Appendix A
Justification of the FCM Algorithm

We present here a justification for the formulas (3.58) and (3.59). Recall that these formulas represent a minimum of the target function.

We formulate a Lagrangian in order to determine the elements $u_{ij}$ of the assignment matrix

$$L(J_\alpha, \lambda) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij}^\alpha d^2(x_i, \mu_j) - \sum_{i=1}^{m} \lambda_i \left( \sum_{j=1}^{k} u_{ij} - 1 \right) \quad (A.1)$$

We obtain the equation system, given below, by setting partial derivatives of the Lagrangian to zero.

$$\frac{\partial L}{\partial u_{ij}} = 0 \Leftrightarrow \alpha u_{ij}^{\alpha-1} d^2(x_i, \mu_j) = \lambda_i \quad (a)$$

$$\frac{\partial L}{\partial \lambda_i} = 0 \Leftrightarrow \sum_{j=1}^{k} u_{ij} = 1 \quad (b)$$

The equation (a) is equivalent to the following

$$u_{ij} = \left[ \frac{\lambda_i}{\alpha d^2(x_i, \mu_j)} \right]^{\frac{1}{\alpha-1}}$$

Using (b), we can write

$$\left( \frac{\lambda_i}{\alpha} \right)^{\frac{1}{\alpha-1}} = \frac{1}{\sum_{j=1}^{k} d^{2/(1-\alpha)}(x_i, \mu_j)}$$

which implies the Eq. (3.61).
In order to determine the components of the prototype, let us define

\[ \gamma_j(\mu_j) = \sum_{i=1}^{m} u_{ij}^\alpha \| x_i - \mu_j \|_A^2 \]

Gradient of the function \( \gamma_j(\mu_j) \) with respect to the components of the vector \( \mu_j \) is of the form

\[ \nabla \gamma_j(\mu_j) = \sum_{i=1}^{m} u_{ij}^\alpha \nabla \| x_i - \mu_j \|_A^2 \]

\[ = -2A \left[ \sum_{i=1}^{m} u_{ij}^\alpha (x_i - \mu_j) \right] = 0 \]

By definition of the norm, the matrix \( A \) is positive definite which guarantees the existence of the inverse matrix \( A^{-1} \). Finally, we have

\[ \mu_j = \frac{\sum_{i=1}^{m} u_{ij}^\alpha x_i}{\sum_{i=1}^{m} u_{ij}^\alpha} \]

Of course, in order to check if the solution of the optimisation task, obtained in this way, is really a (local) minimum and not a saddle point, the Hessian of the function \( J_\alpha(U, M) \) needs to be investigated.
Appendix B
Matrix Calculus

A detailed discussion of the topics presented here can be found in the books [354, 398, 403, 413, 464].

B.1 Vectors and Their Properties

We denote with the symbol $\mathbf{e}$ a vector having each component equal 1. We denote with $\mathbf{e}_j$ a vector identical with the $j$th column of a unit matrix.

**Definition B.1.1** Let $\mathbf{x}$, $\mathbf{y}$ be $n$-dimensional vectors. They are

(a) orthogonal, which we denote $\mathbf{x} \perp \mathbf{y}$, if $\mathbf{x}^T \mathbf{y} = 0$,
(b) orthonormal, if they are orthogonal vectors of unit length each.

If $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is a set of linearly independent $n$-dimensional vectors, where $k \leq n$, then an orthogonal set $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ of vectors, spanning the same $k$-dimensional subspace of the space $\mathbb{R}^n$ as $V$ does, is obtained by application of the Gram-Schmidt orthogonalisation procedure described e.g. in [354].

B.2 Matrices and Their Properties

**Definition B.2.1** Let $A = [a_{ij}]_{m \times m}$ be a square matrix with real-valued entries, i.e. $A \in \mathbb{R}^{m \times m}$. We will call it

(a) non-negative (resp. positive), denoted by $A \geq 0$ (resp. $A > 0$), if all its elements are non-negative (resp. positive)
(b) diagonal, denoted by $A = \text{diag}(d_1, \ldots, d_m)$, if $a_{ij} = 0$ whenever $i \neq j$ and $a_{ii} = d_i$, $i = 1, \ldots, m$. If all diagonal elements are identical and equal to 1, $d_i = 1$, then $A$ is called unit matrix and is denoted with the symbol $\mathbb{I}$. 

© Springer International Publishing AG 2018
S. T. Wierzchoń and M. A. Kłopotek, Modern Algorithms of Cluster Analysis, Studies in Big Data 34, https://doi.org/10.1007/978-3-319-69308-8
(c) symmetric, if \( A = A^T \).
(d) orthogonal, if \( A^T A = A A^T = I \).
(e) normal, if \( A A^T = A^T A \). 

If \( A \) is an orthogonal matrix then \( | \text{det}(A) | = 1 \).

**Definition B.2.2** A symmetric matrix \( A \) with dimensions \( m \times m \) is called positive semidefinite, denoted by \( A \preceq 0 \), if for any non-zero vector \( v \in \mathbb{R}^m \) the following holds: \( v^T A v \geq 0 \). If we can sharpen this relation (\( v^T A v > 0 \) for each nonzero vector) then \( A \) is called positive definite, denoted with the symbol \( A \succ 0 \).

On the other hand, if for each non-zero vector \( v \) \( v^T A v \leq 0 \) holds, then we call \( A \) negative semidefinite. And if \( v^T A v < 0 \) holds, then we call \( A \) negative definite.

In practice, the Sylvester theorem is used to decide on definiteness of quadratic forms. Let us denote with \( \Delta_i, i = 1, \ldots, m \) the leading principal sub-determinants (leading principal minors) of the matrix \( A \), that is

\[ \Delta_1(A) = a_{11}, \quad \Delta_2(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \ldots, \quad \Delta_m(A) = \begin{vmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{vmatrix} \]

Matrix \( A \) is positive semidefinite if \( \Delta_i(A) \geq 0, i = 1, \ldots, m \). If, furthermore, all the leading principal minors of the matrix are positive, then it is positive definite. In the next section, see Lemma B.3.5, we present further characterisations of positive semidefinite matrices. For completeness, let us state that if \( (-1)^j \Delta_j(A) > 0 \) then matrix \( A \) is negative definite, and if \( (-1)^j \Delta_j(A) \geq 0 \) then matrix \( A \) is negative semidefinite. It is easily seen that if \( A \) is a positive (semi)definite matrix then \( B = -A \) is a negative (semi)definite matrix.

The Gram matrix is an important example of a positive semidefinite matrix.

**Definition B.2.3** If \( M = (m_1, \ldots, m_k) \) is a matrix of \( k \) column vectors of dimension \( n \) then \( G = M^T M \) is called Gram matrix.

The above-defined matrix \( G \) is a matrix of dimensions \( k \times k \) with elements \( g_{ij} = m_i^T m_j = g_{ji} \). Its determinant is non-negative.

**Definition B.2.4** If a matrix \( A \) with dimensions \( m \times m \) can be represented in the form \( A = B B^T \), where \( B \) is a non-negative matrix of dimensions \( m \times n \), then \( A \) is called a completely positive matrix.\(^1\) The minimal number of columns of \( B \), i.e. \( n \), ensuring the above factorisation of matrix \( A \) is called the factorisation index or \( cp \)-rank of the matrix \( A \).

**Definition B.2.5** The number

\[ \text{tr}(A) = \sum_{i=1}^{m} a_{ii} \]

is called matrix trace. It has the following properties:

(a) \( \text{tr} (A) = \text{tr} (A^T) \),
(b) \( \text{tr} (A + B) = \text{tr} (A) + \text{tr} (B) \),
(c) \( \text{tr} (ABC) = \text{tr} (BCA) = \text{tr} (CAB) \)

**Definition B.2.6** A non-negative matrix \( P \) is called a row-stochastic or right-stochastic matrix, if all elements of each row sum up to one. If the sum of all elements of each of its columns equals one then \( P \) is called column-stochastic or left-stochastic. If \( P \) is both column-stochastic and row-stochastic then we call it doubly stochastic matrix.

If \( A > 0 \) is a symmetric matrix, then by alternating the normalising operators of rows and columns we get a doubly stochastic matrix \( P \) \[433\]. Saying it differently, there exists a diagonal matrix \( D = \text{diag}(d) \) such that \( P = DAD \). Components of the vector \( d \) are equal to \( d_i = \sqrt{p_{ii}/a_{ii}} \). This method of computing the doubly stochastic matrix is called Sinkhorn-Knopp method. The paper [295] presents a quick algorithm for balancing a symmetric matrix \( A \), that is, an algorithm transforming it into doubly stochastic matrix \( P \), and presents a review of other related methods.

**Definition B.2.7** A stochastic matrix is called

(a) stable, if all of its rows are identical,
(b) column-wise accessible, if its each column contains at least one positive element.

**B.3 Eigenvalues and Eigenvectors**

**B.3.1 Basic Facts**

A square matrix \( A \) of dimensions \( m \times m \) possesses an eigen (or characteristic) value \( \lambda \) and an eigen (or characteristic) vector \( \mathbf{w} \neq \mathbf{0} \) if

\[
A\mathbf{w} = \lambda \mathbf{w}
\]

(B.1)

The pair \((\lambda, \mathbf{w})\), satisfying the above conditions is called eigenpair.

Equation (B.1) can be rewritten in the equivalent form

\[
\det(A - \lambda I_m) = 0
\]

(B.2)

where \( I_m \) is a unit matrix of dimensions \( m \times m \). Knowing that formula (B.2) is a \( m \)-degree polynomial we can state that the matrix \( A \) possesses \( m \) (not necessarily distinct) eigenpairs \((\lambda_i, \mathbf{w}_i)\). If all eigenvalues are distinct then we call them non-degenerate.
The set of all distinct eigenvalues
\[ \sigma(A) = \{\lambda_1, \ldots, \lambda_{m'}\}, \quad m' \leq m \]  \hspace{1cm} (B.3)
defines the spectrum of the matrix \(A\), and the quantity
\[ \rho(A) = \max_{1 \leq j \leq m} |\lambda_j| \]  \hspace{1cm} (B.4)
is called spectral radius of matrix \(A\).

If \(\lambda_1, \ldots, \lambda_m\) are eigenvalues of the matrix \(A\) of dimensions \(m\), then
(a) \[ \sum_{i=1}^{m} \lambda_i = \text{tr}(A), \]
(b) \[ \prod_{i=1}^{m} \lambda_i = \det(A), \]
(c) \(\lambda_1, \ldots, \lambda_m\) are eigenvalues of the matrix \(A^k\).

If \(A = \text{diag}(a)\) is a diagonal matrix with the diagonal identical by the vector \(a\), then its eigenvalues are elements of the vector \(a\), with the \(i\)th eigenpair of the form \((a_i, e_i)\), i.e. \(\lambda_i = a_i\), and \(e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T\) is the \(i\)th column of the unit matrix.

Lemma B.3.1 \textit{If} \((\lambda_i, w_i)\) \textit{is the} \(i\textit{th eigenpair of the matrix} A\), \textit{then eigenpairs of the matrix} \(c_1I + c_2A\) \textit{with} \(c_1, c_2\) \textit{being any values, for which} \(c_2 \neq 0\) \textit{holds, are of the form} \((c_1 + c_2\lambda_i, w_i)\) \hspace{1cm} \(\square\)

Let \(\lambda_{\text{max}}\) be the eigenvalue with the biggest module, i.e. \(|\lambda_{\text{max}}| = \max_{i=1,\ldots,m} |\lambda_i|\) and let \(w_{\text{max}}\) be the corresponding eigenvector. The pair \((\lambda_{\text{max}}, w_{\text{max}})\) is called the principal eigenpair.

If we know the eigenvector \(w_i\) of a symmetric matrix \(A\), then we can compute the corresponding eigenvalue \(\lambda_i\) from the equation
\[ \lambda_i = \frac{w_i^T A w_i}{w_i^T w_i} = R(A, w_i) \]  \hspace{1cm} (B.5)
The quantity \(R(A, x)\), defined by Eq. (B.5), where \(x\) is any non-zero vector, is called Rayleigh quotient. One can easily check that \(R(A, cx) = R(A, x)\) for any constant \(c \neq 0\).

Theorem B.3.1 \textit{Let} \(S_k\) \textit{denote} \(k\text{-dimensional subspace of the space} \mathbb{R}^m\) \textit{and let} \(x \perp S_k\) \textit{denote that} \(x\) \textit{is a vector orthogonal to any vector} \(y \in S_k\). \textit{Let} \(A \in \mathbb{R}^{m \times m}\) \textit{be a symmetric matrix with eigenvalues} \(\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m\). \textit{Then}
\[ \lambda_k = \max_{S_k} \min_{0 \neq x \perp S_k} R(A, x) \]  \hspace{1cm} (B.6)
\(\square\)

This is the so-called Courant-Fischer minimax theorem. It implies that
Appendix B: Matrix Calculus

(a) \( \lambda_1 = \min_{x \neq 0} \frac{x^T A x}{x^T x} \), \( w_1 = \arg \min_{x \neq 0} \frac{x^T A x}{x^T x} \)

(b) \( \lambda_2 = \min_{x \neq 0 \perp w_1} \frac{x^T A x}{x^T x} \), \( w_2 = \arg \min_{x \neq 0 \perp w_1} \frac{x^T A x}{x^T x} \)

... ... ...

(c) \( \lambda_m = \max_{x \neq 0} \frac{x^T A x}{x^T x} \), \( w_m = \arg \max_{x \neq 0} \frac{x^T A x}{x^T x} \)

In this way we obtain an estimate \( \lambda_1 \leq R(A, x) \leq \lambda_m \).

**Lemma B.3.2** If \( A \) is a symmetric square matrix, and \( x \neq 0 \) is a vector, then the value \( \mu \), minimising the expression \( \| A x - \mu x \| \), is equal to the Rayleigh quotient. \( \square \)

Let us mention some important properties of the eigenvectors:

(i) If \( A \) is a Hermitian matrix (in particular, a symmetric one with real-valued elements) then all its eigenvalues are real numbers.

(ii) If \( 0 \in \sigma(A) \), then \( A \) is a singular matrix (the one with determinant equal zero).

(iii) A square matrix \( A \) with positive elements has exactly one real-valued principal eigenvalue and all the elements of the corresponding eigenvector are of the same sign (Perron-Frobenius theorem).

(iv) Eigenvectors of a normal matrix with non-degenerate eigenvalues form a complete set and are orthonormal. This means that they are versors spanning an \( m \) dimensional vector space.

One applies Gram-Schmidt orthogonalisation in case of degenerate eigenvalues. In this way one can find a set of eigenvectors that form a complete set and are orthonormal.

(v) (Gershogorin theorem [354, Example 7.1.4]) Each eigenvalue \( \lambda \) of a square matrix \( A \) of dimension \( m \) fulfils at least one of the inequalities:

\[
|\lambda - a_{ii}| \leq r_i = \sum_{1 \leq j \leq m} |a_{ij}|
\]

i.e. \( \lambda \) lies inside at least one (complex) circle with a centre at the point \( a_{ii} \) and radius \( r_i \) being the sum of absolute values of non-diagonal elements of the \( i \)th row. \( \square \)

**Definition B.3.1** Matrices \( A \) and \( B \) are called similar, which we denote \( A \approx B \), if there exists a non-singular matrix \( X \) such that

\[
A = XBX^{-1}
\]

The mapping transforming the matrix \( B \) into the matrix \( A \) is called the similarity mapping. \( \square \)
Lemma B.3.3 If $A$ and $B$ are similar matrices then they have identical eigenvalues, and their eigenvectors satisfy the condition $w_B = X^{-1}w_A$.

Proof: Let $(\lambda, w_A)$ be an eigenpair of the matrix $A$, i.e. $XBX^{-1}w_A = \lambda w_A$ (because $A \approx B$). Let us multiply both sides of this equation by $X^{-1}$. Then we obtain $B(X^{-1}w_A) = \lambda(X^{-1}w_A)$, which implies the thesis. □

Definition B.3.2 The square matrix $A$ is called a diagonalisable matrix if and only if there exists a non-singular matrix $X$ of the same dimension as $A$, such that $X^{-1}AX$ is a diagonal matrix. If, furthermore, $X$ is an orthogonal matrix, i.e. $X^{-1} = X^T$, then $A$ is called an orthogonally diagonalisable matrix. □

Let $(\lambda_i, w_i), i = 1, \ldots, m$ denote the set of eigenpairs of the matrix $A$. The matrix is diagnosable if its eigenvectors are linearly independent. By substituting $X = (w_1, \ldots, w_m)$, i.e. by placing eigenvectors of the matrix $A$ in the columns of the matrix $X$ we obtain $X^{-1}AX = \text{diag}(\lambda_1, \ldots, \lambda_m) = \Lambda$.

Lemma B.3.4 A square matrix $A$ is a symmetric matrix if and only if it is orthogonally diagonalisable.

Proof: Let us restrict our considerations to the simpler necessary condition. If $A$ is an orthogonally diagonalisable matrix then there exists such an orthogonal matrix $X$ and a diagonal matrix $D$ that $A = XDX^T$. Hence $A^T = (XDX^T)^T = A$ □

Let us consider again the positive semi-definite matrices that were introduced in the preceding section.

Lemma B.3.5 The following conditions are equivalent for a symmetric matrix $A$:

(a) $A \succeq 0$.

(b) All eigenvalues of the matrix $A$ are non-negative.

(c) $A = C^T C$, where $C \in \mathbb{R}^{m \times n}$.

Proof: We will show that (a) implies (b), (b) implies (c), and (c) implies (a).

(a) $\Rightarrow$ (b): As $A$ is a symmetric matrix, hence all, its eigenvalues are real numbers. Let $(\lambda, w)$ be an eigenpair bound by the equation $Aw = \lambda w$. Let us perform left multiplication of both sides of this equation with the vector $w^T$. We obtain $w^TAw = \lambda w^Tw$. Both the left side of this equation and $w^Tw$ are non-negative numbers, hence $\lambda \geq 0$.

(b) $\Rightarrow$ (c): Let $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$ and let $W = (w_1, \ldots, w_m)$ be a matrix with columns being eigenvectors corresponding to eigenvalues $\lambda_1, \ldots, \lambda_m$. The Eq. (B.1) can be rewritten in matrix form as $AW = W \Lambda$. By performing right multiplication of this equation with matrix $W^T$ we obtain $AWW^T = W \Lambda W^T$. But eigenvectors of a symmetric matrix $A$ are orthonormal, hence $A = W \Lambda W^T$. As $\lambda_i \geq 0$, hence matrix $\Lambda$ can be represented as $\Lambda^{1/2}(\Lambda^{1/2})^T$. Therefore

2Because then its rank is equal $m$, which is a necessary condition for the existence of the matrix inverse to the given matrix $A$. 

Appendix B: Matrix Calculus

\[ A = W \Lambda^{1/2} (\Lambda^{1/2})^T W^T = W \Lambda^{1/2} (W \Lambda^{1/2})^T = C^T C \]

where \( C^T = W \Lambda^{1/2} \).

(c) ⇒ (a): Because of \( A = C^T C \), we have, for any non-zero vector \( v \),

\[ v^T A v = v^T C^T C v ⇔ v^T A v = y^T y ⇔ v^T A v \geq 0 \]

where \( y = C v \).

Let \( A_k = [a_{ij}] \), \( i, j \in \{1, \ldots, k\} \) denote a submatrix of the positive semi-definite matrix \( A \). Then, for each value \( 1 \leq k \leq m \) and each non-zero vector \( x \in \mathbb{R}^k \), the following conditions hold: (a) \( x^T A_k x \geq 0 \) and (b) \( \det(A_k) \geq 0 \).

**B.3.2 Left- and Right-Hand Eigenvectors**

The eigenvectors \( w \) of a matrix \( A \) that we have been talking about so far are called also right eigenvectors because they stand to the right of the matrix \( A \) in the defining Eq. (B.1). In an analogous way also the left eigenvectors can be defined.

\[ u^T A = \lambda u^T \quad (B.9) \]

By transposing this equation we see that the left eigenvector of the matrix \( A \) is the right eigenvector of the matrix \( A^T \). It is obvious that the distinction between left and right eigenvectors would be pointless if \( A \) were symmetric, because \( u = w \), that is—the left and right eigenvectors are identical. Hence, let us subsequently consider predominantly the non-symmetric or even non-normal matrices. Such a distinction is not necessary for eigenvalues, because due to Eq. (B.2) and due to \( \det(A) = \det(A^T) \) we see that in both cases one obtains the same eigenvalue.

By reasoning like in the proof of the Lemma B.3.3 we infer

**Lemma B.3.6** If \( A, B \) are similar matrices, then they have identical eigenvalues and their left eigenvectors are linked by the interelation \( u_B = X u_A \).

Lemmas B.3.3 and B.3.6 imply that for similar matrices \( A \) and \( B \) with \( A \), being symmetric, the left \( (u_B) \) and right \( (w_B) \) eigenvectors of the matrix \( B \) can be expressed in terms of the eigenvectors of \( v_A \) of the matrix \( A \) as follows:

\[ u_B = X v_A, \quad w_B = X^{-1} v_A \]

where \( X \) is a matrix of coefficients.
Lemma B.3.7  Let $A$ be a matrix of dimension $m$ possessing $m$ distinct left and right eigenvectors $\mathbf{u}_i, \mathbf{w}_i$. Then

$$A = \sum_{i=1}^{m} \lambda_i \mathbf{w}_i \mathbf{u}_i^T$$  \hspace{1cm} (B.10)

The above spectral representation implies the equation

$$A^n = \sum_{i=1}^{m} \lambda_i^n \mathbf{w}_i \mathbf{u}_i^T$$  \hspace{1cm} (B.11)

In particular, if $A$ is a symmetric matrix, we get

$$A^n = \sum_{i=1}^{m} \lambda_i^n \mathbf{v}_i \mathbf{v}_i^T$$  \hspace{1cm} (B.12)

where $\mathbf{v} = \mathbf{u} = \mathbf{w}$.

B.3.3 Determining Eigenvalues and Eigenvectors

Quick algorithms for computation of eigenvectors are indispensable for the spectral clustering methods. Many monographs have been devoted to this topic, e.g. [41, 413, 464]. Projection methods seem to play a key role here. Chapter 9 of the monograph [204] is devoted to them.

Here below we present the simplest method allowing to find quickly the principal eigenvector and then discuss its application for determining eigenpairs of the Laplacians.

