San José State University

Math 253: Mathematical Methods for Data Visualization

Multidimensional Scaling (MDS)

Dr. Guangliang Chen

The MDS problem

Assume a collection of n objects with pairwise distances $\{\ell_{ij}\}_{1\le i,j\le n}$. Represent them as points in some Euclidean space, say $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^k$, such that

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

Remark. Objects can be images, documents, humans, etc., as along as there is a well-defined distance metric on them. The goal of MDS is to represent them as vectors in some Euclidean space (and to visualize their proximity relationships as well as the global appearance).

Multidimensional Scaling (MDS)

Remark. Possible distance metrics that can be used by MDS:

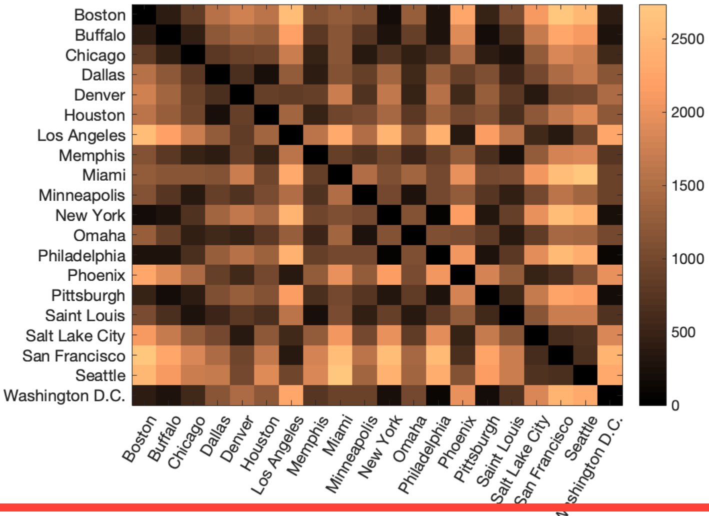
- Euclidean distance (ℓ_2)
- Manhattan/Cityblock distance (ℓ_1)
- Chebyshev/maximum coordinate difference (ℓ_∞)
- Minkowski distance (ℓ_p)
- Cosine of the angle: $\|\frac{\mathbf{x}}{\|\mathbf{x}\|} - \frac{\mathbf{y}}{\|\mathbf{y}\|}\|^2 = 2 - 2\cos\theta$
- Geodesic distance (along curved dimensions)

Multidimensional Scaling (MDS)

We illustrate the MDS problem with some examples.

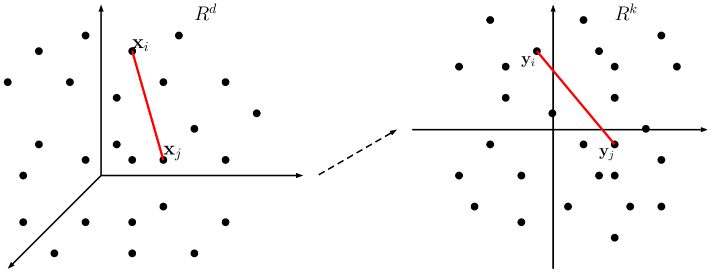
Example 0.1. Given the distances between 20 cities in the U.S., display them on a (two-dimensional) map to preserve, as closely as possible, all the distances.

Multidimensional Scaling (MDS)



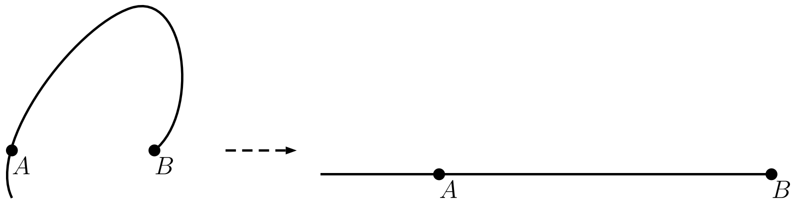
Multidimensional Scaling (MDS)

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



Multidimensional Scaling (MDS)

Example 0.3. If the points are known to lie on a manifold (curve, surface, etc.), then one can use geodesic distance (shortest distance along the manifold) and try to preserve them in a low-dimensional Euclidean space. This is called the **manifold learning** problem.



