Clustering through High Dimensional Data Scaling: Applications and Implementations

Fionn Murtagh and Pedro Contreras

Abstract To analyse very high dimensional data, or large data volumes, we study random projection. Since hierarchically clustered data can be scaled in one dimension, seriation or unidimensional scaling is our primary objective. Having determined a unidimensional scaling of the multidimensional data cloud, this is followed by clustering. In many past case studies we carried out such clustering, using the Baire, or longest common prefix, metric and, simultaneously, ultrametric. In this paper, we examine properties of the seriation, and of the induction of the clustering on the data summarization, through seriation. Simulations are described as well as a small, illustrative example using Fisher's iris data.

1 Introduction

The Baire, or longest common prefix, metric, that is also an ultrametric, was used by us in astronomy, on spectrometric and photometric redshift values for half a million objects. For high dimensional spaces, random projection can be a

Fionn Murtagh
University of Derby and Goldsmiths University of London,
✉ fmurtagh@acm.org

Pedro Contreras
Thinking Safe Ltd.,
✉ pedro.contreras@acm.org

ARCHIVES OF DATA SCIENCE (ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 2, No. 1, 2017

DOI 10.5445/KSP/1000058749/08
ISSN 2363-9881

most effective, and efficient way, to prepare data for clustering that is based on that same principle. Examples of this included clusterwise or nearest neighbour regression on 34,352 proposal documents, crossed by 10,317 other such documents, based on document similarity. A further case was using chemistry data crossing 1,219,553 chemical structures coded through 1052 presence/absence values. In this paper, we aim to exemplify and illustrate the principles of clustering, based on random projection. While small data sets are used, these case studies are intended as being both illustrative of applications and of implementation.

Let us consider scalar numbers, defined by a sequence of digits. For ease of description, and of reproducibility, let each number be $\geq 0, < 1$. The Baire or longest common prefix metric, (Contreras and Murtagh, 2012), allows direct reading of the clusters. Just consider the first partition, using the most significant digit, so that for real values, and not binary nor any other number system, we have 10 clusters labelled 0 to 9. From these clusters, we next look at the next significant digit, and again can read off, for all 10 clusters at the first level, at the second level, 100 clusters. In this way, we can build up a set of partitions, and define a hierarchical clustering. The Baire, or longest common prefix metric, is, by definition, also an ultrametric.

While what has been described is using just scalar numbers, we can benefit very much from properties of high dimensional spaces. Chief among these is the particular importance, and summarizing ability, of random projections. Unlike the work of others that is focused on dimensionality reduction (see discussion in Murtagh and Contreras (2016, 2015)), instead our main interest, with demonstrated success, is how we can determine and make use of a single set of projections, that is, on one axis. Our work has a different focus compared to application of random projection for low dimensional subspace mapping, used in, for example, Bingham and Mannila (2001).

This article sets out foundations of this methodology. The mean set of projections collected from a large number of random axes is examined. We examine relationships between such a mean axis and the dominant eigenvector. The clustering properties of the projections on this mean axis are our primary interest. We seek to describe this work in terms of seriation, or unidimensional scaling. We demonstrate considerable success in the case studies, many of them simulations, that are employed.

In Sect. 2, we study data concentration, through employment of random axes. Sections 3 and 4 consider the inducing of a clustering, in particular a hierarchical clustering. Section 5 relates to data input and hence the context of the implementation. Section 6 considers a central aspect of implementation. Section 7 considers clustering, with an illustrative case study. All work was carried out in R (version 3.2.0).

2 The Mean Random Projection Approximates the Marginal Sum

Our given data is denoted x_{IJ}, that is, $\{x_{ij} | i \in I, j \in J\}, n = |I|, m = |J|$. As a random projection, we use uniformly distributed values in [0,1], denoted u^1_J, that is $\{u^1_j | j \in J\}$. (The superscript, Fig. 1, is used since this is our first, of many, random projections.) A random projection is the matrix-vector product $r^1_I = x_{IJ} u^1_J$. Another random axis is generated, and the projection on it is determined: $r^2_I = x_{IJ} u^2_J$. A third random axis is generated, and the projection on it is determined. This continues in this particular experiment for 100 random projections. Thus we have random projections $r^1_I, r^2_I, r^3_I, \dots, r^{100}_I$.

