Non-Euclidean Multi-Dimensional Scaling

Jie Gao

Computer Science Department
Rutgers University

December 5th 2024

Non-Euclidean Dissimilarities

d-dim vectors:

- Euclidean metric: ℓ_2, $\|p - q\| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$.

Non-Euclidean Dissimilarities

d-dim vectors:

- Euclidean metric: ℓ_2, $\|p - q\| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$.

In applications, many distances/dissimilarities are not Euclidean:

- Minkowski distance (ℓ_p), cosine similarity, Hamming, Jaccard, Mahalanobis, Chebyshev, and KL-divergence.

Non-Euclidean Dissimilarities

d-dim vectors:

- Euclidean metric: ℓ_2, $\|p - q\| = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2}$.

In applications, many distances/dissimilarities are not Euclidean:

- Minkowski distance (ℓ_p), cosine similarity, Hamming, Jaccard, Mahalanobis, Chebyshev, and KL-divergence.
- Some may not even be a metric (e.g., no triangle inequality).
- Human perception of similarity is not a metric. [Tversky & Gati 82]

Non-Euclidean Dissimilarities

d-dim vectors:

- Euclidean metric: ℓ_2, $\|p - q\| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$.

In applications, many distances/dissimilarities are not Euclidean:

- Minkowski distance (ℓ_p), cosine similarity, Hamming, Jaccard, Mahalanobis, Chebyshev, and KL-divergence.
- Some may not even be a metric (e.g., no triangle inequality).
- Human perception of similarity is not a metric. [Tversky & Gati 82]
- Negative entries (e.g.: cosine similarity, correlation, signed distance).

Non-Euclidean Dissimilarities

d-dim vectors:

- Euclidean metric: ℓ_2, $\|p - q\| = \sqrt{\sum_{i=1}^{d}(p_i - q_i)^2}$.

In applications, many distances/dissimilarities are not Euclidean:

- Minkowski distance (ℓ_p), cosine similarity, Hamming, Jaccard, Mahalanobis, Chebyshev, and KL-divergence.
- Some may not even be a metric (e.g., no triangle inequality).
- Human perception of similarity is not a metric. [Tversky & Gati 82]
- Negative entries (e.g.: cosine similarity, correlation, signed distance).

Dimension reduction for non-Euclidean distances/dissimilarities?

Dimension Reduction for Non-Euclidean Data

Given a symmetric dissimilarity matrix $D = \{D_{ij}\}$ of a dataset P of n elements, find k-dimensional vector representation $\hat{P}$ and a distance function f such that $\hat{D}_{ij} = \Phi(\hat{p}_i, \hat{p}_j)$ approximates D_{ij}.

Dimension Reduction for Non-Euclidean Data

Given a symmetric dissimilarity matrix $D = \{D_{ij}\}$ of a dataset P of n elements, find k-dimensional vector representation $\hat{P}$ and a distance function f such that $\hat{D}_{ij} = \Phi(\hat{p}_i, \hat{p}_j)$ approximates D_{ij}.

- Φ is a bilinear form: $\Phi(u, v)$ is linear wrt each variable u or v.

Dimension Reduction for Non-Euclidean Data

Given a symmetric dissimilarity matrix $D = \{D_{ij}\}$ of a dataset P of n elements, find k-dimensional vector representation $\hat{P}$ and a distance function f such that $\hat{D}_{ij} = \Phi(\hat{p}_i, \hat{p}_j)$ approximates D_{ij}.

- Φ is a bilinear form: $\Phi(u, v)$ is linear wrt each variable u or v.
- Wlog, $\Phi(u, v) = u^T A v$ with a symmetric matrix A.

Dimension Reduction for Non-Euclidean Data

Given a symmetric dissimilarity matrix $D = \{D_{ij}\}$ of a dataset P of n elements, find k-dimensional vector representation $\hat{P}$ and a distance function f such that $\hat{D}_{ij} = \Phi(\hat{p}_i, \hat{p}_j)$ approximates D_{ij}.

- Φ is a bilinear form: $\Phi(u, v)$ is linear wrt each variable u or v.
- Wlog, $\Phi(u, v) = u^T A v$ with a symmetric matrix A.
- Euclidean setting: $A = I$ and $\Phi(u, v) = u^T v$ is dot product.
- When A is positive semi-definite: isometric to the Euclidean space.

