the original one.

This limitation is also related to specific properties of the k-means algorithm family. These algorithms aim at optimizing a cost function that has quite a rough landscape. Therefore, frequently the algorithms get stuck at a local minimum. To demonstrate the equivalence of solutions in both original and projected spaces, we show in Theorem 2.4 that the local minima of the original space may under some conditions correspond to local minima in the projected space. This means if we look for local minima in the original space, we can as well limit our attention to the projected space. Theorem 2.5 considers the relationship in the opposite direction—local minima in the projected space may be well local minima in the original

space. This means: if we found a local minimum in the projected space, we have found one in the original space.

Concerning the global optimum, we demonstrate in Theorem 2.6 that the global optimum in the projected space is a constant factor approximation of the global optimum of the original space. Hence it makes sense to look for a global optimum in the projected space.

So let us turn to the general theorem on random projection.

Theorem 2.1 *Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0, 1)$. Let $Q \subset \mathbb{R}^n$ be a set of m points in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let*

$$n' = n'_E \geq 2\frac{-\ln\epsilon + 2\ln(m)}{-\ln(1+\delta)+\delta} \tag{4}$$

Let $\mathcal{C}_{n'}$ be a randomly selected n'-dimensional orthogonal coordinate system. For each $\mathbf{v} \in Q$, let $\mathbf{v}'$ be its projection onto $\mathcal{C}_{n'}$. Then, for all *pairs $\mathbf{u}, \mathbf{v} \in Q$*

$$(1-\delta)\|\mathbf{u}-\mathbf{v}\|^2 \leq \frac{n}{n'}\|\mathbf{u}'-\mathbf{v}'\|^2 \leq (1+\delta)\|\mathbf{u}-\mathbf{v}\|^2 \tag{5}$$

holds with probability of at least $1-\epsilon$.

The proof of this Theorem can be found in Appendix A.1.

The fundamental new insight of this theorem is to explicitly refer to the failure probability ϵ. In the literature, see, e.g. [10–12,20], the failure probability ϵ was not referred to as a control parameter of dimensionality. It was rather derived from other parameters controlling n' in the respective formulas. With our formulation, the user has the freedom to choose as low ϵ as she/he wants to have the probability of success $1-\epsilon$ to get (in a single pass) a random projection fitting the pairwise (relative) distance error to be limited by δ.

Note that the lower bound on dimensionality n' grows with the inverted square of the permissible relative error range δ ($-\ln(1+\delta)+\delta \approx \delta^2/2$). One can see it in Fig. 1 (the black line). The sample size, on the other hand, affects n' logarithmically only, as visible in Fig. 2 (black line). Figure 3 (black line) illustrates the strong dependence of lower bound of n' on the error rate ϵ (logarithmic dependence for values below 1).

The formula (4) provides with an explicit expression for computing a lower bound on n'. We will refer to it sometimes therefore as n'_E, "E" standing for *Explicit*. It is also independent of n. The question may be raised, whether it is possible to reduce n' via exploitation of n.

We propose for this purpose Algorithm 1. The algorithm seeks via bisection the solution of minimalization problem for the function

$$\epsilon_I(n') = \binom{m}{2}\left((1-\delta)^{\frac{n'}{2}}\left(1+\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}} + (1+\delta)^{\frac{n'}{2}}\left(1-\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}\right) \tag{6}$$

The algorithm proceeds as follows: One starts with $n'_L := 1, n'_H := round(n/3) - 1$. If $\epsilon_I(n'_H) > \epsilon$, then seeking n'_I has failed. Otherwise, one determines in a loop $n'_M := round((n'_L + n'_H)/2)$ and computes $\epsilon_I(n'_L), \epsilon_I(n'_M), \epsilon_I(n'_H)$, then if $\epsilon_I(n'_M) < \epsilon$ then one sets $n'_H := n'_M$, otherwise $n'_L := n'_M$ (n'_M is always rounded up to the next integer). This process is continued till n'_M does not change. n'_I is then set to n'_H.

Theorem 2.2 *Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0, 1)$. Let $Q \subset \mathbb{R}^n$ be a set of m points in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let $n' = n'_I$ be computed according to Algorithm 1*

$$n' = n'_I \geq \mathrm{argmin}_b\, \epsilon_I(b) \tag{7}$$

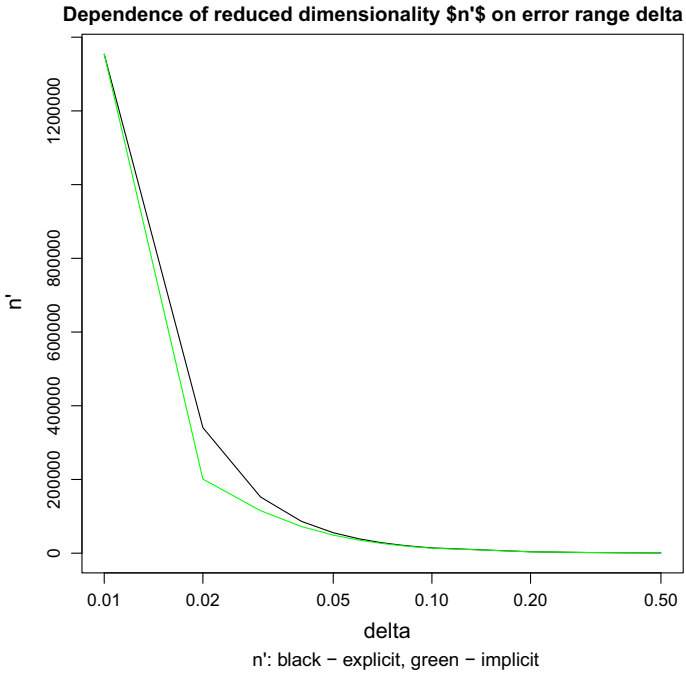

Fig. 1 Dependence of reduced dimensionality n' on error range δ. Other parameters fixed at $m = 2e{+}06\, \epsilon = 0.01\, n = 5e{+}05$ (color figure online)

Algorithm 1 Computing n'_I as $\arg\min \epsilon_I(n')$

Require: $n, m, \epsilon > 0, 0.5 > \delta > 0$
  $n'_L \leftarrow 1, n'_H \leftarrow round(n/3) - 1.$
  if $\epsilon_I(n'_H) > \epsilon$ **then return** ERROR
  end if
  repeat
    $n'_M \leftarrow roundUP((n'_L + n'_H)/2)$
    $e_L \leftarrow \epsilon_I(n'_L)$
    $e_M \leftarrow \epsilon_I(n'_M)$
    $e_H \leftarrow \epsilon_I(n'_H)$
    if $e_M < \epsilon$ **then**
      $n'_H \leftarrow n'_M$
    else
      $n'_L \leftarrow n'_M$
    end if
    $n'_{Mprev} \leftarrow n'_M$
    $n'_M \leftarrow roundUP((n'_L + n'_H)/2)$
  until ($n'_M = n'_{Mprev}$)
  $n'_I \leftarrow n'_H$
  return n'_I

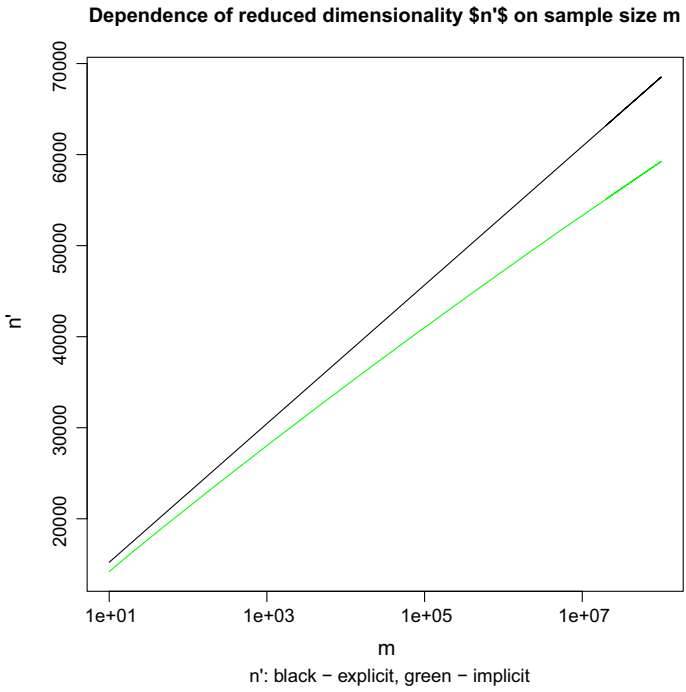

Fig. 2 Dependence of reduced dimensionality n' on sample size m. Other parameters fixed at $\epsilon = 0.01\, \delta = 0.05\, n = 5\text{e}{+}05$ (color figure online)

Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. For each $\mathbf{v} \in Q$, let $\mathbf{v}'$ be its projection onto $\mathcal{C}_{n'}$. Then, for all *pairs $\mathbf{u}, \mathbf{v} \in Q$ the relation* (5) *holds with probability at least $1 - \epsilon$*

Let us stress at this point the significance of these theorems. Earlier forms of JL Lemma required sampling of the coordinates over and over again, with quite a low success rate (e.g. in [20] about $\frac{1}{m}$) until a mapping is found satisfying the error constraints (see, e.g. [39]). In our theorems, we need only one sampling in order to achieve the required success probability of selecting a suitable subspace to perform k-means.

These sampling theorems can be used for any algorithm that requires a dimensionally reduced sample keeping some distortion constraints.

It turns out, however, that the properties of this sampling technique are particularly relevant for k-means.

We make the following claim for k-means objective:

Theorem 2.3 *Let Q be a set of m representatives of objects from $\mathfrak{Q}$ in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$. Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0, 1)$. and let n' satisfy condition* (4) *or* (7)*. Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. Let the set Q' consist of m objects such that for each $i \in \mathfrak{Q}$, $\mathbf{x}'_i \in Q'$ is a projection of $\mathbf{x}_i \in Q$ onto $\mathcal{C}_{n'}$. If $\mathfrak{C}$ is a partition of $\mathfrak{Q}$, then*

$$(1 - \delta)\mathfrak{J}(Q, \mathfrak{C}) \leq \frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}) \leq (1 + \delta)\mathfrak{J}(Q, \mathfrak{C}) \tag{8}$$

holds with probability of at least $1 - \epsilon$.

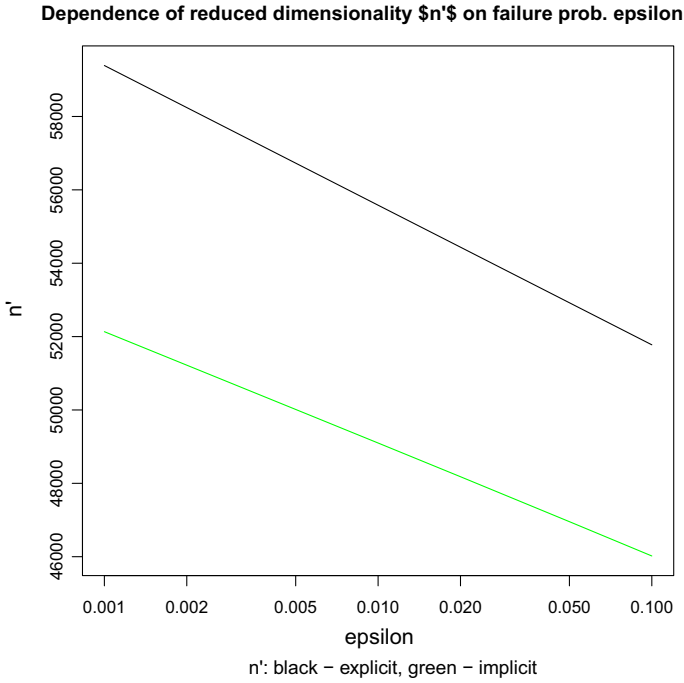

Fig. 3 Dependence of reduced dimensionality n' on failure prob. ϵ. Other parameters fixed at $m = 2\text{e}{+}06\, \delta = 0.05\, n = 5\text{e}{+}05$ (color figure online)

Note that the inequality (8) can be rewritten as

$$\left(1 - \frac{\delta}{1+\delta}\right) \mathfrak{J}(Q', \mathfrak{C}) \leq \frac{n'}{n} \mathfrak{J}(Q, \mathfrak{C}) \leq \left(1 + \frac{\delta}{1-\delta}\right) \mathfrak{J}(Q', \mathfrak{C}) \tag{9}$$

The above theorem tells us how close we can approximate the cost function of k-means in the original space via the solution to k-means problem in the projected space—the relative estimation error is limited by δ.

2.2 Local and global cost function minima in projected space

But the interdependence between solutions to k-means problem in both spaces is even deeper, as the two subsequent theorems show. Let us look at the local minima of k-means at which the k-means algorithms usually get stuck. Theorem 2.4 states conditions under which the local minima of the original space correspond to local minima in the projected space. Theorem 2.5 considers the relationship in the opposite direction.

Before proceeding, let us introduce the concept of the gap between clusters. It is obvious that if there are points at the border of two clusters, then under projection there exists a high risk that the points would move to the other cluster. Therefore, in order to keep cluster membership unchanged under projection, a gap between the clusters needs to exist. We shall understand the gap as follows: k-means assures that for any two clusters C_1, C_2 there exists a hyperplane h orthogonal to the line segment connecting both cluster centres and cutting it at half of the distance between these centres. The points of one cluster lie on the one

side of this plane, the points of the other on the other side. The absolute gap G between the two clusters is understood as twice the smallest distance of any element of these two clusters to this hyperplane h. That is, for an absolute gap G_{C_1,C_2} between the two clusters, the distance between any point of these clusters and the border is larger than $G_{C_1,C_2}/2$. We prefer subsequently to refer to the relative gap g, that is G divided by the distance between the two cluster centres.

Theorem 2.4 *Under the assumptions and notation of Theorem 2.3, if the partition $\mathfrak{C}^*$ yields a local minimum (in the original space) of $\mathfrak{J}(Q, \mathfrak{C})$ over all possible partitions $\mathfrak{C}$ of $\mathfrak{Q}$ and if for any two clusters for some $\alpha \in [0, 1)$ $g = 2(1 - \alpha)$ is lower or equal the relative gap between these clusters, and*

$$\delta \leq \frac{1 - \left(1 - \frac{g}{2}\right)^2}{\left(1 - \frac{g}{2}\right)^2 + (1 + 2p)} \tag{10}$$

(p to be defined later by inequality (38) on page 46), then this same partition is (in the projected space) also a local minimum of $\mathfrak{J}(Q', \mathfrak{C})$ over $\mathfrak{C}$, with probability of at least $1 - \epsilon$.

Theorem 2.4 tells us that we need to meet two conditions if we want that local minima in the original space have their corresponding local minima in the projected space. The one refers to the need of separation of each pair of clusters by a gap—the relative width of area between clusters (where no data points are present) shall amount at least to some g. The second condition imposes restrictions on the upper bound for δ which does not equal to $1/2$ any longer, but may be a smaller value, depending on the mentioned gap (the smaller the gap, the smaller the δ). δ is influenced also by a factor p representing compactness of the clusters. Indirectly of course the projected space dimensionality is affected because the smaller the δ is, the larger the n' will be.

Quite a similar behaviour is observed if we request that the local minimum found in the projected space should correspond to a local minimum in the original space

Theorem 2.5 *Under the assumptions and notation of Theorem 2.3, if the clustering $\mathfrak{C}'^*$ constitutes a local minimum (in the projected space) of $\mathfrak{J}(Q', \mathfrak{C})$ over $\mathfrak{C}$ and if for any two clusters $1 - \alpha$ times the distance between their centres is the gap between these clusters, where $\alpha \in [0, 1)$, and*

$$\frac{\delta}{1 - \delta} \leq \frac{1 - \alpha^2}{(1 + 2p) + \alpha^2} \tag{11}$$

then the very same partition $\mathfrak{C}'^*$ is also (in the original space) a local minimum of $\mathfrak{J}(Q, \mathfrak{C})$ over $\mathfrak{C}$, with probability of at least $1 - \epsilon$.

Note that in both theorems, the conditions on δ are quite similar and converge to one another for vanishing δ. In fact the condition in Theorem 2.5 implies that of Theorem 2.4, so if local minima in the projected space correspond to ones in the original one, then those of the original one correspond to those of the projected one.

The fact that the local minima correspond to each other in both spaces does not automatically mean that the global minimum in the original space is also the global minimum in the projected space. However, the theorem as follows indicates that the values at both optima correspond closely to one another. That is, if we find the global minimum in the projected space then we can estimate the global optimum in the original space quite well.

Theorem 2.6 *Under the assumptions and notation of Theorem 2.3, if* $\mathfrak{C}_{\mathfrak{G}}$ *denotes the clustering reaching the global optimum in the original space, and* $\mathfrak{C}'_{\mathfrak{G}}$ *denotes the clustering reaching the global optimum in the projected space, then*

$$\frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G}}) \leq (1+\delta)\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G}}) \tag{12}$$

$$\frac{n'}{n}\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G}}) \leq (1-\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G}}) \tag{13}$$

with probability of at least $1-\epsilon$.

That is, the perfect k-means algorithm in the projected space is a constant factor approximation of k-means optimum in the original space.

We defer the proof of Theorems 2.3–2.6 till Appendix B. We will derive the basic Theorem 2.1 in Appendix A.1 which is essentially based on the results reported by Dasgupta and Gupta [20]. And the proof of Theorem 2.2 can be found in Appendix A.2.

3 Clusterability and the dimensionality reduction

In Sect. 2, we have stated conditions under which our formulation of JL Lemma (Theorem 2.1) allows to use random projection to perform the clustering in the projected space instead of one in original space using a specific class of clustering algorithms, namely k-means. One may rephrase our results by saying that we have formulated conditions for k-means family under which the transformed data can be clustered the same way as the original data.