Our interest is in the mean of the random projection vectors. The motivation for this is that if the random projections are well correlated, as we have found to be the case for high dimensional data clouds, then the mean is a suitable consensus. So we look at the succession of mean projection vectors: For $K = 1, 2, \dots, 100$: $\frac{1}{K} \sum_{k=1}^{K} r^k_I$. The random axis used on each occasion, $k = 1, 2, \dots, K$, was newly generated.

The marginal sums, $x_I = \{\sum_{j=1}^{m} x_{ij} | i \in I\}$ are a constant times the mean vector. The cloud is the point set, x_{iJ} for $i \in I$. We can expect the mean random projection to approximate very well the cloud mean. That is to say, we have (1) the cloud's mean vector, and (2) the randomized clouds that consist of randomly generated points that are a linear combination of the cloud's given points.

The uniformly distributed random axes result in the projection values being non-negative, when the initial, given data, are non-negatively valued. In this particular study, for expository reasons, we used a 25×12 data array, of uniformly distributed values in [0,1]. Then a second study uses an initial 25×12 data array of Gaussian distributed values, of mean 0 and standard deviation 1. This second study uses both positive and negative valued input data and projections.

Our input data values are not constrained to be non-negative in value. We note this because, in Correspondence Analysis, we are dealing with marginal distributions that are mass distributions of row (observations, I) cloud and column (attributes or properties, J) cloud. Thus in Correspondence Analysis, we must consider non-negative valued, given data.

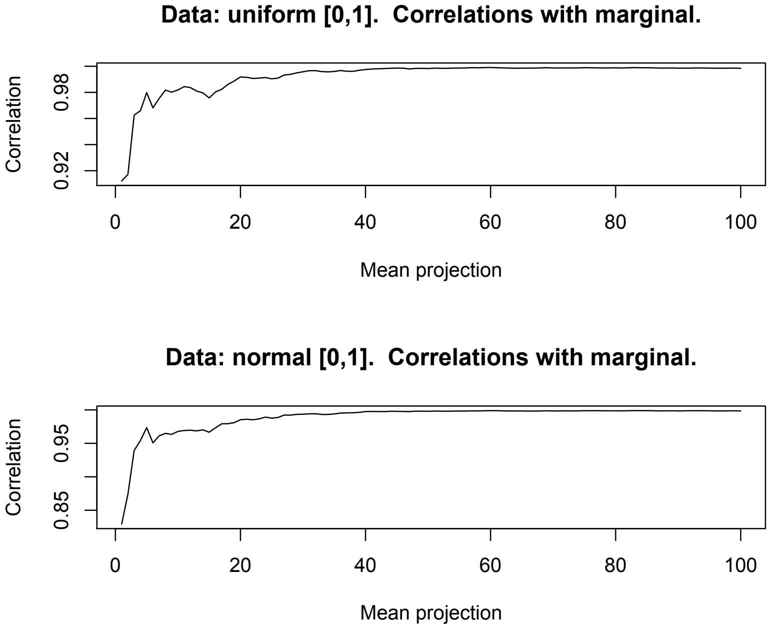


Fig. 1 Correlations between mean random projection, and the marginal sums. The two input data tables used were of dimensions 25 × 12. Maximum and minimum values of the input data tables are, respectively, $0.9991, 0.0012$ and $2.6479, -3.0277$.

Figure 1 demonstrates clearly how the mean random projection (with uniformly distributed random axes) very well approximates the marginal sums, for a sufficient number of random axes. The latter, the marginal sums vector, is proportional to the mean of the point cloud, $I \subset \mathbb{R}^m$. (Here, the cloud's coordinates are real-valued, in the space of dimensionality $m = |J|$.) To illustrate the degree of approximation, that is displayed in Fig. 1, the final three correlation values are as follows. That is, these are correlations between the marginal row sums and the means of 98, 99 and 100 random projections. We have for the two data sets used, respectively, 0.9985497, 0.9985972, 0.9984356, and 0.9985829, 0.9986338, 0.9984509.

