San José State University

Math 250: Mathematical Data Visualization

Multidimensional Scaling (MDS)

Dr. Guangliang Chen

Outline of the presentation

- The MDS problem
- Mathematical setup, derivation and solution
- Numerical issues
- Experiments

The MDS problem

Assume a collection of n objects, $O = \{o_1, \ldots, o_n\}$, which can be points, images, documents, people, etc. Suppose also that we are given the pairwise dissimilarities of the objects in the set O,

$$\mathbf{D} = (d_{ij}) \in \mathbb{R}^{n\times n}, \qquad \text{where} \quad d_{ij} = \text{dissimilarity}(o_i, o_j),\ 1 \le i, j \le n$$

which specify in some way how different the objects are from each other.

Note that these dissimilarities are not necessarily computed based on a valid distance metric and may violate some of the conditions required by a distance metric (e.g., the triangle inequality).

Multidimensional Scaling (MDS)

Remark. When the objects are vectors, the pairwise dissimilarities can be computed based on one of the following distance metrics:

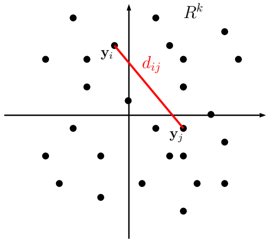
- Euclidean distance (ℓ_2)
- Manhattan/Cityblock distance (ℓ_1)
- Chebyshev/maximum coordinate difference (ℓ_∞)
- Cosine of the angle: $\|\frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|}\|^2 = 2 - 2\cos\theta$
- Correlation coefficient: $1 - \mathrm{corr}(\mathbf{x}, \mathbf{y})$

Multidimensional Scaling (MDS)

Given the matrix of pairwise dissimilarities $\mathbf{D} = (d_{ij}) \in \mathbb{R}^{n\times n}$, the goal of MDS is to represent the objects as vectors in some Euclidean space, say $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^k$, such that

$$\|\mathbf{y}_i - \mathbf{y}_j\|_2 = d_{ij} \text{ (or as closely as possible)}, \quad \forall\, i, j$$

so as to visualize their proximity relationships and the global appearance.

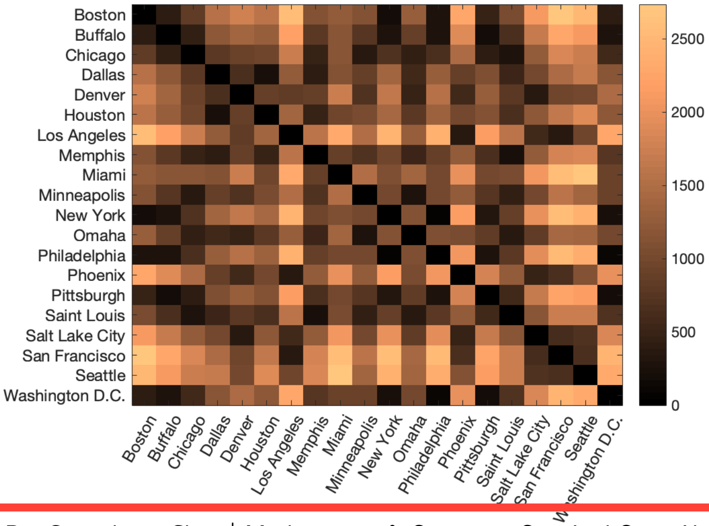


Multidimensional Scaling (MDS)

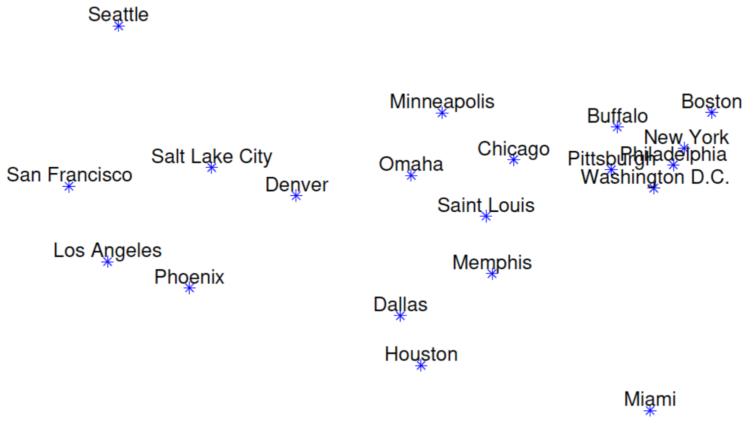
We illustrate the MDS problem with some examples.