Dimension Reduction for Non-Euclidean Data

Given a symmetric dissimilarity matrix $D = \{D_{ij}\}$ of a dataset P of n elements, find k-dimensional vector representation $\hat{P}$ and a distance function f such that $\hat{D}_{ij} = \Phi(\hat{p}_i, \hat{p}_j)$ approximates D_{ij}.

- Φ is a bilinear form: $\Phi(u, v)$ is linear wrt each variable u or v.
- Wlog, $\Phi(u, v) = u^T Av$ with a symmetric matrix A.
- Euclidean setting: $A = I$ and $\Phi(u, v) = u^T v$ is dot product.
- When A is positive semi-definite: isometric to the Euclidean space.
- Minimize STRESS error $\|\hat{D} - D\|_F^2$.

Outline

- Brief review of classical dimension reduction via multi-dimensional scaling (cMDS).
- Problems with cMDS on non-Euclidean data.
- How to fix it with non-Euclidean MDS.

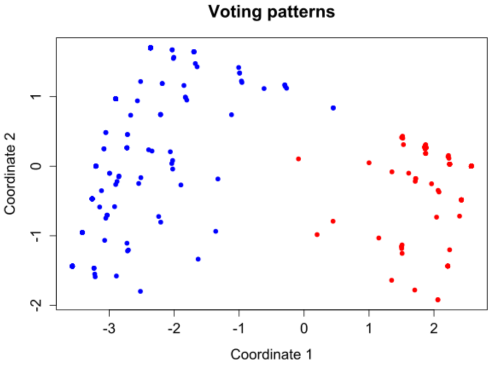
Classical Multi-Dimensional Scaling (cMDS)

Given an input matrix of Euclidean distances between n points in $\mathbb{R}^d$, recover the coordinates of the points. [Torgerson 1958]

- Euclidean distance matrix (EDM) $D = \{d_{ij}^2\}$
- Centering: $B = -\frac{1}{2}CDC$, where $C = I - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$.
- B is the Gram matrix of D. We find its orthogonal diagonalization $U^T \Lambda U$, $\Lambda = diag(\lambda_1, ...\lambda_n)$
- Since D is Euclidean, B is positive semi-definite.
- $X = \sqrt{Diag(\Lambda)}U$ is a $n \times n$ dimensional matrix of rank d (or less), specifying the coordinates of n points in $\mathbb{R}^n$.

cMDS for Dimension Reduction

If we want to find an embedding in $\mathbb{R}^k$ with $k < d$, we keep the dimensions corresponding to the k largest eigenvalues.



Wiki: voting patterns in the United States House of Representatives.

cMDS in Reality

In practice, the input distance matrix D is not necessarily Euclidean and may not be a metric.

cMDS in Reality

In practice, the input distance matrix D is not necessarily Euclidean and may not be a metric.

The standard practice is to use cMDS anyway to find $\hat{D}$, the embedding distance matrix.

cMDS in Reality

In practice, the input distance matrix D is not necessarily Euclidean and may not be a metric.
The standard practice is to use cMDS anyway to find $\hat{D}$, the embedding distance matrix.

- cMDS **does not** minimize STRESS error $\|D - \hat{D}\|_F^2$ – In Euclidean setting, cMDS minimizes STRAIN error (for Gram matrix).

cMDS in Reality

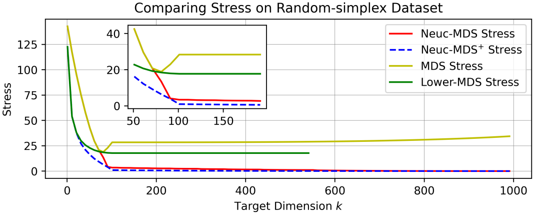
In practice, the input distance matrix D is not necessarily Euclidean and may not be a metric.
The standard practice is to use cMDS anyway to find $\hat{D}$, the embedding distance matrix.

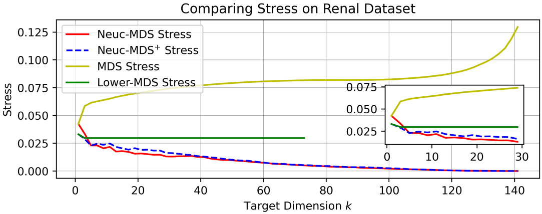
- cMDS **does not** minimize STRESS error $\|D - \hat{D}\|_F^2$ – In Euclidean setting, cMDS minimizes STRAIN error (for Gram matrix).
- Increasing k can give **worse** STRESS error. [SBRG'23, TP'16] – dimensionality paradox.