This study will serve as important background for our further work. To summarize the outcome: *the mean of 40 or more projections of the point cloud on uniformly distributed axes, approximates very well the marginal sums of the point set.* The 40 or more projections finding is observed in Fig. 1 and in

Table 1 Means of 98 random projections, of 99 random projections, and of 100 random projections. The correlation between the mean random projection, and the row sum, is shown here. Parametric and non-parametric correlations are used. In the normalized cases, both the mean of the projections, and the row sum, are normalized, to have maximum value = 1, and to have L_2 norm = 1.

	k = 98	k = 99	k = 100
Pearson	0.9985497	0.9985972	0.9984356
Spearman	0.9961538	0.9961538	0.9953846
Kendall	0.9733333	0.9733333	0.9666667
Pearson/max normalized	0.9985769	0.9986095	0.9984502
Pearson/L_2 norm normalized	0.9985652	0.9985888	0.9984162

many other experiments. The projections on uniformly distributed axes are of the point cloud, that is not centred, or normalized in any way. The uniformly distributed axes means that these axes are defined, as uniformly distributed coordinates, in the given, real, m-dimensional space, $\mathbb{R}^m$. The marginal sums of the point set are directly proportional to the means of the point coordinates. I.e., each point's marginal sum is $x_i = \sum_{j=1}^m x_{ij} = m\frac{1}{m}\sum_{j=1}^m x_{ij}$. Hence we have m times the mean of x_i's coordinate values.

Apart from the Pearson correlation, we looked for any important differences when using non-parametric correlation, e.g. Spearman and Kendall rank correlations between mean random projection, and the marginal sums. We find the Pearson correlation to be the highest in value, followed by the Spearman correlation, that is followed in turn by the Kendall correlation. Nonetheless, the approximation is very good in all cases. For Pearson and Spearman, it is observed to be, with sufficient random axes, 0.99 in all cases. For the Kendall correlation, it is around 0.97.

There are very limited differences when random projections are normalized to unit norm, as we will now show. Assessment is carried out, for normalizations, using the L_∞ or max norm, or Chebyshev norm; and using the L_2 norm. Table 1 shows very little difference between the approaches used.

For the normalized random projection values, the scale will differ, depending on the normalization used. Nonetheless, we do observe a very similar outcome. See Fig. 2.

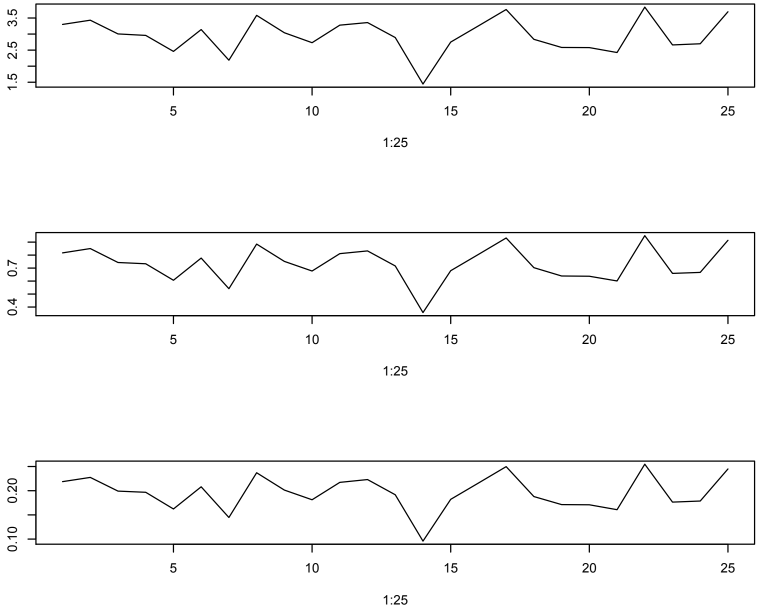


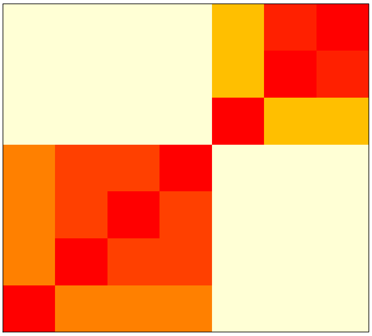
Fig. 2 For the (top to bottom) Pearson correlation, projections not normalized; projections normalized by maximum value (i.e., L_∞ or Chebyshev metric); projections normalized to unit L_2 norm. Correlations between mean random projection, and the marginal sums. Input data: uniformly distributed, of dimensions 25 × 12.