Mathematical setup and derivation

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

We can remove the translational invariance by adding a constraint

$$\sum \mathbf{y}_i = \mathbf{0}.$$

Assuming the equations

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

have a solution, we solve them together with the constraint.

Multidimensional Scaling (MDS)

We square the above equations and expand them to get

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

Summing over i and j separately gives that

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

Multidimensional Scaling (MDS)

Denoting by

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

we can rewrite the equations as

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

Multidimensional Scaling (MDS)

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

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

This implies that

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

Plugging back we then find

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

Multidimensional Scaling (MDS)

and finally

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

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:

- If $\mathbf{L}$ is symmetric, then $\mathbf{G}$ is also symmetric. This is because

$$\mathbf{G} = -\frac{1}{2}\mathbf{JLJ}, \quad \text{where} \quad \mathbf{L} = (\ell_{ij}^2),\ \mathbf{J} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T.$$

Verify:

$$\mathbf{JLJ} = \underbrace{\mathbf{L}}_{\ell_{ij}} - \frac{1}{n}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{L}}_{(\ell_{\cdot j}^2)} - \frac{1}{n}\underbrace{\mathbf{L}\mathbf{1}}_{(\ell_{i\cdot}^2)}\mathbf{1}^T + \frac{1}{n^2}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{L}\mathbf{1}}_{\ell_{\cdot\cdot}^2}\mathbf{1}^T = -2\mathbf{G}.$$

- All the rows (and columns) of $\mathbf{G}$ sum to zero, due to

$$\mathbf{J1} = \mathbf{I}_n\mathbf{1} - \frac{1}{n}\mathbf{1}\underbrace{\mathbf{1}^T\mathbf{1}}_{n} = \mathbf{1} - \mathbf{1} = \mathbf{0}.$$

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

Multidimensional Scaling (MDS)

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

$$(\mathbf{YQ})(\mathbf{YQ})^T = \mathbf{YQQ}^T\mathbf{Y}^T = \mathbf{YY}^T = \mathbf{G}.$$

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

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

$$\mathbf{G} = \mathbf{U\Lambda U}^T = \mathbf{U\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}^T$$

from which we can obtain the following result.

Multidimensional Scaling (MDS)

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

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

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

Remark. If the eigenvalues decay rapidly, we can truncate the columns to obtain an approximate solution $\longleftarrow$ more useful

$$\mathbf{Y}_{r_0} \approx [\sqrt{\lambda_1}\mathbf{u}_1 \ldots \sqrt{\lambda_{r_0}}\mathbf{u}_{r_0}] = \mathbf{U}_{r_0}\Lambda_{r_0}^{1/2} \in \mathbb{R}^{n\times r_0} \quad \longleftarrow \text{embedding}$$

where $r_0 < r$ is the number of dominant (or chosen) eigenvalues.

Multidimensional Scaling (MDS)

Remark. In the setting of vector data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ with the goal of preserving Euclidean distances, the MDS approach is equivalent to PCA.

To see this, note that in this case

$$\mathbf{G} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T,$$

where we used the SVD of $\widetilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Clearly, if $k \geq r$, an exact solution of $\mathbf{Y}\mathbf{Y}^T = \mathbf{G}$ is $\mathbf{Y}_k = \mathbf{U}_k\mathbf{\Sigma}_k$. Otherwise, if $k < r$, it is the “best” approximation (such that $\mathbf{Y}\mathbf{Y}^T$ is the closest to $\mathbf{G}$).