Example 0.1. Given the distances between 20 cities in the U.S., display them on a (two-dimensional) map to preserve, as closely as possible, all the distances.

Multidimensional Scaling (MDS)

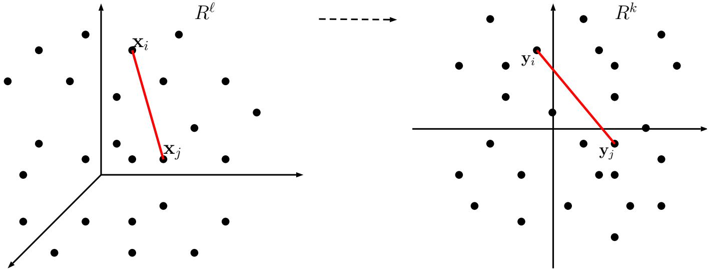


Multidimensional Scaling (MDS)



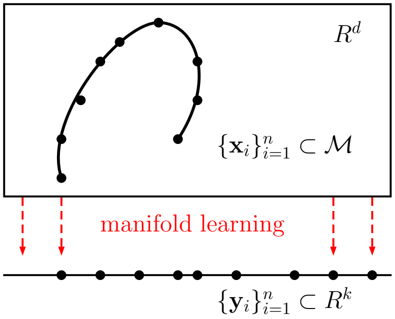
Multidimensional Scaling (MDS)

Example 0.2 (MDS as a dimension reduction method). Suppose we are given points in a very high dimensional space $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^\ell$ (e.g., images, documents) with some kind of distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$. We would like to find low dimensional representations, $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^k$ for some $k < \ell$, which can (approximately) preserve the given distances.



Multidimensional Scaling (MDS)

Example 0.3 (ISOmap). If the high dimensional data points are known to lie on a manifold (curve, surface, etc.), then one can try to preserve the geodesic distance (shortest distance along the manifold) in a low-dimensional Euclidean space.


R^d
$\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{M}$
manifold learning
$\{\mathbf{y}_i\}_{i=1}^n \subset R^k$

Mathematical setup and derivation

To solve the MDS problem, first observe that the solutions are not unique, as any translation of the new points have the pairwise distances:

$$\|(\mathbf{y}_i + \mathbf{c}) - (\mathbf{y}_j + \mathbf{c})\|_2 = \|\mathbf{y}_i - \mathbf{y}_j\|_2 = d_{ij}, \quad \text{for all } i, j$$

We can remove the translational invariance by adding a constraint

$$\sum_{i=1}^{n} \mathbf{y}_i = \mathbf{0}.$$

and solve it together with the MDS equations

$$\|\mathbf{y}_i - \mathbf{y}_j\|_2 = d_{ij} \quad \text{for all } i, j$$

Multidimensional Scaling (MDS)

First, we square the above equations and expand them to get

$$d_{ij}^2 = \|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 - 2\mathbf{y}_i^T\mathbf{y}_j$$

Summing over i and j separately gives that

$$\begin{aligned}
\sum_i d_{ij}^2 &= \sum_i \|\mathbf{y}_i\|^2 + \sum_i \|\mathbf{y}_j\|^2 - 2\sum_i \mathbf{y}_i^T\mathbf{y}_j \\
&= \sum_i \|\mathbf{y}_i\|^2 + n\|\mathbf{y}_j\|^2 \\
\sum_j d_{ij}^2 &= \sum_j \|\mathbf{y}_i\|^2 + \sum_j \|\mathbf{y}_j\|^2 - 2\sum_j \mathbf{y}_i^T\mathbf{y}_j \\
&= n\|\mathbf{y}_i\|^2 + \sum_j \|\mathbf{y}_j\|^2
\end{aligned}$$