3 Clustering Properties of the Row Sums, and Mean Random Projection

In this section we use a small dataset, which is very widely used for test purposes, in order to explain clearly, and to illustrate, the foundations for our methodology. We want to consider the clustering of the data, and the clustering properties of the marginal sums, which, as has been shown, is well approximated by the mean random projection.

Using the hierarchical clustering in Fig. 4, from which the ultrametric distances are shown in Table 2, then in Fig. 3 there is a visual display of the data in that table. This visualization allows structure in the distance array to be seen.

Just as the terminal nodes of the hierarchy in Fig. 4 require a particular ordering for this hierarchical representation, so also an appropriate ordering,



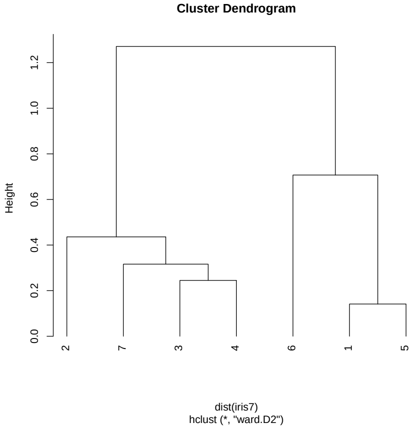


Fig. 3 A visualization of the ultrametric matrix of Table 2, where bright colour = high value, and dark colour = low value.

Fig. 4 Hierarchical clustering of 7 iris flowers – the first 7. No data normalization was used. The agglomerative clustering criterion was the minimum variance or Ward criterion.

or permutation, of rows and columns in Fig. 3 is required. Since this ultrametric distance matrix, derived from the hierarchy, is symmetrical, it follows that the order, i.e. the permutation, of rows and of columns, must be the same. In Murtagh (2009), there is discussion of row and column permuting in the original data table, to achieve such a structure.

Table 2 Ultrametric matrix derived from the dendrogram in Fig. 4.

	2	7	3	4	6	1	5
2	0.000	0.436	0.436	0.436	1.271	1.271	1.271
7	0.436	0.000	0.316	0.316	1.271	1.271	1.271
3	0.436	0.316	0.000	0.245	1.271	1.271	1.271
4	0.436	0.316	0.245	0.000	1.271	1.271	1.271
6	1.271	1.271	1.271	1.271	0.000	0.707	0.707
1	1.271	1.271	1.271	1.271	0.707	0.000	0.141
5	1.271	1.271	1.271	1.271	0.707	0.141	0.000

In Lerman (1981), there is specification of the block diagonal structure that respects the ultrametric inequality. This specifies how, as shown in Fig. 3, values away from the main diagonal increase in value, while respecting the block structure. The ultrametric distance matrix is a symmetric, positive definite matrix, with zero-valued main diagonal entries.

Block structure, as is defined in the above specification, and displayed as in the figure, is a way to express the clustering of the data. That is, by achieving such a block structure, we are hierarchically clustering our data. Such block clustering is used in 2-mode, viz. rows and columns, data clustering, also named biclustering (Vichi, 2015).

The important aspect of this, for our purposes here, is that there is an ordering of terminal nodes in the hierarchy, shown in Fig. 4, and that ordering, a permutation of the object set that we are clustering, is also the ordering of rows and of columns in the display in Fig. 3. That ordering, or permutation, is our major interest. We seek to have our aggregated, or mean or consensus, random projection, to represent well this ordering, or permutation.

In Sect. 7 we will return again to this ordering or permutation, especially relating this to the work of Critchley and Heiser (1988), that a hierarchical clustering, as a hierarchy or tree structure, "can be perfectly scaled in one dimension".

4 Power Iteration Clustering

For Principal Components Analysis, Correspondence Analysis or any other eigen-decomposition, the processing can be carried out iteratively as follows. For a matrix, A, we seek to solve: $Au = \lambda u$. The solution is the first (largest) eigenvalue, λ, associated with the eigenvector, u. A random, non-0, initial vector is chosen: t_0. Then define $t_1, t_2, \ldots$ as follows:

$$\begin{aligned}
At_0 &= x_0, & t_1 &= x_0 / \sqrt{x_0' x_0} \\
At_1 &= x_1, & t_2 &= x_1 / \sqrt{x_1' x_1} \\
At_2 &= x_2, & t_3 &= x_2 / \sqrt{x_2' x_2} \\
&\ldots
\end{aligned}$$

The normalization to unit L_2 norm can be viewed as preventing the vector t_k from getting too large (Lin and Cohen, 2010). However Lin and Cohen (2010) alternatively use the L_1 norm. A justification for the L_2 norm is Principal Components Analysis or Correspondence Analysis where the factor space, defined by the eigenvalue and eigenvectors, is endowed with the Euclidean, L_2 metric.