- How can classical multidimensional scaling go wrong?, NeurIPS'23. [SBRG'23]
- Taking all positive eigenvectors is suboptimal in classical multidimensional scaling. SIAM J. Optim, 2016. [TP'16]

cMDS in Reality

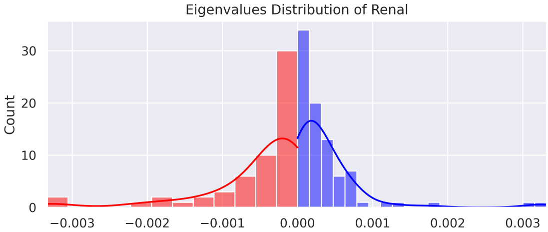
Genomics data from the Curated Microarray Database (CuMiDa)





cMDS in Reality

Eigenvalues of Gram matrix B are no longer non-negative



Non-Euclidean MDS

- If input data is no longer Euclidean, limiting the embedding in Euclidean space is too restrictive and ill-fit.

Non-Euclidean MDS

- If input data is no longer Euclidean, limiting the embedding in Euclidean space is too restrictive and ill-fit.
- cMDS drops all negative eigenvalues which contain important information.

Non-Euclidean MDS

- If input data is no longer Euclidean, limiting the embedding in Euclidean space is too restrictive and ill-fit.
- cMDS drops all negative eigenvalues which contain important information.
- Goal: dimension reduction to k-dim vectors with general bilinear form $f(u, v) = u^T Av$ to minimize STRESS error.

Non-Euclidean MDS

- We still start from the Gram Matrix $B = U^T \Lambda U$.
- Let $X = \sqrt{Diag(|\Lambda|)}U$.

Non-Euclidean MDS

- We still start from the Gram Matrix $B = U^T \Lambda U$.
- Let $X = \sqrt{Diag(|\Lambda|)}U$.
- Redefine Gram Matrix to be $B = X^T AX$ where $A = diag(sgn(\lambda_1), ...sgn(\lambda_n))$.

Non-Euclidean MDS

- We still start from the Gram Matrix $B = U^T \Lambda U$.
- Let $X = \sqrt{Diag(|\Lambda|)}U$.
- Redefine Gram Matrix to be $B = X^T A X$ where $A = diag(sgn(\lambda_1), ...sgn(\lambda_n))$.
- In doing this, we have changed the inner product to :

$$\Phi(u, v) = \sum_{i=1}^{p} u_i v_i - \sum_{i=p+1}^{p+q} u_i v_i.$$

with the addition and subtraction corresponding to the positive and negative eigenvalues. This is called pseudo-Euclidean space with (p, q) signature.

Non-Euclidean Dimension Reduction

If we use only k non-zero eigenvalues, how to choose?

Non-Euclidean Dimension Reduction

If we use only k non-zero eigenvalues, how to choose?

- Analysis of STRESS error.

Non-Euclidean Dimension Reduction

If we use only k non-zero eigenvalues, how to choose?

- Analysis of STRESS error.
- Which k eigenvalues from the input Gram matrix should we take?
- What if we are not limited to eignvalues from the input Gram matrix?

Non-Euclidean MDS: Error Analysis

Suppose we select k out of n eigenvalues S, STRESS$=C_1 + C_2 + C_3$.

- $C_1 = 4\sum_{i \notin S} \lambda_i^2$.

Non-Euclidean MDS: Error Analysis

Suppose we select k out of n eigenvalues S, STRESS$=C_1 + C_2 + C_3$.

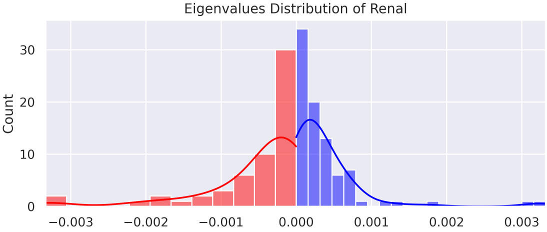
- $C_1 = 4\sum_{i \notin S} \lambda_i^2$.
- $C_2 = 4[\sum_{i \notin S} \lambda_i]^2$.

Non-Euclidean MDS: Error Analysis

Suppose we select k out of n eigenvalues S, STRESS$=C_1 + C_2 + C_3$.

- $C_1 = 4\sum_{i\notin S} \lambda_i^2$.
- $C_2 = 4[\sum_{i\notin S} \lambda_i]^2$.
- $C_3 \geq 0$

Classical MDS: when all $\lambda_i \geq 0$, choosing largest k eigenvalues minimizes $C_1 + C_2$. – No longer true with negative eigenvalues.