B.3.3.1 The Power Method

As stated previously, eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ of a diagonalisable matrix $A$ are linearly independent. Therefore, they constitute a vector base in the space $\mathbb{R}^m$. Any vector $\mathbf{x}^{(0)} \in \mathbb{R}^m$ can be represented in the form

$$\mathbf{x}^{(0)} = c_1 \mathbf{x}_1 + \cdots + c_n \mathbf{x}_m$$

Eigenvector normalization footnote: We deliberately postponed the issue of normalization of the eigenvectors till this point because it becomes only here clear why some choices are done. One should be aware that if a vector $\mathbf{w}$ is a right eigenvector of a matrix $A$, according to Eq. (B.1), then also any vector $c \cdot \mathbf{w}$ for any non zero scalar $c$ is. Similarly, if a vector $\mathbf{u}$ is a left eigenvector of a matrix $A$, according to Eq. (B.9), then also any vector $c \cdot \mathbf{u}$ for any non zero scalar $c$ is. To get rid of such an ambiguity, we assume throughout this book, if not stated otherwise, that the vectors are normalised, that is, we select that right eigenvector $\mathbf{w}$, for which $\mathbf{w}^T \mathbf{w} = 1$ and the left eigenvector $\mathbf{u}$ having the corresponding normalised right eigenvector $\mathbf{w}$, for which $\mathbf{u}^T \mathbf{w} = 1$. Note that the right and left eigenvectors are normalised differently.
where $c_1, \ldots, c_m$ are scalars. By multiplying the above equation by $A^k, k = 1, 2, \ldots$, we get

$$A^k x^{(0)} = c_1 (A^k x_1) + \cdots + c_m (A^k x_m) = c_1 (A^k x_1) + \cdots + c_m (\lambda_m^k x_m) = c_1 \lambda_1^k x_1 + \sum_{j=2}^{k} c_j \left( \frac{\lambda_j}{\lambda_1} \right)^k x_j$$  \hspace{1cm} (B.13)

We exploit here the fact that $A x_j = \lambda_j x_j$, where $\lambda_j$ denotes the eigenvalue corresponding to the eigenvector $x_j$, and that whenever $\lambda_j$ is an eigenvalue of the matrix $A$, then $\lambda_j^k$ is an eigenvalue of the matrix $A^k$. If eigenvalues are sorted according to their decreasing module, then $|\lambda_j / \lambda_1| \leq 1$. So, if only $|\lambda_j / \lambda_1| < 1$ then

$$\lim_{k \to \infty} \frac{x^{(k)}}{\lambda_1^k} = \lim_{k \to \infty} \frac{A^k x^{(0)}}{\lambda_1^k} = c_1 x_1$$

that is, the series $\{x^{(k)}/\lambda_1^k \}$ converges with speed dependent on the quotient $|\lambda_2 / \lambda_1|$, to the vector $c_1 x_1$.

In practice, the successive approximations of the eigenvector are not computed from the equation $x^{(k)} = A^k x^{(0)}$, but rather iteratively, $x^{(k)} = A x^{(k-1)}, k = 1, \ldots$. In this way we do not need to compute successive powers of the matrix $A$. Instead, we multiply each time the matrix $A$ with the vector $x^{(k-1)}$, obtained in the preceding step. As $\|x^{(k)}\| \to 0$ when $|\lambda_1| < 1$ (or $\|x^{(k)}\| \to \infty$ when $|\lambda_1| > 1$), hence, to avoid overflow and underflow, the vector $x^{(k)}$ is normalised. If we denote by $y^{(k)}$ the product $A x^{(k-1)}$ then $x^{(k)} = y^{(k)}/m(y)$, where $m(y)$ is the first element of the vector $y$ with the largest module. If, for example, $y^{(k)} = (1, -5, 2, 5)$, then $m(y^{(k)}) = -5$. Thereby not only the division error is minimized in line 6 of pseudocode B.1, but also the total computational burden is minimized, see [354, p. 534]; [403]. The value $m(y^{(k)}) \to \lambda_1$ when $k \to \infty$, so we get also a method to determine an approximation of the principal eigenvalue. We choose usually, as the stopping criterion, the criterion of minimal correction, i.e. $\|x^{(k+1)} - x^{(k)}\| < \epsilon$, or $|m(y^{(k+1)}) - m(y^{(k)})| \leq \epsilon$. One initialises the vector $x^{(0)}$ randomly. Such an initialisation is particularly recommended if the main diagonal contains elements that are significantly larger than the other elements of the matrix $A$.

The method of computation of the principal eigenpair is presented as the pseudocode B.1.

**Remark B.3.1** The weakness of the power method is that if the principal eigenvalue is a complex number, then the approximation series $x^{(k)}$ is not convergent! \hspace{1cm} □

The power method returns the *principal* eigenpair, which means that the obtained approximation $\tilde{\lambda}$ can be a negative number. If our goal is to find the eigenvector corresponding to the maximal *positive* eigenvalue, then we proceed as follows: We construct a matrix $A' = A - \tilde{\lambda} I$. Its eigenvalues are $\lambda_j' = \lambda_j - \tilde{\lambda} \geq 0$. One can verify
Algorithm B.1 Power method returning the principal eigenpair of the matrix $A$

1: $k = 0.$
2: Initialise the vector $x^{(k)}$. { $x^{(k)}$ cannot be orthogonal to the principal eigenvector. In practice, it is sufficient that all its components have the same sign. }
3: while not done do
4: $y^{(k+1)} = Ax^{(k)}$
5: $\beta^{(k+1)} = m(y^{(k+1)})$
6: $x^{(k+1)} = y^{(k+1)}/\beta^{(k+1)}$
7: if $\|x^{(k+1)} - x^{(k)}\| \leq \epsilon$ then
8: done = true;
9: end if
10: $k = k + 1$
11: end while
12: return eigenvalue $\lambda \approx \beta^{(k+1)}$ and the corresponding eigenvector $x^{(k)}$

that if $\lambda_i$ is the $i$th eigenvalue of the matrix $A$, and $x_i$ is the corresponding eigenvector then

$$(A - \overline{\lambda}I)x_i = \lambda_i x_i - \overline{\lambda}x_i$$

Hence, the matrices $A$ and $A'$ have identical eigenvectors, while the principal eigenvalue of the matrix $A'$ corresponds to the maximal positive eigenvalue of the matrix $A$ increased by $o - \overline{\lambda}$. So, by repeating the power method, this time for the matrix $A'$, we find the eigenvector corresponding to the maximal positive eigenvalue of the matrix $A$. This approach is referred to as the “deflation method”.

**B.3.3.2 Determining the Eigenpairs of the Laplacian**

In case of spectral cluster analysis, we are interested in finding $p < m$ eigenvectors, corresponding to the lowest non-trivial eigenvalues of the Laplacian $L = D - A$, where $A \in \mathbb{R}^{n \times m}$ is a symmetric matrix, and $D$ is a diagonal matrix with elements $d_{ii} = \sum_{j=1}^{m} a_{ij}$. An interesting solution to this problem was proposed by Koren, Carmel and Harel in [302]. The method is outlined in the pseudocode B.2. Originally, the method was developed for drawing graphs, which means that only the first and the second (positive) non-trivial Laplacian eigenvalues are sought.

Let us note, first, that if $(\lambda_i, v_i)$ are eigenpairs of the matrix $A$, then $(g - \lambda_i, v_i)$ are eigenpairs of the matrix $A' = gI - A$. In particular, if $A$ is a Laplacian, and $g \geq \max_{1 \leq j \leq m} |\lambda_j|$, then by applying the deflation method one can determine the eigenpairs corresponding to the lowest eigenvalues of the Laplacian. Gershgorin theorem is used to estimate the value of $g$—see property (iv) from p. 323, according to which the eigenvalues of a matrix $A$ belong to the set sum of discs $K_i$ in the complex plane.

---
4 $A$ is interpreted as a neighbourhood matrix (similarity or adjacency matrix), see Sect. 5.2.
\[ \mathcal{K}_i = \{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|, \quad i = 1, \ldots, m \} \]

Eigenvalues of any Laplacian \( L = [l_{ij}] \) are non-negative. Furthermore \( l_{ii} = \sum_{j \neq i} |l_{ij}| \). Therefore, to estimate the largest eigenvalue, it is sufficient to compute the value

\[
g = 2 \cdot \max_{1 \leq i \leq m} l_{ii} \tag{B.14}
\]

\( L = D - A \), hence diag\((L) = \text{diag}(D) \), which allows to accelerate the computations significantly. It is the step 2 of the Algorithm B.2. Let us sort the eigenvalues of a Laplacian increasingly, and let \( \lambda_i \) be the \( i \)th eigenvalue in this order. Then, values \( \hat{\lambda}_i = g - \lambda_i \) (decreasing with growing \( i \)) are the eigenvalues of the matrix \( \hat{L} = gI - L \). This transformation is performed in step 3 of the Algorithm B.2.

The eigenvector, corresponding to the value \( \hat{\lambda}_1 = g \), is a normalised unit vector, that is—its components are equal \( 1/\sqrt{m} \). Therefore, one determines the eigenvectors corresponding to eigenvalues \( \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_p \)—steps 4–19. The initial, random approximation of the \( i \)th eigenvector (step 5) is subject to Gram-Schmidt orthogonalisation, so that the resultant vector will be orthogonal to the already determined approximations of eigenvectors (step 10). Next one step of the power method is executed.

Let us look at the stopping condition of the loop while–do. Formally, one can iterate the loop till the product \( \hat{v}v \) exceeds the threshold \( 1 - \epsilon \). In practice, one observes stabilisation after a number of iterations, \( \hat{v}^Tv = \text{const} \). Therefore, the formal condition was replaced by a more natural condition \( \text{prev} - \text{next} = 0 \), where \( \text{next} \) (resp. \( \text{prev} \)) is the current (resp. previous) value of the product \( \hat{v}^Tv \).

The only serious computational burden of the algorithm is determination of the product \( \hat{L}v = \hat{D}v - Av \). Notice, however, that \( \hat{D}v \) is a vector with components of the form \( (g - d_i)v_i \), and \( A \) is a sparse matrix.

### B.4 Norms of Vectors and Matrices

A tool allowing to measure the “size” of a vector \( x \in \mathbb{R}^m \) is its norm \( \|x\|_p \), defined as follows:

\[
\|x\|_p = \left( \sum_{i=1}^{m} |x_i|^p \right)^{1/p}, \quad p = 1, 2, \ldots \tag{B.15}
\]

If we set \( p = 1 \), then we obtain the so-called Manhattan norm (called also taxicab metric, rectilinear distance, city block distance, Manhattan distance, or Manhattan length), \( \|x\|_1 = \sum_i |x_i| \), while \( \|x\|_2 \) is called Euclidean norm, or simply—vector
Algorithm B.2 Applying power method to determine the first \( p \leq m \) eigenvectors of the Laplacian

**Input:** \( A \)—neighbourhood matrix, \( V = [v_{ij}]_{m \times p} \)—a matrix of randomly generated and normalised columns; elements of the first column are of the form \( v_{i1} = 1/\sqrt{m}, i = 1, \ldots, m \).

1: \( L = D - A \), where \( D \) is the degree matrix
2: \( g = \max_{1 \leq i \leq m} \left( l_{ii} + \sum_{j \neq i} |l_{ij}| \right) \)
3: \( \hat{L} = gI - L \) \{inversion of the order of the eigenvalues\}
4: \( \text{for } i = 2 \text{ to } p \text{ do} \)
5: \( \hat{v}_i = V(:, i) \) \{ith column of the matrix \( V \)\}
6: \( \text{prev} = 1, \text{next} = 0, \text{done} = \text{false} \)
7: \( \text{while} (\text{not} \ \text{done}) \text{ do} \)
8: \( v \leftarrow \hat{v}_i \)
9: \( \text{for } j = 1 \text{ to } i - 1 \text{ do} \)
10: \( v = v - (v^Tv_j)v_j \) \{Gram-Schmidt orthogonalisation\}
11: \( \text{end for} \)
12: \( \hat{v} = \hat{L}v \)
13: \( \hat{v} = \hat{v}/\|\hat{v}\| \)
14: \( \text{next} = \hat{v}^Tv \)
15: \( \text{done} = |\text{prev} - \text{next}| < \epsilon \)
16: \( \text{prev} = \text{next} \)
17: \( \text{end while} \)
18: \( V(:, i) = \hat{v} \) \{saving the vector \( \hat{v} \) in the ith column of the matrix \( V \)\}
19: \( \text{end for} \)

length. Finally, \( p = \infty \) corresponds to the maximum norm, also known as supremum norm, sup norm, the Chebyshev norm, the infinity norm or the “uniform norm”:

\[
\|x\|_{\infty} = \max_{1 \leq i \leq m} x_i \quad (B.16)
\]

The case of \( p \in (0, 1) \) was investigated exhaustively by Aggarwal, Hinneburg and Keim in the paper [8], where they suggest its high usefulness.

Given a rectangular matrix \( A \in \mathbb{R}^{n \times m} \), one defines either the Frobenius norm

\[
\|A\|_F = \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2} \quad (B.17)
\]

or one uses a matrix norm induced by a vector norm

\[
\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \sup_{\|x\|_p = 1} \|Ax\|_p \quad (B.18)
\]

In particular:

(i) \( \|A\|_1 = \max_{1 \leq j \leq m} \sum_{i=1}^{n} |a_{ij}| \) is the maximum value of the sum of modules of column elements,
(ii) \( \|A\|_2 = \sqrt{\lambda_{\text{max}}} \), where \( \lambda_{\text{max}} \) is the maximal eigenvalue of the matrix \( A^T A \); this norm is called spectral norm,

(iii) \( \|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^m |a_{ij}| \) is the maximum value of the sum of modules of row elements.

If \( A \) is a diagonal matrix, \( A = \text{diag}(a_1, \ldots, a_m) \), then

\[
\|A\|_p = \max_{1 \leq j \leq m} |a_j|, \quad p = 1, 2, \ldots
\]

More information on this subject can be found for example, in Chap. 5 of the monograph [354].

B.5 Kernel Trick and the Euclidean Distance Matrix

Kernel based clustering methods, as described in Sects. 2.5.7 and 3.1.5.5 exploit frequently the so-called kernel-trick, as described by formula (2.49), in that a kernel matrix \( K \) is used. Its elements \( k_{ij} = \Phi(x_i)^T \Phi(x_j) = K(x_i, x_j) \) represent dot-products of coordinate vectors of data points \( x_i, x_j \) transformed via the function \( \Phi \) from the original data space to the feature space. The essence of the kernel trick is not to have the function \( \Phi \) in an explicit form and to work with the entries \( k_{ij} \) instead. But the question should be raised what properties the kernel matrix should have in order to (1) be really a matrix of dot products and (2) to enable to recover function \( \Phi \) at the data points from the kernel matrix.

These questions may seem to be pretty easy: Let \( Y \) be a matrix \( Y = (\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_m))^T \). Then apparently \( K = YY^T \). Hence for any non-zero vector \( u \),

\[
u^T Ku = u^T YY^T u = (Y^T u)^T Y^T u = y^T y \geq 0
\]

where \( y = Y^T u \) so \( K \) must be positive semidefinite. But a matrix is positive semidefinite iff all its eigenvalues are non-negative. Furthermore, all its eigenvectors are real numbers.

So to identify \( \Phi \) at data points, one has to find all eigenvalues \( \lambda_l, l = 1, \ldots, m \) and corresponding eigenvectors \( v_l \) of the matrix \( K \). If all eigenvalues are hereby non-negative, then construct the matrix \( Y \) that has as columns the products \( \sqrt{\lambda_l} v_l \). Rows of this matrix (up to permutations) are the values of the function \( \Phi \) at data points \( 1, \ldots, m \). It may be verified that kernel-\( k \)-means algorithm with the above \( K \) and ordinary \( k \)-means for \( Y \) would yield same results.

Closely related is the following issue: For algorithms like \( k \)-means, instead of the kernel matrix the distance matrix \( D \) between the objects may be available, being the Euclidean distance matrix in the feature space. We will call \( D \) Euclidean matrix. The question is now: (3) can we obtain the matrix \( K \) from such data?

A number of transformations yielding the required kernel matrix has been proposed. The answer to the third question seems to be easily derivable from the paper

---

by Balaji et al. [43]. One should use the transformation

\[ K = -\frac{1}{2} (I - \frac{11^T}{m}) D_{sq} (I - \frac{11^T}{m}) \]

(where \( D_{sq} \) is a matrix containing as entries squared distances from \( D \)) a result going back to a paper by Schoenberg (1932).

A generally accepted proof of a more general transformation can be found in the paper by Gower [208, Theorem 2, p. 5], who generalises the above result of Schoenberg (1935) to

\[ K = (I - 1s^T)(-\frac{D_{sq}}{2})(I - s1^T) \]

for an appropriate choice of \( s \).

Let us recall that a matrix \( D \in \mathbb{R}^{m \times m} \) is an Euclidean distance matrix between points \( 1, \ldots, m \) if and only if there exists a matrix \( X \in \mathbb{R}^{m \times n} \) rows of which \( (x_1^T, \ldots, x_m^T) \) are coordinate vectors of these points in an \( n \)-dimensional Euclidean space and \( d_{ij} = \sqrt{(x_i - x_j)^T(x_i - x_j)} \)

Gower in [209] proposes:

**Lemma B.5.1** \( D \) is Euclidean iff the matrix \( F = (I - 1s^T)(-\frac{1}{2})D_{sq}(I - s1^T) \) is positive semidefinite for any vector \( s \) such that \( s^T 1 = 1 \).

Let \( D \in \mathbb{R}^{m \times m} \) be a matrix of Euclidean distances between objects. Let \( D_{sq} \) be a matrix of squared Euclidean distances \( d_{ij}^2 \) between objects with identifiers \( 1, \ldots, m \). This means that there must exist a matrix \( X \in \mathbb{R}^{m \times n} \) for some \( n \) rows of which represent coordinates of these objects in an \( n \)-dimensional space. If \( E = XX^T \) (\( E \) with dimensions \( m \times m \)), then \( d_{ij}^2 = e_{ii} + e_{jj} - 2e_{ij} \). We will call the matrix \( X \) an embedding of \( D \). A matrix can be called Euclidean if and only if an embedding exists.

As a rigid set of points in Euclidean space can be moved (shifted, rotated, flipped symmetrically) without changing their relative distances, there may exist many other matrices \( Y \) rows of which represent coordinates of these same objects in the same \( n \)-dimensional space after some isomorphic transformation. Let us denote the set of all such embeddings \( \mathcal{E}(D) \). And if a matrix \( Y \in \mathcal{E}(D) \), then for the product \( F = YY^T \) we have \( d_{ij}^2 = f_{ii} + f_{jj} - 2f_{ij} \). We will say that \( F \in \mathcal{E}_{dp}(D) \).

For an \( F \in \mathcal{E}_{dp}(D) \) define a matrix \( G = F + \frac{1}{2} D_{sq} \). Hence \( F = G - \frac{1}{2} D_{sq} \). Obviously then

---

6To follow the convention of Gower and other authors, in the rest of this appendix we use \( e \) instead of \( e \).

7Gower does not consider flipping.
\( d_{ij}^2 = f_{ii} + f_{jj} - 2f_{ij} \)  
\( = (g_{ii} - \frac{1}{2}d_{ii}^2) + (g_{jj} - \frac{1}{2}d_{jj}^2) - 2(g_{ij} - \frac{1}{2}d_{ij}^2) \)  
\( = g_{ii} + g_{jj} - 2g_{ij} + d_{ij}^2 \)  

(B.19)

(B.20)

(B.21)

(as \( d_{jj} = 0 \) for all \( j \)). This implies that

\[ 0 = g_{ii} + g_{jj} - 2g_{ij} \]  

(B.22)

that is

\[ g_{ij} = \frac{g_{ii} + g_{jj}}{2} \]  

(B.23)

So \( G \) is of the form

\[ G = \mathbf{g} \mathbf{1}^T + \mathbf{1} \mathbf{g}^T \]  

(B.24)

with components of \( \mathbf{g} \in \mathbb{R}^m \) equal \( g_i = \frac{1}{2}g_{ii} \).

Therefore, to find \( F \in \mathcal{E}_{dp}(D) \) for an Euclidean matrix \( D \) we need only to consider matrices deviating from \(-\frac{1}{2}D_{sq}\) by \( \mathbf{g} \mathbf{1}^T + \mathbf{1} \mathbf{g}^T \) for some \( \mathbf{g} \). Let us denote with \( \mathcal{G}(D) \) the set of all matrices \( F \) such that \( F = \mathbf{g} \mathbf{1}^T + \mathbf{1} \mathbf{g}^T - \frac{1}{2}D_{sq} \). So for each matrix \( F \) if \( F \in \mathcal{E}_{dp}(D) \) then \( F \in \mathcal{G}(D) \), but not vice versa.

For an \( F \in \mathcal{G}(D) \) consider the matrix \( F^* = (\mathbf{I} - \mathbf{1}s^T)F(\mathbf{I} - \mathbf{1}s^T)^T \). we obtain

\[ F^* = (\mathbf{I} - \mathbf{1}s^T)(\mathbf{1} \mathbf{g}^T + \mathbf{g} \mathbf{1}^T - \frac{1}{2}D_{sq})(\mathbf{I} - \mathbf{1}s^T)^T \]  

(B.25)

(B.26)

(B.27)

Let us investigate \( (\mathbf{I} - \mathbf{1}s^T)\mathbf{1}g^T(\mathbf{I} - \mathbf{1}s^T)^T \):

\[ (\mathbf{I} - \mathbf{1}s^T)\mathbf{1}g^T(\mathbf{I} - \mathbf{s}s^T) = \mathbf{1}g^T - \mathbf{1}g^T \mathbf{s}s^T - \mathbf{1}s^T \mathbf{1}g + \mathbf{1}s^T \mathbf{1}g^T \]  

(B.28)

Let us make the following choice (always possible) of \( \mathbf{s} \) with respect to \( \mathbf{g} \): \( \mathbf{s}^T \mathbf{1} = 1 \), \( \mathbf{s}^T \mathbf{g} = 0 \).

Then we obtain from the above equation

\[ (\mathbf{I} - \mathbf{1}s^T)\mathbf{1}g^T(\mathbf{I} - \mathbf{s}s^T) = \mathbf{1}g^T - 10\mathbf{1}^T - \mathbf{1}g^T + \mathbf{1}s^T \mathbf{1} \cdot \mathbf{1}^T = \mathbf{00}^T \]  

(B.29)

By analogy

\[ (\mathbf{I} - \mathbf{1}s^T)\mathbf{1}g^T(\mathbf{I} - \mathbf{1}s^T)^T = ((\mathbf{I} - \mathbf{1}s^T)\mathbf{1}g^T(\mathbf{I} - \mathbf{s}s^T))^T = \mathbf{00}^T \]  

(B.30)
By substituting (B.29) and (B.30) into (B.27) we obtain
\[ F^* = (I - 1s^T)F(I - 1s^T)^T = -\frac{1}{2}(I - 1s^T)Dsq(I - 1s^T)^T \] (B.31)

So for any \( g \), hence an \( F \in G(D) \) we can find an \( s \) such that:
\[ (I - 1s^T)F(I - 1s^T)^T = -\frac{1}{2}(I - 1s^T)Dsq(I - 1s^T)^T. \]

For any matrix \( F = -\frac{1}{2}(I - 1s^T)Dsq(I - 1s^T)^T \) for some \( s \) with \( 1^Ts = 1 \) we say that \( F \) is in multiplicative form or \( F \in M(D) \).

If \( F = YY^T \), that is \( F \) is decomposable, then also \( F^* = (I - 1s^T)YY^T(I - 1s^T)^T = ((I - 1s^T)Y)((I - 1s^T)Y)^T = Y^*Y^*^T \) is decomposable. But
\[ Y^* = (I - 1s^T)Y = Y - 1s^T \]
where \( v = Y^Ts \) is a shift vector by which the whole matrix \( Y \) is shifted to a new location in the Euclidean space. So the distances between objects computed from \( Y^* \) are the same as those from \( Y \), hence if \( F \in E_{dp}(D) \), then \( Y^* \in E(D) \).