We halt the iterations where there is convergence: $|t_n - t_{n+1}| \leq \varepsilon$. At convergence, we have the approximation of: $t_n = t_{n+1}$. Therefore, $At_n = x_n$. Since $t_{n+1} = x_n / \sqrt{x'_n x_n}$ we may substitute terms in $At_n = x_n$ to give: $At_n = \sqrt{x'_n x_n}\, t_{n+1}$. Since $t_n = t_{n+1}$, that allows us to conclude that t_n is the eigenvector u, and the associated eigenvalue λ is $\sqrt{x'_n x_n}$.

We can partial out the first eigenvector and eigenvalue in order to proceed to the next eigenvector and eigenvalue. (What we do is to redo the analysis on the matrix, $A_{(2)} = A - \lambda u_1 u'_1$.) We will not do this here since our interest is in the first eigenvector. This iterative solution is described in Murtagh and Heck (1987) (Sect. 2.2.6).

In Lin and Cohen (2010), this approach is used with a data table that is normalized through division by the row sum. This furnishes the set of row (object) profiles. For a data table, $x_{IJ} = \{x_{ij} | i \in I, j \in J\}$, the profile is $\{x_{ij}/x_i | i \in I, j \in J\}$, where $x_i = \sum_{j \in J} x_{ij}$. In Correspondence Analysis, frequencies are used: $f_{ij} = x_{ij} / \sum_{i \in I, j \in J} x_{ij}$, where $x_{ij} \geq 0$. Similarly we have $f_i = x_i / \sum_{i \in I} f_i$. In tensor notation (Einstein tensor notation, used in Benzécri (1976)), the row profile is $f^i_J = \{f_{ij}/f_i | j \in J\}$. Now, working with row profiles means that the row sums, $\sum_{j \in J} f_{ij}/f_i = 1$. The row sums are constant. In seeking to solve $f^I_J u_J = \lambda u_J$ we have that $u_J = 1_J$ where the latter term is a $|J|$-length vector of 1s. The expression we want to solve then requires $\lambda = 1$. This is the so-called trivial first eigenvector and eigenvalue in Correspondence Analysis. It is due to the centering of the cloud of row points and column points. Since this is trivial in value, Lin and Cohen (2010) present the case for iterating towards the solution but stopping at a local optimum with regard to convergence. Rather than the first eigenvector, an "intermediate vector" is obtained. The iterative scheme "converges locally within a cluster".

We come now to clustering. Yan et al (2013) note how power iteration clustering can be based on the dominant eigenvalue/eigenvector described above. That points to the relevance of the data array used. It has been noted above how row profiles were used in Lin and Cohen (2010). In Yan et al (2013), reference is made in particular to spectral clustering, described as "a family of methods based on the eigendecomposition of affinity, dissimilarity or kernel matrices". Such spectral clustering is core to Correspondence Analysis, which induces a hierarchical clustering, for example, from the factor space endowed with the Euclidean metric. K-means clustering (partitioning) is used in Lin and Cohen (2010).

Further application of Lin and Cohen (2010) is carried out in Lohk et al (2013). By means of the dominant eigenvector found using the power iteration eigenreduction approach, Lohk et al (2013) use both row and column sets. This allows row and column (of the data array) permutation to yield a block clustered view of the data array. A wide-ranging review of row and column permutation for clustering is in Liiv (2010).

In Lohk et al (2013), comprehensive use is presented of row and column permuting, using power iteration clustering (Lin and Cohen, 2010) applied successively to the rows and to the columns. The input data array used in Lohk et al (2013) is the sums of squares and cross products matrix. I.e. for initial (non-negative dependency values, described in graph terms) matrix, X, this input for row and column permutation is $X'X$.