Non-Euclidean MDS: Choice of Eigenvalues

We developed an optimal algorithm to minimize $C_1 + C_2$: throw away eigenvalues of small magnitude & keep sum to be close to 0.

Non-Euclidean MDS: Choice of Eigenvalues

We developed an optimal algorithm to minimize $C_1 + C_2$: throw away eigenvalues of small magnitude & keep sum to be close to 0.

- iteratively add an eigenvalue to S:
- If $\sum$ remaining eigenvalues is < 0, select the most negative one.
- If $\sum$ remaining eigenvalues is > 0, select the most positive one.

Non-Euclidean MDS: Choice of Eigenvalues

If we are not limited to choose only from input eigenvalues but any k non-zero numbers:

- Equivalently, we allow scaling of the chosen eigenvalues.

Non-Euclidean MDS: Choice of Eigenvalues

If we are not limited to choose only from input eigenvalues but any k non-zero numbers:

- Equivalently, we allow scaling of the chosen eigenvalues.
- In order to minimize $C_1 + C_2$, we minimize:

$$\min_{|W|=k} \left[\sum_{i \notin W} \lambda_i^2 + \frac{1}{1+k} \left(\sum_{i \notin W} \lambda_i \right)^2 \right]$$

Non-Euclidean MDS: Choice of Eigenvalues

If we are not limited to choose only from input eigenvalues but any k non-zero numbers:

- Equivalently, we allow scaling of the chosen eigenvalues.
- In order to minimize $C_1 + C_2$, we minimize:

$$\min_{|W|=k} \left[\sum_{i \notin W} \lambda_i^2 + \frac{1}{1+k} \left(\sum_{i \notin W} \lambda_i \right)^2 \right]$$

- The optimal algorithm now needs to look at marginal gain of adding the most positive or negative eigenvalue instead.

Experiments

Sources of non-Euclidean distances in generated data:

- Random noise: a simplex with random weights.
- Distance between sets: min distance between balls in space.

Dataset	Size	$\# \{\lambda < 0\}$	Classes	Metric
Simplex	1000	900	N.A.	✗
Ball	1000	887	N.A.	✗
Brain	130	53	5	✗
Breast	151	59	6	✗
Colorectal	194	78	2	✗
Leukemia	281	117	7	✗
Renal	143	57	2	✗
MNIST	1000	454	10	✓
Fashion	1000	429	10	✓
CIFAR-10	1000	399	10	✓

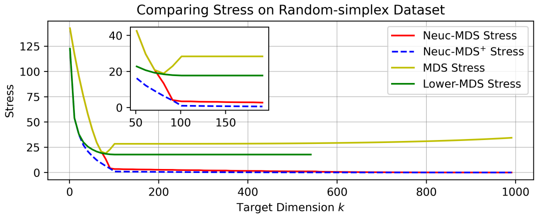
Experiments: Significantly lower STRESS

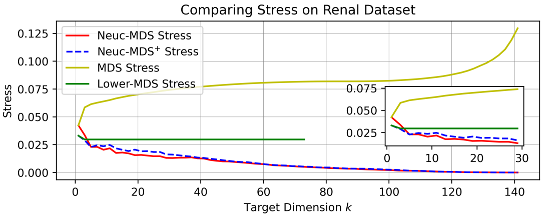
Lower-MDS [Sonthalia et.al'21]: symmetric, low-rank, trace-zero PSD
SMACOF [Scikit-learn]: non-linear optimization using majorization

Dataset	cMDS	Lower-MDS	Neuc-MDS	Neuc-MDS^{+}	SMACOF
Random-Simplex	80.520	31.542	1.179	**0.194**	15.962
Euclidean Ball	36.975	17.303	**1.196**	1.351	4e6
Brain (50161)	2.894	0.289	0.046	**0.045**	0.081
Breast (45827)	2.822	0.423	**0.029**	**0.029**	0.078
Colorectal (44076)	1.464	0.221	**0.017**	0.026	0.036
Leukemia (28497)	2.958	0.624	0.078	0.096	**0.005**
Renal (53757)	0.490	0.090	0.026	0.036	**0.017**
MNIST	65.107	37.896	9.935	**9.885**	2.35e5
Fashion-MNIST	35.235	1.955	0.613	**0.612**	2.80e5
CIFAR10	26.598	1.276	0.858	**0.850**	1.63e5

Experiments: STRESS error drops when k goes up

No dimensionality paradox: STRESS drops monotonically when dimension k is higher.





