Distance-based dimensionality reduction for big data literature review

Adrià Casanova Lloveras

February 12, 2025

Contents

1 Rdimtools: An R Package for Dimension Reduction and Intrinsic Dimension Estimation [4]

1.1 Abstract

Original: Discovering patterns of the complex high-dimensional data is one of the fundamental pillars of modern data science. Dimension reduction and intrinsic dimension estimation are two thematic programs that facilitate geometric characterization of the data. We present `Rdimtools`, an R package that supports 143 dimension reduction and manifold learning methods and 17 dimension estimation algorithms whose unprecedented extent makes multifaceted scrutiny of the data in one place easier. `Rdimtools` is distributed under the MIT license and is accessible from CRAN, GitHub, and its package website, all of which deliver instruction for installation, self-contained examples, and API documentation.

Apple Intelligence summary: `Rdimtools`, an R package, supports 160 dimension reduction and manifold learning methods, making data analysis easier. It is available on CRAN, GitHub, and its package website.

1.2 Key Points

- **R package**: that supports 143 dimension reduction and manifold learning methods and 17 dimension estimation algorithms.
- **Other libraries**: `drtoolbox` in MATLAB, `scikit-learn` in Python, a C++ template library `tapkee` with a known basis of popularity. In R, packages `dimRed`, `dyndimred`, `intrinsicDimension`.
- **Implementation**: mixture of R and C++ that are integrated by `Rcpp`. For numerical operations, `RcppArmadillo` is heavily used to take advantage of `Armadillo` C++ linear algebra library.
- **3 function families**: `do.{algorithm}`, `est.{algorithm}` and `aux.{algorithm}` for DR, IDE, and auxiliary functions.
- Downloaded 1013 times per month on average from CRAN.
- **Future plan**: support for out-of-memory execution in response to the increased needs for big data analysis.

1.3 Example R Code

```
# Documentation example:
# do.idmap (Interactive Document Map)

library(Rdimtools)

## load iris data
data(iris)
set.seed(100)
```

```
subid = sample(1:150,50)
X = as.matrix(iris[subid,1:4])
lab = as.factor(iris[subid,5])
## let's compare with other methods
out1 <- do.pca(X, ndim=2)
out2 <- do.lda(X, ndim=2, label=lab)
out3 <- do.idmap(X, ndim=2, engine="NNP")
## visualize
opar <- par(no.readonly=TRUE)
par(mfrow=c(1,3))
plot(out1$Y, pch=19, col=lab, main="PCA")
plot(out2$Y, pch=19, col=lab, main="LDA")
plot(out3$Y, pch=19, col=lab, main="IDMAP")
par(opar)
```

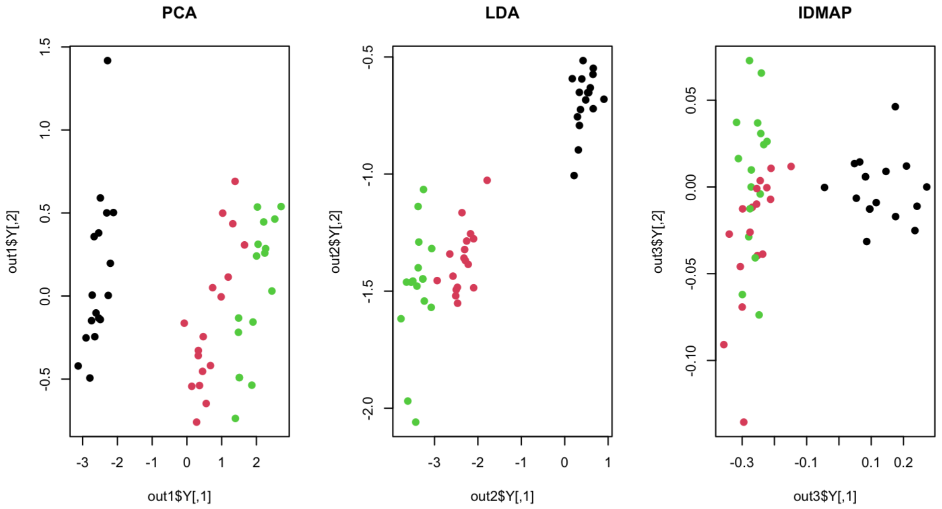

Figure 1: Rdimtools test output.