5 Input Data for Eigen-Reduction

For Correspondence Analysis, we have the following. In Benzécri (1982), there is discussion of analysis of data sets that are unbounded in number of rows (an infinite set), with possibly 1000 columns. For given data k, and factor G on the set J, the eigenvalue and eigenvector decomposition is the solution of this equation:

$$\sum_{j' \in J} \left(\left(\sum_{i \in I} \frac{k_{ij'}}{k_i} \frac{k_{ij}}{k_i} \right) G_\alpha(j') \right) = \lambda_\alpha G_\alpha(j) \tag{1}$$

That, therefore, relates to the column profiles. Following determination of the factors, G, on J, the transition formula relationship allows the determination of the factors on I (denoted F).

The full eigen-reduction, determining the full set of factors, is described in Benzécri (1982, 1997). In the latter, there is discussion of the extensive use by Ludovic Lebart of stochastic approximation of factors and associated contributions to inertia, defined from the eigenvectors and associated eigenvalues. Chapter VI of Lebart et al (1984) provides a comprehensive description, with this eigen-reduction termed a “direct reading” implementation. In Benzécri (1982), since the set of eigenvectors and associated eigenvalues is the objective, a matrix of trial vectors is initialized and converged, in the sequence of iterations, to the desired outcome. (See Eq. (6), page 391, of Benzécri (1982).) This is as in the approach carried out in Clint and Jennings (1970).

In Benzécri (1992), it is to be noted that there is a short review of an alternative approach to efficient processing by finding subtables of the given input data table. Cf. Sect. 3 above. For a subset J_s of J, analyses are carried out on these $I \times J_s$ subtables. This work was primarily due to Brigitte Escofier. For one $I \times J_s$ subtable, there can be added (juxtaposed) a "rest" column with the accumulation of all columns in the set $J - J_s$. In this case, one has a Euclidean representation, i.e. the factor space, of the cloud $N(J)$ that is the same as the analysis carried out on the $I \times J$ table, but relative to axes that are adjusted to the sub-cloud, $N(J_s)$. While the full analysis can be re-assembled from the analyses of the subtables, it may be the case that the subtable analyses are more interesting in their own way. The global analysis, it is stated, may be perturbed by such singular modalities of variables, in particular missing values. While, it is stated, the views of $N(J)$, adjusted to well-chosen sub-clouds $N(J_s)$ can better show the global structure.

For Principal Components Analysis (PCA), depending on applied, the matrix to be diagonalized is one of: the sums of squares and cross-products matrix, the variance-covariance matrix, or the correlation matrix. See, e.g., Murtagh and Heck (1987) for a discussion of these inputs for PCA.

6 Implementation: Equivalence of Iterative Approximation and Batch Calculation

It has been noted in Sect. 4 how a limited number of iterations may be used, rather than convergence. Let us also look now at where and how a fixed number of iterations may be beneficial, as an alternative to convergence.

We use the following illustrative and motivational example from Benzécri (1982). This example relates to estimating the mean of the, potentially unbounded, set of values $x_1, x_2, x_3, \ldots, x_n, \ldots$. If the number of such values is known, denoting it N, then that leads to just determining the estimated mean on all the data. If there were weights involved, then an unbounded sequence becomes more problematic. Now, it can be shown that two successive values in the estimation have this relationship: $\mu_{n+1} = \mu_n + ((x_{n+1} - \mu_n)/(n+1))$. It is seen that the $(n+1)$th value can be considered as correcting the estimate at that iteration. Also we see that if $\mu_{n+1} = \mu_n$, then that correction to the estimate at that iteration would be equal to 0. (It is acknowledged in Benzécri (1982) that successive updates, carried out in this way, may lead to accumulation of rounding errors. On the other hand it is considered that any exceptional or outlying value of x would be very clearly indicated.)

In practice, the iterative estimation can be useful and relevant as an approach to determining the mean, especially of an unbounded sequence. An assumption, to be considered in each case, is the underlying distribution of the x terms.

We conclude the following from this small, illustrative case study. Given an underlying distribution, we can either (i) iterate until convergence, or (ii) assume a fixed value of N and carry out the computation for the sequence of values (or vectors) that are from 1 to N.

7 Inducing a Hierarchical Clustering from Seriation through Unidimensional Representation of Our Observations

The following is based on Critchley and Heiser (1988), which establishes the foundations for inducing a hierarchical clustering from a newly represented, or newly encoded, mapping of our data. This very important result allows us to seek a seriation in order to hierarchically cluster our data in a very computationally efficient manner.

