Manifold Learning with

Extensions to CLCA and CLDA

Thomas Rusch
WU Vienna

Date: 2025-07-01

Outline

I introduce extensions to curvilinear component analysis (CLCA) and curvilinear distance analysis (CLDA) that can be used for manifold learning and nonlinear dimensionality reduction. They add flexibility to CLCA and CLDA.

Outline:

- Idea and purpose of manifold learning
- CLCA and CLDA
- Extensions to CLCA and CLDA
- A Quasi-MM algorithm
- Example: Visualizing topic models
- Conclusion

Manifold Learning

Manifold learning (aka nonlinear dimension reduction) refers to methods that try to unfold a (high-dimensional) manifold in an unsupervised setting.

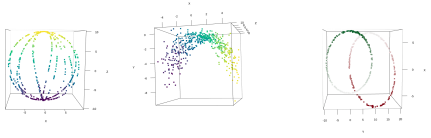
- Underlying assumption: Data exist on (or close to) a submanifold embedded in a higher-dimensional space.
- Goal: Extract the submanifold and remove the unnecessary dimensions (dimensionality reduction).
- Side effect: Represent and visualize the data on the manifold in a (usually) Euclidean space.

Since a manifold is a topological space that locally resembles Euclidean space near each point, this is done by trying to represent the local neighbourhood accurately at expense of the global structure of the manifold.

Example Manifolds

Some examples:

- Points on a sphere sampled along a Clelia curve ($c = 1/6$).
- Points near a U-fold with increasing noise towards the edges.
- Two interlocking rings.



For all of them the goal in manifold learning is to represent the points as a (Euclidean) embedding that respects local structure and removes global structure.

Methods – I

There exist a number of techniques that try to achieve this, e.g., Kernel Methods (Kernel PCA), neural networks (self-organizing map, autoencoders), spectral methods (Laplacian and Hessian eigenmaps), cool kid stuff (t-SNE, UMAP).

We are specifically interested in methods that are proximity preserving and based around multidimensional scaling (MDS). Those that have locality features for manifold learning are:

- Elastic scaling (McGee, 1966)
- Sammon mapping (Sammon, 1969)
- CLCA (Desmartines and Herault, 1994)
- Isomap (Tenenbaum et al., 2000)
- CLDA (Lee et al., 2004)

Methods – II

We developed extensions to CLCA and CLDA (eCLCA, eCLDA, eCLPCA, eCLPDA) that incorporate any or all of

1. Static weights
2. Linear, affine and smooth montonic proximity transformations (optimal scaling)
3. Power transformations of the fitted distances (map to curved Euclidean space)

That way we can simultaneously generalize local methods of MDS and add locality capabilities to global MDS methods, e.g.,

- bring the Isomap idea to Kruskal's MDS framework
- incorporate different scale levels of the proximity measure in CLCA, CLDA
- allow to combine Sammon mapping and elastic scaling with CLCA/CLDA
- give manifold learning versions of ALSCAL, POST-MDS and MULTISCALE

Curvilinear Component/Distance Analysis – I

We are given an $n \times n$ matrix of symmetric proximities δ_{ij} of the n points. In CLCA they are Euclidean distances and in CLDA geodesic distances (distance along the manifold as in Isomap).

We approximate these proximities by pairwise fitted distances $d_{ij}(X)$ where X (the configuration) is $n \times p$. Usually the fitted distance is Euclidean and p is small.

Approximation is based on a badness-of-fit criterion $\sigma(\delta_{ij}, d_{ij}(X)|\tau)$ and we look for

$$\arg\min_X \sigma(\delta_{ij}, d_{ij}(X)|\tau) = \sum_{i<j} \big(\delta_{ij} - d_{ij}(X)\big)^2 \, \mathbb{1}\big(d_{ij}(X) \leq \tau\big)$$

the X that minimizes the approximation error. Note that any i, j for which $d_{ij}(X) > \tau$ is ignored which gives the local flavour. If $\tau > \max(d_{ij}(X))$ this is equivalent to ratio MDS.