2 Global versus local methods in nonlinear dimensionality reduction [3]

2.1 Abstract

Original: Recently proposed algorithms for nonlinear dimensionality reduction fall broadly into two categories which have different advantages and disadvantages: global (Isomap), and local (Locally Linear Embedding, Laplacian Eigenmaps). We present two variants of Isomap which combine the advantages of the global approach with what have previously been exclusive advantages of local methods: computational sparsity and the ability to invert conformal maps.

Apple Intelligence summary: Two new Isomap variants are presented, combining global advantages with local computational sparsity and conformal map inversion.

2.2 Key Points

- **Introduction of LMDS** by applying it to Isomap (L-Isomap).
- ***Landmark points*** ($n << N$) reduce the complexity of computing:
  - the distances matrix with Dijkstra's algorithm with Fibonacci heaps (k = neighborhood size) from $O(kN^2 \log N)$ to $O(knN \log N)$.
  - MDS from $O(n^3)$ to $O(n^2N)$.
- If x is a landmark point, then the embedding given by LMDS is consistent with the original MDS embedding.
- If the distance matrix $D_{n,N}$ can be represented exactly by a Euclidean configuration in $\mathbb{R}^l$, and if the landmarks are chosen so that their affine span in that configuration is l-dimensional (i.e. general position), then LMDS will recover the configuration exactly, up to rotation and translation.

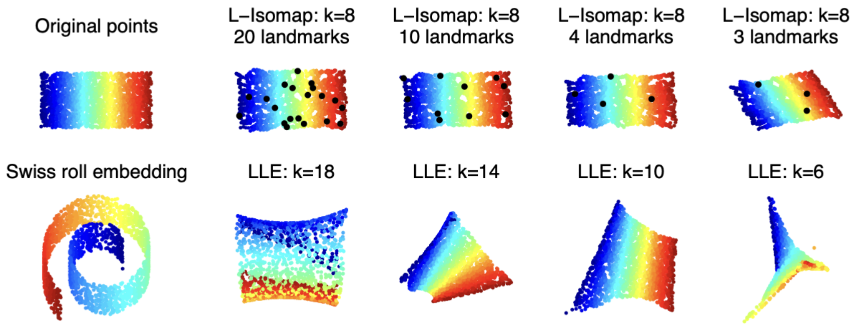

Figure 2: L-Isomap is stable over a wide range of values for the sparseness parameter (the number of landmarks). Results from LLE are shown for comparision[3].

3 dimRed and coRanking - Unifying Dimensionality Reduction in R [2]

3.1 Abstract

Original: "Dimensionality reduction" (DR) is a widely used approach to find low dimensional and interpretable representations of data that are natively embedded in high-dimensional spaces. DR can be realized by a plethora of methods with different properties, objectives, and, hence, (dis)advantages. The resulting low-dimensional data embeddings are often difficult to compare with objective criteria. Here, we introduce the `dimRed` and `coRanking` packages for the R language. These open source software packages enable users to easily access multiple classical and advanced DR methods using a common interface. The packages also provide quality indicators for the embeddings and easy visualization of high dimensional data. The `coRanking` package provides the functionality for assessing DR methods in the co-ranking matrix framework. In tandem, these packages allow for uncovering complex structures high dimensional data. Currently 15 DR methods are available in the package, some of which were not previously available to R users. Here, we outline the `dimRed` and `coRanking` packages and make the implemented methods understandable to the interested reader.

Apple Intelligence summary: Dimensionality reduction (DR) methods create low dimensional data representations, but comparison is challenging. The `dimRed` (figure 3) and `coRanking` R packages are introduced to address this.

3.2 Key Points

- The difficulty in applying DR is that each DR method is designed to maintain certain aspects of the original data and therefore may be appropriate for one task and inappropriate for another. Most methods also have parameters to tune and follow different assumptions.
- **Software packages for other languages**:
  - Python: `scikit-learn`, which contains a module for DR.
  - Julia: `ManifoldLearning.jl` for nonlinear and `MultivariateStats.jl` for linear DR.
  - Matlab: several toolboxes.
  - C++: `Shogun` toolbox, which offers bindings for many high level languages (including R).
- At the time (2018), no comprehensive package for R.
- None of the former provides means to consistently compare the quality of different methods.
- MDS can be seen as kPCA with kernel $x^T y$, since a distance matrix can be transformed to a matrix of inner products.
- `dimRed` wraps `cmdscale`.

- In contrast to a supervised problem, there is no natural way to directly measure the quality of any output or to compare two methods. Every method optimizes a different error function.