Multidimensional Scaling (MDS)

Let $\mathbf{D}_2 = (d_{ij}^2)$, and denote its column, row and overall sums by

$$d_{\cdot j}^2 = \sum_i d_{ij}^2$$
$$d_{i\cdot}^2 = \sum_j d_{ij}^2$$
$$d_{\cdot\cdot}^2 = \sum_i \sum_j d_{ij}^2$$

We can rewrite the equations as

$$d_{\cdot j}^2 = \sum_i \|\mathbf{y}_i\|^2 + n\|\mathbf{y}_j\|^2$$
$$d_{i\cdot}^2 = n\|\mathbf{y}_i\|^2 + \sum_j \|\mathbf{y}_j\|^2$$

Multidimensional Scaling (MDS)

We continue to sum them up (over j, i respectively) to obtain that

$$d_{\cdot\cdot}^2 = n\sum_i \|\mathbf{y}_i\|^2 + n\sum_j \|\mathbf{y}_j\|^2 = 2n\sum_t \|\mathbf{y}_t\|^2$$

This implies that

$$\sum_t \|\mathbf{y}_t\|^2 = \frac{1}{2n} d_{\cdot\cdot}^2$$

Plugging it back, we then find

$$\|\mathbf{y}_j\|^2 = \frac{1}{n} d_{\cdot j}^2 - \frac{1}{2n^2} d_{\cdot\cdot}^2$$
$$\|\mathbf{y}_i\|^2 = \frac{1}{n} d_{i\cdot}^2 - \frac{1}{2n^2} d_{\cdot\cdot}^2$$

Multidimensional Scaling (MDS)

and finally

$$\mathbf{y}_i^T \mathbf{y}_j = \underbrace{\frac{1}{2}\left(\frac{1}{n} d_{i\cdot}^2 + \frac{1}{n} d_{\cdot j}^2 - \frac{1}{n^2} d_{\cdot\cdot}^2 - d_{ij}^2\right)}_{:= g_{ij}}, \quad \forall\, i, j$$

Let

- $\mathbf{G} = (g_{ij}) \in \mathbb{R}^{n\times n}$ (with g_{ij} defined above): gram matrix
- $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T \in \mathbb{R}^{n\times k}$: embedding matrix

Then the last equation may be rewritten as

$$\mathbf{Y}\mathbf{Y}^T = \mathbf{G}.$$

Properties of the $\mathbf{G}$ matrix:

- We have the following matrix representation for $\mathbf{G}$:

$$\mathbf{G} = -\frac{1}{2}\mathbf{C}\mathbf{D}_2\mathbf{C}, \quad \text{where} \quad \mathbf{C} = \mathbf{I}_n - \frac{1}{n}\mathbf{J}_n.$$

Verify:

$$\begin{aligned}\mathbf{C}\mathbf{D}_2\mathbf{C} &= \left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\mathbf{D}_2\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\\ &= \underbrace{\mathbf{D}_2}_{d_{ij}^2} - \frac{1}{n}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{D}_2}_{(d_{\cdot j}^2)} - \frac{1}{n}\underbrace{\mathbf{D}_2\mathbf{1}}_{(d_{i\cdot}^2)}\mathbf{1}^T + \frac{1}{n^2}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{D}_2\mathbf{1}}_{d_{\cdot\cdot}^2}\mathbf{1}^T = -2\mathbf{G}.\end{aligned}$$

Multidimensional Scaling (MDS)

Remark. The above representation is mathematically convenient for studying the properties of $\mathbf{G}$, but should be avoided for computing $\mathbf{G}$ when n is large.

The reason is that it involves matrix multiplications between $n \times n$ matrices and has an overall complexity of $\mathcal{O}(n^3)$.