Extended Curvilinear Component/Distance Analysis – I

We can extend CLCA/CLDA in three ways:

1. Adding static finite weights w_{ij}: $\sum_{i<j} w_{ij} \left(\delta_{ij} - d_{ij}(X)\right)^2 \mathbb{1}\left(d_{ij}(X) \leq \tau\right)$ (incorporates Sammon mapping and elastic scaling with $w_{ij} = \delta_{ij}^{-1}$ or $w_{ij} = \delta_{ij}^{-2}$ respectively, any other weighting also possible).
2. Optimal scaling: Instead of δ_{ij} we use disparities $\hat{\delta}_{ij} = f(\delta_{ij})$ with $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. These transformations are under the constraint that $\hat{\delta}_{ij} \geq 0$:

   Ratio: $\hat{\delta}_{ij} = b\delta_{ij}$ (linear transformation)

   Interval: $\hat{\delta}_{ij} = a + b\delta_{ij}$ (affine transformation)

   Splines: $\hat{\delta}_{ij} = \sum_z a_z I_z(\delta_{ij}|o, v)$ (smooth monotonic nonlinear transformation via I-spline)

Extended Curvilinear Component/Distance Analysis – I

We can extend CLCA/CLDA in three ways:

3. Power transformations for $d_{ij}(X)$: We can fit power functions of the Euclidean distance with exponent $\kappa > 0$, so $\hat{d}_{ij}(X) = d_{ij}(X)^{\kappa}$. This maps to a convex or concave space which can be useful to remove excess curvature of the manifold. This can also express MDS methods like, e.g., ALSCAL ($\kappa = 2$) and MULTISCALE ($\kappa \to 0$) and POST-MDS.

With $\hat{w}_{ij} = w_{ij} h_{ij}(X)$ and $h_{ij}(X) = \mathbb{1}\left(\hat{d}_{ij}(X) \leq \tau\right)$ this combines to an instance of Stress for flexible MDS (Rusch et al. 2021, 2023)

$$\sigma(\delta_{ij}, d_{ij}(X) | \tau, \kappa, w_{ij}) = \sum_{i<j} \hat{w}_{ij} \left(\hat{\delta}_{ij} - \hat{d}_{ij}(X)\right)^2$$

Extended Curvilinear Component/Distance Analysis – II

As the most general specific instance of our extensions we have extended Curvilinear Power Component Analysis (eCLPCA):

$$\arg\min_{X} \sigma_{eCLPCA}(\delta_{ij}, d_{ij}(X)|\tau, \kappa, w_{ij}) = \sum_{i<j} w_{ij} \left(f(\delta_{ij}) - d_{ij}(X)^{\kappa}\right)^2 \mathbb{1}\left(d_{ij}(X)^{\kappa} \leq \tau\right)$$

Special cases include:

- If δ_{ij} is the geodesic distance we have eCLPDA.
- If $\kappa = 1$ this is eCLCA (Euclidean input distance) or eCLDA (geodesic proximity).
- If $f(\cdot)$ is the ratio transformation, we have CLCA (Euclidean input distance) or CLDA (geodesic proximity).
- Sammon and elastic scaling-type weighting with $w_{ij} = \delta_{ij}^{\nu}$ and $\nu \leq 0$.
- Different MDS methods with $\tau > \max(\hat{d}_{ij}(X))$ or local versions of MDS with setting τ.