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

The (classical) MDS algorithm

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

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

Steps:

1. Compute the matrix $\mathbf{G} = -\frac{1}{2}\mathbf{JLJ}$ (see footnote below[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

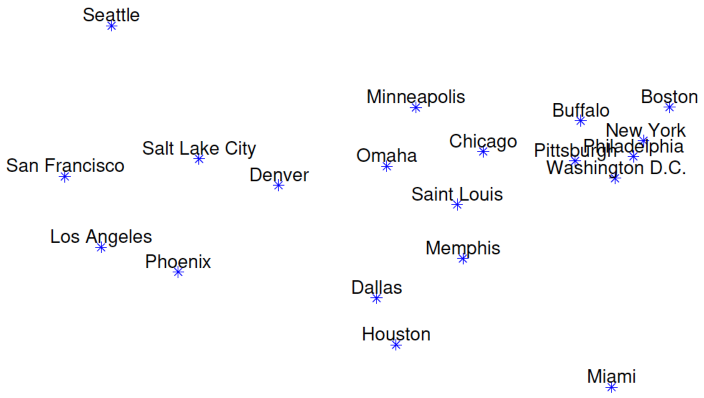
cmdscale Classical Multidimensional Scaling.

Y = cmdscale(D) % D is an n-by-n distance matrix (not squared!)

[Y,e] = cmdscale(D, p) % p specifies the dimensionality of the desired embedding Y

Multidimensional Scaling (MDS)

Example 0.4 (Mapping of the 20 US cities).



Multidimensional Scaling (MDS)

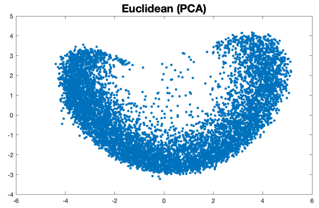
Example 0.5 (MNIST handwritten digit 1)**.** Apply MDS with ℓ_1 metric to embed the images of 1 into 2 dimensions.

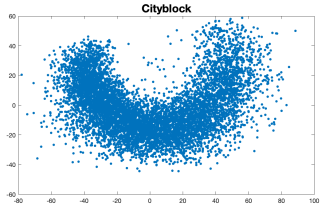
MATLAB script:

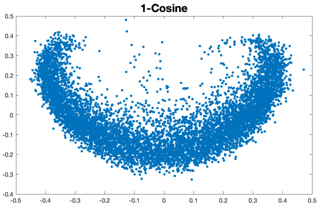
```
% digits1 is a 7877-by-784 matrix containing all the 1's
Y = cmdscale(pdist(digits1, 'cityblock'), 2);
```

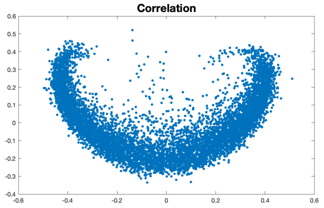
(picture on next slide)

Multidimensional Scaling (MDS)









Multidimensional Scaling (MDS)

In general, to select k and evaluate the quality of approximation by MDS, one can use the following measure.

Def 0.1. The **Kruskal stress** is defined as

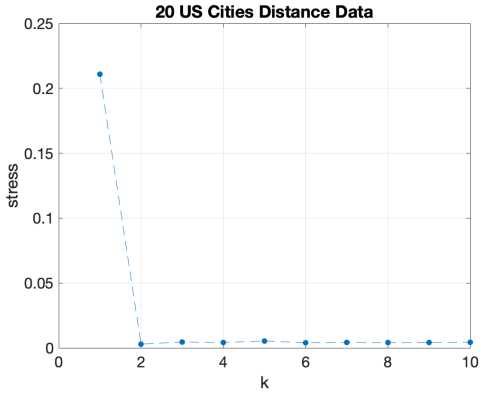
$$\text{Stress} = \sqrt{\frac{\sum_{i,j}(\ell_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|_2)^2}{\sum_{i,j} \ell_{ij}^2}}.$$

Empirically,

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

Multidimensional Scaling (MDS)

Example 0.6. The stress of the 2-D representation of the 20 US cities data is 0.0029.



Further reading

A book chapter on MDS[2]

- Classical MDS (just covered in class): preserve the input distances
- Ordinal MDS: preserve only the rank order

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