Let us turn in this section to the somehow related topic of clusterability of data. The property that the data is *clusterable* usually means that the clustering of that data can be easily found, i.e. with an algorithm of polynomial complexity. This is a very useful property for large data sets. Clusterability conditions usually include requirements of identical clustering under some data transformation. An important question is whether or not adding the extra random projection as data transformation may lead to the loss of clusterability property. The specific contribution of this section is the discussion how well the general property of clusterability is preserved under random projection using our version of JL Lemma in case when the cost function is defined as for k-means.

While clusterability property has some intuitive appeal as a property of data allowing for "easy clustering", concrete formal notions of clusterability differ substantially. The "easiness" refers generally to (low) algorithm complexity given some restrictions imposed on the form of the data, but these restrictions may have different forms hence the various notions of clusterability.

And in fact, in the literature a number of notions of the so-called clusterability have been introduced [2,6–8,14,15,32].[4] Under these notions of clusterability algorithms have been developed clustering the data nearly optimally in polynomial times, when some constraints are matched by the clusterability parameters.

It seems therefore worth to have a look at the issue if the aforementioned projection technique would affect the clusterability property/properties of the data sets.

[4] One says that a data set is *clusterable* if a clustering algorithm can cluster the data quickly, with low complexity, usually polynomial. Formal definitions of clusterability have diverse forms as apparently there exist various conditions under which clustering is an easy task.

3.1 Selected notions of clusterability and issues with JL projection

Let us consider, as representatives, the following notions of clusterability, present in the literature:

- σ-*Separatedness* [32] means that the cost $\mathfrak{J}(Q, \mathfrak{C}_k)$ of optimal clustering $\mathfrak{C}_k$ of the data set Q into k clusters is less than σ^2 $(0 < \sigma < 1)$ times the cost $\mathfrak{J}(Q, \mathfrak{C}_{k-1})$ of optimal clustering $\mathfrak{C}_{k-1}$ into $k-1$ clusters

$$\mathfrak{J}(Q, \mathfrak{C}_k) < \sigma^2 \mathfrak{J}(Q, \mathfrak{C}_{k-1}) \tag{14}$$

- (c, σ)-*Approximation-Stability* [8] means that if the cost function values of two partitions $\mathfrak{C}_a, \mathfrak{C}_b$ differ by at most the factor $c > 1$ (that is $c \cdot \mathfrak{J}(Q, \mathfrak{C}_a) \geq \mathfrak{J}(Q, \mathfrak{C}_b)$ and $c \cdot \mathfrak{J}(Q, \mathfrak{C}_b) \geq \mathfrak{J}(Q, \mathfrak{C}_a)$), then the distance (in some space) between the partitions is at most σ ($d(\mathfrak{C}_a, \mathfrak{C}_b) \leq \sigma$ for some distance function d between partitions). As Ben-David [14] remarks, this implies the uniqueness of optimal solution.
- *Perturbation Robustness* means that small perturbations of distances / positions in space of set elements do not result in a change in the optimal clustering for that data set. Two kinds may be distinguished: additive [2] and multiplicative ones [15] (the limit of perturbation is upper-bounded either by an absolute value or by a coefficient).
  The s-Multiplicative Perturbation Robustness $(0 < s < 1)$ holds for a data set with d_1 being its distance function if the following holds. Let $\mathfrak{C}$ be an optimal clustering of data points with respect to this distance. Let d_2 be any distance function over the same set of points such that for any two points $\mathbf{u}, \mathbf{v}$, $s \cdot d_1(\mathbf{u}, \mathbf{v}) < d_2(\mathbf{u}, \mathbf{v}) < \frac{1}{s} \cdot d_1(\mathbf{u}, \mathbf{v})$. Then, the same clustering $\mathfrak{C}$ is optimal under the distance function d_2.
  The s-Additive Perturbation Robustness $(s > 0)$ holds for a data set with d_1 being its distance function if the following holds. Let $\mathfrak{C}$ be an optimal clustering of data points for this distance. Let d_2 be any distance function over the same set of points such that for any two points $\mathbf{u}, \mathbf{v}$, $d_1(\mathbf{u}, \mathbf{v}) - s < d_2(\mathbf{u}, \mathbf{v}) < d_1(\mathbf{u}, \mathbf{v}) + s$. Then, the same clustering $\mathfrak{C}$ is optimal under the distance function d_2.
  Next, we are interested only in the multiplicative version.
- β-*Centre Stability* [7] means, for any centric clustering, that the distance of an element to its cluster centre is $\beta > 1$ times smaller than the distance to any other cluster centre under optimal clustering.
- $(1 + \beta)$ *Weak Deletion Stability* [6] $(\beta > 0)$ means that given an optimal cost function value OPT for k centric clusters, the cost function of a clustering obtained by deleting one of the cluster centres and assigning elements of that cluster to one of the remaining clusters should be bigger than $(1 + \beta) \cdot OPT$.

Subsequently, we consider the question what is the impact of random projection under our proposal of dimensionality reduction on preservation of various forms of clusterability. In Theorem 3.1, we show that the σ-Separatedness in the projected space increases as a function of δ which means that the larger error δ is allowed, the larger must be the advantage of (optimal) clustering into k clusters over clustering into $k-1$ clusters in order to be able to take advantage of clusterability property by clustering algorithms (smaller σ indicates a bigger advantage of clustering into k clusters over one into $k-1$ clusters, and usually σ must lie below some threshold for optimal algorithms to be applicable). Conversely, if this advantage is lower in the original space, then also δ has to be lower and therefore also a smaller dimensionality reduction is permitted.

Theorem 3.2 points out that the random projection narrows (with increasing δ) the range c of clustering cost functions that have to lie within the same distance σ from a given

clustering for (c, σ)-Approximation-Stability. The decrease in c, as shown by Balcan et al. [8], is disadvantageous as it worsens the approximation complexity of the optimal clustering of a data set, possibly to the point of NP-hardness. Therefore, low error values δ are preferred.

The next notion of clusterability that we will investigate is the Multiplicative Perturbation Stability. The effect of random projection on this property will be explained in Theorem 3.6. In order to prove it, we will need two lemmas: Lemma 3.3 on double perturbation and Lemma 3.5 on conditions when multiplicative perturbation stability ensures that the global k-means optima in the original and the projected space are identical.

Finally, we will turn to theorems on β-Centre-Stability (Theorem 3.7) and $1+\beta$ Weak-Deletion-Stability (Theorem 3.8). The discussion of both these properties will be performed in conjunction with Multiplicative Perturbation Stability.

3.2 σ-separatedness in the projected space

Let us first have a look at the σ-Separatedness. Assume that the data Q in the original space have the property of σ-Separatedness for some σ. Let $\mathfrak{C}_{\mathfrak{G},\mathfrak{k}}$ denote an optimal clustering into k clusters in the original space and $\mathfrak{C}'_{\mathfrak{G},\mathfrak{k}}$ in the projected space.

From Theorem 2.6, we know that

$$\frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}}) \leq (1+\delta)\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G},\mathfrak{k}}) \tag{15}$$

and

$$\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G},\mathfrak{k}-1}) \leq \frac{n}{n'}(1-\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}-1}) \tag{16}$$

σ-Separatedness (14) implies that

$$\sigma^2 \geq \frac{\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G},\mathfrak{k}})}{\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G},\mathfrak{k}-1})} \geq \frac{\frac{n}{n'}(1+\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}_{\mathfrak{G},\mathfrak{k}})}{\mathfrak{J}(Q, \mathfrak{C}_{\mathfrak{G}',\mathfrak{k}-1})}$$

Note that the latter inequality was obtained by applying inequality (15) in the nominator.

$$\geq \frac{\frac{n}{n'}(1+\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}})}{\frac{n}{n'}(1-\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}-1})} = \frac{(1-\delta)\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}})}{(1+\delta)\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}-1})}$$

which was obtained by applying inequality (16) in the denominator

This implies

$$\sigma^2\frac{1+\delta}{1-\delta} \geq \frac{\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}})}{\mathfrak{J}(Q', \mathfrak{C}'_{\mathfrak{G},\mathfrak{k}-1})}$$

We have thus proved

Theorem 3.1 *Under the assumptions and notation of Theorem 2.3, if the data set Q has the property of σ-Separatedness in the original space, then with probability at least $1-\epsilon$ it has the property of $\sigma\sqrt{\frac{1+\delta}{1-\delta}}$-Separatedness in the projected space.*

The fact that this Separatedness increases under projection is of course a deficiency, because clustering algorithms require as low Separatedness as possible (because the clusters are then better separated). Recall that Ostrovsky et al. [32] developed a series of algorithms for efficient computation of near-to-optimal k-means clustering in polynomial time in case the data possesses the σ-Separatedness property (with required σ being small numbers, usually 0.01 or lower). The above theorem establishes conditions to what extent the random projection keeps this property so that the special algorithms are still applicable to the projected data.

3.3 (c, σ)-approximation-stability in the projected space

Let us turn to the (c, σ)-Approximation-Stability property. We can reformulate it as follows: If the distance (in some space) between the partitions is more than σ, then the cost function values of two partitions differ by more than the factor $c > 1$. That is, $d(\mathfrak{C}_a, \mathfrak{C}_b) > \sigma$ implies either $c \cdot \mathfrak{J}(Q, \mathfrak{C}_a) < \mathfrak{J}(Q, \mathfrak{C}_b)$ or $c \cdot \mathfrak{J}(Q, \mathfrak{C}_b) < \mathfrak{J}(Q, \mathfrak{C}_a)$).

So assume we have the (c, σ)-Approximation-Stability property in the original space. Consider two partitions $\mathfrak{C}_1, \mathfrak{C}_2$ with $d(\mathfrak{C}_1, \mathfrak{C}_2) > \sigma$. Without loss of generality, assume that therefore in the original space the following holds:

$$\mathfrak{J}(Q, \mathfrak{C}_1) > c \cdot \mathfrak{J}(Q, \mathfrak{C}_2)$$

By applying Theorem 2.3 to both clusterings $\mathfrak{C}_1, \mathfrak{C}_2$, we get

$$(1-\delta)^{-1}\frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}_1) > c \cdot (1+\delta)^{-1}\frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}_2)$$

and hence:

$$\mathfrak{J}(Q', \mathfrak{C}_1) > \left(c \cdot \frac{1-\delta}{1+\delta}\right)\mathfrak{J}(Q', \mathfrak{C}_2)$$

This result holds for any two clusterings $\mathfrak{C}_1, \mathfrak{C}_2$. So if $c \cdot \frac{1-\delta}{1+\delta} > 1$, that is $c > \frac{1+\delta}{1-\delta}$, then we have $\left(c \cdot \frac{1-\delta}{1+\delta}, \sigma\right)$-Approximation-Stability in the projected space, with appropriate probability. Thus, we have

Theorem 3.2 *Under the assumptions and notation of Theorem 2.3, if the data set Q has the property of (c, σ)-Approximation-Stability in the original space, and $c > \frac{1+\delta}{1-\delta}$, then it has the property of $\left(c \cdot \frac{1-\delta}{1+\delta}, \sigma\right)$-Approximation-Stability property in the projected space with probability at least $1 - \epsilon$.*

Recall that Balcan et al. [8] developed a series of algorithms, including one for k-means (see their Lemma 4.1) that solve the clustering problem efficiently for data with (c, σ)-Approximation-Stability. The above theorem states under what conditions the very same algorithms are applicable to projected space. Note, however, the probabilistic nature of this stability. Note also that the property can be lost altogether if c is too small.

3.4 s-multiplicative perturbation robustness in the projected space

Let us now consider s-Multiplicative Perturbation Robustness. First, we need two auxiliary results. The first one concerns transitiveness of this kind of robustness. We claim that:

Lemma 3.3 *For a set Q of representatives of objects from Ω with the distance function d_1, define the set Q_p of representatives of the very same Ω with distance function d_2 as ν-(multiplicative) perturbation ($0 < \nu < 1$) of Q iff $\nu d_1 \leq d_2 \leq \frac{1}{\nu} d_1$.*

If the data set Q has the property of s-Multiplicative Perturbation Robustness under the distance d_1, and the set Q_p is its ν-perturbation with distance d_2 and $s = \nu \cdot s_p$, where $0 < \nu, s_p < 1$, then set Q_p has the property of s_p-Multiplicative Perturbation Robustness.

Proof Q_p is a ν-perturbation of Q and as $\nu > s$, it is also an s-perturbation of Q, therefore by definition of Multiplicative Perturbation Robustness, both share same optimal clustering. Let Q_q be an s_p-perturbation of Q_p, that is one with distance d_3, such that $s_p d_2 \leq d_3 \leq \frac{1}{s_p} d_2$.

Then, $sd_1 = s_p \nu d_1 \leq s_p d_2 \leq d_3 \leq \frac{1}{s_p} d_2 \leq \frac{1}{s_p \nu} d_1 = \frac{1}{s} d_1$, that is Q_q is a perturbation of Q such that both share same optimal clustering. So Q_p and Q_q share common optimal clustering, hence Q_p has the property of s_p-Multiplicative Perturbation Robustness □

Let us note here in passing that Additive Perturbation Robustness implies Multiplicative Perturbation Robustness. Consider a data set Q which has the property of s-Additive Perturbation Robustness. Let $l = \max_{\mathbf{u},\mathbf{v} \in Q} \|\mathbf{u} - \mathbf{v}\|$. Then, for $\mathbf{u} \neq \mathbf{v}; \mathbf{u}, \mathbf{v} \in Q$ $\|\mathbf{u} - \mathbf{v}\| - s = \|\mathbf{u} - \mathbf{v}\|(1 - \frac{s}{\|\mathbf{u}-\mathbf{v}\|}) < \|\mathbf{u} - \mathbf{v}\| \cdot (1 - \frac{s}{l+s}) = \|\mathbf{u} - \mathbf{v}\| \cdot \frac{l}{l+s} < \|\mathbf{u} - \mathbf{v}\| < \|\mathbf{u} - \mathbf{v}\| \cdot \frac{l+s}{l} = \|\mathbf{u} - \mathbf{v}\| \cdot (1 + \frac{s}{l}) \leq \|\mathbf{u} - \mathbf{v}\|(1 + \frac{s}{\|\mathbf{u}-\mathbf{v}\|}) = \|\mathbf{u} - \mathbf{v}\| + s$.

We can summarize this result via

Lemma 3.4 *s-Additive Perturbation Robustness of a data set Q implies $\frac{l}{l+s}$-Multiplicative Perturbation Robustness, where $l = \max_{\mathbf{u},\mathbf{v} \in Q} \|\mathbf{u} - \mathbf{v}\|$.*

The second lemma, that is needed to prove the Theorem 3.6 on Multiplicative Robustness, relates the global minima of the original and projected space when multiplicative perturbation robustness is present in the data. We claim that:

Lemma 3.5 *Under the assumptions and notation of Theorem 2.3, if the data set Q has the property of s-Multiplicative Perturbation Robustness with $s^2 < 1 - \delta$, and if $\mathfrak{C}_{\mathfrak{G}}$ is the global optimum of k-means in Q, then it is also the global optimum in Q' with probability at least $1 - \epsilon$*

Proof Assume to the contrary that is that in Q' some other clustering $\mathfrak{C}'_{\mathfrak{G}}$ is the global optimum. Let us define the distance $d_1(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ and $d_2(i, j) = \frac{n}{n'} \|\mathbf{x'}_i - \mathbf{x'}_j\|$. The distance d_2 is a realistic distance in the coordinate system $\mathcal{C}$ as we assume $n > n'$. As the k-means optimum does not change under rescaling, so $\mathfrak{C}'_{\mathfrak{G}}$ is also an optimal solution for clustering task under d_2. But

$$\begin{aligned} s^2 d_1^2(i, j) &< (1 - \delta) d_1^2(i, j) \leq d_2^2(i, j) \\ &\leq (1 + \delta) d_1^2(i, j) < (1 - \delta)^{-1} d_1^2(i, j) < s^{-2} d_1^2(i, j) \end{aligned}$$

hence the distance d_2 is a s-multiplicative perturbation of d_1 and hence $\mathfrak{C}_{\mathfrak{G}}$ should be optimal under d_2 also, as we assumed robustness of s-multiplicative perturbation of the data set. Thus, we arrived at a contradiction and the claim of the lemma follows. □

This implies that

Theorem 3.6 *Under the assumptions and notation of Theorem 2.3, if the data set Q has the property of s-Multiplicative Perturbation Robustness with factor $s^2 < s_p^2 \nu \frac{(1-\delta)^2}{1+\delta}$ ($0 < s_p, \nu < 1$) in the original space, then with probability at least $1 - 2\epsilon$ it has the property of s_p-Multiplicative Perturbation Robustness in the projected space.*

Proof Let Q' be the projection of the data set Q. In order to demonstrate that Q' has the property s_p-Multiplicative Perturbation Robustness, we need to show that for each s_p-perturbation of Q' this perturbation has the same optimum as Q'. Obviously, any data set in the projected space, so also each perturbation of Q', is a projection of some set from the original space. So we take any set Q_o from the original space and look at its projection Q'_o. If Q'_o happens to be an s_p-perturbation of Q', then we show that Q_o is an s-perturbation of Q. We demonstrate that, due to Q's s-robustness, Q_o, Q'_o and Q' have the same optima. As this holds for all s_p perturbations of Q', Q' is s_p-robust. In detail, we proceed as follows:

As $s^2 \leq 1-\delta$, Lemma 3.5 implies that the global optima of Q in the original and Q' in projected spaces are identical. So assume that in the original space for the distance $d_1(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|$ $\mathfrak{C}_{\mathfrak{G}}$ is the optimal clustering. Then, under projection $d_1'(i,j) = \|\mathbf{x'}_i - \mathbf{x'}_j\|$ we have the same optimal clustering.