The original formula is computationally more efficient, as it involves only row and columns operations on $\mathbf{G}$ and has a complexity of $\mathcal{O}(n^2)$:

$$g_{ij} = \frac{1}{2}\left(\frac{1}{n}d_{i\cdot}^2 + \frac{1}{n}d_{\cdot j}^2 - \frac{1}{n^2}d_{\cdot\cdot}^2 - d_{ij}^2\right), \quad \forall\, i, j$$

Multidimensional Scaling (MDS)

- If $\mathbf{D}_2$ is symmetric, then $\mathbf{G}$ is also symmetric.
- All the rows (and columns) of $\mathbf{C}$ sum to zero (because $\mathbf{C}$ is a centering matrix):

$$\mathbf{C1} = \left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = \mathbf{I}_n\mathbf{1} - \frac{1}{n}\mathbf{1}\,\underbrace{\mathbf{1}^T\mathbf{1}}_{n} = \mathbf{1} - \mathbf{1} = \mathbf{0}.$$

As a result, all the rows (and columns) of $\mathbf{G}$ sum to zero as well:

$$\mathbf{G1} = -\frac{1}{2}\mathbf{C}\mathbf{D}_2\,\underbrace{\mathbf{C1}}_{\mathbf{0}} = \mathbf{0}$$

This also indicates that $\mathbf{G}$ must have an eigenvalue of 0.

Multidimensional Scaling (MDS)

Remark. The solution of the MDS problem ($\mathbf{Y}\mathbf{Y}^T = \mathbf{G}$) if it exists, is still not unique. The reason is that any rotation of $\mathbf{Y}$, i.e., $\mathbf{Y}\mathbf{Q}$ for some orthogonal matrix $\mathbf{Q}$, is also a solution:

$$(\mathbf{Y}\mathbf{Q})(\mathbf{Y}\mathbf{Q})^T = \mathbf{Y}\mathbf{Q}\mathbf{Q}^T\mathbf{Y}^T = \mathbf{Y}\mathbf{Y}^T = \mathbf{G}.$$

In practice, we only need to find one solution to represent the given data in a Euclidean space.

Remark. In order for a solution to exist, $\mathbf{G}$ must be positive semidefinite. In this case, we can write

$$\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}^T = \left(\mathbf{U}\mathbf{\Lambda}^{1/2}\right)\left(\mathbf{U}\mathbf{\Lambda}^{1/2}\right)^T$$

from which we can obtain the following result.

Result

Theorem 0.1. If the matrix $\mathbf{G} = -\frac{1}{2}\mathbf{C}\mathbf{D}_2\mathbf{C}$ is positive semidefinite and $k \geq r = \operatorname{rank}(\mathbf{G})$, then the MDS problem has the following exact solution

$$\mathbf{Y} = \mathbf{U}_k \mathbf{\Lambda}_k^{1/2} = [\sqrt{\lambda_1}\mathbf{u}_1, \ldots, \sqrt{\lambda_r}\mathbf{u}_r, \underbrace{\sqrt{\lambda_{r+1}}\mathbf{u}_{r+1}}_{=\mathbf{0}}, \ldots, \underbrace{\sqrt{\lambda_k}\mathbf{u}_k}_{=\mathbf{0}}] \in \mathbb{R}^{n\times k}$$

where $(\lambda_i, \mathbf{u}_i)$ are the eigenpairs of the $\mathbf{G}$ matrix.

Remark. The "smallest" exact solution is when $k = r$:

$$\mathbf{Y}_r = \mathbf{U}_r \mathbf{\Lambda}_r^{1/2} = \begin{bmatrix} \sqrt{\lambda_1}\mathbf{u}_1 & \cdots & \sqrt{\lambda_r}\mathbf{u}_r \end{bmatrix}$$

Connection to PCA

In the special setting in which the objects are Euclidean vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^\ell$, and the input dissimilarities are Euclidean distances, $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, MDS as a dimension reduction approach is identical to PCA.