- **Quality criteria implemented in `coRanking`:**

  - **Co-ranking matrix based measures**: the co-ranking matrix Q is the 2d-histogram of the distance ranks. q_{ij} is an integer which counts how many points of distance rank j became rank i. In a perfect DR, this matrix will only have non-zero entries in the diagonal. In R, the co-ranking matrix can be calculated using the the `coRanking::coranking` function. The `dimRed` package contains the functions `Q_local, Q_global, Q_NX, LCMC`, and `R_NX` to calculate the above quality measures in addition to `AUC_lnK_R_NX`. If R_{NX} is high for low values of K, then local neighborhoods are maintained well; if R_{NX} is high for large values of K, then global gradients are maintained well (see fig. 4).
  - Cophenetic correlation.
  - **Reconstruction error**: the fairest one when the method provides an inverse mapping. RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n} d\left(x'_i, x_i\right)^2}$, with $x'_i = f^{-1}(y_i) = f^{-1}(f(x_i))$.

- **Test datasets**: Common ones being the 3d S-curve and the Swiss roll. Real world examples usually have more dimensions and often are much noisier and we cannot be sure if we can observe all the relevant variables. Can be retrieved with `dimRed::loadDataSet`

- **Main functions:** `embed, quality, plot, plot_R_NX`.

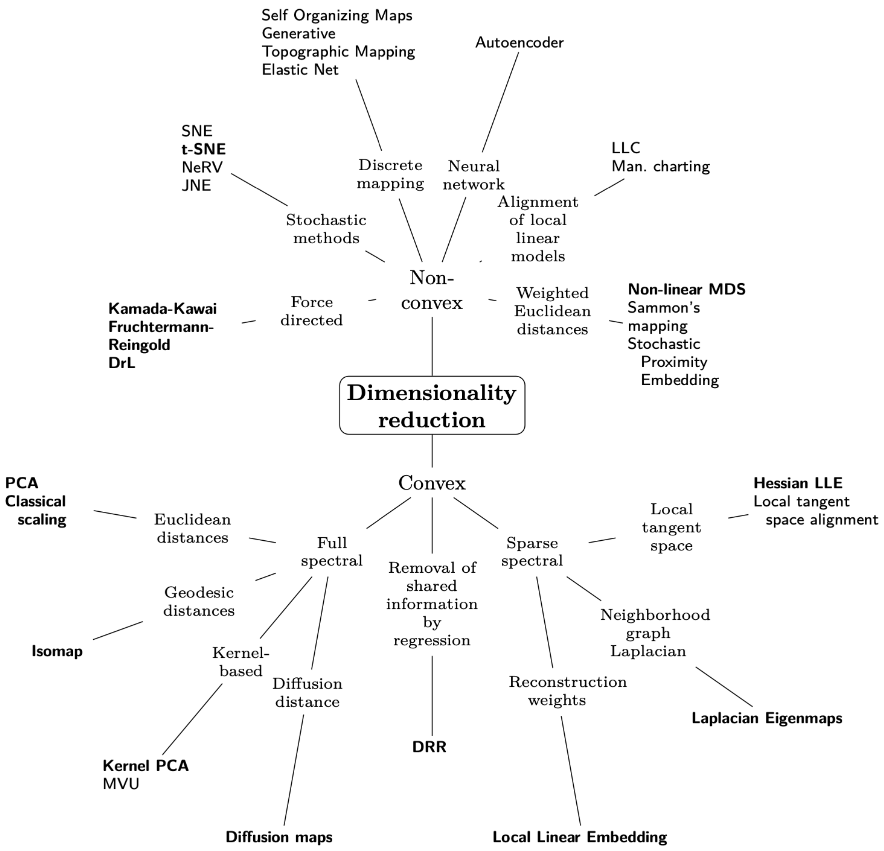

Figure 3: DR methods implemented in `dimRed` [2].

4 Sparse multidimensional scaling using landmark points [1]

4.1 Abstract

Original: In this paper, we discuss a computationally efficient approximation to the classical multidimensional scaling (MDS) algorithm, called Landmark MDS (LMDS), for use when the number of data points is very large. The first step of the algorithm is to run classical MDS to embed a chosen subset of the data, referred to as the 'landmark points', in a low-dimensional space. Each remaining data point can be located within this space given knowledge of its distances to the landmark points. We give an elementary and explicit theoretical analysis of this procedure, and demonstrate with examples that LMDS is effective in practical use.