Consider a dendrogram and the terminal nodes in a sequence, $\pi(I)$, a permutation of the object set, I. For $i \in I, i = 1, 2, 3, \ldots, n = |I|$, consider π_i, i.e. $\pi_1, \pi_2, \ldots, \pi_n$. Now define the $(n-1)$-vector, t, with general element $t_j = x_{j+1} - x_j$. Such a one-dimensional ordering, t, is compatible with the given dendrogram ordering if $j \leq k$ implies $x_j \leq x_k$. Then we define a matrix of interpoint distances in the unidimensional ordering as follows: $d_{jj} = 0; j < k \longrightarrow d_{kj} = d_{jk} = \sum_{l=j}^{k-1} t_l$.

As noted in Benzécri (1997), although quite likely to be fully justified in practice, any iterative refinement algorithm is unable to deliver an optimal solution. The non-uniqueness of the seriation or unidimensional scaling, that can be the starting point for inducing a hierarchical clustering, is a limitation in practice, since many alternatives may (or may not) be relevant for the hierarchy to be induced.

Using our approach on the Fisher iris data, (Fisher, 1936), 150 flowers crossed by petal and sepal width and breadth, provides the following outcome. We determine row sums, of the initial 150×4 data matrix, and the mean random projection of projections on 100 uniformly generated axes. From our previous results, we know that these are very highly correlated. We construct hierarchical clusterings on (i) the original 150×4 data matrix, (ii) the mean random projection, and (iii) the row sums. The cophenetic correlation coefficient is determined. (This uses ultrametric distances derived from the hierarchical tree, or dendrogram.) We find the cophenetic correlation of the hierarchies constructed on the row sums, and on the mean random projection, to be equal to 1 (as anticipated). Then between the hierarchy constructed on the 150×4 data matrix, and the mean random projection, the cophenetic correlation coefficient is 0.8798. For the given data and the row sums, it is 0.9885. The hierarchical clustering used was the average method; and other methods, including single link, provided very similar results. The distance used, as input to the hierarchical agglomerative clustering, was the square root of the squared Euclidean distance. Other alternatives were looked at, from the point of view of the distance used, and from the point of view of the agglomerative hierarchical clustering criterion.

All in all, these results are both supportive of our overall perspectives, and they are consistent with our overall perspectives, as discussed in this work.

We also looked at uniformly distributed, on [0,1], data of dimensions 2500×12. The correlation between row sums and mean of 100 random projections was 0.99. However, for the correlation between the hierarchical clustering on the original data, and the mean random projection, this correlation was 0.58. The correlation with the row sums was 0.578. The performance on this randomly generated data is seen to be not as good as that on the real valued, Fisher data. For data which is not strongly clustered, quantization is relevant. In the k-means clustering (partitioning) context, see e.g. Lloyd (1982). Descriptively expressed, in quantization, in addition to cluster compactness, approximating identical cluster sizes is an objective.

We conclude this practical case study with the following remarks. (i) For real data, we found a very good result. (ii) The lack of uniqueness of the seriation (or unidimensional representation) means that various possibilities may, or may not, be most appropriate. But (iii) we determined a good outcome, that respected (correlation between induced hierarchical clusterings of around 0.88) the clustering properties in the data.

8 Conclusions

We have set out the following objectives, algorithms, and implementations. In all cases important properties and characteristics have been discussed. Our case studies were chosen primarily for expository reasons. We conclude with the following practical outcomes.

1. The use of iterative approximation, to convergence; to optimally summarize the data cloud by means of the dominant eigenvector.
2. Data cloud centroid, for its role in summarizing the cloud, and relationships with the dominant eigenvector.
3. How such summarization induces seriation of our data cloud.
4. General clustering properties of such a seriation. This is motivated by the awareness that an ultrametric topological embedding of our data cloud in a space of arbitrary dimensionality (alternatively expressed: a hierarchical structuring of our data) can be perfectly scaled in one dimension.
5. Noting the importance of the following: no approximate scheme can be guaranteed to provide an optimal outcome (emphasized in Benzécri (1997)); selection of data table normalization plays an important role; convergence, consistency and stability properties of implementations.

Current work, pursuing this work, includes the following. Murtagh (2016b) seeks to address challenges in computation and in storage requirements. In Murtagh (2016a), it is sought to exploit the dual space relationship of the point cloud in observation (or row) space, and the point cloud in attribute (or column) space, in order to cluster massive numbers of row points in moderate to small dimensions.