Furthermore, let the set Q_o with elements $\mathbf{y}_i$, $i = 1, \ldots, m$ be another representation of the set $\mathfrak{Q}$ in the original space. Let the set Q_o' be its image under the very same projection that was applied to Q and let this projection keep the error range δ as well. The probability, that something like this happens, amounts to $1-2\epsilon$ as we have now twice as many projected points as was originally considered when calculating n' for Q. $d_2(i,j) = \|\mathbf{y}_i - \mathbf{y}_j\|$ and for the projected points of Q_o' we have $d_2'(i,j) = \|\mathbf{y'}_i - \mathbf{y'}_j\|$. Let it happen that Q_o' is an s_p-perturbation of Q'.

Then, $(1+\delta)^{-1}\frac{n}{n'}d_2'^2(i,j) \leq d_2^2(i,j) \leq (1-\delta)^{-1}\frac{n}{n'}d_2'^2(i,j)$ holds. As $s_p d_1'(i,j) \leq d_2'(i,j) \leq (s_p)^{-1}d_1'(i,j)$ and $(1-\delta)d_1^2(i,j) \leq \frac{n}{n'}d_1'^2(i,j) \leq (1+\delta)d_1^2(i,j)$, we obtain

$$
\begin{aligned}
s^2 d_1^2(i,j) &< (1+\delta)^{-1}s_p^2(1-\delta)d_1^2(i,j) \leq (1+\delta)^{-1}s_p^2\frac{n}{n'}d_1'^2(i,j) \leq (1+\delta)^{-1}\frac{n}{n'}d_2'^2(i,j) \\
&\leq d_2^2(i,j) \leq (1-\delta)^{-1}\frac{n}{n'}d_2'^2(i,j) \\
&\leq (1-\delta)^{-1}s_p^{-2}\frac{n}{n'}d_1'^2(i,j) \leq (1-\delta)^{-1}s_p^{-2}(1+\delta)d_1^2(i,j) < \frac{1}{s^2}d_1^2(i,j)
\end{aligned}
$$

So d_2 is a perturbation of d_1 with the factor s.

As Q with d_1 is s-multiplicative perturbation robust, therefore (by definition of multiplicative perturbation robustness) both Q and Q_o have the same optimal clustering $\mathfrak{C}_{\mathfrak{G}}$. What is more, the above derivation shows also that Q_o with d_2 is a $s_p\sqrt{\frac{1-\delta}{1+\delta}}$-perturbation of Q with d_1. As Q with d_1 is s-multiplicative perturbation robust, it is also (by assumption on s) $s_p\sqrt{\nu(1-\delta)\frac{1-\delta}{1+\delta}}$-multiplicative perturbation robust.

Hence, according to Lemma 3.3, Q_o with d_2 has the property of $\sqrt{\nu(1-\delta)}$-Multiplicative Robustness

Therefore, its counterpart Q_o' with d_2' has the same optimum clustering $\mathfrak{C}_{\mathfrak{G}}$ as Q_o with d_2 (see Lemma 3.5). As we already showed, Q_o has the same optimal clustering as Q, Q—same as Q', so Q_o' has the same optimal clustering as Q'.

Note that for any s_p-perturbation Q_o' of Q', there exists a Q_o in the original space such that Q_o' is its image under the projection. And it turned out that it yields the same optimal solution as d_1' for Q'. So with high probability (factor 2 is taken as we deal with two data sets, comprising points $\mathbf{x}_i$ and $\mathbf{y}_i$), Q' with d_1' possesses s_p-Multiplicative Perturbation Robustness in the projected space. □

Let us note at this point that Balcan et al. [9] developed a special polynomial clustering algorithm suitable among others for k-means exploiting the Multiplicative Perturbation Robustness. The above theorem shows that with careful choice of dimensionality reduction we can uphold applicability of such algorithms. Also via Lemma 3.4 it is possible to extend these results to Additive Perturbation Robustness.

3.5 β-centre- and $(1+\beta)$-weak-deletion-stability in the projected space

Let us discuss now two remaining clusterability properties, β-Centre-Stability and $1+\beta$-Weak-Deletion-Stability. They differ substantially from the previously discussed ones, if

we look from the perspective of k-means under projection. The former allowed to establish a kind of link between the clustering in the original space and the projected space. We were able, for example, to say for Multiplicative Perturbation Robustness, when the optimal clusterings in the original and in the projected spaces are identical. σ-Separatedness dealt with optimal clustering costs for clustering into k and $k-1$ clusters and we knew from Theorem 2.6 what was the relationship between optimal clustering costs in the original and the projected spaces. In the case of c, σ-Approximation-Stability, we considered all the possible clusterings, not just the optimal ones. In these two types of stability that we shall consider now, we have to handle the optimal clusterings explicitly, and we do not have a possibility to derive a relationship between the optimal clusterings of the original and the projected space from the stability property alone. Hence, we need some additional knowledge about the optimal clusterings. We have chosen here the Multiplicative Perturbation Robustness, as it establishes a straightforward relation between the optimal clusterings in both spaces. Hence, our formulation of the subsequent results.

We claim that:

Theorem 3.7 *Under the assumptions and notation of Theorem 2.3, if the data set Q has both the property of β-Centre Stability and s-Multiplicative Perturbation Robustness with $s^2 < 1-\delta$ in the original space, then with probability at least $1-\epsilon$ it has the property of $\beta\sqrt{\frac{1-\delta}{1+\delta}}$-Centre Stability in the projected space.*

Proof The s-Multiplicative Perturbation Robustness ensures that both the original and the projected space share same optimal clustering $\mathfrak{C}$ (see Lemma 3.5). To prove the claim, we need hence to explore for each data point to what extent the distance to its own cluster centre will relatively increase while the distance to the other cluster centres will decrease upon projection. When considering the relationship of a point to its own cluster centre, we will use the same technique as in the proof of Lemma B.1. When considering the relationship of a point to some other cluster centre, we will proceed as if we would merge the point with the other cluster, and so the relationship of Lemma B.1 applies again to the extended cluster and we need only to explore the relationship between the centre of the other cluster and the centre of the extended cluster.

Consider a data point $\mathbf{x}_i$ and a cluster $C \in \mathfrak{C}$ not containing i. Then, $\mathbf{x}_i, \boldsymbol{\mu}(C)$ and $\boldsymbol{\mu}(C \cup \{i\})$ are co-linear. So are $\mathbf{x'}_i$, $\boldsymbol{\mu}'(C)$ and $\boldsymbol{\mu}'(C \cup \{i\})$, that is the respective (linear) projections. Furthermore, $\frac{\|\mathbf{x}_i - \boldsymbol{\mu}(C\cup\{i\})\|}{\|\boldsymbol{\mu}(C) - \boldsymbol{\mu}(C\cup\{i\})\|} = \frac{|C|}{1}$, hence

$$\|\mathbf{x}_i - \boldsymbol{\mu}(C)\| = \frac{|C|+1}{|C|}\|\mathbf{x}_i - \boldsymbol{\mu}(C \cup \{i\})\| \tag{17}$$

Likewise, in the projected space, $\frac{\|\mathbf{x'}_i - \boldsymbol{\mu}'(C\cup\{i\})\|}{\|\boldsymbol{\mu}'(C) - \boldsymbol{\mu}'(C\cup\{i\})\|} = \frac{|C|}{1}$, hence

$$\|\mathbf{x'}_i - \boldsymbol{\mu}'(C)\| = \frac{|C|+1}{|C|}\|\mathbf{x'}_i - \boldsymbol{\mu}'(C \cup \{i\})\| \tag{18}$$

Upon projection, the distance to own cluster centre can increase relatively by $\sqrt{1+\delta}$ and to the $C \cup \{i\}$ centre can decrease by $\sqrt{1-\delta}$, see Lemma B.1. That means

$$\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq (1+\delta)\frac{n'}{n}\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \tag{19}$$

and

$$\|\mathbf{x'}_i - \boldsymbol{\mu}'(C \cup \{i\})\|^2 \geq (1-\delta)\frac{n'}{n}\|\mathbf{x}_i - \boldsymbol{\mu}(C \cup \{i\})\|^2 \tag{20}$$

Due to the aforementioned relations, that is if we multiply both sides of (20) with $\frac{|C|+1}{|C|}$ substitute (18) on the left-hand side and (17) on the right-hand side into the relation (20), we will obtain

$$\|\mathbf{x}'_i - \boldsymbol{\mu}'(C)\|^2 \geq (1-\delta)\frac{n'}{n}\|\mathbf{x}_i - \boldsymbol{\mu}(C)\|^2 \tag{21}$$

Due to β-Centre-Stability in the original space we have: $\beta^2\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 < \|\mathbf{x}_i - \boldsymbol{\mu}(C)\|^2$. Hence, by substituting on the right-hand side of (21), we get

$$\|\mathbf{x}'_i - \boldsymbol{\mu}'(C)\|^2 > \beta^2(1-\delta)\frac{n'}{n}\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 = \beta^2(1-\delta)\frac{1+\delta}{1+\delta}\frac{n'}{n}\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \tag{22}$$

By substituting with (19) on the right-hand side of (22), we get

$$\|\mathbf{x}'_i - \boldsymbol{\mu}'(C)\|^2 > \beta^2\frac{1-\delta}{1+\delta}\|\mathbf{x}'_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \tag{23}$$

That is

$$\|\mathbf{x}'_i - \boldsymbol{\mu}'(C)\| > \beta\sqrt{\frac{1-\delta}{1+\delta}}\|\mathbf{x}'_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|$$

Hence, the data centre stability can drop to $\beta\sqrt{\frac{1-\delta}{1+\delta}}$. □

Awasthi et al. [7] developed algorithms to find optimal k-means clustering in polynomial time if the data fits the requirements of β-Centre Stability. The results of the above theorem indicate to what extent their algorithms can be applied to randomly projected data given the original data fit the requirements.

β-Centre Stability implies that in the optimal solution each data point preserves some proportion of distances to the neighbouring cluster centres. Awasthi et al. considered also somewhat weakened condition in that a constraint is imposed on cost function under deletion of a cluster centre. Also in this case, we could find conditions when random projection upholds the new deletion-based stability condition.

We claim that:

Theorem 3.8 *Under the assumptions and notation of Theorem 2.3, if the data set Q has both the property of $(1+\beta)$ Weak Deletion Stability and s-Multiplicative Perturbation Robustness with $s^2 < 1-\delta$ in the original space, then with probability at least $1-\epsilon$ it has the property of $(1+\beta)\frac{1-\delta}{1+\delta}$ Weak Deletion Stability in the projected space.*

Proof The s-Multiplicative Perturbation Robustness ensures that both the original and the projected space share same optimal clustering (see Lemma 3.5). Let this optimal clustering be called $\mathfrak{C}_o$. By $\mathfrak{C}$, denote any clustering obtained from $\mathfrak{C}_o$ by deletion of one cluster centre and assigning cluster elements to one of the remaining clusters.

Theorem 2.3 implies that $(1-\delta)\frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}) \leq \mathfrak{J}(Q',\mathfrak{C})$. Therefore,

$$\mathfrak{J}(Q',\mathfrak{C}) \geq (1-\delta)\frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C})$$

By the assumption of $(1+\beta)$-Weak Deletion stability $(1+\beta)\mathfrak{J}(Q,\mathfrak{C}_o) \leq \mathfrak{J}(Q,\mathfrak{C})$. Therefore,

$$(1-\delta)\frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}) \geq (1+\beta)(1-\delta)\frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}_o)$$

Theorem 2.3 implies that $(1+\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}_o) \leq \frac{n'}{n}\mathfrak{J}(Q, \mathfrak{C}_o)$. Hence,

$$(1+\beta)(1-\delta)\frac{n'}{n}\mathfrak{J}(Q, \mathfrak{C}_o) \geq (1+\beta)(1-\delta)(1+\delta)^{-1}\mathfrak{J}(Q', \mathfrak{C}_o)$$

So we can conclude that

$$\mathfrak{J}(Q', \mathfrak{C}) \geq \left((1+\beta)(1-\delta)(1+\delta)^{-1}\right)\mathfrak{J}(Q', \mathfrak{C}_o)$$

which implies the claim. □

Awasthi et al. [6] have demonstrated among others for k-means that the data having the property of $(1+\beta)$ Weak Deletion Stability possess also the so-called PTAS property (existence of Polynomial-time approximation scheme). The above theorem states conditions under which this property is preserved under random projection.

Summarizing this section we can say that the random projection, under suitable conditions, may preserve at least several clusterability properties, known in the literature, and thus conditions may be identified when efficient clustering, according to k-means cost function, is applicable in the projected space if it is applicable in the original space. So not only distance relations, but also clusterability can be maintained under projection according to JL Lemma.

4 Experiments

In the experimental part of this work, in a series of numerical experiments, we want to demonstrate the validity of various aspects of JL Theorem applied to projected data from the perspective of k-means algorithm.

4.1 Numerical experiments on importance of differences between various dimensionality reduction formulas

It is a frequently stated question that to what extent the concrete formula for dimensionality reduction overshoots the real need for embedding dimensions. In order to give an idea how effective the random projection is, see Fig. 4. Figure 4 illustrates a typical distribution of distortions of distances between pairs of points in the projected space for one of the runs characterized by figure caption. It illustrates the distribution of discrepancies between squared distances in the projected and in the original spaces. The distortions are expressed as

$$\frac{\|f(\mathbf{u}) - f(\mathbf{v})\|^2}{\|\mathbf{u} - \mathbf{v}\|^2}$$

One can see that they correspond quite well to the imposed constraints. The vast majority of point pairs have a distortion much lower than δ. There exist, however, sufficiently many pairs for which the distortion is close to δ (in terms of the order of magnitude); therefore, one can assume that not much more can be gained.

Another important question related to δ is its relation to k-means clustering under projection. Figure 5 illustrates the role of the intrinsic gap between clusters in the original space and the permitted value of δ. As one would expect, the bigger the relative gap between clusters, the larger the error value δ is permitted, if class membership shall not be distorted by the projection.

Finally, when discussing the various formulas on dimensionality reduction under random projection, the question may be raised whether or not, under realistic values of the parameters

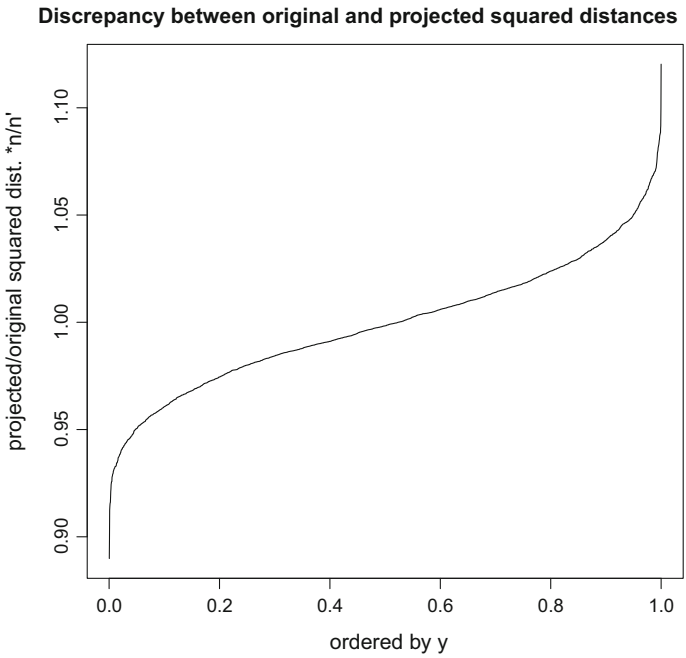

Fig. 4 Discrepancy between projected and original squared distances between points in the sample expressed as their quotient adjusted by n/n'. Parameters fixed at m= 5000 $\epsilon = 0.1$ $\delta = 0.2$ $n = 5000$ $n' = 2188$

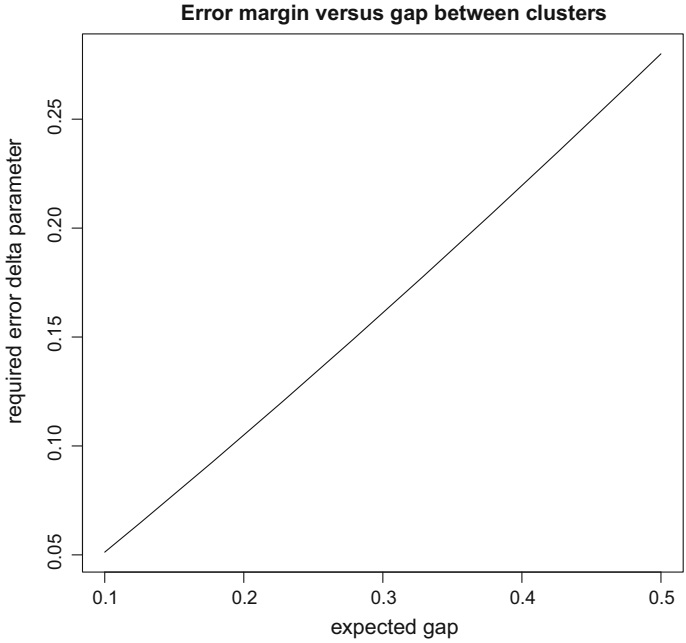

Fig. 5 Permissible error range δ under various assumed gaps between the clusters

Table 1 Dependence of reduced dimensionality n' on sample size m. Other parameters fixed at $\epsilon = 0.01\ \delta = 0.05\ n = 5e{+}05$

m	n'_E	n'_I	n'_E/n'_I	n'_G	n'_I/n'_G	r
10	15,226	14,209	1.07	3879	3.7	44
20	17,518	16,389	1.07	5046	3.3	90
50	20,547	19,191	1.07	6589	3	228
100	22,839	21,269	1.07	7757	2.8	459
200	25,131	23,323	1.08	8924	2.7	919
500	28,160	26,016	1.08	10467	2.5	2301
1000	30,452	28,030	1.09	11,635	2.5	4603
2000	32,744	30,027	1.09	12,802	2.4	9209
5000	35,773	32,648	1.1	14,345	2.3	23,024
10,000	38,065	34,609	1.1	15,513	2.3	46,050
20,000	40,357	36,554	1.1	16,680	2.2	92,102
50,000	43,386	39,097	1.11	18,223	2.2	230,257
1e+05	45,678	41,017	1.11	19,391	2.2	460,515
2e+05	47,970	42,910	1.12	20,558	2.1	921,032
5e+05	50,999	45,392	1.12	22,101	2.1	2,302,583
1e+06	53,291	47,250	1.13	23,269	2.1	4,605,168
2e+06	55,582	49,099	1.13	24,436	2.1	9,210,339
5e+06	58,612	51,515	1.14	25,979	2	23,025,849
1e+08	68,516	59,243	1.16	31,025	2	460,517,014
2e+07	63,195	55,127	1.15	28,314	2	92,103,402
5e+07	66,225	57,480	1.15	29,857	2	230,258,508
1e+08	68,516	59,243	1.16	31,025	2	460,517,014

Symbols: n'_I—implicit n', n'_E—explicit n', n'_G—for comparison n' as computed by Dasgupta and Gupta [20], r—the number of repetitions of sampling needed to compute k-means under Dasgupta and Gupta dimensionality reduction approach

δ, ϵ, m and n there is a real advantage of our newly derived formulas of computing n' over the ones provided by the literature, in particular that of Gupta and Dasgupta [20] and whether or not the implicit n' computation gives us an advantage over the explicit n' formula. The practical reason for an interest in getting as low n' as possible is the following: the lower the dimensionality, the lower numerical effort for computing distances between the objects.