Optimization with Q-SMACOF – I

For optimization we developed an algorithm for eCLPCA that works any input proximity, a Quasi-MM algorithm called Q-SMACOF. Nutshell:

- Let $\hat{w}_{ij} := w_{ij}\mathbb{1}\left(d_{ij}(X)^\kappa \leq \tau\right)$. In iteration k, we treat $\hat{w}_{ij}$ as static.
- Use surrogate majorization function for the alternating least squares objectives

$$\vartheta^{(k)} = \arg\min_{\vartheta} \sigma_\kappa\left(\vartheta, X^{(k)} | \hat{W}\right)$$

$$X^{(k+1)} = \arg\min_{vec(X)^\top vec(X)=1} \sigma_\kappa\left(\vartheta^{(k)}, X | \hat{W}\right)$$

  over all X in $\mathbb{R}^{n\times p}$ and for all values of $\kappa > 0$.
- Minimize the majorization function in each iteration step $k+1$ (MM principle).
- At each iteration k we re-calculate the $\mathbb{1}\left(d_{ij}(X)^\kappa \leq \tau\right)$ which may possibly change $\hat{w}_{ij}$.
- At each iteration k do optimal scaling in an inner optimization step given $\hat{w}_{ij}^{(k)}$.

Optimization with Q-SMACOF – II

For optimization we developed an algorithm for eCLPCA that works any input proximity, a Quasi-MM algorithm called Q-SMACOF.

- eCLPCA objective is not smooth due in X to the $\mathbb{1}\left(d_{ij}(X)^\kappa \leq \tau\right)$.
- The dynamic nature of $\hat{w}_{ij}$ can introduce jump discontinuities in the optimization path and thus a possible increase of the objective (also in the CLCA algorithm)
- The algorithm minimizes the objective but is only converging to a stationary point once the $\hat{w}_{ij}$ does no longer change.
- The larger τ is, the more quickly the $\hat{w}_{ij}$ stabilizes.

That is why we do not necessarily have monotone convergence of MM (hence Quasi-MM).

Self-organization

CLCA is proposed as a self-organizing ANN, which uses a decreasing sequence of τs to automatize the use. We can adapt this to our situation.

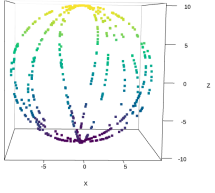
We start with any τ_s, $\tau_s > \min(\hat{d}_{ij}(X))$ and fit the eCLPCA variant of choice with τ_s. Then:

1. Use a new $\tau_t < \tau_s$, $\tau_t > \min(\hat{d}_{ij}(X))$ and fit the model with current τ_t with the solution for τ_s as the starting configuration.
2. Repeat this for a strictly decreasing sequence of τs.
3. Use the optimal X obtained for the smallest τ_t.

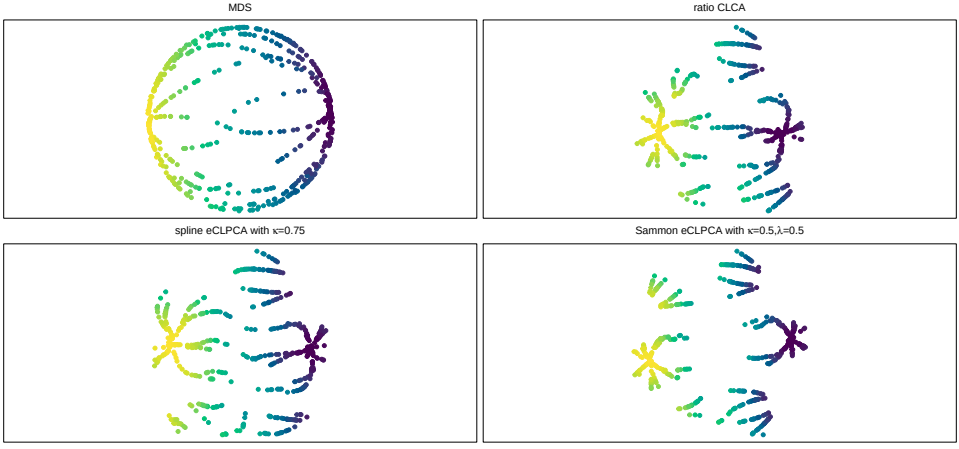
This is self-organizing as it gradually refines the solution for increasingly smaller distances. Distances that are set to 0 for τ_s are also set to 0 for τ_t and with every epoch, the configuration is rearranged based on an increasingly narrower subset of observations.