Therefore, to find a matrix \( F \in E_{dp}(D) \), yielding an embedding of \( D \) in the Euclidean \( n \)-dimensional space we need only to consider matrices of the form
\[ -\frac{1}{2}(I - 1s^T)Dsq(I - 1s^T)^T, \]
subject to the already stated constraint \( s^T1 = 1 \), that is ones from \( M(D) \).

So we can conclude: If \( D \) is a matrix of Euclidean distances, then there must exist a positive semidefinite matrix \( F = -\frac{1}{2}(I - 1s^T)Dsq(I - 1s^T) \) for some vector \( s \) such that \( s^T1 = 1, \det((I - 1s^T)) = 0 \) and \( Dsqs \neq 0 \). So if \( D \) is an Euclidean distance matrix, then there exists an \( F \in M(D) \cap E_{dp}(D) \).

Let us investigate other vectors \( t \) such that \( t^T1 = 1 \),

Note that
\[ (I - 1t^T)(I - 1s^T) = I - 1t^T - 1s^T + 1t^T1s^T \] (B.33)
\[ = I - 1t^T - 1s^T + 1s^T \] (B.34)
\[ = I - 1t^T \] (B.35)

Therefore, for a matrix \( F \in M(D) \)
\[ (I - 1t^T)F(I - 1t^T)^T = -\frac{1}{2}(I - 1t^T)(I - 1s^T)Dsq(I - 1s^T)^T(I - 1t^T)^T \] (B.36)
\[ = -\frac{1}{2}(I - 1t^T)Dsq(I - 1t^T)^T \] (B.37)

But if \( F = YY^T \in E_{dp}(D) \), then

\[ F' = (I - 1t^T)F(I - 1t^T)^T \quad (B.38) \]
\[ = (I - 1t^T)YY^T(I - 1t^T)^T \quad (B.39) \]
\[ = (Y - 1(t^TY))(Y - 1(t^TY))^T \quad (B.40) \]

and hence each \( -\frac{1}{2}(I - 1t^T)D_{sq}(I - 1t^T)^T \) is also in \( \mathcal{E}_{dp}(D) \), though with a different placement (by a shift) in the coordinate systems of the embedded data points. So if one element of \( \mathcal{M}(D) \) is in \( \mathcal{E}_{dp}(D) \), then all of them are.

So we have established that: if \( D \) is an Euclidean distance matrix, this means there is matrix \( X \) such that rows are coordinates of objects \( i \) an Euclidean space with distances as in \( D \), then there exists a decomposable matrix \( F = YY^T \in \mathcal{E}_{dp}(D) \) which is in \( \mathcal{G}(D) \), \( \mathcal{E}_{dp}(D) \subset \mathcal{G}(D) \). For each matrix in \( \mathcal{G}(D) \cap \mathcal{E}_{dp}(D) \) there exists a multiplicative form matrix in \( \mathcal{M}(D) \cap \mathcal{E}_{dp}(D) \). But if it exists, all multiplicative forms are there: \( \mathcal{M}(D) \subset \mathcal{E}_{dp}(D) \)

In this way we have proven the only-if-part of Theorem B.5.1 of Gower.

Gower [208] makes the following remark: \( F = (I - 1s^T)(-\frac{1}{2}D_{sq})(I - 1s^T) \) is to be positive semidefinite for Euclidean \( D \). However, for non-zero vectors \( u \)

\[ u^TFu = -\frac{1}{2}u^T(I - 1s^T)D_{sq}(I - 1s^T)^Tu = -\frac{1}{2}((I - 1s^T)^Tu)^TD_{sq}((I - 1s^T)^Tu) \quad (B.41) \]

But \( D_{sq} \) is known to be not negative semidefinite, so that \( F \) would not be positive semidefinite in at least the following cases: \( \det((I - 1s^T)) \neq 0 \) and \( D_{sq}s = 0 \). So neither can hold.

As we can see from the first case above, \( F \), given by \( F = -0.5(I - 1s^T)D_{sq}(I - 1s^T)^T \) does not need to identify uniquely a matrix \( D \), as \( I - 1s^T \) is not invertible. Though of course it identifies a \( D \) that is an Euclidean distance matrix.

Let us now demonstrate the missing part of Gower’s proof that \( D \) is uniquely defined given a decomposable \( F \).

So assume that for some \( D \) (of which we do not know if it is Euclidean, but is symmetric and with zero diagonal), \( F = -0.5(I - 1s^T)D_{sq}(I - 1s^T)^T \) and \( F \) is decomposable that is \( F = YY^T \). Let \( D(Y) \) be the distance matrix derived from \( Y \) (that is the distance matrix for which \( Y \) is an embedding). That means \( F \) is decomposable into properly distant points with respect to \( D(Y) \). And \( F \) is be in additive form with respect to it, that is \( F \in \mathcal{G} \cap \mathcal{D}(Y) \). Therefore there must exist some \( s' \) such that the \( F' = -0.5(I - 1s'^T)D(Y)_{sq}(I - s'^1T) \) as valid multiplicative form with respect to \( D(Y) \), and it holds that \( F' = (I - 1s'^T)F(I - s'^1T) \). But recall that \( (I - 1s'^T)F(I - s'^1T) = (I - 1s'^T)(-0.5(I - 1s'^T)D_{sq}(I - s'^1T))(I - s'^1T) = -0.5((I - 1s'^T)(I - 1s'^T))D_{sq}((I - 1s'^T)(I - 1s'^T))^T = -0.5(I - 1s'^T)D_{sq}(I - s'^1T). \)

Hence \(-0.5(I - 1s'^T)D_{sq}(I - s'^1T) = -0.5(I - 1s'^T)D(Y)_{sq}(I - s'^1T). \)

So we need to demonstrate that for two two symmetric matrices with zero diagonals \( D, D' \) such that
\[-\frac{1}{2}(\mathbf{I} - \mathbf{s}s^T)D_{sq}(\mathbf{I} - \mathbf{s}s^T) = -\frac{1}{2}(\mathbf{I} - \mathbf{s}s^T)D'_{sq}(\mathbf{I} - \mathbf{s}s^T)\]

the equation \( \mathbf{D} = D' \) holds.

It is easy to see that

\[-\frac{1}{2}(\mathbf{I} - \mathbf{s}s^T)(D_{sq} - D'_{sq})(\mathbf{I} - \mathbf{s}s^T) = \mathbf{00}^T\]

Denote \( \Delta = D_{sq} - D'_{sq} \).

\[(\mathbf{I} - \mathbf{s}s^T)\Delta(\mathbf{I} - \mathbf{s}s^T) = \mathbf{00}^T\]

\[
\Delta - \mathbf{s}s^T\Delta - \Delta\mathbf{s}s^T + \mathbf{s}s^T\Delta\mathbf{s}s^T = \mathbf{00}^T
\]

With \( \overline{\Delta} \) denote the vector \( \Delta\mathbf{s} \) and with \( c \) the scaler \( \mathbf{s}s^T\Delta\mathbf{s} \). So we have

\[
\Delta - \overline{\Delta}^T - \overline{\Delta}\mathbf{1}\mathbf{1}^T + c\mathbf{1}\mathbf{1}^T = \mathbf{00}^T
\]

So in the row \( i \), column \( j \) of the above equation we have: \( \delta_{ij} + c - \overline{\delta}_i - \overline{\delta}_j = 0 \). Let us add cells \( ii \) and \( jj \) and subtract from them cells \( ij \) and \( ji \). \( \delta_{ii} + c - \overline{\delta}_i - \overline{\delta}_i + \delta_{jj} + c - \overline{\delta}_j - \delta_{ij} - c + \overline{\delta}_i + \overline{\delta}_j - \delta_{ji} + c + \overline{\delta}_j + \overline{\delta}_i = \delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji} = 0 \).

But as the diagonals of \( \mathbf{D} \) and \( \mathbf{D}' \) are zeros, hence \( \delta_{ii} = \delta_{jj} = 0 \). So \( -\delta_{ij} - \delta_{ji} = 0 \).

But \( \delta_{ij} = \delta_{ji} \) because \( \mathbf{D}, \mathbf{D}' \) are symmetric. Hence \(-2\delta_{ij} = 0\) so \( \delta_{ji} = 0 \). This means that \( \mathbf{D} = \mathbf{D}' \).

This means that \( \mathbf{D} \) and \( \mathcal{D}(\mathbf{Y}) \) are identical, hence that decomposition is sufficient to prove Euclidean space embedding and yields this embedding. This proves the if-part of Gower’s Theorem B.5.1.
Appendix C
Personalized PageRank Vector

In case of undirected graphs, the PageRank vector corresponds to a stationary distribution of the random walk, described by a transfer matrix of a special form, see the Remark C.1.1 below. We are interested in a special case of such a random walk, which starts from a particular initial node. Knowledge of such stationary distributions allows to construct a stationary distribution for any stochastic initial vector. We discuss below some specific features of such a random walk. Review of recent developments and various forms of the PageRank can be found e.g. in [202].

C.1 Basic Notions and Interdependences

Let $\Gamma = (V, E)$ be an undirected and connected graph, and $P = AD^{-1}$ be a (column) stochastic transition matrix, describing random walk in this graph. As before, $A$ is the neighbourhood matrix representing links in graph $\Gamma$, and $D$ is the diagonal degree matrix. The PageRank vector is defined, as proposed by Page and Brin, in [379], as the vector $\rho(s, \beta)$ being the solution to the equation

$$\rho(s, \beta) = \beta s + (1 - \beta) P \rho(s, \beta) \quad \text{(C.1)}$$

where $\beta \in (0, 1]$ is the so-called damping factor, and $s$ is the starting vector. In the original formulation, $s$ is a vector with all elements equal to $1/|V|$. If only some elements of this vector are positive, while the other ones are equal zero, then we talk about the personalized PageRank vector. The set $\text{supp}(s) = \{v \in V : s(v) > 0\}$ indicates the range of personalization. Subsequently, we will denote a personalised PageRank vector with the symbol $p(s, \beta)$.

Lemma C.1.1 Let $G(V, E)$ be a connected and aperiodic undirected graph and let $d = (d_1, \ldots, d_m)^T$ denote the vector of degrees. Then the vector
\[ \pi = \frac{d}{\sum_i d_i} = \frac{d}{\text{vol} V} \]  

(C.2)

represents a stationary distribution of a Markov chain with transfer matrix \( P \).

Proof: \( D^{-1}d = e \), hence

\[ Pd = AD^{-1}d = Ae = d \]

i.e. \( d \) is a right eigenvector of the matrix \( P \) (note that the last equality is a consequence of the symmetry of the matrix \( A \)). By normalising this vector we obtain a stochastic vector \( \pi \) of the above-mentioned form.

Lemma C.1.2 PageRank vector possesses the following properties:

(a) \( \rho(s, 1) = s \),
(b) If \( \pi \) is a stationary distribution with components of the form (C.2) then \( \rho(\pi, \beta) = \pi \) holds for any value of the parameter \( \beta \in (0, 1] \).
(c) \( \|\rho(s, \beta)\|_1 = 1 \) if only \( \|s\|_1 = 1 \).

Remark C.1.1 Please pay attention that in the original formulation of Page and Brin \( \Gamma \) is a directed graph, which requires a more careful transformation of the connection matrix into a stochastic matrix. Furthermore, in order to ensure that the corresponding transfer matrix has a single principal value, the authors modify the stochastic matrix \( P \) to the form \( P' = \beta se^T + (1 - \beta)P \). If we denote with \( \rho(s, \beta) \) the eigenvector, corresponding to the principal eigenvalue (equal 1) of the matrix \( P' \), we can easily check that, in fact, it fulfills the conditions of the Eq. (C.1), i.e.

\[ P'\rho(s, \beta) = \beta se^T \rho(s, \beta)) + (1 - \beta)P\rho(s, \beta) \]

because \( e^T \rho(s, \beta) = 1 \).

Many authors assume that \( \rho \) and \( s \) are row vectors. In such a case the Eq. (C.1) is of the form

\[ \rho(s, \beta) = \beta s + (1 - \beta)\rho(s, \beta)P^T \]

i.e. the matrix \( P^T \) is defined as the product \( D^{-1}A \).

Remark C.1.2 Let us stress that the symbols \( \rho(s, \beta) \) and \( p(s, \beta) \) denote the same solution of the equation system (C.1). We introduce them solely for the convenience of the reader. When we use the symbol \( \rho(s, \beta) \), we have in mind the global PageRank vector, that is, a solution obtained for the positive vector \( s \), whereas the symbol \( p(s, \beta) \) is intended to mean the personalised PageRank vector that is the solution of equation system (C.1) for the nonnegative starting vector \( s \).

The transfer matrix \( P \), which appears in equation (C.1), was replaced in the paper [22] by the lazy random walk matrix of the form

\[ \hat{P} = \frac{1}{2}(I + AD^{-1}) \]

(C.3)
In such a case, the personalised PageRank vector is a vector \( p(s, \alpha) \), for which the equation below holds:

\[
p(s, \alpha) = \alpha s + (1 - \alpha) \hat{P} p(s, \alpha) = \alpha s + (1 - \alpha) \frac{1}{2}(I + \hat{P}) p(s, \alpha)
\]  

(C.4)

**Lemma C.1.3** If \( p(s, \alpha; \hat{P}) \) is a solution of the equation system (C.4), and \( p(s, \beta; P) \) is a solution of the system (C.1), and if \( \beta = \frac{2\alpha}{1 + \alpha} \), then \( p(s, \alpha; \hat{P}) = p(s, \beta; P) \).

**Proof:** By transforming the equation (C.4), we obtain

\[
\frac{1 + \alpha}{2} p(s, \alpha; \hat{P}) = \alpha s + \frac{(1 - \alpha)}{2} P p(s, \alpha; \hat{P})
\]

By multiplying both sides of the above equation by \( \frac{2}{1 + \alpha} \) and noticing that \( (1 - \alpha)/(1 + \alpha) = 1 - \beta \), where \( \beta = 2\alpha/(1 + \alpha) \), we obtain the thesis. □

The above lemma demonstrates that the Eqs. (C.1) and (C.4) possess identical solutions if we apply in both cases the same starting vector and the damping factors satisfy the condition \( \beta = 2\alpha/(1 + \alpha) \), and matrices \( P \) and \( \hat{P} \) correspond to one another, according to Eq. (C.3). Furthermore, Eqs. (C.1) and (C.4), defining \( p(s, \beta) \) and \( p(s, \alpha) \), have exactly the same formal form. Hence, algebraic considerations, as well as algorithms derived for \( p(s, \beta) \) and \( \beta, P \), can be mapped by simple symbol substitution into \( p(s, \alpha) \) and \( \alpha, \hat{P} \) and vice versa. Therefore, we will limit ourselves to seeking the distributions of \( p \) for \( \alpha, \hat{P} \) only.

**Lemma C.1.4** A PageRank vector, personalised with respect to the starting vector \( s \), is a sum of the geometric series:

\[
p(s, \alpha) = \alpha \sum_{t=0}^{\infty} (1 - \alpha)^t \hat{P}^t s = \alpha s + \alpha \sum_{t=1}^{\infty} (1 - \alpha)^t \hat{P}^t s
\]

(C.5)

**Proof:** Let us rewrite the Eq. (C.4) in the form

\[
[I - (1 - \alpha)\hat{P}]p(s, \alpha) = \alpha s
\]

This implies that \( p(s, \alpha) = \alpha[I - (1 - \alpha)\hat{P}]^{-1} s \) if there exists the matrix \( [I - (1 - \alpha)\hat{P}]^{-1} \). Theorem 1.7 in [270] leads to the conclusion that if \( X \) is a square matrix such that \( X^n \to 0 \) when \( n \to \infty \), then the matrix \( (I - X) \) possesses an inverse matrix of the form

\[
(I - X)^{-1} = \sum_{t=0}^{\infty} X^t
\]

Substituting the matrix \( (1 - \alpha)\hat{P} \) for \( X \) we can check that the condition of the theorem is fulfilled, hence the conclusion. □
Note that in the proof of the above lemma we represented the personalised PageRank vector in the form \( p(s, \alpha) = \alpha[\mathbb{I} - (1 - \alpha)\hat{P}]^{-1}s \).

By substituting
\[
R_\alpha = \alpha\mathbb{I} + \alpha \sum_{i=1}^{\infty} (1 - \alpha)^i \hat{P}^i
\]  
(C.6)
we can rewrite the Eq. (C.5) in the form
\[
p(s, \alpha) = R_\alpha s
\]  
(C.7)
This equation makes apparent that the presonalised PageRank vector is a linear transformation of the vector \( s \), and \( R_\alpha \) is the matrix of this transformation. Two important properties of the PageRank vector can be derived from it.

**Lemma C.1.5** The PageRank vector possesses the following properties:

(a) It is linear with respect to the starting vector, i.e. \( p(s_1 + s_2, \alpha) = p(s_1, \alpha) + p(s_2, \alpha) \) for any vectors \( s_1, s_2 \). This, in turn, means that the standard PageRank vector \( \rho(s, \alpha) \) is a weighted average of personalised PageRank vectors \( p(\chi_v, \alpha), \ v \in V \). Here, the symbol \( \chi_v \) denotes the characteristic function of the set \( S = \{v\} \).

(b) The operator \( p(s, \alpha) \) commutes with the matrix \( \hat{P} \), i.e. \( \hat{P}p(s, \alpha) = p(\hat{P}s, \alpha) \). The equality below is a consequence of this property
\[
p(s, \alpha) = \alpha s + (1 - \alpha)p(\hat{P}s, \alpha)
\]  
(C.8)

**Proof:** The property (a) results directly from the formula \( p(s, \alpha) = R_\alpha s \). The property (b) results from a simple transformation
\[
\hat{P}p(s, \alpha) = \alpha(\hat{P}s) + (1 - \alpha)\hat{P}(\hat{P}s) = p(\hat{P}s, \alpha)
\]

Taking this into account, we transform the definition of the PageRank vector into the form
\[
p(s, \alpha) = \alpha s + (1 - \alpha)\hat{P}p(s, \alpha) = \alpha s + (1 - \alpha)p(\hat{P}s, \alpha)
\]

□

### C.2 Approximate Algorithm of Determining the Personalized PageRank Vector

The PageRank vector represents, in fact, a stationary distribution of the matrix \( P' = (1 - \beta)P + \beta s e^T \), see Remark C.1.1. If \( \Gamma = (V, E) \) is a connected graph, then \( P \) is
a regular matrix as in a finite number of steps one can pass from any node \( u \in V \) to any other node \( v \in V \). According to the theory of regular Markov chains, see e.g. [270], the stationary distribution of the matrix \( P' \) has the form

\[
p_i(s, \alpha) = \frac{\Delta_i}{\sum_{j=1}^{m} \Delta_j}, \quad i = 1, \ldots, m \tag{C.9}
\]

where \( \Delta_i \) is the algebraic complement of the \( i \)th element, lying on the main diagonal in the matrix \( I - P' \), and \( p_i(s, \alpha) \) is the \( i \)th element of the personalised vector \( p(s, \alpha) \).

Application of the above formula requires computation of \( m \) determinants of matrices of dimensions \((m-1) \times (m-1)\), which is quite expensive in practice. The Eq. (C.5) implies a conceptually simple method of determining the approximate value of the personalised PageRank vector. It is illustrated by the Algorithm C.1.

**Algorithm C.1** Algorithm of determination of the approximation to the personalised PageRank vector on the basis of Eq. (C.5)

1: Input parameters: Starting vector \( s \), damping coefficient \( \alpha \), column-stochastic matrix \( P \), precision \( \epsilon \).
2: \( p_{\text{old}} = 0, \ p_{\text{new}} = 0, \ s_1 = s \).
3: \( \beta = 1, \ \alpha_1 = 1 - \alpha \).
4: \( \text{res} = 1 \).
5: while \( \text{res} > \epsilon \) do
6: \( s_1 = Ps_1 \).
7: \( \beta = \alpha_1 \beta \).
8: \( p_{\text{new}} = p_{\text{new}} + \beta s_1 \).
9: \( \text{res} = \|p_{\text{new}} - p_{\text{old}}\|_1 \).
10: \( p_{\text{old}} = p_{\text{new}} \).
11: end while
12: Return the vector \( p = \alpha(s + p_{\text{new}}) \).

A disadvantage of the algorithm is the necessity to perform multiple multiplications of a vector by a matrix. Taking into account that \( P \) is a sparse matrix, this algorithm can be applied to the middle sized graphs.

In case of large graphs, a simulation method, suggested in the paper [22], can prove to be much more convenient. The authors exploited there in an interesting way the idea of Berkhina, who proposed in [65] the following metaphor: let us place a portion of paint in the starting node \( v \), which is spilled over the neighbouring nodes. In each step, a fraction \( \alpha \) of the paint, present in a given node \( v \) dries out, half of the still wet paint remains in the node \( v \), and the remaining part spills in equal proportions onto the neighbours of the node \( v \). Let \( p^t \) be a vector with elements \( p^t(v) \), representing the amount of dried paint at node \( v \in V \) at the time point \( t \), and let \( r^t \) be the vector having the elements \( r^t(v) \) representing the amount of wet paint present at the time point \( t \) in the node \( v \). The process of graph “colouring” is described by two equations

\[
\begin{align*}
p^{t+1} &= p^t + \alpha r^t \\
r^{t+1} &= (1 - \alpha) \hat{P} r^t
\end{align*}
\tag{C.10}
\]
where $\hat{P}$ is a matrix of lazy random walk of the form (C.3), $p^0 = 0$, and $r^0 = e_u$.

It is not difficult to see that the vector $p^{t+1}$ assumes the form

$$p^{t+1} = \alpha \sum_{k=0}^{t} r^k = \alpha \sum_{k=0}^{t} (1 - \alpha)^k \hat{P}^k r^0$$

By substituting $r^0 = s$ and $t \to \infty$, we obtain the Eq. (C.5).

In the algorithm, presented in the paper [22], further simplifications were proposed, consisting in ignoring the time and in local perspective on the colouring process. The details of this approach are available in the cited paper. Let us only mention here that the essence of the algorithm is the creation of the so-called $\epsilon$-approximation of the vector $p(s, \alpha)$, that is, a vector $p$, for which the equation

$$p + p(r, \alpha) = p(s, \alpha)$$

holds, where $s$ is the starting vector, and $r$ is a non-negative vector with components $r(v) < \epsilon d(v)$, representing the amount of non-dried (wet) paint at nodes $v \in V$.

A further improvement to this algorithm was proposed by Chung and Zhao in [118]. Its time complexity was reduced to $O(\alpha m \log(1/\epsilon))$. The essence of this idea is illustrated by the pseudocode C.2.