No Dimension Reduction for Random Dissimilarities

[Theorem] Consider a random symmetric, centered matrix $B \in \mathbb{R}^{n \times n}$ where B_{ij} is iid with second moments σ^2. Let e_C denote the $C_1 + C_2$ error for cMDS and e_N for Non-Euclidean MDS,

1. when $k = o(n)$, $e_C \approx n^2\sigma^2(1 + \frac{4k^2}{n} - \frac{4k}{n})$, $e_N \approx n^2\sigma^2(1 - \frac{4k}{n})$
2. when $k = cn$, with $c \to 1$, $e_N \approx 0$. When $c \geq 1/2$, $e_C \approx 0.1801 \cdot n^3\sigma^2$.

No Dimension Reduction for Random Dissimilarities

[Theorem] Consider a random symmetric, centered matrix $B \in \mathbb{R}^{n \times n}$ where B_{ij} is iid with second moments σ^2. Let e_C denote the $C_1 + C_2$ error for cMDS and e_N for Non-Euclidean MDS,

1. when $k = o(n)$, $e_C \approx n^2\sigma^2(1 + \frac{4k^2}{n} - \frac{4k}{n})$, $e_N \approx n^2\sigma^2(1 - \frac{4k}{n})$
2. when $k = cn$, with $c \to 1$, $e_N \approx 0$. When $c \geq 1/2$, $e_C \approx 0.1801 \cdot n^3\sigma^2$.

- No agressive dimension reduction with $k = o(n)$ — In contrast, ℓ_2 distances in $\mathbb{R}^n$ enjoy dimension reduction to dimension $O(\log n)$.

No Dimension Reduction for Random Dissimilarities

[Theorem] Consider a random symmetric, centered matrix $B \in \mathbb{R}^{n \times n}$ where B_{ij} is iid with second moments σ^2. Let e_C denote the $C_1 + C_2$ error for cMDS and e_N for Non-Euclidean MDS,

1. when $k = o(n)$, $e_C \approx n^2\sigma^2(1 + \frac{4k^2}{n} - \frac{4k}{n})$, $e_N \approx n^2\sigma^2(1 - \frac{4k}{n})$
2. when $k = cn$, with $c \to 1$, $e_N \approx 0$. When $c \geq 1/2$, $e_C \approx 0.1801 \cdot n^3\sigma^2$.

- No agressive dimension reduction with $k = o(n)$ — In contrast, ℓ_2 distances in $\mathbb{R}^n$ enjoy dimension reduction to dimension $O(\log n)$.
- Dimensionality paradox for cMDS: error reaches a plateau $\approx 0.1801 \cdot n^3\sigma^2$.

Practical Considerations

- Non-Euclidean MDS, like cMDS, asks for solving top k largest/smallest eigenvalues, which requires time $O(kn^2)$ using SVD.

Practical Considerations

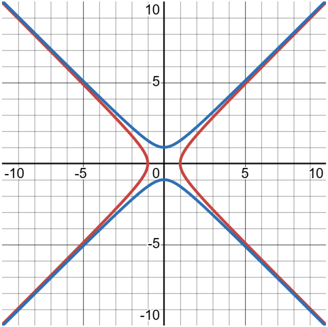
- Non-Euclidean MDS, like cMDS, asks for solving top k largest/smallest eigenvalues, which requires time $O(kn^2)$ using SVD.
- Landmark MDS [Silva,Tenenbaum 04]: embedding s landmarks first and the rest placed wrt landmarks can significantly speed up the process.

Practical Considerations

- Non-Euclidean MDS, like cMDS, asks for solving top k largest/smallest eigenvalues, which requires time $O(kn^2)$ using SVD.
- Landmark MDS [Silva,Tenenbaum 04]: embedding s landmarks first and the rest placed wrt landmarks can significantly speed up the process.
- The current Github: https://github.com/KLu9812/MDSPlus

Future Work

- Applications to machine learning models and tasks.
- Further study of $\mathbb{R}^d$ under general bilinear forms.



Unit disk of $(1, 1)$ signature in the plane.

Acknowledgements

Chengyuan Deng, Jie Gao, Kevin Lu, Feng Luo, Hongbin Sun, Cheng Xin, to appear at NeurIPS 2024. https://arxiv.org/abs/2411.10889

- NSF IIS-2229876, DMS-2220271, DMS-2311064, CCF-2208663, CCF-2118953, and CRCNS2207440.

- Questions?