We have considered the following value ranges for these parameters: $\delta \in [0.01, 0.5]$, $\epsilon \in [0.001, 0.1]$, $m \in [10, 10^8]$ and $n \in [4 \cdot 10^5, 10 \cdot 10^5]$.

Let us recall that under application of k-means it is vital that we have a high success probability $(1 - \epsilon)$ of selecting a random subspace such that under projection of data points onto this subspace the distortion of distances between pairs of points is at most the assumed δ.

We investigate the behaviour of n'_I versus n'_E and at the ration to $n'_G/'_I$ (n'_G being the reduced dimensionality of Gupta/Dasupta). We also want to know how many times the random projection has to be repeated under Gupta/Dasgupta proposal in order to get the assumed success probability $1 - \epsilon$. We investigated the impact of the original data dimensionality n (Table 4 and Fig. 6), the sample size m (Table 1 and Fig. 2), the accepted error δ (Table 3 and Fig. 1, the assumed failure rate ϵ (Table 2 and Fig. 3). In these experiments, we

Table 2 Dependence of reduced dimensionality n' on failure prob. ϵ. Other parameters fixed at $m = 2\text{e+}06\ \delta = 0.05\ n = 5\text{e+}05$

ϵ	n'_E	n'_I	n'_E/n'_I	n'_G	n'_I/n'_G	r
0.1	51,776	46,020	1.13	24,436	1.9	4,605,170
0.05	52,922	46,955	1.13	24,436	2	5,991,464
0.02	54,437	48,180	1.13	24,436	2	7,824,045
0.01	55,582	49,099	1.13	24,436	2.1	9,210,339
0.005	56,728	50,014	1.13	24,436	2.1	10,596,633
0.002	58,243	51,221	1.14	24,436	2.1	12,429,214
0.001	59,389	52,134	1.14	24,436	2.2	13,815,508

Symbols: n'_I—implicit n', n'_E—explicit n', n'_G—for comparison n' as computed by Dasgupta and Gupta [20], r—the number of repetitions of sampling needed to compute k-means under Dasgupta and Gupta dimensionality reduction approach

Table 3 Dependence of reduced dimensionality n' on error range δ. Other parameters fixed at $m = 2\text{e+}06\ \epsilon = 0.01\ n = 5\text{e+}05$

δ	n'_E	n'_I	n'_E/n'_I	n'_G	n'_I/n'_G	r
0.5	712	697	1.02	465	1.5	9,210,339
0.4	1059	1032	1.03	605	1.8	9,210,339
0.3	1787	1745	1.02	922	1.9	9,210,339
0.2	3804	3692	1.03	1814	2.1	9,210,339
0.1	14,339	13,640	1.05	6449	2.2	9,210,339
0.09	17,593	16,631	1.06	7874	2.2	9,210,339
0.08	22,128	20,742	1.07	9857	2.2	9,210,339
0.07	28,721	26,604	1.08	12,736	2.1	9,210,339
0.06	38,846	35,329	1.1	17,150	2.1	9,210,339
0.05	55,582	49,099	1.13	24,436	2.1	9,210,339
0.04	86,291	72,387	1.19	37,783	2	9,210,339
0.03	152,415	115,298	1.32	66,478	1.8	9,210,339
0.02	340,701	201,059	1.69	148,048	1.4	9,210,339
0.01	1,353,858	1,353,859	1	586,209	2.4	9,210,339

Symbols: n'_I—implicit n', n'_E—explicit n', n'_G—for comparison n' as computed by Dasgupta and Gupta [20], r—the number of repetitions of sampling needed to compute k-means under Dasgupta and Gupta dimensionality reduction approach

investigated only the theoretical values (checking for low sample sizes (below 1000) and low dimensionality (below 50,000) whether or not the values are confirmed in multiple (10) simulation runs—they were confirmed on each run so no extra reporting is done). We did not experiment whether lower values of n' than those suggested by our formulas on n'_E, n'_I and their corresponding n'_G would be sufficient, though it is a good subject for further research.

Note that we have two formulas for computing the reduced space dimensionality n', the formula (7) for n'_I and (4) for n'_E. The latter does not engage the original dimensionality n, while it is explicit in n'. The value of n' in the former depends on n, however n' can be only computed iteratively.

Table 4 Dependence of reduced dimensionality n' on original dimensionality n. Other parameters fixed at $m = 2e{+}06\ \epsilon = 0.01\ \delta = 0.05$

n	n'_E	n'_I	n'_E/n'_I	n'_G	n'_I/n'_G	r
4e+05	55,582	47,891	1.16	24,436	2	9,210,339
5e+05	55,582	49,099	1.13	24,436	2.1	9,210,339
6e+05	55,582	49,933	1.11	24,436	2.1	9,210,339
7e+05	55,582	50,551	1.1	24,436	2.1	9,210,339
8e+05	55,582	51,025	1.09	24,436	2.1	9,210,339
9e+05	55,582	51,399	1.08	24,436	2.2	9,210,339
1e+06	55,582	51,703	1.08	24,436	2.2	9,210,339

Symbols: n'_I—implicit n', n'_E—explicit n', n'_G—for comparison n' as computed by Dasgupta and Gupta [20], r—the number of repetitions of sampling needed to compute k-means under Dasgupta and Gupta dimensionality reduction approach

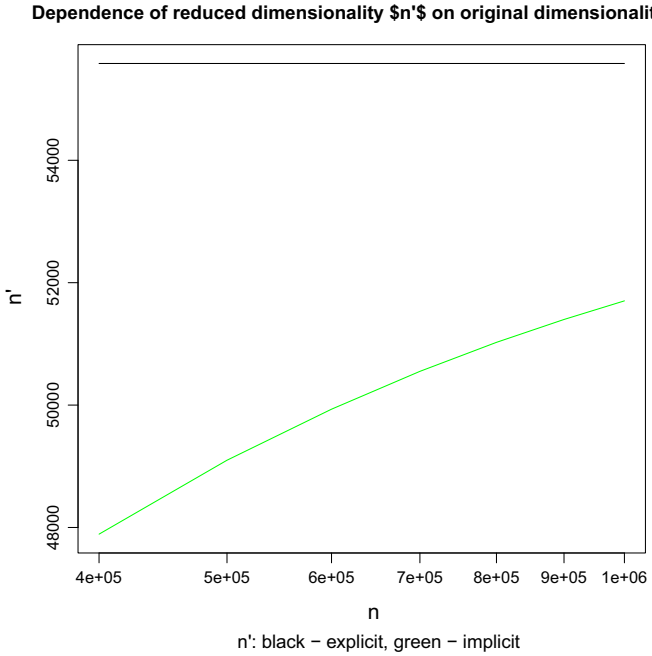

Fig. 6 Dependence of reduced dimensionality n' on original dimensionality n. Other parameters fixed at $m = 2e{+}06\ \epsilon = 0.01\ \delta = 0.05$ (color figure online)

The content of the tables indicates the limitations of dimensionality reduction via JL . There is no point of applying dimensionality reduction via JL Lemma if the dimensionality lies below 1000 (for $\delta < 0.1, \epsilon < 0.5, m = 10$).

Let us investigate the differences between n' computation in implicit and explicit cases. Let us check the impact of the following parameters: n—the original dimensionality (see Table 4 and Fig. 6), δ—the limitation of deviation of the distances between data points in the original and the reduced space (see Table 3 and Fig. 1), m—the sample size (see Table 1 and

Fig. 2), as well as ϵ—the maximum failure probability of the JL transformation (see Table 2 and Fig. 3). Note that in all figures, the X-axis is on log scale.

Concerning the differences between n'_E and n'_I , we see from Table 1 (see also Fig. 2), that increase in sample size in reasonable range 10–10^8 increases the advantage of n'_I over n'_E to even 15%. On the other hand, in Table 2 (see also Fig. 3) we see that the failure probability ϵ does not give a particular advantage to any of these dimensionality sizes, which may be partially attributed to the considered sample size m, though the implicit one approaches the explicit one with increase in ϵ. We see also that when we increase the acceptable failure rate ϵ, the requested dimensionality n' drops.

The decrease in error rate δ, as visible from Table 3 (see also Fig. 1), increases the gap between n'_E and n'_I up to the point when the original dimensionality n is exceeded and hence usage of dimensionality "reduction" is pointless (see the last line of Table 3). Figure 1 shows that the requested dimensionality drops quite quickly with increased relative error range δ till a kind of saturation is achieved.

As visible in Fig. 6, the value of n' from the explicit formula does not depend on the original dimensionality n, while the implicit one does. From Table 4 (see also Fig. 6), we see that the increase in the original dimensionality gradually reduces the advantage of n'_I over n'_E. In fact, the value computed from the implicit formula approaches the explicit value with the growing dimensionality n.

On the other hand, the implicit n' departs from the explicit one with growing sample size m, as visible in Fig. 2. Both grow with increasing m.

In summary, these differences are not very big, but nonetheless can be of significant advantage when the computations over large data sets are likely to have long run times.

The behaviour of explicit n' is not surprising, as it is visible directly from the formula (4). The important insight here is, however, the required dimensionality of the projected data, of hundreds of thousands for realistic ϵ, δ. Thus, the random projection via the Johnson–Lindenstrauss Lemma is not yet another dimensionality reduction technique. It is suitable for very large data only, and it proved to be a useful ingredient to techniques such as PCA, see, e.g. the so-called compressive PCA [37].

The behaviour of implicit n' for the case of increasing original dimensionality n is as expected—the explicit n' reflects the "in the limit" behaviour of the implicit formulation. The discrepancy for ϵ and the divergence for growing m indicate that there is still space for better explicit formulas on n'. In particular, it is worth investigating for increasing m as the processing becomes more expensive in the original space when m is increasing.

With regard to n'_I/n'_G quotient, one shall stress that n'_G has always numerically a clear advantage, of up to 400%, but one shall take into account that the warranty of obtaining a useful projection from the point of view of k-means is low according to theoretical considerations. Hence, while we can perform random projection only once for our dimensionality reduction method, the $Dasgupta/Gupta$ projection needs to be repeated for a multitude of times (r column in the tables). So in Table 1, the number of needed repetitions r is already 44 in the most advantageous case of n'_I/n'_G. with increase in sample size m the quotient falls down to 100% advantage, while r increases radically to hundreds of millions. This fact renders $Dasgupta/Gupta$ projection useless. The same disadvantage of $Dasgupta/Gupta$ projection is visible in other tables. However, note that according to Table 2 with decrease in ϵ the advantage of n'_G over n'_I raises from 90 to 120%. However, the increase in r is disproportional with respect to this advantage. The increase in original dimensionality n gives advantage to the value of n'_G. Error range δ does not exhibit such an obvious pattern when influencing the quotient.

4.2 Impact of projection on correctness of clustering

In this series of experiments, the validity of Theorem 2.6 along with Theorem 2.3 was verified.

The problem with an experimental investigation results from the fact that k-means and its variants are known to usually stick at local minima (at the cost of speed), especially in the high-dimensional landscape, and our Theorem 2.6 relates to the global optimum. So for an arbitrary data set, we would not know the optimum in advance, neither from experimental runs nor from theoretical considerations. Therefore, we created a special generator, providing with well-separated clusters in some sense for which we knew the theoretical solution in the original space (being a parameter of the generator). The solution in the projected space was not known in advance, and it was considered to be the same as in the original space due to theorems on perturbative robustness, and k-means and k-means++ were run in order to discover it experimentally. The experiment was considered as a failure when either the deemed clustering was not discovered or the inequalities of the theorems were violated.

A series of experiments consisting in generating samples of n-dimensional data ($n = 4900$) consisting of m records ($m = 5000$), which had a cluster structure known in advance, was performed. In order to know the clustering in the original space in advance, the sample generator proceeded as follows: m points from n-dimensional normal distribution centred at zero, with unit standard deviation and zero correlation between dimensions were generated. Then, in a random manner, these data points were assigned to the desired number of clusters so that sizes of any two clusters differ only by at most 1. Then, the required distance between balls enclosing the clusters was computed. The required distance was set to twice the largest intrinsic cluster radius (measured from the centre to the most exterior cluster element) multiplied by square root of the number of clusters k. Then, each cluster was moved away from the zero point along a different direction (which was possible as $k < n$) by the required distance divided by $\sqrt{2}$. This division by $\sqrt{2}$ ensures that the distances between cluster enclosing balls were equal to the required distance.

The data was projected onto a lower dimensional subspace with dimensionality according to the formula (4), and k-means clustering procedure was executed both in the original space and in the target space. k-means was executed at least $k/2$ times up to $k^2/2$ times till an agreement with the true clustering was obtained. If the true clustering was not obtained, k-means++ was applied up to k times. If still there was a disagreement, it was checked whether the total within sum of squares of the intrinsic clustering was lower than the one obtained from k-means++. If it were not the case in the projected space, then it would mean a failure of the theory. It would be a failure because it would mean that k-means++ was able to find a better clustering than one predicted by our theoretical considerations.

Multi-start k-means was applied because the algorithm is known to get stuck in local optima, while we were checking whether the global minimum is achieved. k-means++ is known to have guaranteed vicinity to the intrinsic optimum (while k-means does not) though it is more time-consuming than k-means. It turned out that for larger values of $k \geq 9$, the k-means++ had to be called always to get the intrinsic clustering, while for $k = 2$ one or more calls of k-means were sufficient.

Counts of complete and incomplete matches of clusterings in the original scape and in the projected space have been performed for various numbers of clusters $k = 2, 3, 9, 81, 243, 729$. The other parameter of the experiments, ϵ, was fixed at 0.05, which implied $\delta \approx 0.23$ and $n_p = 1732$.

The results are presented in Table 5. It is clearly visible that the probability of not violating projection cost constraints is even higher than the respective theorem predicts.

Table 5 Impact of projection on correctness of clustering for 100 runs

$k =$	2	3	9	81	243	729
No. of disagreements between clusterings of X and X'	0	0	0	0	0	0
No. of violation of relationship (8)	0	0	0	0	0	0

Table 6 Impact of projection on Multiplicative Perturbation Stability for 100 runs

$k =$	2	3	9	81	243	729
No. of disagreements between clusterings of X and Y	0	0	0	0	0	0
No. of disagreements between clusterings of X' and Y'	0	0	0	0	0	0
No. of violation of formulas of Theorem 3.6	0	0	0	0	0	0

4.3 Impact of projection onto multiplicative perturbation stability

The experimental set-up was exactly the same as in Sect. 4.2 except that now multiplicative perturbations were performed. So we had an original data set X, that was projected resulting into the set X'. Furthermore, we had a set Y being a multiplicative perturbation of X and a set Y' being a multiplicative perturbation of X'. We clustered each of them as indicated in the previous section via k-means or k-means++ and checked if all these clusterings agree with the intrinsic one.

We assumed that the maximal multiplicative perturbation permissible s is the one not violating the criterion on knowing in advance the intrinsic clustering. Then, we perturbed the elements of X by a randomly selected factor from the permissible range ($1/s$ to s) with respect to the respective cluster centre. This perturbation was effectively bigger than the theoretical value allowed by perturbation definition, nonetheless it is obvious that if the properties hold for stronger perturbation then they will hold also for proper ones. In this way, the set Y was obtained. We computed the actual value of perturbation s_o of Y, by the respective formula we computed maximal theoretically possible perturbation s_p in the projected space and applied this perturbation to X' to obtain the set Y'. We expected that under this perturbation to X' the result of (optimal) clustering should be the same.

The results are presented in Table 6. It is clearly visible that the probability of not violating multiplicative perturbation robustness constraints is even higher than the respective theorem predicts.

4.4 Impact of projection onto σ-separatedness

We performed the experiment in the same way as described in 4.3 computing additionally the σ-separation both in X and in X' and checked if they fit the restrictions of the respective theorem.

The results are presented in Table 7.

Table 7 Impact of projection on σ-Separatedness for 100 runs

$k =$	2	3	9	81	243	729
No. of disagreements between clusterings of X and X'	0	0	0	0	0	0
No. of violation of relationship from Theorem 3.1	0	0	0	0	0	0

Table 8 Impact of projection on β-centric Stability for 100 runs

$k =$	2	3	9	81	243	729
No. of disagreements between clusterings of X and X'	0	0	0	0	0	0
No. of violation of relationship from Theorem 3.7	0	0	0	0	0	0

Table 9 Impact of projection on $1+\beta$-Weak Deletion Stability for 100 runs

$k =$	2	3	9	81	243	729
No. of disagreements between clusterings of X and X'	0	0	0	0	0	0
No. of violation of relationship from Theorem 3.8	0	0	0	0	0	0

4.5 Impact of projection onto β-centric Stability

We performed the experiment in the same way as described in 4.3 computing additionally the β-centric Stability both in X and in X' and checked if they fit the restrictions of the respective theorem.