To see this, first note that in this case we can repeat the steps of the mathematical derivation with the centered data $\widetilde{\mathbf{X}} = \mathbf{C}\mathbf{X}$,

$$\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\|_2 = d_{ij},\ \forall i, j, \qquad \text{and} \quad \sum_{i=1}^{n} \widetilde{\mathbf{x}}_i = \mathbf{0}$$

to obtain that

$$\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T = \mathbf{G}.$$

Multidimensional Scaling (MDS)

Thus, the matrix equation for the desired embedding matrix $\mathbf{Y} \in \mathbb{R}^{n\times k}$ becomes

$$\mathbf{Y}\mathbf{Y}^T = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T.$$

To solve for $\mathbf{Y}$, consider the compact SVD of $\widetilde{\mathbf{X}}$:

$$\widetilde{\mathbf{X}} = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T, \quad r = \operatorname{rank}(\widetilde{\mathbf{X}})$$

Plug it into the preceding equation to get

$$\mathbf{Y}\mathbf{Y}^T = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T\mathbf{V}_r\mathbf{\Sigma}_r\mathbf{U}_r^T = \mathbf{U}_r\mathbf{\Sigma}_r^2\mathbf{U}_r^T.$$

Multidimensional Scaling (MDS)

Clearly,

- if $k = r$, an exact solution is $\mathbf{Y}_r = \mathbf{U}_r \mathbf{\Sigma}_r$;
- otherwise, if $k < r$, the equation has no exact solution, but the "best" approximation (such that $\mathbf{Y}\mathbf{Y}^T$ is the closest to $\mathbf{G}$ under the Frobenius norm) would be $\mathbf{Y}_k = \mathbf{U}_k \mathbf{\Sigma}_k$, the matrix of first k principal components of the original data.

This also shows that **PCA tries to preserve, as much as possible, the pairwise Euclidean distances of vector data**.

Numerical issues and approximate solutions

In this section we address some numerical issues that may arise in MDS and meanwhile study cases where **approximate solutions** are preferred (even when the exact solution exists):

- The eigenvalues of $\mathbf{G}$ have a rapid decay
- $\mathbf{G}$ is not symmetric or positive semidefinite

Multidimensional Scaling (MDS)

Case 1: The eigenvalues of the matrix $\mathbf{G}$ have a fast decay, i.e.,

$$\lambda_1 \geq \cdots \geq \lambda_{r_0} \gg \lambda_{r_0+1} \geq \cdots \geq \lambda_r > 0$$

where $r = \operatorname{rank}(\mathbf{G})$ and $r_0 < r$ is a positive integer representing the number of dominant eigenvalues of $\mathbf{G}$.

In such a case, we can truncate the columns of

$$\mathbf{Y}_r = \left[\sqrt{\lambda_1}\mathbf{u}_1 \ \cdots \ \sqrt{\lambda_{r_0}}\mathbf{u}_{r_0} \ \sqrt{\lambda_{r_0+1}}\mathbf{u}_{r_0+1} \ \cdots \ \sqrt{\lambda_r}\mathbf{u}_r\right] = \mathbf{U}_r\mathbf{\Lambda}_r^{1/2} \in \mathbb{R}^{n\times r}$$

to obtain an approximate solution to the MDS equation

$$\mathbf{Y}_{r_0} = \left[\sqrt{\lambda_1}\mathbf{u}_1 \quad \cdots \quad \sqrt{\lambda_{r_0}}\mathbf{u}_{r_0}\right] = \mathbf{U}_{r_0}\mathbf{\Lambda}_{r_0}^{1/2} \in \mathbb{R}^{n\times r_0}$$

and use it as a configuration of the original objects.

Multidimensional Scaling (MDS)

There are two different ways to measure the quality of the approximation of $\mathbf{Y}_r$ by $\mathbf{Y}_{r_0}$.

First, we can directly examine the ratio of their squared magnitudes, i.e.,

$$\frac{\|\mathbf{Y}_{r_0}\|_F^2}{\|\mathbf{Y}_r\|_F^2} = \frac{\operatorname{trace}\left(\mathbf{Y}_{r_0}\mathbf{Y}_{r_0}^T\right)}{\operatorname{trace}\left(\mathbf{Y}_r\mathbf{Y}_r^T\right)} = \frac{\operatorname{trace}\left(\mathbf{U}_{r_0}\mathbf{\Lambda}_{r_0}\mathbf{U}_{r_0}^T\right)}{\operatorname{trace}\left(\mathbf{U}_r\mathbf{\Lambda}_r\mathbf{U}_r^T\right)} = \frac{\lambda_1 + \cdots + \lambda_{r_0}}{\lambda_1 + \cdots + \lambda_r}.$$

The approximation is consider to be good if this fraction is close to 1.