**Algorithm C.2** Fast algorithm of determining an $\epsilon$-approximation of the personalised PageRank vector [118]

1: Input parameters: Undirected graph $\Gamma = (V, E)$, initial vector (distribution) $s$, coefficient $\alpha \in (0, 1)$, precision $\epsilon$.
2: Substitute $p = 0$, $r = s$, $e = 1$.
3: while ($e > \epsilon$) do
4: $e = e/2$
5: $p' = 0$
6: while ($\exists v \in V : r(v) \geq \epsilon d(v)$) do
7: Choose node $u$ such that $r(u) \geq \epsilon d(u)$
8: $p'(u) = p'(u) + \alpha r(u)$
9: $r(v) = r(v) + \frac{1 - \epsilon}{d(u)} r(u)$, $\forall v \in N(u)$
10: $r(u) = 0$
11: end while
12: $p = p + p'$
13: end while
14: Return the vector $p$

Let us underline that the method of choosing the node influences only the speed measured in terms of the number of iterations within the while loop. When lazy random walk is applied, about 50% more iterations are needed to find an approximation with required precision. The abovementioned results were obtained for the random walk with the random walk matrix $P$, teleportation coefficient $\alpha = 0.15$ and precision...
$\epsilon = 10^{-12}$. As comparisons show further on, the variant C.2 can be considered as the quickest one.

It turns out that in the graph $\Gamma$ there are not many links between the nodes with high PageRank values and those with a low PageRank [21]. More precisely, if we sort the graph nodes by the decreasing value $p(s, \alpha)$, and the $k$th node in this order is assigned a higher portion of probability than the node of rank $k(1 + \delta)$, then there exist few connections between nodes with ranks $\{1, \ldots, k\}$ and those with ranks from the set $\{k(1 + \delta) + 1, \ldots, |V|\}$. 
Appendix D

Axiomatic Systems for Clustering

A considerable amount of research work has been devoted to understanding the essentials of clustering, as briefly discussed in Sect. 2.6.\textsuperscript{8}

A number of axiomatic frameworks\textsuperscript{9} have been devised in order to capture the nature of the clustering process, the most often cited being probably the Kleinberg’s system [293].

In general, the axiomatic frameworks of clustering address either:

- the required properties of clustering functions, or
- the required properties of the values of a clustering quality function, or
- the required properties of the relation between the qualities of different partitions (ordering of partitions for a particular set of objects and a given similarity/dissimilarity function).

We will now briefly overview Kleinberg’s axioms and some work done around their implications.

D.1 Kleinberg’s Axioms

As a justification for his axiomatic system, Kleinberg [293] claims that a good clustering may only be a result of a reasonable method of clustering. His axioms are dealing with distance-based cluster analysis. He defines the clustering function as follows:


\textsuperscript{9}Axiomatic systems may be traced back to as early as 1973, when Wright proposed axioms of weighted clustering functions. This means that every domain object was attached a positive real-valued weight, that could be distributed among multiple clusters, like in later fuzzy systems. See: W.E. Wright. A formalization of cluster analysis. Pattern Recognition, 5(3):273–282, 1973.
Definition D.1.1 A function \( f \) is a *clustering function* if it takes as its argument a distance function \( d \) on the set \( S \) and returns a partition \( C \) of \( X \). The sets in \( C \) will be called its *clusters*.

He postulated that some quite “natural” axioms need to be met, when we manipulate the distances between objects. These are:

**Axiom D.1.1** The clustering method should allow to obtain any clustering of the objects (so-called *richness property*). More formally, if \( \text{Range}(f) \) denotes the set of all partitions \( C \) such that \( f(d) = C \) for some distance function \( d \) then \( \text{Range}(f) \) should be equal to the set of all partitions of \( X \).

**Axiom D.1.2** The method should deliver clusterings invariant with respect to distance scale (so-called (scale)-*invariance property*). Formally: for any distance function \( d \) and any \( \alpha > 0 \), \( f(d) = f(\alpha \cdot d) \) should hold.

**Axiom D.1.3** The method should deliver the same clustering if we move objects closer to cluster centres to which they are assigned (so-called *consistency property*). Formally, for a partition \( C \) of \( X \), and any two distance functions \( d \) and \( d' \) on \( X \) such that (a) for all \( i, j \in S \) belonging to the same cluster of \( C \), we have \( d'(i, j) \leq d(i, j) \), and (b) for all \( i, j \in S \) belonging to different clusters of \( C \), we have \( d'(i, j) \geq d(i, j) \) (we will say that \( d' \) is a \( C \)-transformation of \( d \)), the following must hold: if \( f(d) = C \) then \( f(d') = C \).

### D.1.1 Formal Problems

Though the axioms may seem to be reasonable, Kleinberg demonstrated that they cannot be met all at once (but only pair-wise).

**Theorem D.1.1** No clustering function can have at the same time the properties of richness, invariance and consistency

The axiom set is apparently not sound. The proof is achieved via contradiction. For example, take a set of \( n + 2 \) elements. The richness property implies that under two distinct distance functions \( d_1, d_2 \) the clustering function \( f \) may form two clusterings, respectively \( C_1, C_2 \), where the first \( n \) elements form one cluster and in \( C_1 \) the remaining two elements are in one cluster, and in \( C_2 \) they are in two separate clusters. By the invariance property, we can derive from \( d_2 \) the distance function \( d_4 \) such that no distance between the elements under \( d_4 \) is lower than the biggest distance under \( d_1 \). By the invariance property, we can derive from \( d_1 \) the distance function \( d_3 \) such that the distance between elements \( n + 1, n + 2 \) is bigger than under \( d_4 \). We have then \( f([1, \ldots, n + 2]; d_4) = C_2, f([1, \ldots, n + 2]; d_3) = C_1 \). Now let us apply the consistency axiom. From \( d_4 \) we derive the distance function \( d_6 \) such that for elements 1, \ldots, \( n \) the \( d_1 \) and \( d_6 \) are identical, the distance between \( n + 1, n + 2 \) is the same as in \( d_4 \), and the distances between any element of 1, \ldots, \( n \) and any of \( n + 1, n + 2 \)
Appendix D: Axiomatic Systems for Clustering

is some $l$ that is bigger than any distances between any elements under $d_1, \ldots, d_4$. From $d_3$ we derive the distance function $d_5$ such that for elements $1, \ldots, n$ the $d_1$ and $d_5$ are identical, the distance between $n + 1, n + 2$ is the same as in $d_4$ and the distances between any element of $1, \ldots, n$ and any of $n + 1, n + 2$ is the same $l$ as above. We have, then, $f([1, \ldots, n + 2]; d_6) = C_2, f([1, \ldots, n + 2]; d_5) = C_1$. But this means a contradiction, because by construction, $d_5$ and $d_6$ are identical.

At the same time, Kleinberg demonstrated that there exist algorithms that satisfy any pair of the above conditions. He uses for the purpose of this demonstration the versions of the well known statistical single-linkage procedure. The versions differ by the stopping condition:

- $k$-cluster stopping condition (which stops adding edges as soon as the linkage graph for the first time consists of $k$ connected components)—not “rich”
- distance-$r$ stopping condition (which adds edges of weight at most $r$ only)—not scale-invariant
- scale-stopping condition (which adds edges of weight being at most some percentage of the largest distance between nodes)—not consistent

Notice that also $k$-median and $k$-means, as well as the MDL clustering do not fulfill the consistency axiom.

Note, however, that Ben-David [62] drew attention by an illustrative example (their Fig. 2), that consistency alone is a problematic property as it may give rise to new clusters at micro or macro-level.

### D.1.2 Common Sense Problems

As there are practical limitations on the measurement precision, scale-invariance may not be appropriate for the whole range of scaling factors.

The consistency axiom precludes a quite natural phenomenon that under stretching of distances new clusters may be revealed.

Finally, we are really not interested in getting clusters of cardinality say equal one, so that the richness axiom is in fact not intuitive.

### D.1.3 Clusterability Theory Concerns

A set of objects is not all we want to know when saying that we discovered a cluster. We want to see that objects belonging to different clusters differ substantially from one another, for example that the clusters are separated from one another by some space.

But this separating space is something what the majority of clustering methods do not take into account and about which Kleinberg’s axiomatisation does not care.
Speaking differently, a clustering algorithm may provide us with clusters even when there are none in the data.

The very existence of clusters in the data, or, more precisely, the separation of clusters, was the subject of research on the so-called clusterability,\(^{10}\) see e.g. [2, 3].

**Definition D.1.2** [3] Clusterability is a function of the set \(X \subseteq \mathbb{R}^m\) mapping it into the set of real numbers that specifies how a set \(X\) is clusterable.

These functions are constructed in such a way that the higher the value of the function the stronger is the evidence that high quality clusterings may occur.

Methods of cluster analysis, examined by Ackerman [3], on which we base our reflection in this section, are limited to the so-called centre based (or centric) clustering,

**Definition D.1.3** We say that a partition is a centric clustering if each cluster is distinguished by the centre or several centres, where the distance of any element to the nearest centre of its own cluster is not greater than that to any other centre (of any other cluster).

Centric clusterings are a special case of results from the distance driven clustering functions (Definition D.1.1) so that Kleinberg’s axioms should be applicable.

To express the inadequacy of data clustering, Ackerman and Ben-David introduce the concept of loss function for this class of clustering. An optimal clustering minimizes the loss function for a particular data set.

Ackerman introduced the following concepts:

**Definition D.1.4** Two centric clusterings are \(\epsilon\)-close, if you can create a set of disjoint pairs of centres from both clusterings such that the distance between the centres of each pair is less than \(\epsilon\).

**Definition D.1.5** Data is \(\epsilon, \delta\)-clusterable if the loss function for any clustering, that is \(\epsilon\)-close to an optimal clustering, is not greater than \((1 + \delta)\) times the value of the loss function of the optimal clustering (perturbational clusterability).

The study [3] proposed algorithms detecting such clusterability for \(\epsilon = \frac{\text{radius}(X)}{\sqrt{l}}\), where \(l\) is the cardinality of subsets of \(X\), which all have to be considered.

Let us recall two concepts on which clusterability evaluation is based.

**Definition D.1.6** The separation in the clustering is the minimum distance between the elements of different classes. Clustering diameter is the maximum distance between elements of the the same class.

It is argued in [3] that there is at most one clustering such that the separation is larger than the diameter.

With these concepts, Ackerman introduces in [3] various kinds of clusterability.

\(^{10}\)Instead of clusterability, that is, the issue of the very existence of clusters, the cluster validity, that is—their agreement with some prior knowledge about expected clusters, may be investigated, as discussed already in Chap. 4. Cluster validity was also studied in papers [119, 364].
Definition D.1.7 (Clusterability and related concepts)

- **Worst-pair-quality of a clustering** is the ratio of separation and diameter.
- **Worst-pair-clusterability** is the minimal worst-pair-quality over all centric clusterings of the set $X$.
- **$k$-separable clusterability** is the decrease in the loss function while passing from $k - 1$ clusters to $k$ clusters.
- **Variance-clusterability** is the quotient of the variance between clusters to the variance within clusters.
- **Target-clusterability** is measured as the distance to the clustering defined manually.

It is claimed that in polynomial time, one can calculate only the worst-pair-clusterability.

Note, first of all, that all these clusterability measures are sensitive to outliers because (most) clustering methods do not allow for an element to be left unclustered.

But more interesting is the relationship to the Kleinberg’s axioms. We would naively expect that if a clustering algorithm returns a clustering, then there exists an intrinsic clustering. Furthermore, one would expect that if there is an intrinsic clustering, then the clustering algorithm returns it. And it returns (approximately) the clustering that is there in the data.

But any algorithm, seeking clusters fitting the requirements of $\epsilon, \delta$-clusterability, will fail under distance-scaling, so it will not satisfy the Kleinberg’s invariance-axiom.

On the other hand, a clustering algorithm, seeking to match the worst-pair-clusterability or variance-clusterability or target-clusterability criterion, will perform well under scaling.

The $k$-separable clusterability may or may not depend on scaling this being determined by the loss function.

The axiom of consistency seems not to constitute any obstacle in achieving any of the mentioned clusterability criteria. However, the Kleinberg’s $C$-transformation may lead to emergence of new clusters according to at least the variance-based clusterability criterion.

The Kleinberg’s axiom of richness is, however, hard to fulfil at least by the variance-based clusterability. This clusterability criterion puts preference on clusters with small numbers of elements, but if the variance is to be estimated with any basic scrutiny, the minimum number of three elements within each cluster is necessary, which precludes matching of the richness axiom.

### D.1.4 Learnability Theory Concerns

Learnability theory [473] defines learnability as the possibility to generalize from the sample to the population. Its basic postulate is: the concept is learnable if it can be falsified. In order to learn, the algorithm must be able to conclude that the data do
not belong to the space of concepts. If the algorithm is able to assign to each set of data a concept from the concept space, it does not learn anything.

In the light of the learnability theory

- the classical cluster analysis does not reveal any “natural clusters”, but provides a mixture of the intrinsic structure of the data and of the structure induced by the clustering algorithm
- the classical cluster analysis is not a method of learning without supervision (“without a teacher”)\(^\text{11}\)
- the capability of a clustering algorithm to recreate the external (“manual”) clustering is not a criterion for the correctness of the algorithm, if it has not learning abilities
- if the cluster analysis has something to contribute, it cannot satisfy the axioms of Kleinberg that is to
  - produce any clustering possible (via manipulation of distance)
  - produce invariant results while scaling distance
  - produce invariant results with respect to reduction of distance to cluster centre.

because these axioms give a much too big number of degrees of freedom for manipulation of the function of distance.

\subsection*{D.2 Cluster Quality Axiomatisation}

Ackerman and Ben-David\(^\text{12}\) propose to resolve the problem of Kleinberg’s axiomatics by axiomatising not the clustering function, but rather cluster quality function. We base the following on their considerations.

\textbf{Definition D.2.1} Let \(\mathcal{C}(X)\) be the set of all possible clusterings over the set of objects \(X\), and let \(\mathcal{D}(X)\) be the set of all possible distance functions over the set of objects \(X\).

A clustering-quality measure (CQM) \(J : X \times \mathcal{C}(X) \times \mathcal{D}(X) \rightarrow \mathbb{R}^+ \cup \{0\}\) is a function that, given a data set (with a distance function) and its partition into clusters, returns a non-negative real number representing how strong or conclusive the clustering is.

Ackerman and Ben-David propose the following axioms:

\textbf{Axiom D.2.1 (Scale Invariance)} A quality measure \(J\) satisfies scale invariance if for every clustering \(\mathcal{C}\) of \((X, d)\),\(^\text{13}\) and every positive \(\beta\), \(J(X, \mathcal{C}, d) = J(X, \mathcal{C}, \beta d)\).

\(^{11}\)There is, in fact, always a teacher who provides an aesthetic criterion of what is “similar”/“different”.


\(^{13}\)\(X\) is the set of objects, \(d\) is a distance function.
Note that if we define a clustering function in such a way that it maximises the quality function, then the clustering function has also to be invariant.

**Axiom D.2.2** (*Consistency*) A quality measure $J$ satisfies consistency if for every clustering $C$ over $(X, d)$, whenever $d'$ is a $C$-transformation of $d$, then $J(X, C, d') \geq J(X, C, d)$.

Note that if we define a clustering function in such a way that it maximises the quality function, then the clustering function *does not need to be consistent*.

**Axiom D.2.3** (*Richness*) A quality measure $m$ satisfies richness if for each non-trivial clustering $C^*$ of $X$, there exists a distance function $d$ over $X$ such that $C^* = \arg\max_C \{J(X, C, d)\}$.

Note that if we define a clustering function in such a way that it maximises the quality function, then the clustering function has also to be rich.

Ackerman and Ben-David claim that

**Theorem D.2.1** Consistency, scale invariance, and richness for clustering-quality measures form a consistent set of requirements.

and prove this claim by providing a quality measure that matches all these axioms.

The quality measure they propose here is the “Relative Margin”. First, one needs to compute the ratio of distance of a data point to its closest centre to that to the second closest centre. Then, a representative set of a clustering is defined as a set $K$ containing exactly one element from each cluster. One computes the average of the ratio, mentioned before, for each potential representative set. The minimum over these averages is the cluster quality function called Relative Margin. The lower the value of the Relative Margin, the higher the clustering quality. This does not match quite their axiomatisation because the axiomatisation assumed that increasing quality is related to increasing quality function value. So to satisfy the axiomatic system, axioms have to be inverted, so that quality increase is related to function decrease.

A disadvantage of this measure is that it ranks highly a clustering in which each element is a separate cluster.

### D.3 Relaxations for Overcoming the Kleinberg’s Problems

In the preceding section one way of clustering axiomatisation improvement was presented. It consisted in shifting axiomatisation from clustering function to clustering quality, which implicitly relaxed at least one Kleinberg’s axiom (consistency).

Let us look now at other proposals, concerning now the explicit relaxation of the Kleinberg’s axioms, as summarised by Ben-David.

**Axiom D.3.1** (*Relaxation of Kleinberg’s richness*) For any partition $C$ of the set $X$, consisting of exactly $k$ clusters, there exists such a distance function $d$ that the clustering function $f(d)$ returns this partition $C$.
This relaxation allows for the algorithms splitting the data into a fixed number of clusters, like $k$-means. But it still leaves the open problem of a minimal number of elements in a cluster.

**Axiom D.3.2 (Local Consistency)** Let $C_1, \ldots, C_k$ be the clusters of $f(d)$. For every $\beta_0 \geq 1$ and positive $\beta_1, \ldots, \beta_k \leq 1$, if $d'$ is defined by: $d'(a, b) = \beta_i d(a, b)$ for $a$ and $b$ in $C_i$, $d'(a, b) = \beta_0 d(a, b)$ for $a, b$ not in the same $f(d)$-cluster, then $f(d) = f(d')$.

This axiom does not guarantee that $d'$ is in fact a distance, so that it is hard to satisfy.

**Axiom D.3.3 (Refinement Consistency)** is a modification of the consistency axiom obtained by replacing the requirement that $f(d) = f(d')$ with the requirement that one of $f(d), f(d')$ is a refinement of the other.

Obviously, the replacement of the consistency requirement with refinement consistency breaks the impossibility proof of Kleinberg’s axiom system. But there is a practical concern: assume a data set of points uniformly randomly distributed in a plane on line segments $((a, 0), (2a, 6a)), ((a, 0), (2a, -6a)), ((-a, 0), (-2a, 6a)), ((-a, 0), (-2a, -6a))$. A $k$-means algorithm with $k = 2$ would create two clusters: (1) segments $((a, 0), (2a, 6a)), ((a, 0), (2a, -6a)), (2) ((-a, 0), (-2a, 6a)), ((-a, 0), (-2a, -6a))$. But if cluster (2) is shrunk as follows: all points are rotated around $(-a, 0)$ so that they lie on the X axis to the left of $(-a, 0)$, then $k$-means would change class allocation of a part of points from this cluster, violating refinement consistency.

### D.4 $k$-means Algorithm—Popular, but not a Clustering Algorithm?

The well-known $k$-means clustering algorithm seeks to minimize the function

$$Q(C) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \| x_i - \mu_j \|^2 = \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x_i, x_l \in C_j} \| x_i - x_l \|^2 \quad \text{(D.1)}$$

for a dataset $X$ under some partition $C$ into the predefined number $k$ of clusters, where $u_{ij}$ is an indicator of the membership of data point $x_i$ in the cluster $C_j$ having the centre at $\mu_j$. It is easily seen that it is scale-invariant, but one sees immediately

---

14The considerations would apply also to kernel $k$-means algorithm using the quality function

$$Q(C) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \| \Phi(x_i) - \mu_j^\Phi \|^2$$

where $\Phi$ is a non-linear mapping from the original space to the so-called feature space.
that it is not rich (only partitions with $k$ clusters are considered), it has also been
demonstrated by Kleinberg that it is not consistent. So the widely used algorithm
violates in practice two of three Kleinberg’s axioms, so that it cannot be considered to
be a “clustering function”. We perceive this to be at least counter-intuitive. Ben-David
and Ackerman in [62] in Sect. 4.2., raised also similar concern from the perspective of
what an axiomatic system should accomplish. They state that one would expect, for
the axiomatised set of objects, a kind of soundness that is that most useful clustering
algorithms would fit the axioms and completeness that is that apparent non-clustering
algorithms would fail on at least one axiom. While Kleinberg’s axioms explicitly
address the distance-based clustering algorithms (and not e.g. density based ones),
you fall apparently short of reaching this goal. In this Appendix we demonstrate that
even for a narrower set of algorithms, ones over data embedded in Euclidean space,
the axioms fail.

There exist a number of open questions on why it is so. Recall that in [476] it has
been observed that Kleinberg’s proof of Impossibility Theorem stops to be valid in
case of graph clustering. This raises immediately the question of its validity in $\mathbb{R}^m$
Euclidean space. Note that Kleinberg did not bother about embedding the distance in
such a space. So one may ask whether or not $k$-means does not fit Kleiberg’s axioms
because this is a peculiar property of $k$-means or because any algorithm embedded
in Euclidean space would fail to fit.

Therefore we made an effort to identify and overcome at least some reasons for the
difficulties connected with axiomatic understanding of research area of cluster analy-
thesis and hope that this may be a guidance for further generalisations to encompass
if not all then at least a considerable part of the real-life algorithms. This Appendix
investigates why the $k$-means algorithm violates the Kleinberg’s axioms for cluster-
ing functions. We claim that the reason is a mismatch between informal intuitions
and formal formulations of these axioms. We claim also that there is a way to rec-
concile $k$-means with Kleinberg’s consistency requirement via introduction of centric
consistency which is neither a subset nor superset of Kleinberg’s consistency, but
rather a $k$-means clustering model specific adaptation of the general idea of shrinking
the cluster.

To substantiate our claim that there is a mismatch between informal intuitions
and formal formulations of Kleinberg’s axioms, we present a series of carefully
constructed examples. First of all Kleinberg claims that his axioms can be fulfilled
pair-wise. But we demonstrate that richness and scaling-invariance alone may lead
to a contradiction for a special case. Further we provide example showing that the
consistency and scale-invariance axioms, even ignoring richness axiom, yield a con-
tradiction to the intuitions behind clustering. We demonstrate this by showing that
these axioms may lead to moving clusters closer to one another instead of away from
one another. If the clusters are moved closer, it does not wonder when a clustering
algorithm puts them together. Furthermore we show that consistency alone leads to
contradictions. We demonstrate that in practical settings of application of many algo-
rithms, that is in a metric $m$-dimensional space where $m$ is the number of features,
it is impossible to contract a single cluster without moving the other ones and as a
consequence running at risk of moving some clusters closer together. We show in
particular that $k$-means version where we allow for $k$ to range over a set, will change the optimal clustering $k$ when Kleinberg’s $C$ operation (consistency operation) is applied.

Finally to substantiate our claim that there exists a way to reconcile the formulation of the axioms with their intended meaning and that under this reformulation the axioms stop to be contradictory and we can reconcile $k$-means with Kleinberg’s consistency requirement we first introduce the concept of centric consistency which is neither a subset nor superset of Kleinberg’s consistency, but rather a $k$-means clustering model specific adaptation of the general idea of shrinking the cluster. It relies simply in moving cluster elements towards its centre. Then we provide an example of a clustering function that fits the axioms of near-richness, scale-invariance and possesses the property of centric consistency, so that it is clear that they are not contradictory. And then we prove mathematically that $k$-means fits the centric-consistency property in that both local and global optima of its target function are retained. Additionally we prove that $k$-means is not Kleinberg-consistent even when we restrict ourselves to $m$-dimensional metric space.

Let us stress, however, that the proposed reformulation is not sufficient to be raised to the status of a set of sound and complete axioms. Therefore, following a number of authors, e.g. [4] we will rather talk about properties that may or may not be fulfilled by a clustering algorithm, just characterising some features of $k$-means that should possibly be covered, and at least not neglected by future development of axiomatisation in the domain of cluster analysis.