Toy Example: Sphere

For the sphere data, I fit a ratio MDS, a ratio eCLCA model with $\tau = 0.02$ (equivalent to standard CLCA), a spline eCLPCA model with $\kappa = 0.75$, $\tau = 0.05$ (the I-spline has degree of 2 and 2 interior knots) and a ratio eCLPCA model for $\sqrt{\delta_{ij}}$ with $\kappa = 0.5$ and $\tau = 0.15$ and Sammon weighting (i.e., $w_{ij} = \delta_{ij}^{-1}$).

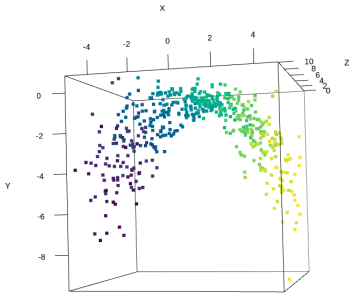


Toy Example: Sphere

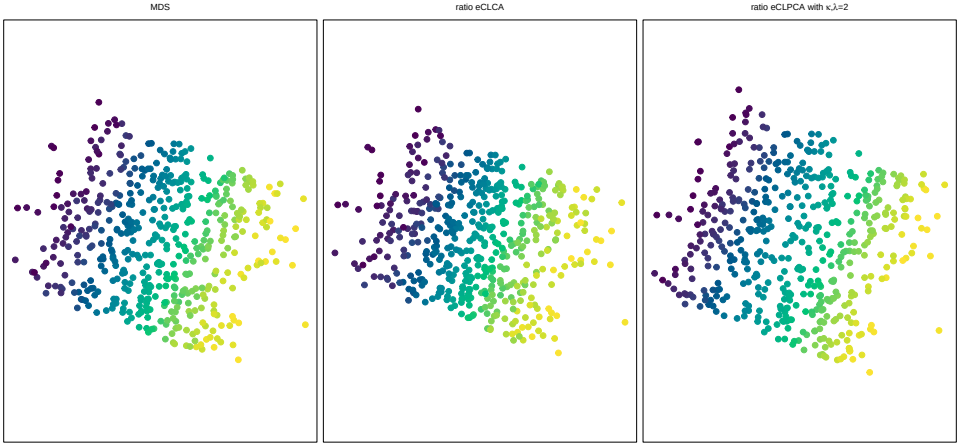


Toy Example: U-fold

Here we fit ratio MDS, the self-organizing versions of standard CLCA and a ratio eCLPCA with $\delta_{ij}^2, \kappa = 2$ starting with $\tau_s = 0.5$ and 0.1 respectively.