In the special setting of Euclidean vector data and Euclidean distances (where MDS is identical to PCA), this criterion coincides with the relative amount of scatter preserved by PCA, because the eigenvalues of $\mathbf{G} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T$ are the squared singular values of the centered data matrix $\widetilde{\mathbf{X}}$, i.e., $\lambda_i = \sigma_i^2$ for all $1 \leq i \leq r$.

Multidimensional Scaling (MDS)

The second measure is the *Kruskal stress*,

$$\text{Stress} = \frac{\|\mathbf{D} - \widehat{\mathbf{D}}\|_F}{\|\mathbf{D}\|_F}.$$

which examines the relative approximation error of $\mathbf{D}$ by the Euclidean distances of the solution $\mathbf{Y}$,

$$\widehat{\mathbf{D}} = (\hat{d_{ij}}) \in \mathbb{R}^{n\times n}, \quad \hat{d_{ij}} = \|\mathbf{y}_i - \mathbf{y}_j\|$$

This is more general measure of the goodness of the approximate solution $\mathbf{Y}_{r_0}$, and can be used in other settings. Empirically,

- the fit is good if stress < 0.1, and
- unacceptable if stress > 0.15.

Multidimensional Scaling (MDS)

Case 2: The positive semidefiniteness or even the symmetry of $\mathbf{G}$ does not hold true, so an exact solution does not exist.

First, $\mathbf{D}$ may lose the exact symmetry because the dissimilarity measure used is not symmetric, or there is noise in the entries of $\mathbf{G}$. In those cases, we can apply the following symmetry correction:

$$\mathbf{D} \longleftarrow \frac{1}{2}\left(\mathbf{D} + \mathbf{D}^T\right).$$

Second, the matrix $\mathbf{G}$ may be symmetric but not positive semidefinite. In fact, $\mathbf{G} \in S_+^n(\mathbb{R})$ if and only if $\mathbf{D}$ corresponds to Euclidean geometry, i.e., Euclidean distances on Euclidean vectors.

Multidimensional Scaling (MDS)

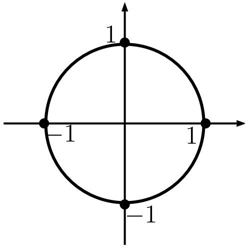
We give an example of non-Euclidean geometry here. Consider a collection of four points on the unit circle in $\mathbb{R}^2$

$$\mathbf{x}_1 = (1,0)^T, \quad \mathbf{x}_2 = (0,1)^T, \quad \mathbf{x}_3 = (-1,0)^T, \quad \mathbf{x}_4 = (0,-1)^T$$

and for any pair of points, their distance is defined as the length of the shortest path between them along the circle.

The pairwise dissimilarity matrix is

$$\mathbf{D} = \begin{pmatrix} 0 & \frac{\pi}{2} & \pi & \frac{\pi}{2} \\ \frac{\pi}{2} & 0 & \frac{\pi}{2} & \pi \\ \pi & \frac{\pi}{2} & 0 & \frac{\pi}{2} \\ \frac{\pi}{2} & \pi & \frac{\pi}{2} & 0 \end{pmatrix}.$$



Multidimensional Scaling (MDS)

It follows that

$$\mathbf{G} = -\frac{1}{2}\mathbf{C}\,\mathbf{D}_2\,\mathbf{C} = \frac{\pi^2}{16}\begin{pmatrix} 3 & 1 & -5 & 1 \\ 1 & 3 & 1 & -5 \\ -5 & 1 & 3 & 1 \\ 1 & -5 & 1 & 3 \end{pmatrix}$$

This matrix is symmetric but not positive semidefinite because $\mathbf{G}$ has a negative eigenvalue:

$$\lambda_1 = \lambda_2 = \frac{\pi^2}{2},\ \lambda_3 = 0,\ \lambda_4 = -\frac{\pi^2}{4}.$$

What can we do when $\mathbf{G}$ is symmetric but not positive semidefinte because $\mathbf{D}$ corresponds to non-Euclidean geometry?