The results are presented in Table 8.

4.6 Impact of projection onto $1+\beta$-weak deletion stability

We performed the experiment in the same way as described in 4.3 computing additionally the $1+\beta$-Weak Deletion Stability both in X and in X' and checked if they fit the restrictions of the respective theorem.

The results are presented in Table 9.

Note that we did not experiment with c, σ-Approximation-Stability because it requires not only a substantially larger set of experiments (not only the optimal clusterings need to be investigated but also all the other), but also generation of samples (for k-means algorithm) exhibiting c, σ-Approximation-Stability is rather tedious because in typical large samples this property is violated (e.g. local minima exist with close cost function and radically different structures).

5 Previous work

As already mentioned in the introduction, there exists a vast number of research papers that explored the consequences of the Johnson–Lindenstrauss Lemma in various dimensions.

We shall focus here on those aspects that are relevant for the research results presented in this paper. The original formulation of JL Lemma was rather existential without the primary goal to apply it to machine learning tasks. The applied research was therefore concentrated around the issue of actual computational burden related to use of random projection. First of all, an attempt was made to reduce maximally the dimensionality of the projected space.

The denominator of the expression for n' of original JL Lemma was δ^2, [25,28,31]. Later, papers suggest $\delta^2 - \delta^3$ [1,20] (decreasing the nominator) so that a slight decrease in the number of dimensions is achieved. We suggest the denominator $-\ln(1+\delta)+\delta$ that reduces the allowed dimensionality slightly more.

Larsen and Nelson [29] concentrate on finding the largest value of n' for which Johnson–Lindenstrauss Lemma does not hold demonstrating that the value they found is tight even for nonlinear mappings f. Though not directly related to our research, they discuss the flip side of the problem, that is the dimensionality below which at least one point of the data set has to violate the constraints.

Kane and Nelson [26] and Cohen et al. [19] pursue a research on Sparse Johnson–Lindenstrauss transform (SJLT). The SJLT deals with the problem that the original JL transform densifies vectors coming from sparse spaces. Also Clarkson et al. [18] are proposing an algorithm for low dimensionality embedding for sparse matrices that has low computational complexity in the number of nonzero entries in the data matrix. This is an interesting research direction for sparse matrices because the traditional random projection usually densifies the projected vectors causing losses to efficiency gained by dimensionality reduction. We do not pursue this problem though the densification may be an issue for versions of clustering algorithms that explicitly address sparse spaces. k-means in its original version densifies in fact the cluster centre vectors. So in fact k-means itself would require some changes.

Note that if we would set ϵ close to 1, and expand by Taylor method the ln function in denominator of the inequality (4) to up to three terms then we get the value of n' from equation (2.1) from the paper [20]:

$$n' \geq 4\frac{\ln m}{\delta^2 - \delta^3}$$

Note, however, that setting ϵ to a value close to 1 does not make sense as we want to keep rare the event that the data does not fit the interval we are imposing.

Though one may be tempted to view our results as formally similar to those of Dasgupta and Gupta, there is one major difference. Let us first recall that the original proof of Johnson and Lindenstrauss [25] is probabilistic, showing that projecting the m -point subset onto a random subspace of $O(\ln m/\epsilon^2)$ dimensions only changes the (squared) distances between points by at most $1-\delta$ with positive probability. Dasgupta and Gupta showed that this probability is at least $1/m$, which is not much indeed. That is, if we pick with their method one random projection, we may fail to obtain the projection with required properties with probability $1-1/m$. For $m = 1000$, we will fail with probability of 99.9%. In order to get failure probability ϵ below say 0.05%, one needs to proceed as follows: repeat the process of picking the random projection r times, r to be specified below. Then, among the resulting r projections, with probability $P_s(r) < 1-\epsilon$, at least one will have the required properties. But we do not know which of the r projections. So for each projection, we need to check whether or not the distances under projection have the desired properties. In our method, we need to pick only one projection and the check is not needed. Let us go over to the estimation of r and of $P_s(r)$. Note that each choice of a random projection is independent of the other. Therefore, the probability $P_f(r)$ of failing to pick a projection with desired properties in

each of the r trials, amounts to $(1-\frac{1}{m})^r$, as $1-\frac{1}{m}$ is the failure probability in a single experiment. (Note that by definition $P_s(r) = 1 - P_f(r)$.) If we want to ensure that $P_f(r) < \epsilon$, that is

$$\left(1-\frac{1}{m}\right)^r < \epsilon \tag{24}$$

we need an $r > \ln(\epsilon)/\ln(1-\frac{1}{m})$.

In case of $m = 1000$, this means over $r = 2995$ repetitions, and with $m = 1,000,000$—over $r = 2,995,000$ repetitions,

In this paper, we have shown that this success probability can be increased to $1-\epsilon$ for an ϵ given in advance. Hereby, the increase in target dimensionality is small enough compared to Dasgupta and Gupta formula, that our random projection method is orders of magnitude more efficient. A detailed comparison is contained in Tables 1, 2, 3, 4 in the last three columns. We compare in these tables n' computed using our formulas with those proposed by Dasgupta and Gupta as well as we present the required number of repetitions of projection onto sampled subspaces in order to obtain a faithful distance discrepancies with reasonable probability. Dasgupta and Gupta generally obtain several times lower number of dimensions. However, as stated in the introduction, the number of repeated samplings nullifies this advantage and in fact a much higher burden when clustering is to be expected.

Note that the choice of n' has been estimated by [1]

$$n' \geq (4+2\gamma)\frac{\ln m}{\delta^2-\delta^3}$$

where γ is some positive number. They propose a projection based on two or three discrete values randomly assigned instead of ones from normal distribution. With the quantity γ, they control the probability that a single element of the set Q leaves the predefined interval $\pm\delta$. However, they do not control the probability that none of the elements leaves the interval of interest. Rather, they derive expected values of various moments.

Though in passing, a similar result to ours is claimed in Lemma 5.3 by Bandeira [10], that is that he requires $n' \geq (2+\tau)\frac{2\ln m}{\delta^2-\delta^3}$. In fact, if one substitutes in (4) the failure probability ϵ with $m^{-\tau}$ and $-\ln(1+\delta)+\delta$ with $\delta^2-\delta^3$, then purely formally we get Bandeira's formula. However:

- Bandeira refrains from proving his formula.
- His lower bound on n' is higher than ours, because $-\ln(1+\delta)+\delta > \delta^2-\delta^3$.
- As he does not investigate the proof of his formulation, he also fails to find still lower bound on n' as we have done investigating the implicit formula for n' in Sect. A.2. We provided possibilities to reduce n' via our implicit formulation of conditions for n' and proving that the implicit function is invertible. As Table 1 shows, for example, the dimensionality reduction may be up to 15%. *We are unaware of this being observed by anyone else*.
- From a practical point of view, Bandeira's formulation is misleading as to the nature of increase in n' with increase in the sample size. The formula above would superficially suggest that it grows linearly with the logarithm of the sample size m, while our formulation clearly shows that this growth is slower when keeping the failure probability ϵ. See, for example, the results in Table 1. For a practical illustration, consider the case of $m = 100$, $\delta = 0.1$, $\epsilon = 0.05$ and accordingly $\tau = 0.65$. Bandeira's n' would amount to 5230. Our explicit n' would be 5205 which are pretty close (though our is lower (by 0.5%), while at the same time their failure probability $m^{-\tau} > 0.0501$ is slightly bigger. If we increase, however, m to say 10,000, while keeping respective parameters of Ban-

deira's relation, he gets $n' = 10,460$, while we require only $n' = 9134$ (12% less), and for $m = 1,000,000$ he needs 15,691 dimensions and we only 13,061 (16% less). In this sense, our formulation is more user-friendly.
- His parameter τ has no practical meaning from the point of view of a user, while our ϵ has a clear-cut semantics.
- He does not draw conclusions with respect to the clustering task and clusterability as done in this paper in our theorems, especially in Theorem 3.6 which rely on the fact that ϵ is explicitly mentioned in the formula for n'. It would be really hard to explain the user of Bandeira's formula what it means to having to double our ϵ.

The publications [11,12] by Baraniuk et al. present a theorem similar in spirit to [10], in a bit different area of transformations with the so-called RIP (restricted isometry property). Our above remarks apply also to those publications, respectively.

Our research was oriented towards applicability of Johnson–Lindenstrauss Lemma to the task of clustering. This application area has been explored already by Schulman [36]. He optimized the intracluster sum of weights in graph clustering and used JL Lemma to reduce the computational time. Recently, a number of papers pursued the research path of applying JL Lemma to improve clustering process for graphs [27]. It is a combination of compressive sampling with spectral clustering [33,34,38,41], spectral compressive principal component analysis [22] and similar approaches. Our research differs from these developments. The graph clustering explores the fact that data items (graph nodes) are interdependent. Therefore, it is possible to explore these dependencies to reduce the number of attributes (they are dependent as they are the nodes of the graph) and at the same time the sample size and hence automatically reduce the dimensionality further (as it depends on the sample size). We considered here only the case of independent data items (objects) and therefore could not benefit from their results. Note also that typical graph-based clustering methods ignore long distances between nodes and their usage of JL Lemma keeps rather distances between eigenvectors of Laplacians of such graph.

While we insisted on avoiding multiple repetitions of projections, Cannings and Samworth [16] explicitly use multiple projections for purposes of classification.

See also Fedoruk et al. [21] for an overview of theoretical and practical bounds for dimensionality reduction via JL Lemma.

6 Conclusions

In this paper, we investigated a novel aspect of the well-known and widely investigated and applied Johnson–Lindenstrauss lemma on the possibility of dimensionality reduction by projection onto a random subspace.

In this paper, we made three main claims:

- JL Lemma can be enhanced in such a way that, in the process of dimensionality reduction, with user-controlled probability, all the projected points keep error bounds;
- the proposed framework can identify the suitable subspace by random projection that preserves the cluster structure of higher dimension in the embedding with some controllable error;
- in the proposed framework, we derived deterioration degrees of a number of clusterability properties under the projection.

With respect to claim one, the original formulation of JL Lemma means in practice that we have to check whether or not we have found a proper transformation f leading to error

bounds within required range for all pairs of points, and if necessary (and it is theoretically necessary very frequently), to repeat the random projection process over and over again.

We have shown here that it is possible to determine in advance the choice of dimensionality in the random projection process as to assure with desired certainty that none of the points of the data set violates restrictions on error bounds. This new formulation, expressed in Theorem 2.1, proven in Sect. A.1 and empirically validated in Sect. 4.1, can be of importance for many data mining applications, such as clustering, where the distortion of distances influences the results in a subtle way (e.g. k-means clustering). Via some numerical examples, we have pointed at the real application areas of this kind of projections, that is problems with high number of dimensions, starting with dozens of thousands and hundreds of thousands of dimensions. Also the superiority of our approach to that of Gupta/Dasgupta was demonstrated by pointing at the computational burden resulting from the need to repeat the projections multiple times.

As the second claim is concerned, we have broadened the analysis of JL Lemma-based random projection for k-means algorithms in that we identified the conditions under which clusterings yielding local minima of k-means objective coincide in the original and the projected spaces, and also conditions when the values of global optima of this objective for the original and projective spaces are close to one another. This has been expressed in Theorems 2.3–2.6, proven in Appendix B. An empirical investigation was performed in the Sect. 4.2.

Additionally, as stated in the third claim, we have formulated Theorems 3.1–3.8 and proved them in Sect. 3 showing, when our reformulation of the JL Lemma permits to uphold five well-known clusterability properties at the projection. An empirical investigation was performed in Sects. 4.3–4.6 on four of them.

The scope of this investigation was restricted in a number of ways. Hence, there exist numerous possibilities to extend the research. First of all, this research and papers of other (e.g. [41,42]) indicate that the JL Lemma-induced dimensionality reduction is too conservative. We have seen this, for example, by comparison of explicit and implicit dimensionality reduction differences. Various empirical studies suggest that the dimensionality could be radically reduced, though no analytical results are yet available.

We restricted ourselves to studying the impact of JL Lemma on k-means clustering algorithm cost function. It may be of interest to study particular brands of k-means algorithms in terms of not the theoretical optimum, but rather the practically achievable ones (an "in the limit behaviour study"). An extension to other families of algorithms, based on other principles, may turn out to be interesting.

Furthermore, we insisted on keeping bounds for distance distortions for all pairs of points. From the perspective of clustering algorithms, it may not be so critical if the distance distortion bounds are violated by a sufficiently small number of points. This may be an interesting study direction on JL Lemma itself. And also for the study of clustering algorithms based on subsampling rather than on the whole data set. One may suspect, for example, that k-means algorithm will stabilize under increasing sample size. But the sample size increase delimits the possibilities of dimensionality reduction. Hence, the subsampling may be an interesting research direction for generalizations of JL Lemma.

A Proof of Theorems 2.1 and 2.2

A.1 Derivation of the set-friendly Johnson–Lindenstrauss lemma (Theorem 2.1)

Let us recall the process of seeking the mapping f from Theorem 1.1, as proposed by Dasgupta and Gupta [20]. We then switch to our target of selecting the size of the subspace guaranteeing that the projected distances maintain their proportionality in the required range.

Let us consider first a single vector $\mathbf{x} = (x_1, .x_2, \ldots, x_n)$ of n independent random variables drawn from the normal distribution $\mathcal{N}(0, 1)$ with mean 0 and variance 1. Let $\mathbf{x}' = (x_1, .x_2, \ldots, x_{n'})$, where $n' < n$, be its projection onto the first n' coordinates.[5]

Dasgupta and Gupta [20] in their Lemma 2.2 demonstrated that for a positive β

– if $\beta < 1$ then

$$Pr\left(\|\mathbf{x}'\|^2 \leq \beta\frac{n'}{n}\|\mathbf{x}\|^2\right) \leq \beta^{\frac{n'}{2}}\left(1+\frac{n'(1-\beta)}{n-n'}\right)^{\frac{n-n'}{2}} \tag{25}$$

– if $\beta > 1$ then

$$Pr\left(\|\mathbf{x}'\|^2 \geq \beta\frac{n'}{n}\|\mathbf{x}\|^2\right) \leq \beta^{\frac{n'}{2}}\left(1+\frac{n'(1-\beta)}{n-n'}\right)^{\frac{n-n'}{2}} \tag{26}$$

As probabilities are non-negative numbers, and we seek only possible projections, this latter bound makes sense only if $1+\frac{n'(1-\beta)}{n-n'} \geq 0$, that is $n' \leq n/\beta$. But what happens if this condition is violated? Rewrite $Pr(\|\mathbf{x}'\|^2 \geq \beta\frac{n'}{n}\|\mathbf{x}\|^2)$ as $Pr(x_1^2+\cdots+x_{n'}^2 \geq \beta\frac{n'}{n}\left(x_1^2+\cdots+x_n^2\right)) = Pr\left(\left(1-\beta\frac{n'}{n}\right)\left(x_1^2+\cdots+x_{n'}^2\right) \geq \beta\frac{n'}{n}\left(x_{n'+1}^2+\cdots+x_n^2\right)\right)$ If $n' > n/\beta$, then $\left(1-\beta\frac{n'}{n}\right) \leq 0$ which means that the mentioned probability is equal to zero, as it is the case when $n = n'\beta$. Therefore, the Gupta/Dasgupta Lemma is formally correct if we assume subsequently that there exists some n_{TRUE} being the true dimensionality and we take

$$n = \max(n_{TRUE}, n'\beta+\epsilon)\ ,\ \epsilon > 0 \tag{27}$$

Now, imagine we want to keep the error of squared length of $\mathbf{x}$ bounded within a range of $\pm\delta$ (relative error) upon projection, where $\delta \in (0, 1)$. Hence, $1-\delta \leq \beta \leq 1+\delta$. Then, we get the probability

$$\begin{aligned} &Pr\left((1-\delta)\|\mathbf{x}\|^2 \leq \frac{n}{n'}\|\mathbf{x}'\|^2 \leq (1+\delta)\|\mathbf{x}\|^2\right) \\ &\geq 1-(1-\delta)^{\frac{n'}{2}}\left(1+\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}} \\ &\quad -(1+\delta)^{\frac{n'}{2}}\left(1-\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}} \end{aligned}$$

[5] This is the random projection technique proposed by Dasgupta and Gupta [20]. In fact, if we first choose randomly a vector and then we would project it onto randomly selected n'-dimensional subspace, we would get the very same probability distribution for the vector and its projection as with this Dasgupta/Gupta approach.

This implies

$$Pr\left((1-\delta)\|\mathbf{x}\|^2 \le \frac{n}{n'}\|\mathbf{x}'\|^2 \le (1+\delta)\|\mathbf{x}\|^2\right)$$
$$\ge 1 - 2\max\left((1-\delta)^{\frac{n'}{2}}\left(1+\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}},\right.$$
$$\left.(1+\delta)^{\frac{n'}{2}}\left(1-\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}\right)$$
$$= 1 - 2\max_{\delta^*\in\{-\delta,+\delta\}}\left(\left(1-\delta^*\right)^{\frac{n'}{2}}\left(1+\frac{\delta^* n'}{n-n'}\right)^{\frac{n-n'}{2}}\right) \tag{28}$$

The same holds if we scale the vector $\mathbf{x}$.

Let us consider a sample consisting of m points in space, without however a guarantee that coordinates are independent between the vectors. We want that the probability that squared distances between all of them lie within the relative range $\pm\delta$ is larger than

$$1-\epsilon \le 1 - \binom{m}{2}\left(1 - Pr\left((1-\delta)\|\mathbf{x}\|^2 \le \frac{n}{n'}\|\mathbf{x}'\|^2 \le (1+\delta)\|\mathbf{x}\|^2\right)\right) \tag{29}$$

for some failure probability[6] term $\epsilon \in (0,1)$.