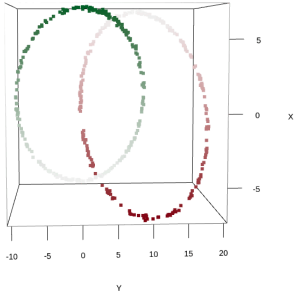


Toy Example: U-fold

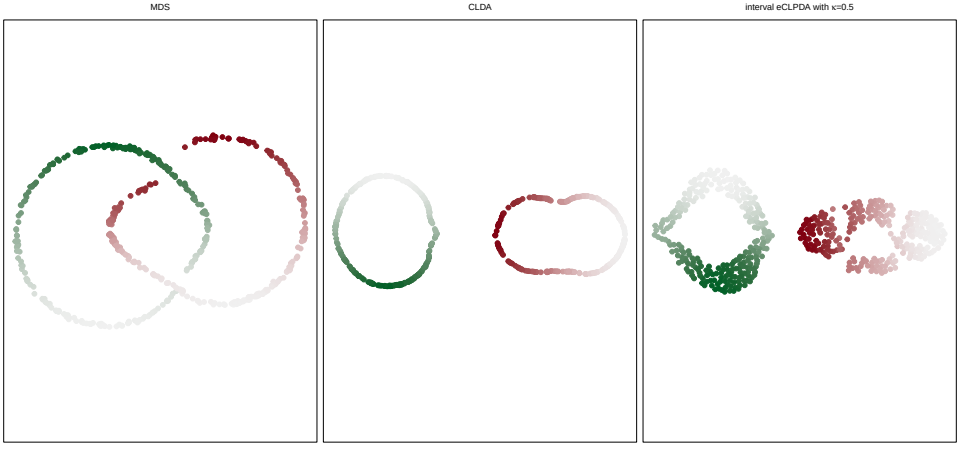


Toy Example: 2 Rings

I fit two models: CLDA with $k = 58, \tau = 0.03$ and an interval eCLPDA with $k = 58, \kappa = 0.5, \tau = 0.03^{\kappa}$.



Toy Example: 2 Rings



Real Example: Topic modeling – I

I revisit Sablica et al. (2025) results.

- Data are the abstracts of the 129 scientific papers of Fritz Leisch.
- Abstracts were embedded on a hypersphere with 256 dimensions via OpenAI's text-embedding-3-large model
- Model-based clustering with spherical Cauchy distributions gave an 8 cluster solution.

We visualize and explore this with a number of eCLCA/eCLDA versions.

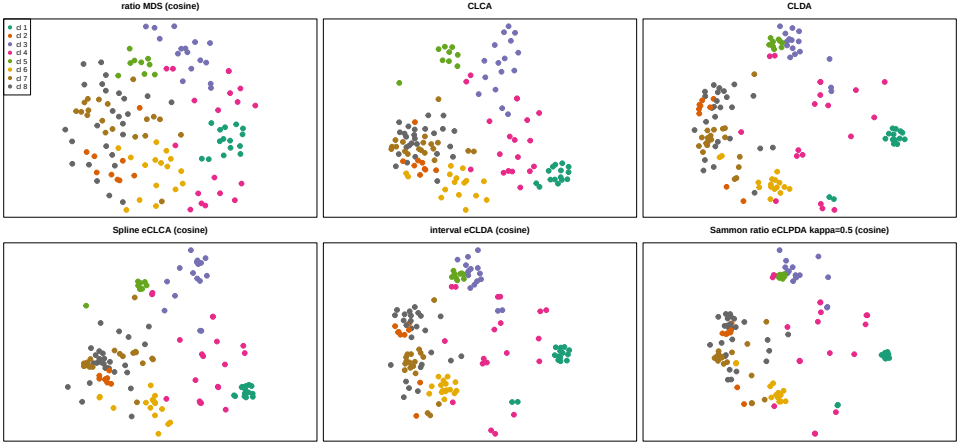
Real Example: Topic modeling - II

Specifically we look at

1. Ratio MDS with cosine input distance (our reference)
2. Standard CLCA (uses Euclidean input distance)
3. Standard CLDA (geodesic Euclidean input distance)
4. Spline eCLCA with cosine input distance
5. Interval eCLDA with cosine input distance (geodesic)
6. Sammon ratio CLPDA with $\kappa = 0.5$ and cosine input distance (geodesic)

The geodesic distances (the shortest path between poinst over the weighted neighbourhood graph) were used with either Euclidean distances (3) or cosine distance (4, 5, 6) for the edge weights. The neighbourhood graph was calculated for $k = 10$ nearest neighbours.

Real Example: Topic modeling – III



Example: Topic Model – Exploration

Some of the observations we may make:

- Geodesic cosine input distance seems to work best
- Results supports the clustering relatively well
- Sammon eCLPDA allows visualisation most aligend with the clustering result, also interval CLDA, CLDA and perhaps spline eCLCA
- Cluster 1 seems to generally be well separated in all but MDS
- Cluster 3 and Cluster 5 look connected when using geodesic distances, also Cluster 2, Cluster 7, Cluster 8
- For Cluster 2, Cluster 6, Cluster 7, Cluster 8 it looks as if they overlap or are density-connected in every setup. Sammon eCLPDA suggests nested clusters.
- Cluster 4 “Environmental and Biological Effects of Agrochemicals” seems to be difficult as its own cluster (low cohesion, broken-up)
- Some abstracts do not fit neatly to the topics → manual inspection

Software

We have implemented these functions R in the package **smacofx**.