Multidimensional Scaling (MDS)

A simple approach would be to ignore the negative eigenvalue(s) of $\mathbf{G}$ and focus on the corresponding eigenvectors of the positive eigenvalues.

That is, letting $\lambda_i, 1 \le i \le r^+$ be the only positive eigenvalues of $\mathbf{G}$ with corresponding unit-vector eigenvectors $\mathbf{v}_i, 1 \le i \le r^+$, we use

$$\mathbf{Y}_{r^+} = \begin{bmatrix} \sqrt{\lambda_1}\mathbf{v}_1 & \cdots & \sqrt{\lambda_{r^+}}\mathbf{v}_{r^+} \end{bmatrix}$$

as a low-dimensional embedding of the data.

Such an approximate solution satisfies

$$\mathbf{Y}_{r^+}\left(\mathbf{Y}_{r^+}\right)^T = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \cdots + \lambda_{r^+} \mathbf{v}_{r^+}\left(\mathbf{v}_{r^+}\right)^T.$$

which is the positive component of the symmetric matrix $\mathbf{G}$ and also the closest positive semidefinite matrix to $\mathbf{G}$ under the Frobenius norm.

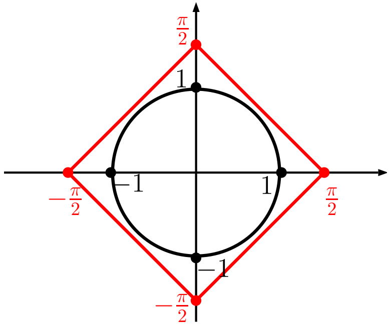
Multidimensional Scaling (MDS)

In the above example, we would just focus on $\lambda_1 = \lambda_2 = \frac{\pi^2}{2}$ and corresponding eigenvectors

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}}(1, 0, -1, 0)^T, \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}}(0, 1, 0, -1,)^T$$

and use them to form an embedding matrix

$$\mathbf{Y}_2 = \begin{bmatrix} \sqrt{\lambda_1}\mathbf{v}_1 & \sqrt{\lambda_2}\mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{2} & 0 \\ 0 & \frac{\pi}{2} \\ -\frac{\pi}{2} & 0 \\ 0 & -\frac{\pi}{2} \end{bmatrix}.$$



Multidimensional Scaling (MDS)

The embedded points have the following Euclidean distances:

$$\widehat{\mathbf{D}} = \begin{pmatrix} 0 & \frac{\pi}{\sqrt{2}} & \pi & \frac{\pi}{\sqrt{2}} \\ \frac{\pi}{\sqrt{2}} & 0 & \frac{\pi}{\sqrt{2}} & \pi \\ \pi & \frac{\pi}{\sqrt{2}} & 0 & \frac{\pi}{\sqrt{2}} \\ \frac{\pi}{\sqrt{2}} & \pi & \frac{\pi}{\sqrt{2}} & 0 \end{pmatrix}$$

To evaluate the quality of the MDS configuration, we compute the Kruskal stress

$$\text{stress} = \frac{\|\mathbf{D} - \widehat{\mathbf{D}}\|_F}{\|\mathbf{D}\|_F} = \frac{\sqrt{2}-1}{\sqrt{3}} = .2391$$

This indicates that the MDS embedding cannot adequately capture the nonlinear geometry.

The (classical) MDS algorithm

Input: Matrix of squared pairwise distances $\mathbf{D}_2 \in \mathbb{R}^{n\times n}$ and integer $k \geq 1$

Output: A k-dimensional vector representation of the objects: $\mathbf{Y} \in \mathbb{R}^{n\times k}$.

Steps:

1. Compute the matrix $\mathbf{G} = -\frac{1}{2}\mathbf{C}\mathbf{D}_2\mathbf{C}$ (see footnote[1])
2. Find the top-k eigenvectors of $\mathbf{G}$: $\mathbf{G} \approx \mathbf{U}_k\mathbf{\Lambda}_k\mathbf{U}_k^T$
3. Form the embedding matrix $\mathbf{Y} = \mathbf{U}_k\mathbf{\Lambda}_k^{1/2}$.

[1] Do not use this formula to compute $\mathbf{G}$; this is only for mathematical convenience.