To achieve this, it is sufficient that the following holds:

$$\epsilon \ge \epsilon_I(n')$$
$$= \binom{m}{2}\left((1-\delta)^{\frac{n'}{2}}\left(1+\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}} + (1+\delta)^{\frac{n'}{2}}\left(1-\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}\right) \tag{30}$$

Let us now depart from the path of reasoning of [20] because we do not want to have a failure rate as big as $\frac{m-1}{m}$, but rather of say 0.05 or less, whichever we desire.

The formulas (28) and (29) allow us to conclude, that it is sufficient if ϵ satisfies:

$$\epsilon \ge 2\binom{m}{2}\max_{\delta^*\in\{-\delta,+\delta\}}\left(\left(1-\delta^*\right)^{\frac{n'}{2}}\left(1+\frac{\delta^* n'}{n-n'}\right)^{\frac{n-n'}{2}}\right)$$

Taking logarithm, we obtain:

$$\ln\epsilon \ge \ln(m(m-1))$$
$$+\max_{\delta^*\in\{-\delta,+\delta\}}\left(\frac{n'}{2}\ln(1-\delta^*) + \frac{(n-n')}{2}\ln\left(1+\frac{\delta^* n'}{n-n'}\right)\right)$$

Hence,

$$\ln\epsilon - \ln(m(m-1))$$
$$\ge \max_{\delta^*\in\{-\delta,+\delta\}}\left(\frac{n'}{2}\ln(1-\delta^*) + \frac{(n-n')}{2}\ln\left(1+\frac{\delta^* n'}{n-n'}\right)\right)$$

[6] We speak about a success if all the projected data point pairs fit formula (1). Otherwise, we speak about failure (even if only one data point lies outside this range).

We know[7] that $\ln(1+x) < x$ for $x > -1$ and $x \neq 0$, hence[8]

$$\ln\epsilon - \ln(m(m-1)) \geq \max_{\delta^* \in \{-\delta,+\delta\}} \left(\frac{n'}{2} \ln(1-\delta^*) + \frac{(n-n')}{2} \frac{\delta^* n'}{n-n'} \right)$$

It can be simplified to

$$\begin{aligned} \ln\epsilon - \ln(m(m-1)) &\geq \max_{\delta^* \in \{-\delta,+\delta\}} \left(\frac{n'}{2} \ln(1-\delta^*) + \frac{1}{2}(\delta^*) n' \right) \\ &= \frac{n'}{2} \max_{\delta^* \in \{-\delta,+\delta\}} \left(\ln(1-\delta^*) + \delta^* \right) \end{aligned}$$

Recall that also we have $\ln(1-x) + x < 0$ for $x < 1$ and $x \neq 0$, therefore

$$\max_{\delta^* \in \{-\delta,+\delta\}} \left(2 \frac{\ln\epsilon - \ln(m(m-1))}{\ln(1-\delta^*) + \delta^*} \right) \leq n'$$

So finally, realizing that $-\ln(1-\delta) - \delta \geq -\ln(1+\delta) + \delta > 0$, and that $\ln(m(m-1)) < 2\ln(m)$ we get as *sufficient* condition[9]

$$n' \geq 2 \frac{-\ln\epsilon + 2\ln(m)}{-\ln(1+\delta) + \delta}$$

This proves Theorem 2.1. Note that this expression does not depend on n that is the number of dimensions in the projection is chosen independently of the original number of dimensions. This is in spite of the fact that it was present in the formula (30) and it vanished "in the limit", when n was increasing. One may therefore ask if for lower values of n, n' can be lower than in that formula. We will handle this issue in Sect. A.2.

A.2 Computing n' from implicit representation in formula (7): Proof of Theorem 2.2

Having determined the explicit way of computing n', let us turn to the derivation of the algorithm for implicit computation for n' from formula (30).

Wherever we refer in our theorems and lemmas to the relation (4) *for* n', *we can use always alternatively* (7) *for* n'.

Let us now show that $\epsilon_I(n')$ is a decreasing function of n'. If it is so, then the above formula (30) can serve as a way of implicit n' computation via $\epsilon_I^{-1}(\epsilon)$, as explained subsequently.

$\epsilon_I(n')$ is a decreasing function if both $(1-\delta)^{\frac{n'}{2}} \left(1 + \frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}$ and $(1+\delta)^{\frac{n'}{2}} \left(1 - \frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}$ are decreasing in n'.

[7] Recall at this point the Taylor expansion $\ln(1+x) = x - x^2/2 + x^3/3 - x^5/5 \ldots$ which converges in the range $(-1,1)$ and hence implies $\ln(1+x) < x$ for $x \in (-1,0) \cup (0,1)$ as we will refer to it discussing difference to JL theorems of other authors.

[8] This step requires that $\frac{\delta^* n'}{n-n'} > -1$ that is $\delta^* > -\frac{n-n'}{n'}$. In case of $\delta^* = \delta$, it holds obviously. In case of $\delta^* = -\delta$, we require $\delta < \frac{n}{n'} - 1$, that is $1 + \delta < \frac{n}{n'}$. From condition (27), we have that $n > n'\beta$, so that $\frac{n}{n'} > \beta$ for every $\beta \in [1-\delta, 1+\delta]$. Hence, also $\frac{n}{n'} > 1 + \delta$ holds.

[9] We substituted the denominator with a smaller positive number and the nominator with a larger positive number so that the fraction value increases so that a higher n' will be required than actually needed.

Let us investigate $(1+\delta)^{\frac{n'}{2}}\left(1-\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}$. It is decreasing if its logarithm is decreasing too. So let us look at

$$\frac{n'}{2}\log(1+\delta)+\frac{n-n'}{2}\log\left(1-\frac{n'\delta}{n-n'}\right)$$

Recall that $\log(1+x)=-\sum_{j=1}^{\infty}\frac{(-x)^j}{j}$ for $x\in(-1,1)$. Therefore, the above expression can be rewritten as:

$$\begin{aligned}
&=\frac{n'}{2}\left(-\sum_{j=1}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{n-n'}{2}\left(-\sum_{j=1}^{\infty}\left(\frac{n'\delta}{n-n'}\right)^j/j\right)\\
&=-\frac{n'}{2}(-\delta)+\frac{n'}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)-\frac{n-n'}{2}\frac{n'\delta}{n-n'}+\frac{n-n'}{2}\left(-\sum_{j=2}^{\infty}\left(\frac{n'\delta}{n-n'}\right)^j/j\right)\\
&=\frac{n'}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{n-n'}{2}\left(-\sum_{j=2}^{\infty}\left(\frac{n'\delta}{n-n'}\right)^j/j\right)\\
&=\frac{n'}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{n'^j\delta^j}{(n-n')^{j-1}\cdot j}\right)
\end{aligned}$$

Let us compute the derivative of the latter expression

$$\begin{aligned}
&\frac{d}{dn'}\left[\frac{n'}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{n'^j\delta^j}{(n-n')^{j-1}\cdot j}\right)\right]\\
&=\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{\delta^j}{j}\frac{jn'^{j-1}\cdot(n-n')^{j-1}-(j-1)n'^j(n-n')^{j-2}}{(n-n')^{2(j-1)}}\right)\\
&=\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{\delta^j}{j}\frac{jn'^{j-1}\cdot(n-n')-(j-1)n'^j}{(n-n')^j}\right)\\
&=\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{\delta^j}{j}n'^{j-1}\frac{j\cdot(n-n')-(j-1)n'}{(n-n')^j}\right)\\
&=\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{\delta^j}{j}\left(\frac{n'}{n-n'}\right)^{j-1}\frac{jn-(2j-1)n'}{n-n'}\right)
\end{aligned}$$

In the same way, we can obtain the derivative of the logarithm of $(1-\delta)^{\frac{n'}{2}}\left(1+\frac{n'\delta}{n-n'}\right)^{\frac{n-n'}{2}}$ as

$$\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{\delta^j}{j}\right)+\frac{1}{2}\left(-\sum_{j=2}^{\infty}\frac{(-\delta)^j}{j}\left(\frac{n'}{n-n'}\right)^{j-1}\frac{jn-(2j-1)n'}{n-n'}\right)$$

Convergence of both is granted (because the summands in absolute value decrease quicker than in power series) if $\frac{n'}{n-n'} \leq \frac{1}{2}$ (or equivalently $3n' < n$)[10] which is a reasonable value for n', if we would bother about random projection.

Recall that $\left(-\sum_{j=2}^{\infty} \frac{(-\delta)^j}{j}\right) = \log(1+\delta) - \delta < 0$ and that $\left(-\sum_{j=2}^{\infty} \frac{\delta^j}{j}\right) = \log(1-\delta) + \delta < 0$. Furthermore, $-\sum_{j=2}^{\infty} \frac{\delta^j}{j} \left(\frac{n'}{n-n'}\right)^{j-1} \frac{jn-(2j-1)n'}{n-n'} < 0$ because all summands are of the same sign. And $-\sum_{j=2}^{\infty} \frac{(-\delta)^j}{j} \left(\frac{n'}{n-n'}\right)^{j-1} \frac{jn-(2j-1)n'}{n-n'} < 0$ because this is a sum of decreasing elements of alternating signs so the sign of the sum is identical with that of the first summand (which is positive, and the whole sum is negated).

Hence, both derivatives are negative. So the respective expressions are decreasing in n' so that the expression $\epsilon \geq \epsilon_I(n')$ can be exploited to find the lowest n' (the n'_I) that this expression is satisfied for a fixed ϵ. This search can be performed using the bisectional method. One starts with $n'_L := 1, n'_H := round(n/3) - 1$. If $\epsilon_I(n'_H) > \epsilon$, then seeking n'_I has failed. Otherwise, one determines in a loop $n'_M := round((n'_L + n'_H)/2)$ and computes $\epsilon_I(n'_L), \epsilon_I(n'_M), \epsilon_I(n'_H)$, then if $\epsilon_I(n'_M) < \epsilon$ then one sets $n'_H := n'_M$, otherwise $n'_L := n'_M$ (n'_M is always rounded up to the next integer). This process is continued till n'_M does not change. n'_I is set to n'_H.

B Proofs of Theorems 2.3–2.6

The proofs of theorems 2.3–2.6 require several intermediate lemmas, establishing stepwise partial results of these theorems.

B.1 Proof of Theorem 2.3

So the proof of Theorem 2.3 on cost function under projection requires first demonstration of Lemma B.1 showing that our Theorem 2.1 establishes limitations not only on distortions of point-to-point distances but also on point-its-own-cluster-centre distances.

Lemma B.1 *Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0,1)$. Let $Q \subset \mathbb{R}^n$ be a set of m representatives of elements of $\mathfrak{Q}$ in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let the inequality (4) or (7) hold. Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. For each $\mathbf{x}_i \in Q$, let $\mathbf{x'}_i \in Q'$ be its projection onto $\mathcal{C}_{n'}$. Let $\mathfrak{C}$ be a partition of $\mathfrak{Q}$. Then, for* all *data points $\mathbf{x}_i \in Q$*

$$(1-\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \leq \frac{n}{n'}\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq (1+\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \tag{31}$$

hold with probability of at least $1-\epsilon$,

Proof As we know, data points under k-means are assigned to clusters having the closest cluster centre. On the other hand, the cluster centre $\boldsymbol{\mu}$ is the average of all the data point representatives in the cluster.

[10] Then $\frac{jn-(2j-1)n'}{n-n'} \leq j \leq 2^{j-1} \leq \left(\frac{n'}{n-n'}\right)^{-(j-1)}$ for $j \geq 2$.

Hence, the cluster element i has the squared distance to its cluster centre $\boldsymbol{\mu}(\mathfrak{C}(i))$ amounting to

$$\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 = \frac{1}{|\mathfrak{C}(i)|} \sum_{j \in \mathfrak{C}(i)} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

But according to Theorem 2.1

$$(1-\delta) \sum_{j \in \mathfrak{C}(i)} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \frac{n}{n'} \sum_{j \in \mathfrak{C}(i)} \|\mathbf{x'}_i - \mathbf{x'}_j\|^2 \leq (1+\delta) \sum_{j \in \mathfrak{C}(i)} \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

Hence,

$$(1-\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \leq \frac{n}{n'}\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq (1+\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2$$

Note that here $\boldsymbol{\mu}'(\mathfrak{C}(i))$ is, by intention, not the projective image of $\boldsymbol{\mu}(\mathfrak{C}(i))$, but rather the centre of projected images of cluster elements. However, due to the linear nature of the projection, $\boldsymbol{\mu}'(\mathfrak{C}(i))$ and the centre of projected images of cluster elements coincide.

Lemma B.1 permits us to prove Theorem 2.3

Proof Theorem 2.3 According to formula (31):

$$(1-\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \leq \frac{n}{n'}\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq (1+\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2$$

Hence,

$$\sum_{i \in \mathfrak{Q}} (1-\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \leq \sum_{i \in \mathfrak{Q}} \frac{n}{n'}\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq \sum_{i \in \mathfrak{Q}} (1+\delta)\|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2$$

$$(1-\delta) \sum_{i \in \mathfrak{Q}} \|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2 \leq \sum_{i \in \mathfrak{Q}} \frac{n}{n'}\|\mathbf{x'}_i - \boldsymbol{\mu}'(\mathfrak{C}(i))\|^2 \leq (1+\delta) \sum_{i \in \mathfrak{Q}} \|\mathbf{x}_i - \boldsymbol{\mu}(\mathfrak{C}(i))\|^2$$

Based on defining equations (2) and (3), we get the formula (8)

$$(1-\delta)\mathfrak{J}(Q, \mathfrak{C}) \leq \frac{n}{n'}\mathfrak{J}(Q', \mathfrak{C}) \leq (1+\delta)\mathfrak{J}(Q, \mathfrak{C})$$

B.2 Proof of Theorem 2.4

In order to prove Theorem 2.4 on preservation of local minima under projection, we need to go beyond point-its-own-cluster-centre distance considerations and investigate the cluster-cluster distance change under random projection, as described by Lemma B.2. Then, in Lemma B.3 we establish conditions under which a point does not change locally optimal assignment to cluster under projection (as the change would not decrease the cost function) if we concentrate on two clusters only. An alternative formulation of no cluster change condition is expressed in Lemma B.4.

Let us derive now first Lemma B.2 on distances between projected cluster centres. Let us investigate the distance between centres of two clusters, say C_1, C_2. Let their cardinalities amount to m_1, m_2, respectively. Denote $C_{12} = C_1 \cup C_2$. Consequently $m_{12} = |C_{12}| = m_1 + m_2$. For a set C_j, let $\mathrm{VAR}(C_j) = \frac{1}{|C_j|}\sum_{i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}(C_j)\|^2$ and $\mathrm{VAR}'(C_j) = \frac{1}{|C_j|}\sum_{i \in C_j} \|\mathbf{x'}_i - \boldsymbol{\mu}'(C_j)\|^2$.

Therefore,

$$\begin{aligned}\mathrm{VAR}(C_{12}) &= \frac{1}{|C_{12}|}\sum_{i\in C_{12}}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\\ &= \frac{1}{|C_{12}|}\left(\left(\sum_{i\in C_1}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\end{aligned}$$

By inserting a zero

$$\begin{aligned}&= \frac{1}{|C_{12}|}\left(\left(\sum_{i\in C_1}\|\mathbf{x}_i-\boldsymbol{\mu}(C_1)+\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12})\|^2\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\\ &= \frac{1}{|C_{12}|}\left(\left(\sum_{i\in C_1}\left((\mathbf{x}_i-\boldsymbol{\mu}(C_1))^2+(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2\right.\right.\right.\\ &\quad\left.\left.\left.+2(\mathbf{x}_i-\boldsymbol{\mu}(C_1))\circ(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))\right)\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\\ &= \frac{1}{|C_{12}|}\left(\left(\left(\sum_{i\in C_1}(\mathbf{x}_i-\boldsymbol{\mu}(C_1))^2\right)+\left(\sum_{i\in C_1}(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2\right)\right.\right.\\ &\quad\left.\left.+2\left(\sum_{i\in C_1}\mathbf{x}_i-\sum_{i\in C_1}\boldsymbol{\mu}(C_1)\right)\circ(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\\ &= \frac{1}{|C_{12}|}\left(\left(\left(\sum_{i\in C_1}(\mathbf{x}_i-\boldsymbol{\mu}(C_1))^2\right)+|C_1|(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2\right.\right.\\ &\quad\left.\left.+2(|C_1|\boldsymbol{\mu}(C_1)-|C_1|\boldsymbol{\mu}(C_1))\circ(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\\ &= \frac{1}{|C_{12}|}\left(\left(VAR(C_1)|C_1|+|C_1|(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2\right)+\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_{12})\|^2\right)\right)\end{aligned}$$

Via the same reasoning, we get:

$$\begin{aligned}&= \frac{1}{|C_{12}|}\left(\left(\mathrm{VAR}(C_1)|C_1|+|C_1|(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2\right)\right.\\ &\quad\left.+\left(\mathrm{VAR}(C_2)|C_2|+|C_2|(\boldsymbol{\mu}(C_2)-\boldsymbol{\mu}(C_{12}))^2\right)\right)\\ &= \frac{1}{|C_{12}|}\left(\mathrm{VAR}(C_1)|C_1|+\mathrm{VAR}(C_2)|C_2|\right.\\ &\quad\left.+|C_1|(\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_{12}))^2+|C_2|(\boldsymbol{\mu}(C_2)-\boldsymbol{\mu}(C_{12}))^2\right)\end{aligned}$$