We demonstrate in Sect. D.8 that the richness axiom denies common sense by itself. We claim that there is a clash between the intended meaning and formalism of the Kleinberg’s axioms and that there exists a way to reconcile them. We propose a reformulation of the axioms in accordance with the intuitions and demonstrate that under this reformulation the axioms stop to be contradictory (Sect. D.9), and even a real-world algorithm like $k$-means conforms to this augmented axiomatic system (Sect. D.10).

We start with some remarks on the relationship between Kleinberg’s axioms and $k$-means (Sect. D.5).

### D.5 Kleinberg’s Axioms and $k$-means—Conformance and Violations

Let us briefly discuss here the relationship of $k$-means algorithm to the already mentioned axiomatic systems, keeping in mind that we apply it in $\mathbb{R}^m$ Euclidean space.

Scale-invariance is fulfilled because $k$ means qualifies objects into clusters based on relative distances to cluster centres and not their absolute values as may be easily seen from Eq. (D.1).\(^\dagger\)

\(^\dagger\)However, this quality function fails on the axiom of Function Scale Invariance, proposed in [62].
On the other hand richness, a property denial of which has nothing to do with distances, hence with embedding in an Euclidean space, as already known from mentioned publications, e.g. [526], is obviously violated because $k$-means returns only partitions into $k$ clusters.

But what about its relaxation that is $k$-richness. Let us briefly show here that

**Theorem D.5.1** $k$-means algorithm is $k$-rich

**Proof:** We proceed by constructing a data set for each required partition. Let us consider $n$ data points arranged on a straight line and we want to split them into $k$ clusters with a concrete partition cluster. For this purpose arrange the clusters on the line in non-increasing order of their cardinality. Each cluster shall occupy (uniformly) a unit length. The space between the clusters (distance between $i$th and $(i+1)$st cluster) should be set as follows: For $i = 1, \ldots, k-1$ let $dce(j, i)$ denote the distance between the most extreme data points of clusters $j$ and $i$, $\text{cardc}(j, i)$ shall denote the combined cardinality of clusters $j, j + 1, \ldots, i$. The distance between clusters $i$ and $i + 1$ shall be then set to $2 \times dce(1, i) \times \frac{\text{cardc}(1, i) + \text{cardc}(i+1, i+1)}{\text{cardc}(i+1, i+1)}$. In this case application of $k$-means algorithm will lead the desired split into clusters.

Let us stress here that there exist attempts to upgrade $k$-means algorithm to choose the proper $k$. The portion of variance explained by the clustering is used as quality criterion.\footnote{Such a quality function would satisfy axiom of Function Scale Invariance, proposed in [62].} It is well known that increase of $k$ increases the value of this criterion. The optimal $k$ is deemed to be one when this increase stops to be “significant”. The above construction could be extended to cover a range of $k$ values to choose from. However, the full richness is not achievable because a split into two clusters will be better than keeping a single cluster, and the maximum is attained for this criterion if $k = n$. So either the clustering will be trivial or quite a large number of partitions will be excluded. However, even $k$-richness offers a large number of partitions to choose from.

Kleinberg himself proved via a bit artificial example (with unbalanced samples and an awkward distance function) that $k$-means algorithm with $k = 2$ is not consistent. Kleinberg’s counter-example would require an embedding in a very high dimensional space, non-typical for $k$-means applications. Also $k$-means tends to produce rather balanced clusters, so Kleinberg’s example could be deemed to be eccentric.

Let us illustrate by a more realistic example (balanced, in Euclidean space) that this is a real problem. Let $A, B, C, D, E, F$ be points in three-dimensional space with coordinates: $A(1, 0, 0), B(33, 32, 0), C(33, -32, 0), D(-1, 0, 0), E(-33, 0, -32), F(-33, 0, 32)$. Let $X_{AB}, X_{AC}, X_{DE}, X_{DF}$ be sets of say 1000 points randomly uniformly distributed over line segments (except for endpoints) $AB, AC, DE, EF$ resp. Let $X = X_{AB} \cup X_{AC} \cup X_{DE} \cup X_{DF}$, $k$-means with $k = 2$ applied to $X$ yields a partition $\{X_{AB} \cup X_{AC}, X_{DE} \cup X_{DF}\}$. But let us perform a $C$ transformation consisting in rotating line segments $AB, BC$ around the point $A$ in the plane spread by the first two coordinates towards the first coordinate axis so that the angle between this
axis and $AB'$ and $AC'$ is say one degree. Now the $k$-means with $k = 2$ yields a different partition, splitting line segments $AB'$ and $AC'$.\textsuperscript{17} With this example not only consistency violation is shown, but also refinement-consistency violation.

\section*{D.6 Problems with Consistency in Euclidean Space}

How does it happen that seemingly intuitive axioms lead to such a contradiction. We need to look more carefully at the consistency axiom in conjunction with scale-invariance. $C$-transform does not do what Kleinberg claimed it should that is describing a situation when moving elements from distinct clusters apart and elements within a cluster closer to one another.\textsuperscript{18}

For example, let $X$ consist of four elements $e_1, e_2, e_3, e_4$ and let a clustering function partition it into $\{e_1\}, \{e_2, e_3\}, \{e_4\}$ under some distance function $d_1$. One can easily construct a distance function $d_2$ being a $C$-transform of $d_1$ such that $d_2(e_2, e_3) = d_1(e_2, e_3)$ and $d_2(e_1, e_2) + d_2(e_2, e_3) = d_2(e_1, e_3) + d_2(e_2, e_3) + d_2(e_3, e_4) = d_2(e_2, e_4)$ which implies that these points under $d_2$ can be embedded in the space $\mathbb{R}$ that is the straight line. Without restricting the generality (the qualitative illustration) assume that the coordinates of these points in this space are located at points 0, 0.4, 0.6, 1 resp. Now assume we want to perform $C$-transformation of Kleinberg (obtaining the distance function $d_3$) in such a manner that the data points remain in $\mathbb{R}$ and move elements of the second set i.e. $\{e_2, e_3\}$ ($d_2(e_2, e_3) = 0.2$) closer to one another so that $e_2 = (0.5)$, $e_3 = (0.6)$ ($d_3(e_2, e_3) = 0.1$). $e_1$ may then stay where it is but $e_4$ has to be shifted at least to (1.1) (under $d_3$ the clustering function shall yield same clustering). Now apply rescaling into the original interval that is multiply the coordinates (and hence the distances, yielding $d_4$) by $1/1.1$. $e_1$ stays at (0), $e_2 = (\frac{5}{11}), e_3 = \frac{6}{11}, e_4 = (1)$. $e_3$ is now closer to $e_1$ than before. We could have made the things still more drastic by transforming $d_2$ to $d'_2$ in such a way that instead of $e_4$ going to (1.1), as under $d_3$, we set it at (2). In this case the rescaling would result in $e_1 = (0), e_2 = (0.25), e_3 = (0.3), e_4 = (1)$ (with the respective distances $d'_2$) which means a drastic relocation of the second cluster towards the first—the distance between clusters decreases instead of increasing as claimed by Kleinberg. This is a big surprise. The $C$ transform should have moved elements of a cluster closer together and further apart those from distinct clusters and rescaling should not disturb the proportions. It turned out to be the other way. So something is wrong

\textsuperscript{17}In a test run with 100 restarts, in the first case we got clusters of equal sizes, with cluster centres at (17,0,0) and (−17,0,0), (between_SS / total_SS = 40\%) whereas after rotation we got clusters of sizes 1800, 2200 with centres at (26,0,0), (−15,0,0) (between_SS / total_SS = 59\%).

\textsuperscript{18}Recall that the intuition behind clustering is to partition the data points in such a way that members of the same cluster are “close” to one another, that is their distance is low, and members of two different clusters are “distant” from one another, that is their distance is high. So it is intuitively obvious that moving elements from distinct clusters apart and elements within a cluster closer to one another should make a partition “look better”.

either with the idea of scaling or of $C$-transformation. We shall be reluctant to claim the scaling, except for the practical case when scaling down leads to indescernability between points due to measurement errors.

### D.7 Counter-Intuitiveness of Consistency Axiom Alone

So we will consider counter-intuitiveness of consistency axiom. To illustrate it, recall first the fact that a large portion of known clustering algorithms uses data points embedded in an $m$ dimensional feature space, usually $\mathbb{R}^m$ and the distance is the Euclidean distance therein. Now imagine that we want to perform a $C$-transform on a single cluster of a partition that is the $C$-transform shall provide distances compatible with the situation that only elements of a single cluster change position in the embedding space. Assume the cluster is an “internal” one that is for a point $e$ in this cluster any hyperplane containing it has points from some other clusters on each side. Furthermore assume that other clusters contain together more than $m$ data points, which should not be an atypical case. Here the problem starts. The position of $e$ is determined by the distances from the elements of the other clusters in such a way that the increase of distance from one of them would necessarily decrease the distance to some other (except for strange configurations). Hence we can claim

**Theorem D.7.1** Under the above-mentioned circumstances it is impossible to perform $C$-transform reducing distances within a single cluster.

So the $C$-transform enforces either adding a new dimension and moving the affected single cluster along it (which does not seem to be quite natural) or to change positions of elements in at least two clusters within the embedding space. Therefore vast majority of such algorithms does not meet not only the consistency but also inner consistency requirement.

Why not moving a second cluster is so problematic? Let us illustrate the difficulties with the original Kleinberg’s consistency by looking at an application of the known $k$-means algorithm, with $k$ being allowed to cover a range, not just a single value, to the two-dimensional data set visible in Fig. D.1.19 This example is a mixture of data points sampled from 5 normal distributions. The $k$-means algorithm with $k = 5$, as expected, separates quite well the points from various distributions. As visible from the second column of Table D.1, in fact $k = 5$ does the best job in reducing the unexplained variance. Figure D.2 illustrates a result of a $C$-transform on the results of the former clustering. Visually we would tell that now we have two clusters. A look into the third column of the Table D.1 convinces that really $k = 2$ is the best choice for clustering these data with $k$-means algorithm. This of course contradicts Kleinberg’s consistency axiom. And demonstrates the weakness of outer-consistency concept as well.

---

19 Already Ben-David [62] indicated problems in this direction.
Appendix D: Axiomatic Systems for Clustering

Fig. D.1  A mixture of 5 normal distributions as clustered by \( k \)-means algorithm (Voronoi diagram superimposed)

Table D.1  Variance explained (in percent) when applying \( k \)-means algorithm with \( k = 2, \ldots, 6 \) to data from Figs. D.1 (Original), D.2 (Kleinberg) and D.3 (Centric)

<table>
<thead>
<tr>
<th>( k )</th>
<th>Original</th>
<th>Kleinberg</th>
<th>Centralised</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>54.3</td>
<td>98.0</td>
<td>54.9</td>
</tr>
<tr>
<td>3</td>
<td>72.2</td>
<td>99.17</td>
<td>74.3</td>
</tr>
<tr>
<td>4</td>
<td>83.5</td>
<td>99.4</td>
<td>86.0</td>
</tr>
<tr>
<td>5</td>
<td>90.2</td>
<td>99.7</td>
<td>92.9</td>
</tr>
<tr>
<td>6</td>
<td>91.0</td>
<td>99.7</td>
<td>93.6</td>
</tr>
</tbody>
</table>

Fig. D.2  Data from Fig. D.1 after Kleinberg’s \( C \)-transformation clustered by \( k \)-means algorithm into two groups
D.8 Problems of Richness Axiom

As already mentioned, richness or near-richness forces the introduction of “refinement-consistency” which is a too weak concept. But even if we allow for such a resolution of the contradiction in Kleinberg’s framework, it still does not make it suitable for practical purposes. The most serious drawback of Kleinberg’s axioms is the richness requirement.

But we may ask whether or not it is possible to have richness, that is for any partition there exists always a distance function that the clustering function will return this partition, and yet if we restrict ourselves to $\mathbb{R}^m$, the very same clustering function is not rich any more, or even it is not anti-chain.

Consider the following clustering function $f()$. If it takes a distance function $d()$ that takes on only two distinct values $d_1$ and $d_2$ such that $d_1 < 0.5d_2$ and for any three data points $a, b, c$ if $d(a, b) = d_1, d(b, c) = d_1$ then $d(a, c) = d_1$, it creates clusters of points in such a way that $a, b$ belong to the same cluster if and only if $d(a, b) = d_1$, and otherwise they belong to distinct clusters. If on the other hand $f()$ takes a distance function not exhibiting this property, it works like $k$-means.

Obviously, function $f()$ is rich, but at the same time, if confined to $\mathbb{R}^m$, if $n > m + 1$ and $k \ll n$, then it is not rich—it is in fact $k$-rich, and hence not anti-chain.

Can we get around the problems all three Kleinberg’s axioms in a similar way in $\mathbb{R}^m$? Regrettably,

**Theorem D.8.1** If $C$ is a partition of $n > 2$ elements returned by a clustering function $f$ under some distance function $d$, and $f$ satisfies Consistency, then there exists a distance function $d_E$ embedded in $\mathbb{R}^m$ for the same set of elements such that $C$ is the partition of this set under $d_E$.

The consequence of this theorem is of course that the constructs of contradiction of Kleinberg axioms are simply transposed from the domain of any distance functions to distance functions in $\mathbb{R}^m$.

**Proof**: To show the validity of the theorem, we will construct the appropriate distance function $d_E$ by embedding in the $\mathbb{R}^m$. Let $d_{max}$ be the maximum distance between the considered elements under $d$. Let $C_1, \ldots, C_k$ be all the clusters contained in $C$. For each cluster $C_i$ we construct a ball $B_i$ with radius $r_i$ equal to $r_i = \frac{1}{2} \min_{x,y \in C_i, x \neq y} d(x, y)$. The ball $B_1$ will be located in the origin of the coordinate system. $B_{1 \ldots i}$ be the ball of containing all the balls $B_1 \ldots, B_i$. Its centre be at $c_{1 \ldots i}$ and radius $r_{1 \ldots i}$. The ball $B_i$ will be located on the surface of the ball with centre at $c_{1 \ldots i-1}$ and radius $r_{1 \ldots i-1} + d_{max} + r_i$. For each $i = 1, \ldots, k$ select distinct locations for elements of $C_i$ within the ball $B_i$. The distance function $d_E$ define as the Euclidean distances within $\mathbb{R}^m$ in these constructed locations.

Apparently, $d_E$ is a $C$-transform of $d$, as distances between elements of $C_i$ are smaller than or equal to $2r_i = \min_{x,y \in C_i, x \neq y} d(x, y)$, and the distances between elements of different balls exceed $d_{max}$. \qed
But richness is not only a problem in conjunction with scale-invariance and consistency, but rather it is a problem by itself.

It has to be stated first that richness is easy to achieve. Imagine the following “clustering function”. You order nodes by average distance to other nodes, on tights on squared distance and so on, and if no sorting can be achieved, the unsortable points are set into one cluster. Then we create an enumeration of all clusters and map it onto unit line segment. Then we take the quotient of the lowest distance to the largest distance and state that this quotient mapped to that line segment identifies the optimal clustering of the points. Though the algorithm is simple in principle (and useless also), and meets axioms of richness and scale-invariance, we have a practical problem: As no other limitations are imposed, one has to check up to \( \sum_{k=2}^{n} \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^n \) possible partitions (Bell number) in order to verify which one of them is the best for a given distance function because there must exist at least one distance function suitable for each of them. This is prohibitive and cannot be done in reasonable time even if each check is polynomial (even linear) in the dimensions of the task (\( n \)).

Furthermore, most algorithms of cluster analysis are constructed in an incremental way. But this can be useless if the clustering quality function is designed in a very unfriendly way. For example as an XOR function over logical functions of class member distances and non-class member distances (e.g. being true if the distance rounded to an integer is odd between class members and divisible by a prime number for distances between class members and non-class members, or the same with respect to class centre or medoid).

Just have a look at sample data from Table D.2. A cluster quality function was invented along the above line and exact quality value was computed for partitioning first \( n \) points from this data set as illustrated in Table D.3. It turns out that the best partition for \( n \) points does not give any hint for the best partition for \( n + 1 \) points

<table>
<thead>
<tr>
<th>id</th>
<th>( x ) coordinate</th>
<th>( y ) coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.022346</td>
<td>5.142886</td>
</tr>
<tr>
<td>2</td>
<td>3.745942</td>
<td>4.646777</td>
</tr>
<tr>
<td>3</td>
<td>4.442992</td>
<td>5.164956</td>
</tr>
<tr>
<td>4</td>
<td>3.616975</td>
<td>5.188107</td>
</tr>
<tr>
<td>5</td>
<td>3.807503</td>
<td>5.010183</td>
</tr>
<tr>
<td>6</td>
<td>4.169602</td>
<td>4.874328</td>
</tr>
<tr>
<td>7</td>
<td>3.557578</td>
<td>5.248182</td>
</tr>
<tr>
<td>8</td>
<td>3.876208</td>
<td>4.507264</td>
</tr>
<tr>
<td>9</td>
<td>4.102748</td>
<td>5.073515</td>
</tr>
<tr>
<td>10</td>
<td>3.895329</td>
<td>4.878176</td>
</tr>
</tbody>
</table>
Table D.3  Partition of the best quality (the lower the value the better) after including \( n \) first points from Table D.2

<table>
<thead>
<tr>
<th>( n )</th>
<th>Quality</th>
<th>Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1270</td>
<td>{1, 2}</td>
</tr>
<tr>
<td>3</td>
<td>1270</td>
<td>{1, 2} {3}</td>
</tr>
<tr>
<td>4</td>
<td>823</td>
<td>{1, 3, 4} {2}</td>
</tr>
<tr>
<td>5</td>
<td>315</td>
<td>{1, 4} {2, 3, 5}</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>{1, 5} {2, 4, 6} {3}</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>{1, 6} {2, 7} {3, 5} {4}</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>{1, 2, 4, 5, 6, 8} {3} {7}</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>{1, 2, 4, 5} {3, 8} {6, 9} {7}</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>{1, 2, 3, 5, 9} {4, 6} {7, 10} {8}</td>
</tr>
</tbody>
</table>

therefore each possible partition needs to be investigated in order to find the best one.20

Summarizing these examples, the learnability theory points at two basic weaknesses of the richness or even near-richness axioms. On the one hand the hypothesis space is too big for learning a clustering from a sample (it grows too quickly with the sample size). On the other hand an exhaustive search in this space is prohibitive so that some theoretical clustering functions do not make practical sense.

There is one more problem. If the clustering function can fit any data, we are practically unable to learn any structure of data space from data [294]. And this learning capability is necessary at least in the cases: either when the data may be only representatives of a larger population or the distances are measured with some measurement error (either systematic or random) or both. Note that we speak here about a much broader aspect than so-called cluster stability or cluster validity, pointed at by Luxburg [483, 485].

D.9  Correcting Formalisation of Kleinberg Axioms

It is obvious that richness axiom of Kleinberg needs to be replaced with a requirement of the space of hypotheses to be “large enough”. For \( k \)-means algorithm it has been shown that \( k \)-richness is satisfied (and the space is still large, a Bell number of partitions to choose from). \( k \) means satisfies the scale-invariance axiom, so that only the consistency axiom needs to be adjusted to be more realistic.

Therefore a meaningful redefinition of Kleinberg’s \( C \)-transform is urgently needed. It must not be annihilated by scaling and it must be executable.

---

20Strict separation [78] mentioned earlier is another kind of a weird cluster quality function, requiring visits to all the partitions.
Let us create for \( \mathbb{R} \) a working definition of the \( C^* \) transform as follows: Distances in only one cluster \( X \) are changed by moving a point along the axis connecting it to cluster \( X \) centre reducing them within the cluster \( X \) by the same factor, the distances between any elements outside the cluster \( X \) are kept [as well as to the gravity centre of the cluster \( X \)]\(^{21}\).

Now consider the following one-dimensional clustering function: For a set of \( n \geq 2 \) points two elements belong to the same cluster if their distance is strictly lower than \( \frac{1}{n+1} \) of the largest distance between the elements. When \( a, b \) belong to the same cluster and \( b, c \) belong to the same cluster, then \( a, c \) belong to the same cluster. As a consequence, the minimum distance between elements of distinct clusters is \( \frac{1}{n+1} \) of the largest distance between the elements of \( X \). It is easily seen that the weakened richness is fulfilled. The scale-invariance is granted by the relativity of inter-cluster distance. And the consistency under redefined \( C \) transform holds also. In this way all three axioms hold.

A generalization to an Euclidean space of higher dimensionality seems to be quite obvious if there are no ties on distances (the exist one pair of points the distance between which is unique and largest among distances\(^{22}\)). We embed the points in the space, and then say that two points belong to the same cluster if the distance along each of the dimensions is lower than \( \frac{1}{n+1} \) of the largest distance between the elements along the respective dimension. The distance is then understood as the maximum of distances along all dimensions.

Hence

**Theorem D.9.1** For each \( n \geq 2 \), there exists clustering function \( f \) that satisfies Scale-Invariance, near-Richness, and \( C^* \) based Consistency (which we term “centric-consistency”\(^{23}\)).

This way of resolving Kleinberg’s contradictions differs from earlier approaches in that a realistic embedding into an \( \mathbb{R}^m \) is considered and the distances are metric.

We created herewith the possibility of shrinking a single cluster without having to “move” the other ones. As pointed out, this was impossible under Kleinberg’s \( C \) transform, that is under increase of all distances between objects from distinct clusters. In fact intuitively we do not want the objects to be more distant but rather the clusters. We proposed to keep the cluster centroid unchanged while decreasing distances between cluster elements proportionally, insisting that no distance of other

\(^{21}\)Obviously, for any element outside the cluster \( X \) the distance to the closest element of \( X \) before the transform will not be smaller than its distance to the closest element of \( X \) after the transform. Note the shift of attention. We do not insist any longer that the distance to each element of other cluster is increased, rather only the distance to the cluster as a “whole” shall increase. This is by the way a stronger version of Kleinberg’s local consistency which would be insufficient for our purposes.

\(^{22}\)otherwise some tie breaking measures have to be taken that would break the any symmetry and allow to choose a unique direction.

\(^{23}\)Any algorithm being consistent is also refinement-consistent. Any algorithm being inner-consistent is also consistent. Any algorithm being outer-consistent is also consistent. But there are no such subsumions for the centric-consistency.
elements to the closest element of the shrunk cluster should decrease. This approach is pretty rigid. It assumes that we are capable to embed the objects into some Euclidean space so that the centroid has a meaning.

**D.10  \( k \)-means Fitting Centric-Consistency Axiom**

Our proposal of centric-consistency has a practical background. Kleinberg proved that \( k \)-means does not fit his consistency axiom. As shown experimentally in Table D.1, \( k \)-means algorithm behaves properly under \( C^* \) transformation. Figure D.3 illustrates a two-fold application of the \( C^* \) transform (same clusters affected as by \( C \)-transform in the preceding figure). As recognizable visually and by inspecting the forth column of Table D.1, here \( k = 5 \) is the best choice for \( k \)-means algorithm, so the centric-consistency axiom is followed.