- Fitting functions: eCLCA, eCLDA, eCLPCA, eCLPDA. Self-organizing versions are prefixed with so_, so e.g., so_eCLCA.
- Standard S3 Generics: `print`, `summary`, `plot`, `coef` (gives the parameters)
- Plotting: Configuration plot, Shepard plot, bubbleplot, stress-decomposition plot, residual plot, transformation plot (argument `plot.type`), Biplots (`biplotmds`)
- Uncertainty quantification: MDS jackknife (`jackmds`), MDS bootstrap (`bootmds`), permutation test (`permtest`)
- Local Minima Diagnostic: Effect of starting configuration (`icGenExplore`), different starting configuration (`multistart`)

Conclusions

We introduced extensions to curvilinear component analysis and curvilinear distance analysis for manifold learning. These extensions cover:

- Optimal scaling with linear, affine, spline transfromations
- Inclusion of static weighting (enabling Sammon or elastic scaling type weighting, and more)
- Power transformations for the configuration distances
- Manual specification of τ or self-organization

To use these new variants we developed

- A Q-SMACOF algorithm for optimization
- Software in R that can utilize the object-oriented approach and post-fit infrastructure of the Smacofverse.

Conclusions

We introduced extensions to curvilinear component analysis and curvilinear distance analysis for manifold learning. These extensions cover:

- Optimal scaling with linear, affine, spline transfromations
- Inclusion of static weighting (enabling Sammon or elastic scaling type weighting, and more)
- Power transformations for the configuration distances
- Manual specification of τ or self-organization

To use these new variants we developed

- A Q-SMACOF algorithm for optimization
- Software in R that can utilize the object-oriented approach and post-fit infrastructure of the Smacofverse.

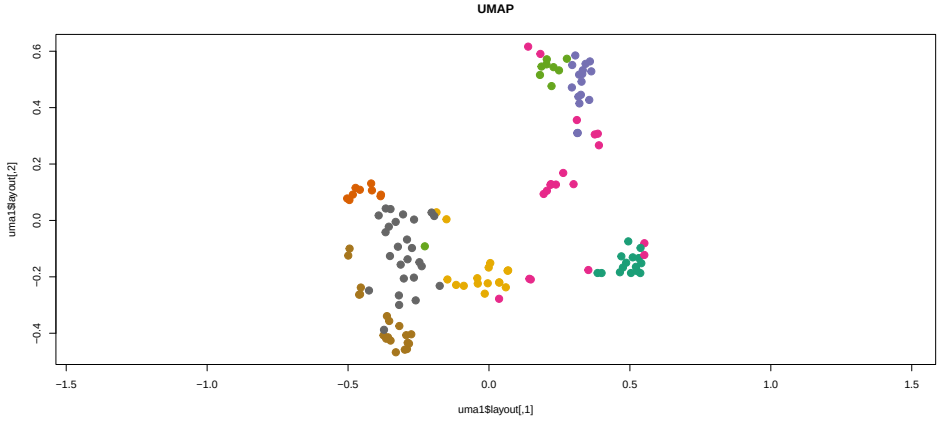
These extensions allow for more flexible learning and exploration of data that exist on or near a manifold.