As $\boldsymbol{\mu}(C_{12}) = \frac{1}{|C_{12}|}\sum_{i\in C_{12}}\mathbf{x}_i = \frac{1}{|C_{12}|}\left((\sum_{i\in C_1}\mathbf{x}_i) + (\sum_{i\in C_2}\mathbf{x}_i)\right) = \frac{1}{|C_{12}|}(|C_1|\boldsymbol{\mu}(C_1) + |C_1|\boldsymbol{\mu}(C_1))$ that is $\boldsymbol{\mu}(C_{12}) = \frac{|C_1|}{|C_{12}|}\boldsymbol{\mu}(C_1) + \frac{|C_2|}{|C_{12}|}\boldsymbol{\mu}(C_2$, we get

$$
\begin{aligned}
&= \frac{1}{|C_{12}|}\left(\mathrm{VAR}(C_1)|C_1| + \mathrm{VAR}(C_2)|C_2| + |C_1|\left(\boldsymbol{\mu}(C_1) - \frac{|C_1|}{|C_{12}|}\boldsymbol{\mu}(C_1) - \frac{|C_2|}{|C_{12}|}\boldsymbol{\mu}(C_2\right)^2\right.\\
&\quad \left. + |C_2|\left(\boldsymbol{\mu}(C_2) - \frac{|C_1|}{|C_{12}|}\boldsymbol{\mu}(C_1) - \frac{|C_2|}{|C_{12}|}\boldsymbol{\mu}(C_2)\right)^2\right)\\
&= \frac{1}{|C_{12}|}\left(\mathrm{VAR}(C_1)|C_1| + \mathrm{VAR}(C_2)|C_2| + |C_1|\left(\frac{|C_2|}{|C_{12}|}\boldsymbol{\mu}(C_1) - \frac{|C_2|}{|C_{12}|}\boldsymbol{\mu}(C_2)\right)^2\right.\\
&\quad \left. + |C_2|\left(-\frac{|C_1|}{|C_{12}|}\boldsymbol{\mu}(C_1) + \frac{|C_1|}{|C_{12}|}\boldsymbol{\mu}(C_2)\right)^2\right)\\
&= \frac{1}{|C_{12}|}\left(\mathrm{VAR}(C_1)|C_1| + \mathrm{VAR}(C_2)|C_2| + \frac{|C_1||C_2|^2 + |C_1|^2|C_2|}{|C_{12}|^2}(\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2))^2\right)
\end{aligned}
$$

hence

$$
\mathrm{VAR}(C_{12}) = \frac{1}{|C_{12}|}\left(\mathrm{VAR}(C_1)|C_1| + \mathrm{VAR}(C_2)|C_2| + \frac{|C_1||C_2|}{|C_{12}|}(\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2))^2\right)
$$

This leads immediately to

$$
\mathrm{VAR}(C_{12})\cdot m_{12} = \mathrm{VAR}(C_1)\cdot m_1 + \mathrm{VAR}(C_2)\cdot m_2 + m_1\cdot m_2/m_{12}\cdot\|\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2)\|^2 \tag{32}
$$

which implies

$$
\mathrm{VAR}(C_{12})\cdot\frac{m_{12}^2}{m_1\cdot m_2} = \mathrm{VAR}(C_1)\cdot\frac{m_{12}}{m_2} + \mathrm{VAR}(C_2)\cdot\frac{m_{12}}{m_1} + \|\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2)\|^2
$$

By analogy, we can derive

$$
\mathrm{VAR}'(C_{12})\cdot m_{12} = \mathrm{VAR}'(C_1)\cdot m_1 + \mathrm{VAR}'(C_2)\cdot m_2 + m_1\cdot m_2/m_{12}\cdot\|\boldsymbol{\mu}'(C_1) - \boldsymbol{\mu}'(C_2)\|^2 \tag{33}
$$

Note that $\mathfrak{J}(Q_{12}, \{C_{12}\}) = \mathrm{VAR}(C_{12})\cdot m_{12}, \mathfrak{J}(Q'_{12}, \{C_{12}\}) = \mathrm{VAR}'(C_{12})\cdot m_{12}$. According to Lemma B.1, applied to the set C_{12} as a cluster,

$$
(1-\delta)\mathfrak{J}(Q_{12}, \{C_{12}\}) \le \frac{n}{n'}\mathfrak{J}(Q'_{12}, \{C_{12}\}) \le (1+\delta)\mathfrak{J}(Q_{12}, \{C_{12}\})
$$

that is after the substitution (32), (33), and its complement for the projected space

$$
\begin{aligned}
&(1-\delta)\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2 + \mathrm{VAR}(C_2)\cdot m_{12}/m_1 + \|\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2)\|^2\right)\\
&\le \frac{n}{n'}\left(\mathrm{VAR}'(C_1)\cdot m_{12}/m_2 + \mathrm{VAR}'(C_2)\cdot m_{12}/m_1 + \|\boldsymbol{\mu}'(C_1) - \boldsymbol{\mu}'(C_2)\|^2\right)\\
&\le (1+\delta)\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2 + \mathrm{VAR}(C_2)\cdot m_{12}/m_1 + \|\boldsymbol{\mu}(C_1) - \boldsymbol{\mu}(C_2)\|^2\right)
\end{aligned} \tag{34}
$$

According to Lemma B.1, applied to the set C_1 as a cluster, and C_2 as a cluster,

$$
\begin{aligned}
(1-\delta)\mathfrak{J}(Q_1, \{C_1\}) &\le \frac{n}{n'}\mathfrak{J}(Q'_1, \{C_1\}) \le (1+\delta)\mathfrak{J}(Q_1, \{C_1\})\\
(1-\delta)\mathfrak{J}(Q_2, \{C_2\}) &\le \frac{n}{n'}\mathfrak{J}(Q'_2, \{C_2\}) \le (1+\delta)\mathfrak{J}(Q_2, \{C_2\})
\end{aligned}
$$

which implies

$$\begin{aligned}(1-\delta)(\mathfrak{J}(Q_1,\{C_1\})+\mathfrak{J}(Q_2,\{C_2\})) &\le \frac{n}{n'}(\mathfrak{J}(Q_1',\{C_1\})+\mathfrak{J}(Q_2',\{C_2\}))\\ &\le (1+\delta)(\mathfrak{J}(Q_1,\{C_1\})+\mathfrak{J}(Q_2,\{C_2\}))\end{aligned} \tag{35}$$

Recall also that $\mathfrak{J}(Q_1,\{C_1\}) = \mathrm{VAR}(C_1)\cdot m_1$, $\mathfrak{J}(Q_1',\{C_1\}) = \mathrm{VAR}'(C_1)\cdot m_1$, $\mathfrak{J}(Q_2,\{C_2\}) = \mathrm{VAR}(C_2)\cdot m_2$, $\mathfrak{J}(Q_2',\{C_2\}) = \mathrm{VAR}'(C_2)\cdot m_2$. These equations combined with the relation (35) imply:

$$\begin{aligned}&(1-\delta)\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1\right)\\ &\le \frac{n}{n'}\left(\mathrm{VAR}'(C_1)\cdot m_{12}/m_2+\mathrm{VAR}'(C_2)\cdot m_{12}/m_1\right)\\ &\le (1+\delta)\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1\right)\end{aligned} \tag{36}$$

The two inequalities (34) and (36) mean that

$$\begin{aligned}&-2\delta\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1\right)+(1-\delta)\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2\\ &\le \frac{n}{n'}\left(\|\boldsymbol{\mu}'(C_1)-\boldsymbol{\mu}'(C_2)\|^2\right)\\ &\le 2\delta\left(\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1\right)+(1+\delta)\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2\end{aligned} \tag{37}$$

Let us assume that the quotient

$$\frac{\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1}{\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2}\le p \tag{38}$$

where p is some positive number. So substituting this relation into (37), we have in effect

$$\begin{aligned}(1-\delta(1+2p))\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2 &\le \frac{n}{n'}\left(\|\boldsymbol{\mu}'(C_1)-\boldsymbol{\mu}'(C_2)\|^2\right)\\ &\le (1+\delta(1+2p))\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2\end{aligned}$$

Under balanced ball-shaped[11] clusters, p does not exceed 1. So we have shown the lemma.

Lemma B.2 *Under the assumptions of preceding lemmas for any two clusters C_1, C_2*

$$\begin{aligned}(1-\delta(1+2p))\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2 &\le \frac{n}{n'}\left(\|\boldsymbol{\mu}'(C_1)-\boldsymbol{\mu}'(C_2)\|^2\right)\\ &\le (1+\delta(1+2p))\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\|^2\end{aligned}$$

where the non-negative p, defined by relation (38), depends on degree of balance between cluster distance and cluster shape, holds with probability at least $1-\epsilon$.

Now, let us consider the choice of δ in such a way that with high probability no data point will be classified into some other cluster. We claim the following:

[11] A ball-shaped set of data has the variance of at most the squared radius of the ball. If clusters are balanced, that is of the same enclosing radius ($r_1 = r_2$), so the variance $\mathrm{VAR}(C_1) \le r_1$, $\mathrm{VAR}(C_2) \le r_2$), and the same cardinality $m_1 = m_2$, the nominator of expression for p has the form of at most four times the common squared radius ($\mathrm{VAR}(C_1)\cdot m_{12}/m_2+\mathrm{VAR}(C_2)\cdot m_{12}/m_1 \le r_1^2\cdot 2+r_1^2\cdot 2$). The denominator is bigger than squared sum of radii ($\|\boldsymbol{\mu}(C_1)-\boldsymbol{\mu}(C_2)\| \ge r_1+r_2 = 2r_1$), that is four times the squared common radius. So the quotient does not exceed 1.

Lemma B.3 *Consider two clusters C_1, C_2. Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0, 1)$. Let $Q \subset \mathbb{R}^n$ be a set of m points in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let the inequality (4) or (7) hold. Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. For each $\mathbf{x}_i \in Q$, let $\mathbf{x'}_i$ be its projection onto $\mathcal{C}_{n'}$. For two clusters C_1, C_2, obtained via k-means, in the original space let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ be their centres and $\boldsymbol{\mu}'_1, \boldsymbol{\mu}'_2$ be centres to the corresponding sets of projected cluster members. Furthermore, let d be the distance of the first cluster centre to the common border of both clusters and let the closest point of the first cluster to this border be at the distance of αd from its cluster centre as projected on the line connecting both cluster centres, where $\alpha \in (0, 1)$.*

Then, all projected points of the first cluster are (each) closer to the centre of the set of projected points of the first than to the centre of the set of projected points of the second if

$$\delta \leq \frac{1 - \left(1 - \frac{g}{2}\right)^2}{\left(1 - \frac{g}{2}\right)^2 + (1 + 2p)} = \frac{1 - \alpha^2}{(1 + 2p) + \alpha^2} \tag{39}$$

where $g = 2(1 - \alpha)$, with probability of at least $1 - \epsilon$.

Proof Consider a data point $\mathbf{x}$ "close" to the border between the two neighbouring clusters, on the line connecting the cluster centres, belonging to the first cluster, at a distance αd from its cluster centre, where d is the distance of the first cluster centre to the border and $\alpha \in (0, 1)$. The squared distance between cluster centres, under projection, can be "reduced" by the factor $1 - \delta$ (beside the factor $\frac{n}{n'}$ which is common to all the points), whereas the squared distance of $\mathbf{x}$ to its cluster centre may be "increased" by the factor $1 + \delta$. This implies a relationship between the factor α and the error δ.

If $\mathbf{x}'$ should not cross the border between the clusters, the following needs to hold:

$$\|\mathbf{x}' - \boldsymbol{\mu}'_1\| \leq \frac{1}{2}\|\boldsymbol{\mu}'_2 - \boldsymbol{\mu}'_1\| \tag{40}$$

which implies:

$$\frac{n}{n'}\|\mathbf{x}' - \boldsymbol{\mu}'_1\|^2 \leq \frac{n}{n'}\frac{1}{4}\|\boldsymbol{\mu}'_2 - \boldsymbol{\mu}'_1\|^2$$

As (see Lemma B.1)

$$\frac{n}{n'}\|\mathbf{x}' - \boldsymbol{\mu}'_1\|^2 \leq (1 + \delta)\|\mathbf{x} - \boldsymbol{\mu}_1\|^2 = (1 + \delta)(\alpha d)^2$$

and (see Lemma B.2)

$$\frac{n}{n'}\frac{1}{4}\|\boldsymbol{\mu}'_2 - \boldsymbol{\mu}'_1\|^2 \geq (1 - \delta(1 + 2p))\frac{1}{4}\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|^2 = (1 - \delta(1 + 2p))d^2$$

we know that, for inequality (40) to hold, it is sufficient that:

$$(1 + \delta)(\alpha d)^2 \leq (1 - \delta(1 + 2p))d^2$$

that is

$$\alpha \leq \sqrt{\frac{1 - \delta(1 + 2p)}{1 + \delta}}$$

But $2(1 - \alpha)d$ or $2(1 - \alpha)$ can be viewed as resp. absolute or relative gap between clusters. So if we expect a relative gap $g = 2(1 - \alpha)$ between clusters, we have to choose δ in such a way that

$$1 - \frac{g}{2} \leq \sqrt{\frac{1 - \delta(1+2p)}{1+\delta}}$$

Therefore,

$$\delta \leq \frac{1 - \left(1 - \frac{g}{2}\right)^2}{\left(1 - \frac{g}{2}\right)^2 + (1+2p)} \tag{41}$$

□

So we see that the decision on the permitted error depends on the size of the gap between clusters that we hope to observe.

Lemma B.3 allows us to prove Theorem 2.4 in a straightforward manner.

Proof (Theorem 2.4) Observe that in this theorem we impose the condition of this lemma on each cluster. So all projected points are closer to their set centres than to any other centre. So the k-means algorithm would get stuck at this clustering and hence we get at a local minimum. □

B.3 Proofs of Theorems 2.5 and 2.6

In order to prove Theorem 2.5 on mapping of local minima from the projected space to the original space, we need first a proof of Lemma B.5 on no cluster change when stepping back from the projected space to the original space (keeping locally optimal assignment to a cluster).

Having these results, we complete the section with the proof of Theorem 2.6.

Note that we can make another characterization of the situation of no cluster change, not related to cluster centres but rather to point-to-point distances. Now, let us consider the choice of δ in such a way that with high probability no data point will be classified into some other cluster. We claim the following:

Lemma B.4 *Consider two clusters C_1, C_2. Let $\delta \in (0, \frac{1}{2})$, $\epsilon \in (0,1)$. Let $Q \subset \mathbb{R}^n$ be a set of m points in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let the inequality (4) or (7) hold. Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. For each $\mathbf{x}_i \in Q$, let $\mathbf{x}'_i$ be its projection onto $\mathcal{C}_{n'}$. For two clusters C_1, C_2, obtained via k-means, in the original space let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ be their centres and $\boldsymbol{\mu}'_1, \boldsymbol{\mu}'_2$ be centres to the corresponding sets of projected cluster members.*

Then, all projected points of the first cluster are (each) closer to the centre of the set of projected points of the first than to the centre of the set of projected points of the second if

$$\begin{aligned}
&\frac{1}{|C_1|}\left(2\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \frac{1}{|C_2|+1}\left(\sum_{i\in C_2}\|\mathbf{x}_i - \boldsymbol{\mu}(C_2)\|^2\right) \\
&\leq \frac{1-\delta}{1+\delta}\left(\frac{1}{|C_1|}\left(\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i - \boldsymbol{\mu}(C_1-\{j\})\|^2\right)\right. \\
&\quad \left. + \frac{1}{|C_2|+1}\left(2\sum_{i\in C_2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)\right)
\end{aligned} \tag{42}$$

with probability of at least $1-\epsilon$.

Proof For the change in clusters to be prevented, for any point $j \in C_1$, the combined sum of squared distances to cluster centres in these two clusters C_1, C_2 should be lower or equal to such a sum for clusters resulting from the switch of the element j from C_1 to C_2, that is for clusters $C_1 - \{j\}, C_2 \cup \{j\}$. This means that the following has to hold:

$$\sum_{i \in C_1} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_1)'\|^2 + \sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_2)'\|^2 \leq \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_1 - \{j\})'\|^2 + \sum_{i \in C_2 \cup \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_2 \cup \{j\})'\|^2$$

By replacing the sums of squared distances to centres by squared distances between data points, we get:

$$\sum_{i \in C_1} \sum_{l \in C_1} \frac{1}{|C_1|} \|\mathbf{x}'_i - \mathbf{x}'_l\|^2 + \sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_2)'\|^2 \leq \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_1 - \{j\})'\|^2 + \sum_{i \in C_2 \cup \{j\}} \sum_{l \in C_2 \cup \{j\}} \frac{1}{|C_2 \cup \{j\}|} \|\mathbf{x}'_i - \mathbf{x}'_l\|^2$$

By rearranging terms, we obtain:

$$\frac{1}{|C_1|} \left(\left(\sum_{i \in C_1 - \{j\}} \sum_{l \in C_1 - \{j\}} \|\mathbf{x}'_i - \mathbf{x}'_l\|^2 \right) + \left(2 \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right) \right) + \sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_2)\|^2 \leq \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_1 - \{j\})\|^2 + \frac{1}{|C_2| + 1} \left(\left(\sum_{i \in C_2} \sum_{l \in C_2} \|\mathbf{x}'_i - \mathbf{x}'_l\|^2 \right) + \left(2 \sum_{i \in C_2} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right) \right)$$

We consider now the centric sums of squares for smaller clusters (C_2, $C_1 - \{i\}$)

$$\frac{1}{|C_1|} \left((|C_1| - 1) \left(\sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_1 - \{j\})\|^2 \right) + \left(2 \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right) \right) + \sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_2)\|^2 \leq \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_1 - \{j\})'\|^2 + \frac{1}{|C_2| + 1} \left(|C_2| \left(\sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_2)\|^2 \right) + \left(2 \sum_{i \in C_2} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right) \right)$$

By simplification, we obtain:

$$\frac{1}{|C_1|} (|C_1| - 1) \left(\sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_1 - \{j\})\|^2 \right) + \frac{1}{|C_1|} \left(2 \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right) + \sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_2)\|^2 \leq \sum_{i \in C_1 - \{j\}} \|\mathbf{x}'_i - \boldsymbol{\mu}(C_1 - \{j\})'\|^2 + \frac{1}{|C_2| + 1} |C_2| \left(\sum_{i \in C_2} \|\mathbf{x}'_i - \boldsymbol{\mu}'(C_2)\|^2 \right) + \frac{1}{|C_2| + 1} \left(2 \sum_{i \in C_2} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2 \right)$$