References

Benzécri JP (1976) L'Analyse des Données. II. Correspondances, 2nd edn. Dunod

Benzécri JP (1982) L'approximation stochastique en analyse des correspondances. Les Cahiers de l'Analyse des Données 7(4):387–394, URL http://eudml.org/doc/88060

Benzécri JP (1992) Correspondence Analysis Handbook. Marcel Dekker

Benzécri JP (1997) Approximation stochastique, réseaux de neurones et analyse des données. Les Cahiers de l'Analyse des Données 22(2):211–220, URL http://eudml.org/doc/88541

Bingham E, Mannila H (2001) Random projection in dimensionality reduction: Application to image and text data. In: Proc. Seventh International Conference on Knowledge Discovery and Data Mining, ACM, pp 245–250

Clint M, Jennings A (1970) The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration. The Computer Journal 13(1):76–80, DOI 10.1093/comjnl/13.1.76

Contreras P, Murtagh F (2012) Fast, linear time hierarchical clustering using the Baire metric. Journal of Classification 29(2):118–143, DOI 10.1007/s00357-012-9106-3

Critchley F, Heiser W (1988) Hierarchical trees can be perfectly scaled in one dimension. Journal of Classification 5(1):5–20, DOI 10.1007/BF01901668

Fisher R (1936) The use of multiple measurements in taxonomic problems. The Annals of Eugenics 7(2):179–188, DOI 10.1111/j.1469-1809.1936.tb02137.x

Lebart L, Morineau A, Warwick K (1984) Multivariate Descriptive Statistical Analysis. Wiley, Chapter 6, Direct Reading Algorithms

Lerman I (1981) Classification et Analyse Ordinale des Données. Dunod

Liiv I (2010) Seriation and matrix reordering methods: An historical overview. Statistical Analysis and Data Mining: The ASA Data Science Journal 3(2):70–91, DOI 10.1002/sam.10071

Lin F, Cohen W (2010) Power iteration clustering. In: Proc. 27th International Conference on Machine Learning, Haifa, Israel

Lloyd S (1982) Least-squares quantization in PCM. IEEE Transactions on Information Theory 28(2):129–137, DOI 10.1109/TIT.1982.1056489

Lohk A, Tilk O, Võhandu L (2013) How to create order in large closed subsets of WordNet-type dictionaries. Eesti Rakenduslingvistika Ühingu aastaraamat / Estonian Papers in Applied Linguistics 9, DOI 10.5128/ERYa.1736-2563

Murtagh F (2009) Symmetry in data mining and analysis: a unifying view based on hierarchy. Proceedings of Steklov Institute of Mathematics 265(1):177–198, DOI 10.1134/S0081543809020175

Murtagh F (2016a) Massive data clustering in moderate dimensions from the dual spaces of observation and attribute data clouds. Tech. rep., in preparation

Murtagh F (2016b) Sparse p-adic data coding for computationally efficient and effective big data analytics. P-Adic Numbers, Ultrametric Analysis, and Applications 8(3):236–247, DOI 10.1134/S2070046616030055

Murtagh F, Contreras P (2015) Random projection towards the Baire metric for high dimensional clustering. Proceedings SLDS 2015, Symposium on

Learning and Data Sciences, Lecture Notes in Artificial Intelligence Volume 9047 pp 424–431, DOI 10.1007/978-3-319-17091-6_37

Murtagh F, Contreras P (2016) Linear Storage and Potentially Constant Time Hierarchical Clustering Using the Baire Metric and Random Spanning Paths, Springer International Publishing, Cham, pp 43–52. DOI 10.1007/978-3-319-25226-1_4

Murtagh F, Heck A (1987) Multivariate Data Analysis. Reidel (Kluwer)

Vichi M (2015) Two-mode partitioning and multipartitioning. In: Hennig C, Meila M, Murtagh F, Rocci R (eds) Handbook of Cluster Analysis, Chapman and Hall/CRC, pp 519–544

Yan W, Brahmakshatriya U, Xue Y, Gilder M, Wise B (2013) p-PIC: parallel power iteration clustering for big data. Journal of Parallel and Distributed Computing 73(3):352–359, DOI http://dx.doi.org/10.1016/j.jpdc.2012.06.009