MATLAB implementation of MDS

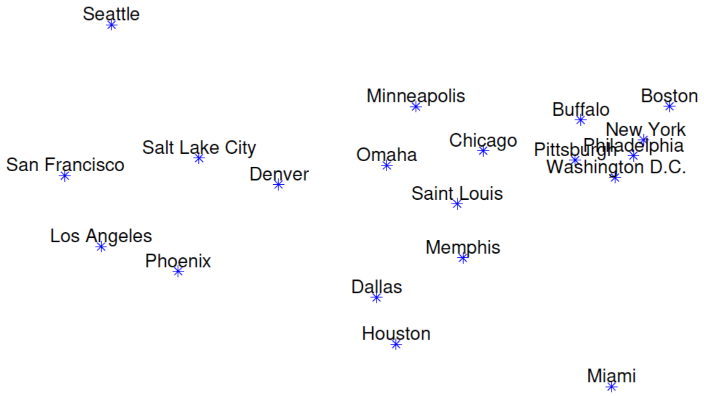
cmdscale Classical Multidimensional Scaling.

$Y = cmdscale(D)$;
% D is an $n \times n$ distance matrix (not squared!);
% Y is the final configuration matrix of size $n \times r$

$[Y, e] = cmdscale(D, p)$;
% p specifies the dimensionality of the desired embedding Y
% e contains the largest p eigenvalues of the Gram matrix $(\mathbf{G})$ that correspond to Y

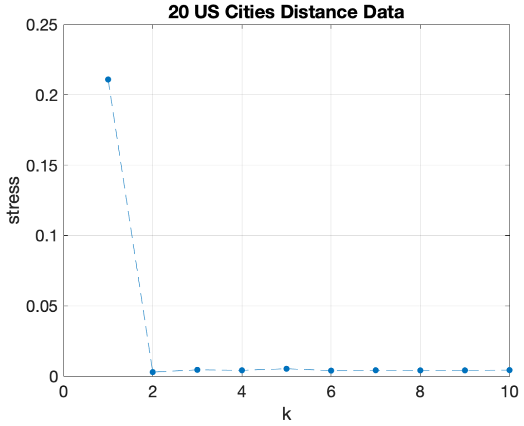
Multidimensional Scaling (MDS)

Example 0.4 (Mapping of the 20 US cities).



Multidimensional Scaling (MDS)

The stress of the 2-D representation of the 20 US cities data is 0.0029.



Multidimensional Scaling (MDS)

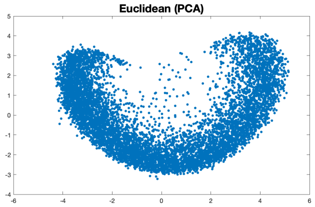
Example 0.5 (MNIST handwritten digit 1)**.** Apply MDS with ℓ_1 metric to embed the images of 1 into 2 dimensions.

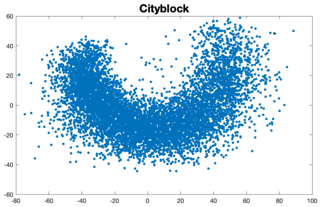
MATLAB script:

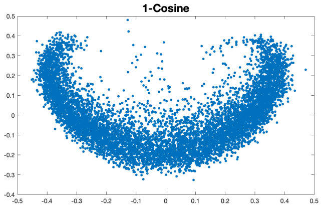
```
% digits1 is a 7877-by-784 matrix containing all the 1's

Y = cmdscale(pdist(digits1, 'cityblock'), 2);
```

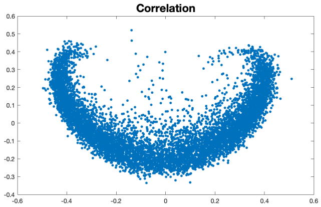
(picture on next slide)

Multidimensional Scaling (MDS)









Further reading

A book chapter on MDS[2]

- Classical MDS (covered in this lecture): preserve the input distances
- Ordinal MDS (not covered): preserve only the rank order

[2] `http://www.bristol.ac.uk/media-library/sites/cmm/migrated/documents/chapter3.pdf`