We subtract from both sides $\sum_{i\in C_2}\|\mathbf{x'}_i-\boldsymbol{\mu}'(C_2)\|^2+\sum_{i\in C_1-\{j\}}\|\mathbf{x'}_i-\boldsymbol{\mu}(C_1-\{j\})'\|^2$:

$$+\frac{1}{|C_1|}\left(2\sum_{i\in C_1-\{j\}}\|\mathbf{x'}_i-\mathbf{x'}_j\|^2\right)+\frac{1}{|C_2|+1}\left(\sum_{i\in C_2}\|\mathbf{x'}_i-\boldsymbol{\mu}'(C_2)\|^2\right)$$
$$\leq\frac{1}{|C_1|}\left(\sum_{i\in C_1-\{j\}}\|\mathbf{x'}_i-\boldsymbol{\mu}'(C_1-\{j\})\|^2\right)+\frac{1}{|C_2|+1}\left(2\sum_{i\in C_2}\|\mathbf{x'}_i-\mathbf{x'}_j\|^2\right)$$

In order that the above formula is valid, it suffices that

$$(1+\delta)\frac{n'}{n}\left(\frac{1}{|C_1|}\left(2\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)+\frac{1}{|C_2|+1}\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_2)\|^2\right)\right)$$
$$\leq(1-\delta)\frac{n'}{n}\left(\frac{1}{|C_1|}\left(\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i-\boldsymbol{\mu}(C_1-\{j\})\|^2\right)+\frac{1}{|C_2|+1}\left(2\sum_{i\in C_2}\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)\right)$$

because of the relationship between the original and projected space distances from Theorems. 2.1. and 2.3

Now, we obtain the claim of the Lemma, by dividing both sides with $(1+\delta)\frac{n'}{n}$:

$$\frac{1}{|C_1|}\left(2\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)+\frac{1}{|C_2|+1}\left(\sum_{i\in C_2}\|\mathbf{x}_i-\boldsymbol{\mu}(C_2)\|^2\right)$$
$$\leq\frac{1-\delta}{1+\delta}\left(\frac{1}{|C_1|}\left(\sum_{i\in C_1-\{j\}}\|\mathbf{x}_i-\boldsymbol{\mu}(C_1-\{j\})\|^2\right)+\frac{1}{|C_2|+1}\left(2\sum_{i\in C_2}\|\mathbf{x}_i-\mathbf{x}_j\|^2\right)\right)$$

Note that the above relation can be interpreted as stating that the "mean" squared distance to elements of the own cluster increased by mean squared distance to cluster centre in the opposite cluster must be lower from the "mean" squared distance to elements of the other clusters increased by the mean squared distance to cluster centre within the rest of the cluster reduced the factor $\frac{1-\delta}{1+\delta}$. Note also that when ignoring the $\frac{1-\delta}{1+\delta}$ fraction, the condition is normally satisfied in the original space if clustering gets at a local minimum. Hence, this factor is actually the only one required additionally for the clustering to be kept under projection. □

Lemma B.5 *Let $\delta\in(0,\frac{1}{2})$, $\epsilon\in(0,1)$. Let $Q\subset\mathbb{R}^n$ be a set of m points in an n-dimensional orthogonal coordinate system $\mathcal{C}_n$ and let the inequality (4) or (7) hold. Let $\mathcal{C}_{n'}$ be a randomly selected (via sampling from a normal distribution) n'-dimensional orthogonal coordinate system. For each $\mathbf{x}_i\in Q$, let $\mathbf{x'}_i$ be its projection onto $\mathcal{C}_{n'}$. For any two k-means clusters C_1, C_2 in the projected space, let $\boldsymbol{\mu}'_1, \boldsymbol{\mu}'_2$ be their centres in the projected space and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ be centres to the corresponding sets of cluster members in the original space. Furthermore, let d be the distance of the first cluster centre to the common border of both clusters in the projected space and let the closest point of the first cluster to this border in that space be at the distances of αd from its cluster centre , where $\alpha\in[0,1)$.*

Then, all points of the first cluster in the original space are (each) closer to the centre of the set of points of the first than to the centre of the set of points of the second cluster in the

original space if

$$\delta \leq \frac{1-\left(1-\frac{(2-2\alpha)}{2}\right)^2}{\left(1-\frac{(2-2\alpha)}{2}\right)^2+(1+2p)} = \frac{1-\alpha^2}{(1+2p)+\alpha^2} \tag{43}$$

with probability of at least $1-\epsilon$.

Proof Consider a data point $\mathbf{x}'$ "close" to the border between the two neighbouring clusters in the projected space, on the line connecting the cluster centres, belonging to the first cluster, at a distance αd from its cluster centre, where d is the distance of the first cluster centre to the border and $\alpha \in (0, 1)$. The squared distance between cluster centres, in original space, can be "reduced" by the factor $(1+\delta)^{-1}$ (beside the factor $\frac{n'}{n}$ which is common to all the points), whereas the squared distance of $\mathbf{x}$ to its cluster centre may be "increased" by the factor $(1-\delta)^{-1}$. This implies a relationship between the factor α and the error δ.

If $\mathbf{x}$ (in the original space) should not cross the border between the clusters, the following needs to hold:

$$\|\mathbf{x}-\boldsymbol{\mu}_1\| \leq \frac{1}{2}\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\| \tag{44}$$

which implies:

$$\frac{n'}{n}\|\mathbf{x}-\boldsymbol{\mu}_1\|^2 \leq \frac{n'}{n}\frac{1}{4}\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\|^2$$

As (see Lemma B.1)

$$\frac{n'}{n}\|\mathbf{x}-\boldsymbol{\mu}_1\|^2 \leq (1-\delta)^{-1}\|\mathbf{x}'-\boldsymbol{\mu}_1'\|^2 = (1-\delta)^{-1}(\alpha d)^2$$

and (see Lemma B.2)

$$\frac{n'}{n}\frac{1}{4}\|\boldsymbol{\mu}_2-\boldsymbol{\mu}_1\|^2 \geq (1+\delta(1+2p))^{-1}\frac{1}{4}\|\boldsymbol{\mu}_2'-\boldsymbol{\mu}_1'\|^2 = (1+\delta(1+2p))^{-1}d^2$$

We know that, for inequality (44) to hold, it is sufficient that:

$$(1-\delta)^{-1}(\alpha d)^2 \leq (1+\delta(1+2p))^{-1}d^2$$

that is

$$\alpha \leq \sqrt{\frac{1-\delta}{1+\delta(1+2p)}}$$

But $2(1-\alpha)d$ or $2(1-\alpha)$ can be viewed as absolute or relative gap between clusters. So if we want to have a relative gap $g = 2(1-\alpha)$ between clusters, we have to choose δ in such a way that

$$1-\frac{g}{2} \leq \sqrt{\frac{1-\delta}{1+\delta(1+2p)}}$$

Therefore,

$$\delta \leq \frac{1-\left(1-\frac{g}{2}\right)^2}{\left(1-\frac{g}{2}\right)^2+(1+2p)} \tag{45}$$

□

Lemma B.5 allows us to prove Theorem 2.5 in a straightforward manner.

Proof (Theorem 2.5) Observe that in this theorem we impose the condition of this lemma on each cluster. So all original space points are closer to their set centres than to any other centre. So the k-means algorithm would get stuck at this clustering and hence we get at a local minimum. □

Having these results, we can go over to the proof of Theorem 2.6.

Proof (Theorem 2.6) Let $\mathfrak{C}_{\mathfrak{G}}$ denote the clustering reaching the global optimum in the original space. Let $\mathfrak{C}'_{\mathfrak{G}}$ denote the clustering reaching the global optimum in the projected space. From Theorem 2.3, we have that

$$(1-\delta)\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}}) \leq \frac{n}{n'}\mathfrak{J}(Q',\mathfrak{C}_{\mathfrak{G}}) \leq (1+\delta)\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}}) \tag{46}$$

On the other hand

$$(1+\delta)^{-1}\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}}) \leq \frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}'_{\mathfrak{G}}) \leq (1-\delta)^{-1}\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}})$$

$\mathfrak{C}'_{\mathfrak{G}}$ is the global minimum in the projected space, hence

$$\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}}) \leq \mathfrak{J}(Q',\mathfrak{C}_{\mathfrak{G}}) \tag{47}$$

So from inequalities (46) and (47)

$$\frac{n}{n'}\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}}) \leq \frac{n}{n'}\mathfrak{J}(Q',\mathfrak{C}_{\mathfrak{G}}) \leq (1+\delta)\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}})$$

So we proved the claim of Theorem 2.6 that is that

$$\frac{n}{n'}\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}}) \leq (1+\delta)\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}})$$

Note additionally, that analogously, $\mathfrak{C}_{\mathfrak{G}}$ is the global minimum in the original space, hence

$$\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}}) \leq \mathfrak{J}(Q,\mathfrak{C}'_{\mathfrak{G}})$$

and therefore

$$\frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}_{\mathfrak{G}}) \leq \frac{n'}{n}\mathfrak{J}(Q,\mathfrak{C}'_{\mathfrak{G}}) \leq (1-\delta)^{-1}\mathfrak{J}(Q',\mathfrak{C}'_{\mathfrak{G}})$$

□

References

1. Achlioptas D (2003) Database-friendly random projections: Johnson–Lindenstrauss with binary coins. J Comput Syst Sci 66(4):671–687
2. Ackerman M, Ben-David S (2009) Clusterability: a theoretical study. In: van Dyk D, Welling M (eds) Proceedings of the twelth international conference on artificial intelligence and statistics, vol. 5 of proceedings of machine learning research, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, pp. 1–8. http://proceedings.mlr.press/v5/ackerman09a.html
3. Ahmadian S, Norouzi-Fard A, Svensson O, Ward J (2017) Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In: 2017 IEEE 58th annual symposium on foundations of computer science (FOCS). pp 61–72
4. Ailon N, Chazelle B (2006) Approximate nearest neighbors and the fast Johnson–Lindenstrauss transform. In: Proceedings of the thirty-eighth annual ACM symposium on theory of computing, STOC 06. ACM, New York, pp 557–563

5. Arthur D, Vassilvitskii S (2007) *k*-means++: the advantages of careful seeding. In: Bansal N, Pruhs K, Stein C (eds) Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA 2007. SIAM, New Orleans, pp 1027–1035
6. Awasthi P, Blum A, Sheffet O (2010) Stability yields a ptas for k-median and k-means clustering. In: Proceedings of the 2010 IEEE 51st annual symposium on foundations of computer science, FOCS 10. IEEE Computer Society, Washington, pp 309–318
7. Awasthi P, Blum A, Sheffet O (2012) Center-based clustering under perturbation stability. Inf Process Lett 112(1–2):49–54
8. Balcan M, Blum A, Gupta A (2009) Approximate clustering without the approximation. In: Proceedings of the twentieth annual ACM-SIAM symposium on discrete algorithms, SODA 2009, New York, NY, USA, January 4–6, 2009, pp 1068–1077
9. Balcan M, Liang Y (2016) Clustering under perturbation resilience. SIAM J Comput 45(1):102–155
10. Bandeira AS (2015) 18.s096: Johnson–Lindenstrauss lemma and Gordons theorems. Lecture Notes. http://math.mit.edu/~bandeira/2015_18.S096_5_Johnson_Lindenstrauss.pdf
11. Baraniuk R, Davenport MA, DeVore R, Wakin M (2007) A simple proof of the restricted isometry property for random matrices. Constr Approx 28(3):253–263
12. Baraniuk R, Davenport MA, Duarte MF, Hegde C (2014) An introduction to compressive sensing. https://legacy.cnx.org/content/col11133/1.5/. Accessed 5 May 2018
13. Baraniuk R, Davenport M, Devore R, Wakin M (2008) A simple proof of the restricted isometry property for random matrices. Constr Approx 28(3):253–263
14. Ben-David S (2015) Computational feasibility of clustering under clusterability assumptions. arXiv:1501.00437
15. Bilu Y, Linial N (2012) Are stable instances easy? Comb Probab Comput 21(5):643–660
16. Cannings TI, Samworth RJ (2017) Random-projection ensemble classification. J R Stat Soc Ser B (Stat Methodol) 79(4):959–1035
17. Chiong, KX, Shum M (2016) Random projection estimation of discrete-choice models with large choice sets. arxiv:1604.06036
18. Clarkson KL, Woodruff DP (2017) Low-rank approximation and regression in input sparsity time. J ACM 63(6):54:1–54:45. https://doi.org/10.1145/3019134
19. Cohen M, Jayram T, Nelson J (2018) Simple analyses of the sparse Johnson–Lindenstrauss transform. In: Seidel R (ed) 1st symposium on simplicity in algorithms (SOSA 2018), Vol. 61 of OpenAccess series in informatics (OASIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp 15:1–15:9
20. Dasgupta S, Gupta A (2003) An elementary proof of a theorem of Johnson and Lindenstrauss. Random Struct Algorithms 22(1):60–65
21. Fedoruk J, Schmuland B, Johnson J, Heo G (2018) Dimensionality reduction via the Johnson-Lindenstrauss lemma: theoretical and empirical bounds on embedding dimension. J Supercomput 74(8):3933–3949
22. Fowler JE (2009) Compressive-projection principal component analysis. IEEE Trans Image Process 18(10):2230–42 (no JL Lemma)
23. Frankl P, Maehara H (1988) The Johnson–Lindenstrauss lemma and the sphericity of some graphs. J Comb Theory Ser B 44(3):355–362
24. Indyk P, Naor A (2007) Nearest-neighbor-preserving embeddings. ACM Trans Algorithms. https://doi.org/10.1145/1273340.1273347
25. Johnson WB, Lindenstrauss J (1982) Extensions of Lipschitz mappings into a Hilbert space. In: Conference in modern analysis and probability (New Haven, Conn., 1982). Also appeared in volume 26 of Contemp. Math. American Mathematical Society, Providence, RI, 1984, pp 189–206
26. Kane DM, Nelson J (2014) Sparser Johnson–Lindenstrauss transforms. J ACM 61(1):4
27. Khoa N, Chawla S (2012) Large scale spectral clustering using resistance distance and Spielman–Teng solvers. In: Ganascia JG, Lenca P, Petit JM (eds) Discovery science. Lecture notes in computer science, vol 7569. Springer, Berlin, Heidelberg, pp 7–21
28. Larsen KG, Nelson J (2014) The Johnson–Lindenstrauss lemma is optimal for linear dimensionality reduction. *CoRR*. arXiv:abs/1411.2404
29. Larsen KG, Nelson J (2016) Optimality of the Johnson–Lindenstrauss lemma. *CoRR*. arXiv:abs/1609.02094
30. Magen A (2002) Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications. In: RANDOM 02: proceedings of the 6th international workshop on randomization and approximation techniques. Springer, London, pp 239–253
31. Matousek J (2008) On variants of the Johnson–Lindenstrauss lemma. Random Struct Algorithms 33(2):142–156

32. Ostrovsky R, Rabani Y, Schulman LJ, Swamy C (2013) The effectiveness of lloyd-type methods for the k-means problem. J ACM 59(6):28:1–28:22 0.0000001 is epsilon so that epsilon square <target kmeans for k/target kmeans for k-1
33. Puy G, Tremblay N, Gribonval R, Vandergheynst P (2015) Random sampling of bandlimited signals on graphs. CoRR. arXiv:1511.05118
34. Sakai T, Imiya A (2009) Fast spectral clustering with random projection and sampling. In: Perner P (ed) Machine learning and data mining in pattern recognition. Lecture notes in computer science, Vol. LNAI 5632. Springer, Berlin, Heidelberg, pp 372–384
35. Sakai T, Imiya A (2011) Practical algorithms of spectral clustering: toward large-scale vision-based motion analysis. In: Wang L, Zhao G, Cheng L, Pietikäinen M (eds) Machine learning for vision-based motion analysis. Advances in pattern recognition. Springer, London, pp 3–26
36. Schulman LJ (2000) Clustering for edge-cost minimization (extended abstract). In: STOC 00: proceedings of the thirty-second annual ACM symposium on theory of computing. ACM, New York, NY, USA, pp 547–555
37. Shahid N, Perraudin N, Puy G, Vandergheynst P (2016) Compressive PCA for low-rank matrices on graphs. CoRR. arXiv:abs/1602.02070. no referernce to JL Lemma
38. Shang F, Jiao LC, Shi J, Gong M, Shang RH (2011) Fast density-weighted low-rank approximation spectral clustering. Data Min Knowl Discov 23(2):345–378
39. Sivakumar D (2002) Algorithmic derandomization using complexity theory. In: Proceedings of the 34th annual ACM symposium on the theory of computing. Montreal, Canada, pp 619–626
40. Terada Y (2014) Strong consistency of reduced k-means clustering. Scand J Stat 41(4):913–931
41. Tremblay N, Puy G, Gribonval R, Vandergheynst P (2016) Compressive spectral clustering. In: Proceedings of the 33rd international conference on machine learning, ICML 2016, New York City, NY, USA, June 19–24, 2016, pp 1002–1011. exploit JL Lemma indirectly
42. Venkatasubramanian S, Wang Q (2011) The Johnson–Lindenstrauss transform: an empirical study. SIAM, pp 164–173

Mieczysław A. Kłopotek studied computer science and obtained his PhD. From Dresden University of Technology (Germany). He habilitated at the Institute of Computer Science of Polish Academy of Science, where he runs research work currently as head of Laboratory for Foundations of Artificial Intelligence. He lectured among others at the Warsaw University of Technology. He co-authored applied analytical systems for business, among others in his capacity as director for advanced data analytics at the Netezza corporation, USA, as well as a number of experimental search engines, including nekst.pl. His areas of interest encompass artificial intelligence, reasoning, decision-making theory, machine learning, data, text and web mining, network mining, cluster analysis, managing uncertainty in knowledge-based systems, Bayesian networks, big data analysis, mathematical theory of evidence, programming on the Internet, computer vision, semantic web search engine construction, massively parallel databases.