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

The \( k \)-means algorithm minimizes the sum\(^{24}\) \( Q \) from Eq. (D.1). \( V(C_j) \) be the sum of squares of distances of all objects of the cluster \( C_j \) from its gravity centre. Hence \( Q(C) = \sum_{j=1}^{k} \frac{1}{n_j} V(C_j) \). Consider moving a data point \( x^* \) from the cluster \( C_{j_0} \) to cluster \( C_{j_1} \). As demonstrated by [164], \( V(C_{j_0} - \{x^*\}) = V(C_{j_0}) - \frac{n_{j_0}}{n_{j_0} + 1} \|x^* - \mu_{j_0}\|^2 \) and \( V(C_{j_1} \cup \{x^*\}) = V(C_{j_1}) + \frac{n_{j_1}}{n_{j_1} + 1} \|x^* - \mu_{j_1}\|^2 \) So it pays off to move a point from

\(^{24}\)We use here the symbol \( Q \) for the cluster quality function instead of \( J \) because \( Q \) does not fit axiomatic system for \( J \)—it is not scale-invariant and in case of consistency it changes in opposite direction, and with respect of richness we can only apply \( k \)-richness.
one cluster to another if \( \frac{n_{j_0}}{n_{j_0} - 1} \| x^* - \mu_{j_0} \|^2 > \frac{n_{j_1}}{n_{j_1} + 1} \| x^* - \mu_{j_1} \|^2 \). If we assume local optimality of \( C \), this obviously did not pay off. Now transform this data set to \( X' \) in that we transform elements of cluster \( C_{j_0} \) in such a way that it has new elements \( x'_j = x_j + \lambda(x_j - \mu_{j_0}) \) for some \( 0 < \lambda < 1 \), see Fig. D.4. Now consider a partition \( C' \) of \( X' \) all clusters of which are the same as in \( C \) except for the transformed elements that form now a cluster \( C'_{j_0} \). The question now is: does it pay off now to move a data point \( x'^* \) between the clusters? Consider the plane containing \( x'^*, \mu_{j_0}, \mu_{j_1} \). Project orthogonally the point \( x'^* \) onto the line \( \mu_{j_0}, \mu_{j_1} \), giving a point \( p \). Either \( p \) lies between \( \mu_{j_0}, \mu_{j_1} \) or \( \mu_{j_0} \) lies between \( p, \mu_{j_1} \). Properties of \( k \)-means exclude other possibilities. Denote distances \( y = \| x'^* - p \|, x = \| \mu_{j_0} - p \|, d = \| \mu_{j_0} - \mu_{j_1} \| \). In the second case the condition that moving the point does not pay off means:

\[
\frac{n_{j_0}}{n_{j_0} - 1}(x^2 + y^2) \leq \frac{n_{j_1}}{n_{j_1} + 1}((d + x)^2 + y^2).
\]

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

\[
\lambda^2 \frac{n_{j_0}}{n_{j_0} - 1}(x^2 + y^2) = \frac{n_{j_0}}{n_{j_0} - 1}((\lambda x)^2 + (\lambda y)^2) \leq \lambda^2 \frac{n_{j_1}}{n_{j_1} + 1}((d + x)^2 + y^2).
\]

\[
= \frac{n_{j_1}}{n_{j_1} + 1}((d + \lambda x)^2 + (\lambda y)^2) \text{ which means that it does not pay off to move the point } x'^* \text{ between clusters either. Consider now the first case and assume that it pays off to move } x'^* \text{. So we would have}
\]

\[
\frac{n_{j_0}}{n_{j_0} - 1}(x^2 + y^2) \leq \frac{n_{j_1}}{n_{j_1} + 1}((d + x)^2 + y^2)
\]

and at the same time:

\[
\frac{n_{j_0}}{n_{j_0} - 1} \lambda^2(x^2 + y^2) > \frac{n_{j_1}}{n_{j_1} + 1}((d - \lambda x)^2 + \lambda^2 y^2)
\]

Subtract now both sides:

\[
\frac{n_{j_0}}{n_{j_0} - 1} \lambda^2(x^2 + y^2) < \frac{n_{j_1}}{n_{j_1} + 1}((d - x)^2 + y^2) - \frac{n_{j_1}}{n_{j_1} + 1}((d - \lambda x)^2 + \lambda^2 y^2).
\]

This implies:

\[
\frac{n_{j_0}}{n_{j_0} - 1}(1 - \lambda^2)(x^2 + y^2) < \frac{n_{j_1}}{n_{j_1} + 1}((1 - \lambda^2)(x^2 + y^2) - 2d\lambda x).
\]

It is a contradiction because:

\[
\frac{n_{j_0}}{n_{j_0} - 1}(1 - \lambda^2)(x^2 + y^2) > \frac{n_{j_1}}{n_{j_1} + 1}((1 - \lambda^2)(x^2 + y^2) - 2d\lambda x).
\]

So it does not pay off to move \( x'^* \), hence the partition \( C' \) remains locally optimal for the transformed data set. If the data have one stable optimum only like in case of “well separated” normally distributed \( k \) real clusters, then both turn to global optima.

However, it is possible to demonstrate that the newly defined transform preserves also the global optimum of \( k \)-means. Let us consider first the simple case of two clusters only (2-means). Let the optimal clustering for a given set of

---

**Fig. D.4** Impact of contraction towards cluster centre by a factor lambda—local optimum maintained.
objects \( X \) consist of two clusters: \( T \) and \( Z \). The subset \( T \) shall have its gravity centre at the origin of the coordinate system. The quality of this partition \( Q((T, Z)) = n_T \text{Var}(T) + n_Z \text{Var}(Z) \) where \( n_T, n_Z \) denote the cardinalities of \( T, Z \) and \( \text{Var}(T), \text{Var}(Z) \) their variances (averaged squared distances to gravity centre).

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

\[
Q([A \cup B, C \cup D]) = n_A \text{Var}(A) + n_B \text{Var}(B) + n_C \text{Var}(C) + n_D \text{Var}(D) + \frac{n_C n_D}{n_C + n_D} (\mathbf{v}_C - \mathbf{v}_D)^2
\]

while

\[
Q([A \cup C, B \cup D]) = n_A \text{Var}(A) + n_D \text{Var}(D) + \frac{n_A n_D}{n_A + n_D} (\mathbf{v}_A - \mathbf{v}_D)^2
\]

and

\[
Q([A' \cup B', C \cup D]) = n_A \lambda^2 \text{Var}(A) + n_B \lambda^2 \text{Var}(B) + n_C \text{Var}(C) + n_D \text{Var}(D) + \frac{n_C n_D}{n_C + n_D} (\lambda \mathbf{v}_C - \mathbf{v}_D)^2
\]

while

\[
Q([A' \cup C, B' \cup D]) = n_A \lambda^2 \text{Var}(A) + n_D \text{Var}(D) + \frac{n_A n_D}{n_A + n_D} (\lambda \mathbf{v}_A - \mathbf{v}_D)^2
\]

The following must hold: \( Q([A' \cup B', C \cup D]) > Q([A' \cup D, B' \cup C]) \) and \( Q([A \cup B, C \cup D]) < Q([A \cup D, B \cup C]) \). Additionally also \( Q([A \cup B, C \cup D]) < Q([A \cup B \cup C, D]) \) and \( Q([A \cup B, C \cup D]) < Q([A \cup B \cup D, C]) \). These two latter inequalities imply:

\[
\frac{n_C n_D}{n_C + n_D} (\mathbf{v}_C - \mathbf{v}_D)^2 < \frac{(n_A + n_B) n_C}{(n_A + n_B) + n_C} \mathbf{v}_C^2
\]
and
\[
\frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2 < \frac{(n_A + n_B) n_D}{(n_A + n_B) + n_D} v_D^2
\]

Consider now an extreme contraction ($\lambda = 0$) yielding sets $A''$, $B''$ out of $A$, $B$. Then we have
\[
Q([A'' \cup B'', C \cup D]) - Q([A'' \cup C, B'' \cup D])
= \frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2 - \frac{n_A n_D}{n_A + n_D} v_D^2 - \frac{n_B n_C}{n_B + n_C} v_C^2
\]
\[
= \frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2
- \frac{n_A n_D}{n_A + n_D} \frac{(n_A + n_B) + n_D}{n_A + (n_A + n_B) + n_D} \left( \frac{n_A + n_B}{n_A + n_B + n_D} \right) v_D^2
- \frac{n_B n_C}{n_B + n_C} \frac{(n_A + n_B) + n_C}{n_A + (n_A + n_B) + n_C} v_C^2
\]
\[
= \frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2
- \frac{n_A}{n_A + n_B} \frac{(1 + \frac{n_B}{n_A + n_D})}{(n_A + n_B) + n_D} \left( \frac{n_A + n_B}{n_A + n_B + n_D} \right) v_D^2
- \frac{n_B}{n_A + n_B} \frac{(1 + \frac{n_A}{n_B + n_C})}{(n_A + n_B) + n_C} v_C^2
\]
\[
< \frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2
- \frac{n_A}{n_A + n_B} \frac{(n_A + n_B) n_D}{(n_A + n_B) + n_D} v_D^2
\]
\(- \frac{n_B}{n_A + n_B} \frac{(n_A + n_B)n_C}{n_A + n_B + n_C} v_C^2 < 0\)

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

\[
    h(x) = n_A x^2 v_A^2 + n_B x^2 v_B^2 + \frac{n_C n_D}{n_C + n_D} (v_C - v_D)^2
    - \frac{n_A n_D}{n_A + n_D} (x v_A - v_D)^2
    - \frac{n_B n_C}{n_B + n_C} (x v_B - v_C)^2
\]

It can be easily verified that \(h(x)\) is a quadratic polynomial with a positive coefficient at \(x^2\). Furthermore \(h(1) = Q(A \cup B, C \cup D) - Q(A \cup C, B \cup D) < 0, h(\lambda) = Q(A' \cup B', C \cup D) - Q(A' \cup C, B' \cup D) > 0, h(0) = Q(A'' \cup B'', C \cup D) - Q(A'' \cup C, B'' \cup D) < 0\). But no quadratic polynomial with a positive coefficient at \(x^2\) can be negative at the ends of an interval and positive in the middle. So we have the contradiction. This proves the thesis that the (globally) optimal 2-means clustering remains (globally) optimal after transformation.

Let us turn to the general case of \(k\)-means. Let the optimal clustering for a given set of objects \(X\) consist of two clusters: \(T\) and \(Z_1, \ldots, Z_{k-1}\). The subset \(T\) shall have its gravity centre at the origin of the coordinate system. The quality of this partition

\[
    Q(T, Z_1, \ldots, Z_{k-1}) =
    n_T V\text{ar}(T) + \sum_{i=1}^{k-1} n_{Z_i} V\text{ar}(Z_i)
\]

We will prove by contradiction that by applying our \(C\) transform we get partition that will be still optimal for the …. We shall assume the contrary that is that we can transform the set \(T\) by some \(1 > \lambda > 0\) to \(T'\) in such a way that optimum of 2-means clustering is not the partition \(\{T', Z\}\) but another one, say \(\{T'_1, Z_1 \cup \ldots \cup Z_{k-1,1}, T'_2 \cup Z_{1,2} \cup \ldots \cup Z_{k-1,2}, \ldots, T'_k \cup Z_{1,k} \cup \ldots \cup Z_{k-1,k}\}\) where \(Z_i = \bigcup_{j=1}^{k} Z_{i,j}, T'_1, \ldots, T'_k\) are transforms of sets \(T_1, \ldots, T_k\) for which in turn \(\bigcup_{j=1}^{k} T_j = T\). It may be easily verified that

\[
    Q(T, Z_1, \ldots, Z_{k-1}) = \sum_{j=1}^{k} n_{T_j} V\text{ar}(T_j) + \sum_{j=1}^{k} n_{T_j} v_{T_j}^2
    + \sum_{i=1}^{k-1} n_{Z_i} V\text{ar}(Z_i)
\]

while (denoting \(Z_{s,j} = \bigcup_{i=1}^{k-1} Z_{s,j}\))
\[ Q(\{T_1 \cup Z_{s,1}, \ldots, T_k \cup Z_{s,k}\}) = \]
\[ = \sum_{j=1}^{k} \left( n_{T_j} \text{Var}(T_j) + n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_{T_j} n_{Z_{s,j}}}{n_{T_j} + n_{Z_{s,j}}} (v_{T_j} - v_{Z_{s,j}})^2 \right) \]

whereas
\[ Q(\{T', Z_1, \ldots, Z_{k-1}\}) = \sum_{j=1}^{k} n_{T_j} \lambda^2 \text{Var}(T_j) + \sum_{j=1}^{k} n_{T_j} \lambda^2 v_{T_j}^2 \]
\[ + \sum_{i=1}^{k-1} n_{Z_i} \text{Var}(Z_i) \]

while
\[ Q(\{T'_1 \cup Z_{s,1}, \ldots, T'_k \cup Z_{s,k}\}) = \]
\[ = \sum_{j=1}^{k} \left( n_{T_j} \lambda^2 \text{Var}(T_j) + n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_{T_j} n_{Z_{s,j}}}{n_{T_j} + n_{Z_{s,j}}} (\lambda v_{T_j} - v_{Z_{s,j}})^2 \right) \]

The following must hold: \( Q(\{T', Z_1, \ldots, Z_{k-1}\}) > Q(\{T'_1 \cup Z_{s,1}, \ldots, T'_k \cup Z_{s,k}\}) \) and \( Q(\{T, Z_1, \ldots, Z_{k-1}\}) < Q(\{T_1 \cup Z_{s,1}, \ldots, Z_{s,k}\}) \). Additionally, also \( Q(\{T, Z_1, \ldots, Z_{k-1}\}) < Q(\{T \cup Z_{s,1}, Z_{s,2}, \ldots, Z_{s,k}\}) \) and \( Q(\{T, Z_1, \ldots, Z_{k-1}\}) < Q(\{T \cup Z_{s,2}, Z_{s,1}, Z_{s,3}, \ldots, Z_{s,k}\}) \) and...and \( Q(\{T, Z_1, \ldots, Z_{k-1}\}) < Q(\{T \cup Z_{s,k}, Z_{s,1}, \ldots, Z_{s,k-1}\}) \). These latter \( k \) inequalities imply that for \( l = 1, \ldots, k \):

\[ Q(\{T, Z_1, \ldots, Z_{k-1}\}) = n_T \text{Var}(T) + \sum_{j=1}^{k} n_{T_j} \text{Var}(T_j) + \sum_{j=1}^{k} n_{T_j} v_{T_j}^2 \]
\[ + \sum_{i=1}^{k-1} n_{Z_i} \text{Var}(Z_i) < \]
\[ < Q(\{T \cup Z_{s,l}, Z_{s,1}, \ldots, Z_{s,l-1}, Z_{s,l+1} \ldots, Z_{s,k}\}) = \]
\[ = n_T \text{Var}(T) + \sum_{j=1}^{k} n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_{T} n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}} (v_T - v_{Z_{s,j}})^2 \]
\[ + \sum_{i=1}^{k-1} n_{Z_i} \text{Var}(Z_i) < \]
\[ \sum_{j=1}^{k} n_{Z_{s,j}} \text{Var}(Z_{s,j}) + \frac{n_T n_{Z_{s,j}}}{n_T + n_{Z_{s,j}}}(v_T - v_{Z_{s,j}})^2 \]

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

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

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

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

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

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

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

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

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

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

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

\[ g(x) = \sum_{j=1}^{k} n_T x^2 T_j^2 + \sum_{i=1}^{k-1} n_{Z_i} \text{Var}(Z_i) \]
\[
- \sum_{j=1}^{k} \left( n_{Z*,j} \text{Var}(Z_{*,j}) + \frac{n_{T_j} n_{Z*,j}}{n_{T_j} + n_{Z*,j}} (x v_{T_j} - v_{Z*,j})^2 \right)
\]

It can be easily verified that \( g(x) \) is a quadratic polynomial with a positive coefficient at \( x^2 \). Furthermore

\[
g(1) = Q(\{T, Z_1, \ldots, z_{k-1}\}) - Q(\{T_1 \cup Z_{*,1}, \ldots, T_k \cup Z_{*,k}\}) < 0, \quad g(\lambda) = Q(\{T', Z_1, \ldots, z_{k-1}\}) - Q(\{T'_1 \cup Z_{*,1}, \ldots, T'_k \cup Z_{*,k}\}) > 0, \quad g(0) = Q(\{T'', Z_1, \ldots, z_{k-1}\}) - Q(\{T''_1 \cup Z_{*,1}, \ldots, T''_k \cup Z_{*,k}\}) < 0.
\]

But no quadratic polynomial with a positive coefficient at \( x^2 \) can be negative at the ends of an interval and positive in the middle. So we have the contradiction. This proves the thesis that the (globally) optimal \( k \)-means clustering remains (globally) optimal after transformation.

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

Hence

**Theorem D.10.1** \( k \)-means algorithm satisfies Scale-Invariance, \( k \)-Richness, and centric Consistency.

Note that \( (C^* \text{-based}) \) centric Consistency is not a specialization of Kleinberg’s consistency as the requirement of increased distance between all elements of different clusters is not required in \( C^* \) based Consistency. Note also that the decrease of distance does not need to be equal for all elements as long as the gravity centre does not relocate. Also a limited rotation of the cluster may be allowed for. But we could strengthen centric-consistency to be in concordance with Kleinberg’s consistency and under this strengthening \( k \)-means would of course still behave properly.

### D.11 Moving Clusters

Kleinberg’s consistency is split by some authors into the inner consistency (decreasing distances within the clusters) and the outer consistency (moving clusters away). We have just reformulated the inner consistency assumptions by Kleinberg. Let us see what can be done with outer consistency.

As we have stated already, in the \( \mathbb{R}^n \) it is actually impossible to move clusters in such a way as to increase distances to all the other elements of all the other clusters. However, we shall ask ourselves if we may possibly move away clusters as whole, via increasing the distance between cluster centres and not overlapping cluster regions, which, in case of \( k \)-means, represent Voronoi-regions.

Let us concentrate on the \( k \)-means case and let us look at two neighbouring clusters. The Voronoi regions are in fact polyhedrons, such that the "outer" polyhedrons (at least one of them) can be moved away from the rest without overlapping any other region.
So is such an operation on regions permissible without changing the cluster structure. A closer look at the issue tells us that it is not. As \( k \)-means terminates, the neighbouring clusters’ polyhedrons touch each other via a hyperplane such that the straight line connecting centres of the clusters is orthogonal to this hyperplane. This causes that points on the one side of this hyperplane lie more closely to the one centre, and on the other to the other one. But if we move the clusters in such a way that both touch each other along the same hyperplane, then if it happens that some points within the first cluster will become closer to the centre of the other cluster and vice versa. So moving the clusters generally will change their structure (points switch clusters) unless the points lie actually not within the polyhedrons but rather within “paraboloids” with appropriate equations. Then moving along the border hyperplane will not change cluster membership (locally). But the intrinsic cluster borders are now “paraboloids”. What would happen if we relocate the clusters allowing for touching along the “paraboloids”? The problem will occur again.

Hence the question can be raised: What shape should have the \( k \)-means clusters in order to be immune to movement of whole clusters?

Let us consider the problem of susceptibility to class membership change within a 2D plane containing the two cluster centres. Let the one cluster centre be located at a point \((0,0)\) in this plane and the other at \((2x_0, 2y_0)\). Let further the border of the first cluster be characterised by a (symmetric) function \( f(x) \) and let the shape of the border of the other one \( g(x) \) be the same, but properly rotated: \( g(x) = 2y_0 - f(x - 2x_0) \) so that the cluster centre is in the same. Let both have a touching point (we excluded already a straight line and want to have convex smooth borders). From the symmetry conditions one easily sees that the touching point must be \((x_0, y_0)\). As this point lies on the surface of \( f() \), \( y_0 = f(x_0) \) must hold. For any point \((x, f(x))\) of the border of the first cluster with centre \((0, 0)\) the following must hold:

\[
(x - 2x_0)^2 + (f(x) - 2f(x_0))^2 - x^2 - f^2(x) \geq 0
\]

That is

\[-2x_0(2x - 2x_0) - 2f(x_0)(2f(x) - 2f(x_0)) \geq 0\]

\[-f(x_0)(f(x) - f(x_0)) \geq x_0(x - x_0)\]

Let us consider only positions of the centre of the second cluster below the \( X \) axis. In this case \( f(x_0) < 0 \). Further let us concentrate on \( x \) lower than \( x_0 \). We get

\[-\frac{f(x) - f(x_0)}{x - x_0} \geq \frac{x_0}{-f(x_0)}\]

In the limit, when \( x \) approaches \( x_0 \).

\[-f'(x_0) \geq \frac{x_0}{-f(x_0)}\]
Now turn to $x$ greater than $x_0$. We get

$$- \frac{f(x) - f(x_0)}{x - x_0} \leq \frac{x_0}{-f(x_0)}$$

In the limit, when $x$ approaches $x_0$.

$$- f'(x_0) \leq \frac{x_0}{-f(x_0)}$$

This implies

$$- f'(x_0) = \frac{-1}{f(x_0)}$$

Note that $\frac{f(x_0)}{x_0}$ is the directional tangent of the straight line connecting both cluster centres. As well as it is the directional tangent of the line connecting the centre of the first cluster to its surface. $f'(x_0)$ is the tangential of the borderline of the first cluster at the touching point of both clusters. The equation above means both are orthogonal. But this property implies that $f(x)$ must be definition (of a part) a circle centred at $(0, 0)$. As the same reasoning applies at any touching point of the clusters, a $k$-means cluster would have to be (hyper)ball-shaped in order to allow the movement of the clusters without elements switching cluster membership.

The tendency of $k$-means to recognize best ball-shaped clusters has been known long ago, but we are not aware of presenting such an argument for this tendency.

It has to be stated however that clusters, even if enclosed in a ball-shaped region, need to be separated sufficiently to be properly recognized. Let us consider, under which circumstances a cluster $C_1$ of radius $r_1$ containing $n_1$ elements would take over $n_{21}$ elements (subcluster $C_{21}$) of a cluster $C_1$ of radius $r_2$ of cardinality $n_2$. Let $n_{22} = n_2 - n_{21}$ be the number of the remaining elements (subcluster $C_{22}$ of the second cluster. Let the enclosing balls of both clusters be separated by the distance (gap) $g$. Let us consider the worst case that is that the centre of the $C_{21}$ subcluster lies on a straight line segment connecting both cluster centres. The centre of the remaining $C_{22}$ subcluster would lie on the same line but on the other side of the second cluster centre. Let $r_{21}, r_{22}$ be distances of centres of $n_{21}$ and $n_{22}$ from the centre of the second cluster. The relations

$$n_{21} \cdot r_{22} = n_{22} \cdot r_{21}, \quad r_{21} \leq r_2, \quad r_{22} \leq r_2$$

must hold. Let denote with $SSC(C)$ the sum of squared distances of elements of the set $C$ to the centre of this set.