References

- Demartines, P., and Herault, J. (1997). Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks, 8(1)*, 148-154.
- Lee, J., Lendasse, A., and Verleysen, M. (2004). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing, 57*, 49-76.
- McGee, V. (1966). The multidimensional analysis of 'elastic' distances. *British Journal of Mathematical and Statistical Psychology, 19*, 181-196.
- Rusch, T., De Leeuw, J., Mair, P., and Hornik, K. (under review). Flexible multidimensional scaling with the package smacofx.
- Rusch, T. (under review). Multidimensional scaling with Heaviside weighting: Extensions of curvilinear component analysis and curvilinear distance analysis.
- Rusch, T., Mair, P., and Hornik, K. (2021). Cluster optimized proximity scaling. *Journal of Computational and Graphical Statistics, 30*, 1156–1167.
- Rusch, T., Mair, P., and Hornik, K. (2023). Structure-based hyperparameter selection with Bayesian optimization in multidimensional scaling. *Statistics and Computing, 33*, 28.
- Sablica, L., Hornik, K., and Gruen, B. (2025). circlus: An R package for circular and spherical clustering using Poisson kernel-based and spherical Cauchy distributions. *Austrian Journal of Statistics, 54*, 27-42.
- Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers, C-18(5)*, 401–409.
- Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290(5500)*, 2319–2323.

Real Example: Topic modeling - UMAP

Comparison with UMAP with (neighbours=10)



Majorization Function - I

We write $h_{ij} = \mathbb{1}\left(d_{ij}(X)^\kappa \leq \tau\right)$ and $\hat{w}_{ij} = w_{ij}h_{ij}$. We treat h_{ij} as static, then we can derive a majorizing function of

$$\sigma_\kappa(X|\hat{W}) = \sum_{i<j} \hat{w}_{ij}\left(\delta_{ij} - d_{ij}(X)^\kappa\right)^2 \tag{1}$$

all X in $\mathbb{R}^{n\times p}$ and for all values of $\kappa > 0$. The disparities are scaled as $\sum_{i<j}\hat{w}_{ij}\delta_{ij}^2 = 1$.

Since Euclidean embeddings are scale invariant we have

$$\min_X \sigma_\kappa(X|\hat{W}) = \min_{\vartheta\geq 0}\min_{vec(X)^\top vec(X)=1} \sigma_\kappa\left(\vartheta, X|\hat{W}\right)$$

This can be solved via an alternating least squares algorithm with first solving over ϑ for fixed X and then over X for fixed ϑ with the steps

$$\vartheta^{(k)} = \arg\min_\vartheta \sigma_\kappa\left(\vartheta, X^{(k)}|\hat{W}\right) \tag{2}$$

$$X^{(k+1)} = \arg\min_{vec(X)^\top vec(X)=1} \sigma_\kappa\left(\vartheta^{(k)}, X|\hat{W}\right) \tag{3}$$

The minimum (2) for fixed X is attained at

$$\vartheta^\kappa = \frac{\rho_\kappa(X|\hat{W})}{\eta_\kappa(X|\hat{W})}$$

which means for (3), one needs to majorize

$$\sigma_\kappa\left(X, \vartheta|\hat{W}\right) = 1 - 2\vartheta^\kappa\rho_\kappa(X|\hat{W}) + \vartheta^{2\kappa}\eta_\kappa(X|\hat{W}) \tag{4}$$

over $vec(X)^\top vec(X) = n$ and $\vartheta^\kappa = \rho_\kappa(X|\hat{W})/\eta_\kappa(X|\hat{W})$ fixed at its current value.

Majorization Function - II

We have a MM algorithm update for X of the form of multiplying the current $X^{(k)}$ with the $n \times n$ matrix $M_K(X|\hat{W})$ for iteration $k+1$ as

$$X^{(k+1)} = M_K\left(X^{(k)}|\hat{W}\right)X^{(k)} \tag{5}$$

under the condition $||M_K(X^{(k)}|\hat{W})X^{(k)}|| = 1$.