Apple Intelligence summary: Landmark MDS approximates classical multidimensional scaling for large datasets. It embeds a subset of data points, called "landmark points", in a low-dimensional space, then locates remaining points based on their distances to these landmarks.

4.2 Key Points

- **LMDS**: Landmark MDS.

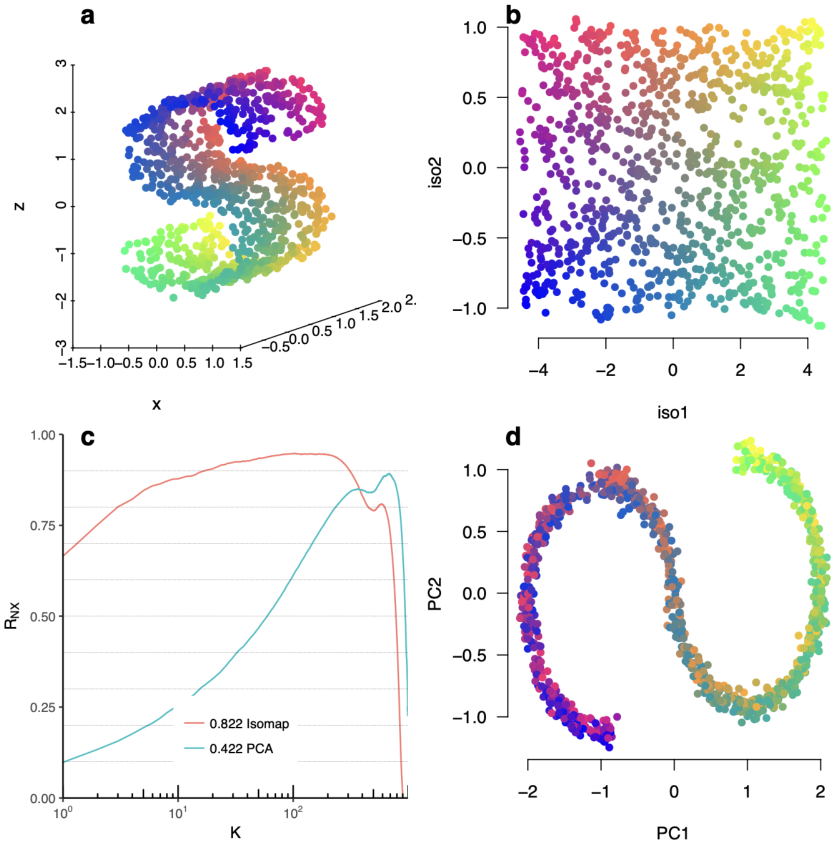

Figure 4: R_{NX} measures the quality of the embedding. [2].

- **Method**:
  1. Select n landmark points from N data points.
  2. Apply MDS to the $n \times n$ distance matrix to obrain L.
  3. Embed remaining points via distance-based triangulation.
- **Complexity**: Classical MDS: $\mathcal{O}(N^2)$ storage, $\mathcal{O}(N^3)$ time. LMDS: $\mathcal{O}(nN)$ storage, lower time complexity.
- It has links with Isomap (L-Isomap), the Nyström method (which finds approximate solutions to a positive semi-definite symmetric eigenvalue problem using just a few of the columns of the matrix) and FastMap.

5 Title TEXstring

5.1 Abstract

Original:

Apple Intelligence summary:

5.2 Key Points

-

References

[1] Vin De Silva and Joshua B Tenenbaum. *Sparse multidimensional scaling using landmark points*. Tech. rep. Technical Report, Stanford University, 2004.

[2] Guido Kraemer, Markus Reichstein, and Miguel D. Mahecha. "dimRed and coRanking - Unifying Dimensionality Reduction in R". In: *R J.* 10 (2018), p. 342. URL: `https://api.semanticscholar.org/CorpusID:62831555`.

[3] Vin Silva and Joshua Tenenbaum. "Global Versus Local Methods in Nonlinear Dimensionality Reduction". In: *Advances in Neural Information Processing Systems*. Ed. by S. Becker, S. Thrun, and K. Obermayer. Vol. 15. MIT Press, 2002. URL: `https://proceedings.neurips.cc/paper_files/paper/2002/file/5d6646aad9bcc0be55b2c82f69750387-Paper.pdf`.

[4] Kisung You and Dennis Shung. "Rdimtools: An R Package for Dimension Reduction and Intrinsic Dimension Estimation". In: *Software Impacts* 14 (2022), p. 100414. ISSN: 26659638. DOI: `10.1016/j.simpa.2022.100414`.