So in order for the clusters to be stable

$$SSC(C_1) + SSC(C_2) \leq SSC(C_1 \cup C_{21}) + SSC(C_{22})$$

must hold. But
Appendix D: Axiomatic Systems for Clustering

\[ SSC(C_2) = SSC(C_{21}) + SSC(C_{22}) + n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2 \]

\[ SSC(C_1 \cup C_2) = SSC(C_1) + SSC(C_{21}) + \frac{n_{1n_{21}}}{n_1 + n_{21}} (r_1 + r_2 + g - r_{21})^2 \]

Hence

\[ SSC(C_1) + SSC(C_{21}) + SSC(C_{22}) + n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2 \leq SSC(C_1) \]

\[ + SSC(C_{21}) + \frac{n_{1n_{21}}}{n_1 + n_{21}} (r_1 + r_2 + g - r_{21})^2 + SSC(C_{22}) \]

\[ n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2 \leq \frac{n_{1n_{21}}}{n_1 + n_{21}} (r_1 + r_2 + g - r_{21})^2 \]

\[ \frac{n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2}{n_{1n_{21}}/n_1 + n_{21}} \leq (r_1 + r_2 + g - r_{21})^2 \]

\[ \sqrt{\frac{n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2}{n_{1n_{21}}/n_1 + n_{21}}} \leq r_1 + r_2 + g - r_{21} \]

\[ \sqrt{\frac{n_{21} \cdot r_{21}^2 + n_{22} \cdot r_{22}^2}{n_{1n_{21}}/n_1 + n_{21}}} - r_1 - r_2 + r_{21} \leq g \]

\[ \sqrt{n_{21} \cdot r_{21}^2 + n_{21} \cdot r_{21} \cdot r_{22}} \cdot (1/n_1 + 1/n_{21}) - r_1 - r_2 + r_{21} \leq g \]

\[ \sqrt{(r_{21}^2 + r_{21} \cdot r_{22}) (n_{21}/n_1 + 1)} - r_1 - r_2 + r_{21} \leq g \]

As \( r_{22} = \frac{r_{21}n_{21}}{n_2 - n_{21}} \)

\[ \sqrt{(r_{21}^2 + r_{21} \cdot \frac{r_{21}n_{21}}{n_2 - n_{21}}) (n_{21}/n_1 + 1)} - r_1 - r_2 + r_{21} \leq g \]

\[ r_{21} \sqrt{(1 + \frac{n_{21}}{n_2 - n_{21}}) (n_{21}/n_1 + 1)} - r_1 - r_2 + r_{21} \leq g \]
Appendix D: Axiomatic Systems for Clustering

Let us consider the worst case when the elements to be taken over are at the “edge” of the cluster region \( (r_{21} = r_2) \). Then

\[
r_{21} \sqrt{\frac{n_2}{n_2 - n_{21}} \left( \frac{n_1 + n_{21}}{n_1} \right)} - r_1 - r_2 + r_{21} \leq g
\]

\[
r_{21} \sqrt{\frac{n_2 n_1 + n_{21}}{n_1 n_2 - n_{21}}} - r_1 - r_2 + r_{21} \leq g
\]

The lower limit on \( g \) will grow with \( n_{21} \), but \( n_{21} \leq 0.5n_2 \), because otherwise \( r_{22} \) would exceed \( r_2 \). Hence in the worst case

\[
r_2 \sqrt{\frac{n_2 n_1 + n_2/2}{n_1 n_2/2}} - r_1 \leq g
\]

\[
r_2 \sqrt{2(1 + 0.5n_2/n_1)} - r_1 \leq g
\]

In case of clusters with equal sizes and equal radius this amounts to

\[
g \geq r_1 (\sqrt{3} - 1) \approx 0.7r_1
\]

D.12 Learnability Oriented Axiomatisation

Let us now look at ways to formulate a learnability based axiomatic framework for clustering algorithms. The clustering algorithm is inevitably connected to the clusterings it can produce. So, first let us state the axioms necessary for clustering.\(^{25}\)

Axiom D.12.1 *(Strong learnability)* Clustering must split the sample space (for each point in sample space we must be able to say to which cluster it belongs or if it belongs to the undecided space),

Axiom D.12.2 *(Weak learnability)* Clustering must be learnable from finite sample (the clustering to be discovered must belong to a class of clusterings such that the class is learnable in the sense of learnability theory),

Axiom D.12.3 *(Separability)* Clusters in the clustering must be separable (for a sufficiently large sample it should be significantly more probable to have the closest neighbour from the same cluster than from a different one),

Axiom D.12.4 *(Enlightening)* Clustering must be enlightening (new information should be obtained via clustering compared to prior knowledge).

Therefore:

Axiom D.12.5 *(Learnability)* A clustering algorithm with high probability shall return a clustering if the intrinsic clustering of the data belongs to the class of clusterings for which the algorithm is designed and this clustering should be close to the intrinsic one,

Axiom D.12.6 *(Strong learnability)* A clustering algorithm with high probability shall return a failure information if the intrinsic clustering of the data does not belong to the class of clusterings for which the algorithm is designed or if there is no clustering behind the data,

Axiom D.12.7 *(Separability)* A clustering algorithm shall state with high reliability what is the separation between clusters,

Axiom D.12.8 *(Enlightening)* A clustering algorithm shall verify if there is a difference between the detected clustering and the prior knowledge of the sample space.

Note that these axioms are in opposition to axioms of Kleinberg. Learnability contradicts the richness axiom. Separability contradicts invariance. Enlightening opposes consistency.

D.13 Graph Clustering Axiomatization

In the preceding sections we considered the case, when the clustered objects are placed in a space, where distances between objects may be defined. But a large portion of this book was devoted to clustering of graphs (Chap. 5).

Hence we cannot overlook axiomatic frameworks developed specially for them. Van Laarhoven and Marchiori\(^{26}\) developed an axiomatic framework, defining the desired properties of the clustering quality function, understood in a similar way as previously.

Their first axiom assumes “Permutation invariance”, namely.

Axiom D.13.1 *(Permutation invariance)* The graph partition quality depends on node similarities but not on node labels.

The second axiom is the “Scale invariance” which means that

**Axiom D.13.2** *(Scale invariance)* Proportional increase/decrease of weights of edges (node similarities) does not change the partition quality.

Authors quoted request also the “Richness”, that is

**Axiom D.13.3** *(Richness)* For any partition of the sets of nodes there exists a graph, for which this partition is optimal with respect to the given quality function among all the partitions of the graph considered.

Further, they request the “Consistent improvement monotonicity”, that is

**Axiom D.13.4** *(Consistent improvement monotonicity)* If the edge weights within clusters are increased and between clusters they are decreased, then the quality function shall increase.

One sees immediately that these axioms parallel those from Sect. D.2—the difference is that we talk now about similarity measures and not distances and that some similarities could be equal zero. A similarity equal to zero breaks Kleinberg’s contradiction proof, because scaling by the lowest and highest distances is necessary, and in this case the highest distance is infinite. However, there are two difficulties here. One is that the graph may have all edges with non-zero weights. In this case the claim of breaking Kleinberg’s contradiction is not valid. Furthermore, the weight zero stems frequently from the fact of thresholding the similarity level between nodes. In this case the Scale invariance is violated, because an increase in similarity would introduce new edges, its decrease would remove the existing ones (crossing the threshold).

Van Laarhoven and Marchiori introduce, as well, an axiom for “Local consistency” (violated by the algorithms with a fixed number of clusters):

**Axiom D.13.5** *(Local consistency)* For graphs agreeing on some subgraphs, if a change of clustering in the common subgraph increases the quality function in one graph, then the same change should cause an increase in the other.

Their final axiom is called “Continuity”. That is

**Axiom D.13.6** *(Continuity)* For each $\epsilon > 0$ there exists a $\delta > 0$ such that for two graphs differing by edge weights for each edge by only $\delta$, the quality functions of the two graphs differ only by $\epsilon$ for each clustering $C$ (of both graphs).

The same authors show that the popular graph partition quality measure called modularity$^{27}$ conforms only to four of their axioms (richness, continuity, scale invariance, permutation invariance) but violates monotonicity and locality.

Therefore, they propose a new quality function, called adaptive scale modularity,

$$Q_{M,C} = \sum_{c \in C} \left( \frac{w(c)}{M + C_{vol}(c)} - \left( \frac{vol(c)}{M + C_{vol}(c)} \right)^2 \right)$$

where $\text{vol}(c)$ is the volume of the cluster $c$, and $w(c)$ is the sum of weights of edges within the cluster $c$. This function satisfies all their axioms for $M = 0, \mathcal{C} \geq 2$, and violates only scale invariance for $M > 0$.

For comparison, the original modularity was of the form

$$Q_{M, \mathcal{C}} = \sum_{c \in \mathcal{C}} \left( \frac{w(c)}{\text{vol}(G)} - \frac{\text{vol}(c)}{\text{vol}(G)} \right)^2$$

where $\text{vol}(G)$ is the volume of the entire graph $G$.

It has been observed by Fortunato and Barthelemy\textsuperscript{28} that modularity suffers from the so-called resolution limit. Consider a ring of cliques in which cliques are interconnected by a single link only. One would expect that an optimal clustering would contain exactly one clique in one cluster. However, optimality with respect to modularity does not behave in this way. Therefore, Fortunato and Barthelemy require that graph partition quality measure should be Resolution-limit-free. This means that: If a partition $\mathcal{C}$ of a graph $G$ is optimal with respect to such a measure among all partitions of $G$, then for each subgraph $G'$ of $G$ the partition $\mathcal{C}'$ induced from $\mathcal{C}$ by $G'$ should also be optimal among all partitions of $G'$.

Adaptive scale modularity does not fulfil this requirement but it does not suffer, nonetheless, from the resolution limit. Hence, the axiom of resolution-limit-freedom remains an open question for further investigations.

Let us make a further remark on modularity. It has a clear interpretation as a difference between the current link structure and a random one. This is in concordance with the enlightening axiom, mentioned before. The adaptive scale modularity lacks such an interpretation and one can suspect that it will promote big clusters (high volume compared to border, that is the sum of cluster node degrees being much larger than the number of edges linking the given cluster with other clusters). Hence, it seems that lack of contradictions in an axiomatic system is a necessary condition for the adequacy of this system, but it is not sufficient.

Furthermore, it turns out that finding a partition optimising the value of modularity is NP hard. Hence, in fact approximate, but fast algorithms are used instead, without even any guarantees of giving a solution within some bounds with respect to the optimal one.

Therefore, it seems reasonable to press on learnability rather than on richness in axiomatic systems.

This concluding remark would be conform with the opinion of Estivill-Castro, expressed in his position paper\textsuperscript{29} that the lack of common definition of a cluster is the primary reason for the multitude of the clustering algorithms. The required properties of a clustering depend on the application. That is we have to define our notion of clustering and then prove that it is learnable from data using one’s algorithm.

---


Appendix E
Justification for the $k$-means++ Algorithm

The ultimate goal of the $k$-means algorithm is to minimise the partition cost function stated in formula (3.1)\(^ {30}\)

\[
J(U, M) = \sum_{i=1}^{m} \sum_{j=1}^{k} u_{ij} \|x_i - \mu_j\|^2
\]

(E.1)

where $u_{ij}$ is equal 1 if among all of cluster centres $\mu_j$ is the closest to $x_i$, and is 0 otherwise.

As already stated, Arthur and Vassilvitskii [32] prove the following property of their $k$-means++ algorithm.

**Theorem E.0.1 [32]** The expected value of the partition cost, computed according to Eq. (3.1), is delimited in case of $k$-means++ algorithm by the inequality

\[
\mathbb{E}(J) \leq 8(\ln k + 2) J_{\text{opt}}
\]

(E.2)

where $J_{\text{opt}}$ denotes the optimal value of the partition cost.

(For $k = 3$ this amounts to more than 24). Note that the proof in [32] refers to the initialisation step of the algorithm only so that subsequently we assume that $k$-means++ clustering consists of this initialisation only. In the full algorithm $\mathbb{E}(J)$ is lower than after initialisation step only as $k$-means can partially eliminate effects of poor initialisation.

The motivation for the invention of the $k$-means++ algorithm is, as [32] states, the poor performance of $k$-means for many real examples (i.e., $\frac{J}{J_{\text{opt}}}$ is unbounded even when $m$ and $k$ are fixed). However, $k$-means turns out to exhibit an appealing simplicity. Therefore it makes sense to try to improve its clustering quality. Arthur and Vassilvitskii created an algorithm with some goodness guarantees.


© Springer International Publishing AG 2018
S. T. Wierzchoń and M. A. Klopotek, Modern Algorithms of Cluster Analysis, Studies in Big Data 34, https://doi.org/10.1007/978-3-319-69308-8
On the other hand, Pollard [395] proved an interesting property of a hypothetical \( k \)-means\(_{opt} \) algorithm that would produce a partition yielding \( J_{opt} \) for finite sample. He demonstrated the convergence of \( k \)-means\(_{opt} \) algorithm to proper population clustering with increasing sample size. So one can state that such an algorithm would be strongly consistent. The problem of that algorithm is that it is \( NP \)-hard [395].

So let us investigate whether or not a realistic algorithm, \( k \)-means++, featured by reasonable complexity and reasonable expectation of closeness to optimality, is also strongly consistent.

Note that there existed and exists interest in strong consistency results concerning variants of \( k \)-means algorithm. However, like in [457] or [338], an abstract global optimiser is referred to rather than an actual algorithm performance.

In what follows we extend the results of Pollard.

Assume that we are drawing independent samples of size \( m = k, k+1, \ldots \) from some probability distribution \( P \). Assume further, that the number of disjoint balls of radius \( \epsilon \) with non-zero \( P \) measure is much larger than \( k \).

Furthermore let

\[
\int \|x\|^{2(k+1)} P(dx) \]

be finite.

Given a sample consisting of \( m \) vectors \( x_i, i = 1, \ldots, m \), it will be convenient to consider an empirical probability distribution \( P_m \) assigning a probability mass of \( m^{-1} \) at each of the sample points \( x_i \). Let further \( M \) denote any set of up to \( k \) points (serving as an arbitrary set of cluster centres).

Let us introduce the function \( J_{cp}(., .) \), that for any set \( M \) of points in \( \mathbb{R}^n \) and any probability distribution \( Q \) over \( \mathbb{R}^n \), computes \( k \)-means quality function normalised over the probability distribution:

\[
J_{cp}(M, Q) = \int D(x, M)^2 Q(dx) \quad (E.3)
\]

where \( D(x, M) = \min_{\mu \in M} \|x - \mu\| \).

Then \( J_{cp}(M, P_m) \) can be seen as a version of the function \( J(\cdot) \) from Eq. (3.1).

Note that for any sets \( M, M' \) such that \( M \subset M' \) we have \( J_{cp}(M, Q) \geq J_{cp}(M', Q) \).

Note also that if \( M \) has been obtained via the initialisation step of \( k \)-means++, then the elements of \( M \) can be considered as ordered, so we could use indexes \( M_{[p:q]} \) meaning elements \( \mu_p, \ldots, \mu_q \) from this sequence. As the cluster around \( M_{[k]} \) will contain at least one element (\( \mu_k \) itself), we can be sure that \( J_{cp}(M_{[1:k-1]}, P_m) \geq J_{cp}(M_{[1:k]}, P_m) \) as no element has a chance to be selected twice. Hence also

\[
\mathbb{E}_M(J_{cp}(M_{[1:k-1]}, P_m)) > \mathbb{E}_M(J_{cp}(M_{[1:k]}, P_m)).
\]

If we assume that the distribution \( P \) is continuous, then for the expectation over all possible \( M \) obtained from initialisation procedure of \( k \)-means++, for any \( j = 2, \ldots, k \) the relationship

\[
\mathbb{E}_M(J_{cp}(M_{[1:j-1]}, P)) > \mathbb{E}_M(J_{cp}(M_{[1:j]}, P))
\]

holds. The reason is as follows: The last point, \( \mu_k \), was selected from a point of non-zero density, so that in ball of radius \( \epsilon_b > 0 \) such that \( \epsilon_b \) is lower than \( 1/3 \) of the
distance of $\mu_j$ to each $\mu \in M_{[1:j-1]}$ around it, each point would be re-clustered to $\mu_j$ diminishing the overall within-cluster variance. As it holds for each set $M$, so it holds for the expected value too.

But our goal here is to show that if we pick the sets $M$ according to $k$-means++ (initialization) algorithm, then $E_M(J_{cp}(M, P_m)) \rightarrow_{m \rightarrow \infty} E_M(J_{cp}(M, P))$, and, as a consequence, $E_M(J_{cp}(M, P)) \leq 8(\ln k + 2)J_{opt}$, where $J_{opt}$ is the optimal quality for the population.

We show this below.

Let $T_{m,k}$ denote the probability distribution for choosing the set $M$ as cluster centres for a sample from $P_m$ using $k$-means++ algorithm. From the $k$-means++ algorithm we know that the probability of choosing $\mu_j$ given $\mu \in M_{[1:j-1]}$ have already been picked amounts to

$$
\text{Prob} (\mu_j | M_{[1:j-1]}) = \frac{D(\mu_j, M_{[1:j-1]})^2}{\int D(x, M_{[1:j-1]})^2 P_m(dx)}
$$

whereas the point $\mu_1$ is picked according to the probability distribution $P_m$. Obviously, the probability of selecting the set $M$ amounts to:

$$
\text{Prob}(M) = \text{Prob}(\mu_1) \cdot \text{Prob}(\mu_2 | M_{[1]}) \cdot \text{Prob}(\mu_3 | M_{[1:2]}) \cdots \text{Prob}(\mu_k | M_{[1:k-1]})
$$

So the probability density $T_{m,k}$ may be expressed as

$$
T_{m,k}(dM) = P_m(\mu_1) \cdot \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{\int D(x, M_{[1:j-1]})^2 P_m(dx)}
$$

For $P$ we get accordingly the $T_k$ distribution.

Now assume we ignore with probability of at most $\delta > 0$ a distant part of the potential locations of cluster centres from $T_k$ via a procedure described below. Denote with $M_{\delta}$ the subdomain of the set of cluster centres left after ignoring these elements. Instead of expectation of

$$
E_M(J_{cp}(M, P_m)) = \int J_{cp}(M, P_m)T_{m,k}(dM)
$$

we will be interested in

$$
E_{M,\delta}(J_{cp}(M, P_m)) = \int_{M \in M_{\delta}} J_{cp}(M, P_m)T_{m,k}(dM)
$$

and respectively for $P$ instead of

$$
E_M(J_{cp}(M, P)) = \int J_{cp}(M, P)T_k(dM)
$$

we want to consider

$$
E_{M,\delta}(J_{cp}(M, P)) = \int_{M \in M_{\delta}} J_{cp}(M, P)T_k(dM)
$$
\( \mathbb{E}_M(J_{cp}(M, P)) \) is apparently finite.

Let us concentrate on the difference \( |\mathbb{E}_{M, \delta}(J_{cp}(M, P)) - \mathbb{E}_{M, \delta}(J_{cp}(M, P_m))| \) and show that it converges to 0 when \( m \) is increased for a fixed \( \delta \), that is for any \( \epsilon \) and \( \delta \) there exists such an \( m_{\delta, \epsilon} \) that for any larger \( m \) almost surely (a.s.) this difference is lower than \( \epsilon \). Thus by decreasing \( \delta \) and \( \epsilon \) we find that the expectation on \( P_m \) converges to that on \( P \).

The subdomain selection (ignoring \( \delta \) share of probability mass of \( T_m \)) shall proceed as follows. Let \( \delta_1 = \sqrt{1 + \frac{1}{\delta} - 1} \).

Let us denote \( J_{m, opt} = \min_M J_{cp}(M, P_m), J_{opt} = \min_M J_{cp}(M, P) \), where \( M \) is a set of cardinality at most \( k \). Let us denote \( J_{m, opt, j} = \min_M J_{cp}(M, P_m), J_{opt, j} = \min_M J_{cp}(M, P) \), where \( M \) be a set of cardinality at most \( j, j = 1, \ldots, k \).

Let \( R_1 \) be a radius such that \( \int_{\|x\| > R_1} P(d\mathbf{x}) < \delta_1 \). Further, let \( R_j \) for \( j = 2, \ldots, k \) be such that \( \int_{\|x\| > R_j} (\|x\| + R_j)^2 P(d\mathbf{x}) < \delta_1 \).

We reject all \( M \) such that for any \( j = 1, \ldots, k \|\mathbf{\mu}_j\| > R_j \).

Let us show now that the rejected \( M \)s constitute only \( \delta \) of probability mass or less.

Consider the difference (for any \( l = 2, \ldots, k \)) between the probability mass before and after rejection.

\[
\int_{\|\mathbf{\mu}_1\| \leq R_1} \cdots \int_{\|\mathbf{\mu}_l\| \leq R_1} \prod_{j=2}^l \frac{D(\mathbf{\mu}_j, M_{[1:j-1]}^2}{D(\mathbf{x}, M_{[1:j-1]}^2} P(d\mathbf{\mu}_1) \cdots P(d\mathbf{\mu}_1)
\]

\[
- \int_{\|\mathbf{\mu}_1\| \leq R_1} \cdots \int_{\|\mathbf{\mu}_l\| \leq R_1} \prod_{j=2}^l \frac{D(\mathbf{\mu}_j, M_{[1:j-1]}^2}{D(\mathbf{x}, M_{[1:j-1]}^2} P(d\mathbf{\mu}_1) \cdots P(d\mathbf{\mu}_1)
\]

which is obviously non-negative.

Let us introduce the following notation: \( \mathbf{r} \) be a vector of relations from \( \{\leq, >\}^l \). So the above can be rewritten as

\[
\sum_{\mathbf{r} \in \{\leq, >\}^l} \int_{\|\mathbf{\mu}_1\| \leq \mathbf{r}_1 R_1} \cdots \int_{\|\mathbf{\mu}_l\| \leq \mathbf{r}_l R_l} \prod_{j=2}^l \frac{D(\mathbf{\mu}_j, M_{[1:j-1]}^2}{D(\mathbf{x}, M_{[1:j-1]}^2} P(d\mathbf{\mu}_1) \cdots P(d\mathbf{\mu}_1)
\]

\[
- \int_{\|\mathbf{\mu}_1\| \leq R_1} \cdots \int_{\|\mathbf{\mu}_l\| \leq R_1} \prod_{j=2}^l \frac{D(\mathbf{\mu}_j, M_{[1:j-1]}^2}{D(\mathbf{x}, M_{[1:j-1]}^2} P(d\mathbf{\mu}_1) \cdots P(d\mathbf{\mu}_1)
\]

which is equivalent to

\[
\sum_{\mathbf{r} \in \{\leq, >\}^l \setminus \{\leq\}^l} \int_{\|\mathbf{\mu}_1\| \leq \mathbf{r}_1 R_1} \cdots \int_{\|\mathbf{\mu}_l\| \leq \mathbf{r}_l R_l} \prod_{j=2}^l \frac{D(\mathbf{\mu}_j, M_{[1:j-1]}^2}{D(\mathbf{x}, M_{[1:j-1]}^2} P(d\mathbf{\mu}_1) \cdots P(d\mathbf{\mu}_1)
\]

Consider that for any \( j \geq 2 \)
Appendix E: Justification for the \(k\)-means++ Algorithm

\[
\int_{\|\mu_j\| \leq R_j} \frac{D(\mu_j, M_{[1:j-1]})^2}{\int D(x, M_{[1:j-1]})^2 P(dx)} P(d\mu_j) \leq 1
\]

On the other hand

\[
\int_{\|\mu_j\| > R_j} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{, \text{opt}, j-1}} P(d\mu_j) \leq \frac{(\|\mu_j\| + R_1)^2}{J_{, \text{opt}, j-1}} \leq \frac{1}{\delta_1}
\]

So the above sum is lower equal to:

\[
\leq \sum_{r \in \{\leq, >\} - \{\leq\}} \delta_1 \text{count}(>, r)
\]

where \(\text{count}(>, r)\) is the number of \(>\) relations in the vector \(r\).

\[
= (1 + \delta_1)^l - 1 \leq \delta
\]

This confirms that upon the above restriction on \(R_j\) we reject at most \(\delta\) of the mass of \(\mu_l\) points under \(T_k\).