The form of $M_K(X|\hat{W})$ differs based on the value of κ. For the case of $\kappa > 1$ we have

$$M_K(X|\hat{W}) = B_K(X|\hat{W}) - \frac{\rho_K(X|\hat{W})}{\eta_K(X|\hat{W})}\left(C_K(X|\hat{W}) - \zeta_K(X|\hat{W})I_n\right) \tag{6}$$

and for the case of $0 < \kappa \leq 1$ we have

$$M_K(X|\hat{W}) = \left(B_K(X|\hat{W}) - \beta_K(X|\hat{W})I_n\right) - \frac{\rho_K(X|\hat{W})}{\eta_K(X|\hat{W})}\left(C_K(X|\hat{W}) - \gamma_K(X|\hat{W})I_n\right).$$

with I_n denoting the $n \times n$ identity matrix.

Majorization Function - III

The terms used are:

- Three scalar valued terms $\beta_K(X|\hat{W}) = (\kappa-1)\sum_{i<j}\hat{w}_{ij}\hat{\delta}_{ij}2^{K/2}$, $\zeta_K(X|\hat{W}) = (2\kappa-1)\sum_{i<j}\hat{w}_{ij}4^{K/2}$ and $\gamma_K(X|\hat{W}) = 2\sum_{i<j}\hat{w}_{ij}d_{ij}(X)^{2(\kappa-1)}$
- The $n \times n$ matrix $B_K(X|\hat{W})$ that has elements $b_K(X|\hat{W})_{ii} = \sum_j \hat{w}_{ij}\hat{\delta}_{ij}d_{ij}(X)^{K-2}$ in the main diagonal and $b_K(X|\hat{W})_{ij} = -w_{ij}\hat{\delta}_{ij}d_{ij}(X)^{K-2}$ if $i \neq j$
- The $n \times n$ matrix $C_K(X|\hat{W})$ that has elements $c_K(X|\hat{W})_{ii} = \sum_j \hat{w}_{ij}d_{ij}(X)^{2K-2}$ in the main diagonal and $c_K(X|\hat{W})_{ij} = -\hat{w}_{ij}d_{ij}(X)^{2K-2}$ if $i \neq j$.

Q-SMACOF Algorithm

Let $r := \kappa/2$ and $k^{\max}$ be the maximum iteration number.

1. Set $k = 0$. Set $X^{(0)}$ to an initial X. This can be the configuration of a Torgerson scaling.
2. Set $\hat{w}_{ij}^{(0)}(\hat{d}_{ij}(X^{(0)})) = 0$ if $d_{ij}(X^{(0)})^{\kappa} > \tau$.
3. Optimally scale the δ_{ij} by calculating $\hat{\delta}_{ij} = f(\delta_{ij}^{\lambda})$ for fixed distances $\hat{d}_{ij}(X^{(0)})$.
4. Compute initial $\sigma_r(X^{(0)}; \hat{W}^{(0)}(\hat{D}(X^{(0)}))$.
5. Repeat until convergence
   - 5.1 Compute $M_r\left(X^{(k)}; \hat{W}(\hat{D}(X^{(k)})\right)$.
   - 5.2 Update $X^{(k+1)} = M_r\left(X^{(k)}; \hat{W}(\hat{D}(X^{(k)}))\right)X^{(k)}$.
   - 5.3 Compute $d_{ij}^{2r}(X^{(k+1)})$.
   - 5.4 Set $\hat{w}_{ij}^{(k+1)}(\hat{d}_{ij}(X^{(0)})) = 0$ if $d_{ij}(X^{(k+1)})^{\kappa} > \tau$.
   - 5.5 Optimally scale the δ_{ij} by calculating $\hat{\delta}_{ij} = f(\delta_{ij}^{\lambda})$ for fixed distances $\hat{d}_{ij}(X^{(k)})$.
   - 5.6 Compute $\sigma_r\left(X^{(k+1)}; W^{(k+1)}(\hat{D}(X^{(k+1)}))\right)$.
   - 5.7 Stop if $|\sigma_r\left(X^{(k+1)}; W^{(k+1)}\right) - \sigma_r\left(X^{(k)}; W^{(k)}(\hat{D}(X^{(k+1)}))\right)| < \epsilon$ or $k > k^{\max}$.
   - 5.8 Set $k := k + 1$