So let us consider

\[
|\mathbb{E}_{M, \delta}(J_{cp}(M, P)) - \mathbb{E}_{M, \delta}(J_{cp}(M, P_m))| = \left| \int_{M \in M_k} J_{cp}(M, P_m) T_{m,k}(dM) - \int_{M \in M_k} J_{cp}(M, P) T_k(dM) \right|
\]

\[
= \left| \int_{M \in M_k} J_{cp}(M, P_m) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{\int D(x, M_{[1:j-1]})^2 P_m(dx)} P_m(d\mu_k) \ldots P_m(d\mu_1) \right|
\]

\[
- \left| \int_{M \in M_k} J_{cp}(M, P) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{\int D(x, M_{[1:j-1]})^2 P(dx)} P(d\mu_k) \ldots P(d\mu_1) \right|
\]

\[
= \left| \int_{M \in M_k} J_{cp}(M, P_m) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P_m)} P_m(d\mu_k) \ldots P_m(d\mu_1) \right|
\]
Appendix E: Justification for the $k$-means++ Algorithm

For a sufficiently large $m$ \( \prod_{j=2}^{k} \frac{1}{J_{cp}(M_{[1:j-1]}, \mu_{j-1})} \) differs from \( \prod_{j=2}^{k} \frac{1}{J_{opt}(M_{[1:j-1]}, \mu_{j-1})} \) by at most \( \epsilon_{opt} \) (due to the result of Pollard). This difference can be made also that small between \( \prod_{j=2}^{k} \frac{1}{J_{cp}(M_{[1:j-1]}, \mu_{j-1})} \) and \( \prod_{j=2}^{k} \frac{1}{J_{opt}(M_{[1:j-1]}, \mu_{j-1})} \) by increasing $m$. The argument runs as follows: We can always find a radius $R$ for which \( J_{cp}(x, M_{[1:j-1]})^2 P(dx) \) can be made as small as we want so that it is negligible for our purposes and within the limited domain of radius $R$ with increase of $m$ both \( J_{cp}(x, M_{[1:j-1]})^2 P_m(dx) \) and \( J_{cp}(x, M_{[1:j-1]})^2 P(dx) \) become as close as we want. Therefore

\[
\leq \left| \int_{M \in M_k} J_{cp}(M, P_m) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2 (1 \pm \epsilon_{opt})}{J_{cp}(M_{[1:j-1]}, P)} P_m(d\mu_j) \ldots P_m(d\mu_1) \right|
\]

\[
- \int_{M \in M_k} J_{cp}(M, P) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P(d\mu_j) \ldots P(d\mu_1)
\]

\[
\leq \epsilon_{opt} \left| \int_{M \in M_k} J_{cp}(M, P_m) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P_m(d\mu_j) \ldots P_m(d\mu_1) \right|
\]

\[
+ \left| \int_{M \in M_k} J_{cp}(M, P_m) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P_m(d\mu_j) \ldots P_m(d\mu_1) \right|
\]

\[
- \int_{M \in M_k} J_{cp}(M, P) \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P(d\mu_j) \ldots P(d\mu_1)
\]

The first addend, the product of an \( \epsilon_{opt} \) and a finite quantity, can be set as low as needed by choosing a sufficiently low \( \epsilon_{opt} \).

Now following Pollard, we will decrease as much as required the second addend, which we will rewrite as.

\[
\left| \int_{M \in M_k} J_{cp}(x, M)^2 \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P_m(dx) P_m(d\mu_j) \ldots P_m(d\mu_1) \right|
\]

\[
- \int_{M \in M_k} J_{cp}(x, M)^2 \prod_{j=2}^{k} \frac{D(\mu_j, M_{[1:j-1]})^2}{J_{cp}(M_{[1:j-1]}, P)} P(dx) P(d\mu_j) \ldots P(d\mu_1)
\]
So select a finite set \( T_\beta \) of points from \( M_\delta \) such that each element of \( M_\delta \) lies within a distance of \( \beta \) from a point of \( T_\beta \).

Let us define the function \( g_M(x) = D(x, M)^2 \), the distance of \( x \) from any point \( M \), as follows: \( M^* \) be the set of elements from \( T_\beta \) such that \( \| \mu_j - M^*[j] \| \) is less than \( \beta \), and \( |J^{-1}_{c_p}(M, P) - J^{-1}_{c_p}(M^*, P)| \) is less than \( \beta \). Then \( D(x, M) = D(x, M^*) + \beta \), where \( D(x, M) = \max(D(x, M^*) - \beta, 0); \quad J^{-1}_{c_p}(M, P) = J^{-1}_{c_p}(M^*) + \beta \) and \( J^{-1}_{c_p}(M, P) = \max(J^{-1}_{c_p}(M^*, P) - \beta, J^{-1}_{c_p}(M, P)) \), where \( J_{\max} \) is the poorest initialisation within the domain \( M_\delta \). Now define the function

\[
g_M(x) = D(x, M)^2 \prod_{j=2}^{k} D(\mu_j, M_{1:j-1})^2 J_{c_p}(M_{1:j-1}, P)
\]

and

\[
\overline{g}_M(x) = D(x, M)^2 \prod_{j=2}^{k} \overline{D}(\mu_j, M_{1:j-1})^2 \overline{J}_{c_p}(M_{1:j-1}, P)
\]

As \( \|x, M^*[j]\| - \beta \leq \|x - \mu_j\| \leq \|x, M^*[j]\| + \beta \), it is easily seen that \( \overline{D}(x, M) \leq D(x, M) \leq \overline{D}(x, M) \), similarly with \( J \), and hence \( \overline{g}_M(x) \leq g_M(x) \leq \overline{g}_M(x) \).

Therefore

\[
\left| \int_{M \in M_k} \int_x g_M(x) P(dx) P(d\mu_k) \ldots P(d\mu_1) \right| - \int_{M \in M_k} \int_x g_M(x) P_m(dx) P_m(d\mu_k) \ldots P_m(d\mu_1)
\]

is bounded from above by

31For a constraint domain of \( M \) this is always possible for the following reason. Let \( M^* \) and \( M^{**} \) be two sets of up to \( k \) distinct “corresponding” cluster centres. “Corresponding” shall mean here that \( \mu^*_j \) and \( \mu^{**}_j \) are much closer to each other than any other centre from \( M^+ = M^* \cup M^{**} \). Let the distance between corresponding cluster centres be at most \( d \). Recall that

\[
J_{c_p}(M, Q) = \int D(x, M) Q(dx) = \int (\min_{\mu \in M} \|x - \mu\|)^2 Q(dx)
\]

This implies that \( J_{c_p}(M^*, Q) \geq J_{c_p}(M^+, Q) \) and \( J_{c_p}(M^{**}, Q) \geq J_{c_p}(M^+, Q) \). Note that \( \|x - \mu^*\|^2 - \|x - \mu^{**}\|^2 \). Hence

\[
J_{c_p}(M^*, Q) - J_{c_p}(M^+, Q) \leq d \int \min_{\mu^* \in M^*} (2\|x - \mu^*\| + d) \cdot Q(dx)
\]

This difference can be kept as low as requested (but positive) by diminishing \( d \). Anallogous result is obtained for \( J_{c_p}(M^{**}, Q) - J_{c_p}(M^+, Q) \). If we keep in mind that \( |J_{c_p}(M^*, Q) - J_{c_p}(M^{**}, Q)| \leq J_{c_p}(M^*, Q) - J_{c_p}(M^+, Q) + J_{c_p}(M^{**}, Q) - J_{c_p}(M^+, Q) \) and that \( J_{c_p} \) has always a positive minimum, then obviously we can keep the mentioned difference in a bounded area below \( \beta \) by an appropriately dense grid.
A sufficient increase on $m$ will make the second addend as small as we like. We will show below that using an argument similar to Pollard, one can demonstrate that the first addend can be diminished as much as we like by appropriate choice of $\beta$. Hence the restricted expectation can be shown to converge as claimed.

For any $R$

$$\int_{M \in M_1} \int_{x \in M} (\bar{g}_M(x) - \underline{g}_M(x)) P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

$$= \int_{M \in M_1} \int_{\|x\| \leq R} (\bar{g}_M(x) - \underline{g}_M(x)) P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

$$+ \int_{M \in M_1} \int_{\|x\| > R} (\bar{g}_M(x) - \underline{g}_M(x)) P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

The second addend has the property:

$$\int_{M \in M_1} \int_{\|x\| > R} (\bar{g}_M(x) - \underline{g}_M(x)) P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

$$\leq \int_{M \in M_1} \int_{\|x\| > R} \bar{g}_M(x) P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

$$\leq \int_{M \in M_1} \int_{\|x\| > R} (2R_k + 2\sigma)^{2k+2} (\|x\| + R_k + 2\sigma)^2 P(dx) P(d\mu_k) \ldots P(d\mu_1)$$

which can be decreased as much as necessary by increasing $R$. The first addend can be transformed.
where $h_M(x)$ is a function of $x$ and $M$. As both $x$ and elements of $M$ are of bounded length, $h_M(x)$ is limited from above. Hence this addend can be decreased to the desired value by appropriate choice of (small) $\sigma$.

Therefore the whole term can be decreased to a desired value (above zero of course), what we had to prove.

Hence our claim about consistency of $k$-means++ has been proven.
References


© Springer International Publishing AG 2018
References

47. G.H. Ball, D.J. Hall, ISODATA: a novel method of data analysis and pattern classification. Technical Report NTIS AD 699616 (Stanford Research Institute, Stanford, CA, 1965)  
74. N. Biggs, Algebraic Graph Theory, 2nd edn. (Cambridge University Press, 1993). https://doi.org/10.1017/CBO9780511608704
75. C.M. Bishop, Neural Networks for Pattern Recognition (Clarendon Press, London, 1995)
78. A. Blum, Thoughts on Clustering. Essay for the 2009 NIPS Workshop “Clustering: Science or Art?” (2009)
80. A. Bonato, A Course on the Web Graph, vol. 89 (Graduate Studies in Mathematics (AMS, New York, 2008)


139. J.V. de Oliveira, W. Pedrycz (eds.), Advances in Fuzzy Clustering and its Applications (Wiley, New York, 2007)
References


147. I.S. Dhillon, Y. Guan, J. Kogan, Iterative clustering of high dimensional text data augmented by local search, in Proceedings of IEEE International Conference on Data Mining (Maebashi City, Japan, 2002), pp. 131–138

148. I.S. Dhillon, Y. Guan, J. Kogan, Refining clusters in high dimensional text data, in 2-nd SIAM ICDM, Workshop on Clustering High Dimensional Data (Arlington, VA, 2002). http://citeseer.ist.psu.edu/dhillon02refining.html


244. D. Haussler, Convolution kernels on discrete structure. Technical Report UCSC-CRL-99-10 (Department of Computer Science, University of California at Santa Cruz, Santa Cruz, CA, USA, 1999)


References


References

297. J. Kogan, Introduction to Clustering Large and High-Dimensional Data (Cambridge University Press, Cambridge, 2007)
323. D.A. Levin, Y. Peres, E.L. Wilmer, Markov Chains and Mixing Times (AMS, 2009)
347. Ch.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval (Cambridge University Press, 1 Apr 2009)
351. M. Meilă, W. Pentney, Clustering by weighted cuts in directed graphs, in Proceedings of the 7th SIAM International Conference on Data Mining, Radisson University Hotel, Minneapolis, MN, 26–28 Apr 2007


370. R. Ng, J. Han, Efficient and effective clustering methods for spatial data mining. In Proceedings of 20-th Conference on VLDB (Santiago, Chile, 1994), pp. 144–155


387. W. Pedrycz, Knowledge-Based Clustering (Wiley, Hoboken, NJ, 2005)
422. V. Schwämmle, O.N. Jensen, A simple and fast method to determine the parameters for fuzzy c-means cluster validation. arxiv:1004.1307v1 [q-bio.qm], Department of Biochemistry and Molecular Biology, University of Southern Denmark, DK-5230 Odense M,Denmark (2010)
428. H. Shinnou, M. Sasaki, Spectral clustering for a large data set by reducing the similarity matrix size, in Proceedings of the 6th International Conference on Language Resources and
References


436. H. Späth, Cluster Analysis Algorithms for Data Reduction and Classification of Objects (Ellis Harwood, Chichester, 1980)


453. P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining (Addison Wesley Longman, 2006)


457. Y. Terada, Strong consistency of reduced k-means clustering (2014)


510. J. Xie, B.K. Szymanski, Towards linear time overlapping community detection in social networks, in *PAKDD. Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Date Mining*, May 29–June 01 2012. https://doi.org/10.1007/978-3-642-30220-6_3


416 References


Index

A
Adamic-Adar index, 266
Affinity propagation, 128
Algebraic connectivity, 212
Algorithm
   affinity propagation, 128
   agglomerative, 29
   anchor grouping, 54
   ARCA, 128
   BM aggregation, 61
   Bregman Hard Clustering, 135
   CLARA, 92
   CLARANS, 92
   CSC, 258
   CSC, 248
   DBSCAN, 51
   EM, 96
   FastGreedy, 276
   FCM, see algorithm, fuzzy c-means, 102
   fine-tuning, 286
   FKCN, 112
   fuzzy c-means, 100
      with Minkowski metric, 112
   Fuzzy c-shells (FCS), 118
   Fuzzy c-varietes (FCV), 116
   Gustafson-Kessel (GK), 114
   HDBSCAN, 54
   k-Bregman Bubble Clustering, 135
   Kernel-based FCM, 120
   KFCM-\(F\), 121
   KFCM-X, 120
   k-means, 68
      k-means||, 78
      balanced, 81
      batch version, 72
      harmonic, 86
      incremental version, 72
   ISODATA version, 80
      kernel-based, 88
      robust sparse, 135
      semi-supervised, 81
      sparse clustering, 134
      spherical, 84
   k-medians, 93
   k-medoids (k-centres), 91
   k-modes, 93, 94
   k-q-flats, 131
   Laplacian leading eigenvectors, 328
   Louvain, 277
   LSC, 250
   Minimum Volume Ellipsoids (MVE), 45
   MM aggregation, 61
   MTV clustering, 232
   Newman, 276
   Nibble, 243
   PageRank-Nibble, 244
   PAM, 92
   partitional, 34
   possibilistic Clustering Method (PCM), 123
   reduced k-means, 132
   relational FCM, 126
   robust single link, 33
   spherical fuzzy c-means (SFCM), 119
   Subspace Clustering, 133
   Tandem Clustering, 132
   U M approximation of data matrix, 42
   Assignment matrix, 69
   Average Commute Time index, 267
   Averaged entropy, 83

B
Bregman divergence, 22

© Springer International Publishing AG 2018
S. T. Wierzchoń and M. A. Kłopotek, Modern Algorithms of Cluster Analysis,
Studies in Big Data 34, https://doi.org/10.1007/978-3-319-69308-8
Bregman information, 23

C
Centroid, 30
Chebyshev distance, 19
Cheeger’s inequality, 224
CLARA, 92
CLARANS, 92
Cluster
compact, 100
Voronoi cluster, 44
well separated, 100
Clusterability, 65
Clustering tree, 33, 150
Cluster tree, 33
Coefficient
participation coefficient, 289
partition, 107
z-score coefficient, 289
Common Neighbour index, 264
Community
auxiliary, 262
in a strong sense, 262
in a weak sense, 262
kernel, 262
Conductance
of a graph, 224
of a set, 224
Connectedness
density connectedness, 52
Consensus partitioning, 59
Constrained k-means algorithm, 154
Contingency table, 173
two way, 174
COP-KMEANS, 154
Correlation coefficient
cophenetic, 31
Pearson (modified), 27
Cosine index, 265
CWS, 100

D
Data representation
embedded, 13
relational, 13
Deflation method, 328
Density forest, 150
Deterministic annealing, 299
Diagonalization, 195
Discrete Schrödinger operator, 188
Dissimilarity relation

definition, 126
Euclidean, 126
Distance
Bhattacharyya, 28
chi-square, 28
k-distance, 52
Distribution
c-separable, 65
stationary, 204
Disturbance of information, 179
Dunbar’s number, 262

E
Eigen-decomposition, 195
Eigenpair, 195, 321
principal, 322
Eigenproblem
definition, 195
generalized, 200
Eigenvalue, 195, 321
non-degenerate, 321
Eigenvector, 195
definition, 321
left, 325
right, 325
Entropy
conditional, see quality indexes
fuzzy partition, 107
joint, 179
of partition, see quality indexes
Error-Correcting Output Codes (ECOC), 237
Expectation Maximisation, see algorithm, EM
Extremal optimisation, 300

F
Factor
damping, 337
Factorisation index (cp-rank), 320
FCM, see algorithm, fuzzy c-means
Feature
extraction, 164
selection, 164
Fiedler
eigenvalue, 212
eigenvector, 212
ordering, 212
Fitness landscape, 191
Flat, 130
Forgy algorithm, see algorithm, k-means,
batch version
Fractional Minkowski distance, 20
Function
    kernel (Mercer kernel), 56
Fuzziness exponent, 101
Fuzzy c-means, see algorithm, fuzzy c-means
Fuzzy ISODATA, 101

F
Genetic algorithm, 300
Global Leicht-Holme-Newman index, 267
Gram-Schmidt orthogonalisation, 319

G
Graph
    empirical, 259
    authority, 275
    hub, 275
    signal, 191
    simple, 185

H
Heuristics
    Lloyd heuristics, 43
generalised, 43
H-MEANS, see algorithm, k-means, batch
version
H-MEANS+, 74
Hub Depressed index, 265
Hub Promoted index, 265

I
Isoperimetric number
    of a graph, 224
    of a set, 224

J
Jaccard coefficient, 176, 279
Jaccard index, 265, 279
Johnson-Lindenstraus
    dimension, 256
    lemma, 256

K
Katz index, 266
KHCM, see algorithm, k-means, kernel-based
KHM, see algorithm, k-means, harmonic
k-means projective clustering algorithm, 136
k-means, see algorithm, k-means
k-prototypes algorithm, 96
Kullback-Leibler divergence, 23

L
Laplace-Beltrami operator (on manifolds), 192
Laplace operator (continuous Laplacian), 192
Laplacian
    combinatorial (un-normalized), 188
discrete, 188
generalized, 188
normalized, 188
Q-Laplacian, 281
Leicht-Holme-Newman index, 265
Local modularity R, 307
Local Path Index, 268
Local Random Walk index, 268
Low-pass filter, 257

M
Manhattan (city block, taxicab) distance, 19
Markov chain
    ergodic, 205
    reversible, 204
Matrix
    adjacency, 184
    adjacency, generalized, 185
column-wise accessible, 321
completely positive, 320
constitutive, 192
diagonally small, 324
diagonally dominant, 190, 242
dispersion, 37
doubly stochastic, 321
Gram, 320
Gram matrix, 58
incidence, 191
incidence (normalized), 194
in-group covariance, 37
inter-group covariance, 37
modularity matrix, 269
non-negative, 319
positive a, 319
positive definite, 320
positive semidefinite, 320
row stochastic, 203
similar (to some other matrix), 323
spectral radius of, 322
spectral representation of, 326
spectrum of, 322
stable, 321
stochastic, 321
Matrix Forest Index, 267
Matrix hermitian, 195
Maximum Entropy SimRank index, 267
Mean field annealing, 299
Mean field theory, 299
Medoid, 35
Method
  average linkage, 30
  centroid (UPGC), 30
  complete linkage, 30
  farthest neighbour, 30
  minimum variance (Ward), 30
  nearest neighbour, 29
  single linkage, 29
  Sinkhorn-Knopp, 321
  weighted centroids (WPGC), 30
  weighted pair-group average, 30
Minkowski distance, 19
Mixing time, 205
Model
  referential (null model), 263
Modularity, 268
Multilayered graph, 309
Mutual information
  definition, 178
  Fred and Jain normalisation, 178
  Strehl and Gosh normalisation, 178
N
Network
  complex, 259
Newman modularity, 268
Nodal set, 239
Node
  degree, 185
  role, 289
Norm
  determined by quadratic form, 18
  Euclidean, 329
  Frobenius, 330
  induced by a vector norm, 330
  spectral, 331
O
One class clustering, 135
Orthogonal initialization, 75
Outlier, 22
P
PAM, 92
Partition cost, 69
PCKmeans, 155
Pearson’s correlation coefficient, 279
p-Laplacian
  combinatorial, 229
  normalized, 229
Power distance, 25
Power method, 328
Preferential Attachment index, 266
Principal component analysis (PCA)
  kernel, 234
  kernel weighted, 235
  linear (classical), 234
Principle of structural equivalence, 279
Problem
  MINcut, 210
Prototype (of a class), 100
Purity index, 107
Q
Quality indexes
  chi-square statistics, 175
  conditional entropy, 178
  Davies-Bouldin index, 171
  external, 170
  F-measure, 176
  Fowlkes-Mallows coefficient, 175
  internal, 170
  Kaufmann and Rousseeuw’s indicator, 171
  MH measure, 177
  Mirkin coefficient, 175
  partition entropy, 177
  purity, 176
  quantisation error, 171
  Rand coefficient, 175
  The Dunn index, 171
  Wallace measure, 177
  Xie-Beni index, 172
  Quantisation error, 171
R
Random walk
  classical, 203
  lazy, 207
  Ruelle-Bowens (maximum entropy), 203
Random Walk with Restart index, 267
Rayleigh quotient, 198, 322
  generalized, 199
Reachability
  density-reachability, 51
  Reclustering, 217
  Reconstruction error, 107
Index

Resolution limit, 271
Resource Allocation index, 266
RKM, 132

S
Salton index, 265
Score variable, 235
Seed, 241
Seeded k-means algorithm, 154
$\sigma$, $\epsilon$-separation, 33
Similarity graph
  $K$-NN, 186
  $\tau$-NN, 186
  $r$-neighbourhood, 186
SimRank index, 267
Simulated annealing, 298
Singular value decomposition truncated, 202
Sorensen index, 265
Sparse clustering $k$-means, 134
Sparse coding, 250
Spectral
  absolute gap, 206
  coordinates, 218
  embedding, 218
  gap, 206
  partitional clustering, 218
repeated bisection clustering, 217
Superposed Random Walk index, 268

T
Tabu search, 300
Tessellation
  Dirichlet, 44
  Voronoi, 44
Theorem
  Courant-Fischer, 200, 322
  discrete nodal domain, 239
  Eckart-Young-Mirsky, 202
  Fan, 201
  Perron-Frobenius, 323
  Transductive tree, 157

V
Value
  singular, 202
Vector
  PageRank (global), 337
  PageRank (personalised), 337
  singular, 202
  stochastic, 204
Volume, 185