Multidimensional scaling

Lecture 5
Cinderella 2.0
If it doesn’t fit, you must acquit!

A pivotal point of the trial: O.J. Simpson is unable to put on the glove

A glove allegedly dropped by Simpson at the crime scene
**Metric model**

Shape = metric space $(X, d)$

Similarity = isometry w.r.t. $d$

**EXTRINSIC GEOMETRY**

- Euclidean metric $d_{\mathbb{R}^3}\mid_X$
- Invariant to rigid motion

Extrinsic geometry is not invariant under inelastic deformations
Numerical geometry of non-rigid shapes  Multidimensional scaling

**Metric model**

Shape = metric space \((X, d)\)

Similarity = isometry w.r.t. \(d\)

**EXTRINSIC GEOMETRY**
- Euclidean metric \(d_{\mathbb{R}^3}|_X\)
- Invariant to rigid motion

**INTRINSIC GEOMETRY**
- Geodesic metric \(d_X\)
- Invariant to inelastic deformation
Intrinsic vs. extrinsic similarity

EXTRINSIC SIMILARITY
Part of the same metric space

INTRINSIC SIMILARITY
Two different metric spaces

SOLUTION: Find a representation of $(X, d_X)$ and $(Y, d_Y)$ in a common metric space
Canonical forms

Isometric embedding

\[ d_X = d_{\mathbb{R}^3} \circ (\varphi \times \varphi) \]
Canonical form distance

\[ d_{\text{int}}(X, Y) = d_{\text{ext}}(\varphi(X), \psi(Y)) \]
Mapmaker’s problem

\[ d_S = d_{\mathbb{R}^2} \]
Mapmaker’s problem

A sphere has non-zero curvature, therefore, it is not isometric to the plane (a consequence of Theorema egregium)

\[ d_S \neq d_{\mathbb{R}^2} \]

Karl Friedrich Gauss
(1777-1855)
**Linial’s example**

Conclusion: generally, isometric embedding does not exist!

No contradiction to Nash embedding theorem: Nash guarantees that any Riemannian structure can be realized as a **length metric** induced by Euclidean metric. We try to realize it using **restricted Euclidean metric**
Minimum distortion embedding

$$\min_{\varphi: X \to \mathbb{R}^3} \left\| d_X - d_{\mathbb{R}^3} \circ (\varphi \times \varphi) \right\|$$
Multidimensional scaling

Find an embedding into $\mathbb{R}^m$ that **distorts the distances the least** by solving the optimization problem

$$\{z_1^*, ..., z_N^*\} = \underset{\{z_1, ..., z_N\} \subset \mathbb{R}^m}{\text{argmin}} \sigma(z_1, ..., z_N)$$

where $z_i = f(x_i)$ are the coordinates of the canonical form

- The function $\sigma$ measuring the distortion of distances is called **stress**
- The problem is called **multidimensional scaling (MDS)**
Stress

The stress is a function of the canonical form coordinates \( \{z_1, \ldots, z_N\} \) and the distances \( d_X(x_i, x_j) \)

- \( L_2 \)-stress

\[
\sigma_2(z_1, \ldots, z_N) = \sum_{i > j} |d_{\mathbb{R}m}(z_i, z_j) - d_X(x_i, x_j)|^2
\]

- \( L_p \)-stress

\[
\sigma_p(z_1, \ldots, z_N) = \sum_{i > j} |d_{\mathbb{R}m}(z_i, z_j) - d_X(x_i, x_j)|^p
\]

- \( L_\infty \)-stress (distortion)

\[
\sigma_\infty(z_1, \ldots, z_N) = \max_{i, j=1, \ldots, N} |d_{\mathbb{R}m}(z_i, z_j) - d_X(x_i, x_j)|
\]
Matrix expression of $L_2$-stress

Some notation:

- $Z$ - an $N \times m$ matrix of canonical form coordinates (each row corresponds to a point)
- Shorthand notation for Euclidean distances $d_{ij}(Z) = d_{\mathbb{R}^m}(z_i, z_j)$

Write the stress as

$$\sigma_2(Z) = \sum_{i>j} |d_{ij}(Z) - d_X(x_i, x_j)|^2$$

$$= \sum_{i>j} \left( d_{ij}^2(Z) - 2d_{ij}(Z)d_X(x_i, x_j) + d_X^2(x_i, x_j) \right)$$
Term 1

\[
\sum_{i > j} d_{ij}^2(Z) = \sum_{i > j} \sum_{k=1}^{m} (z_i^k - z_j^k)^2
\]

\[
= \sum_{i > j} \sum_{k=1}^{m} (z_i^k)^2 - 2z_i^k z_j^k + (z_j^k)^2
\]

\[
= \sum_{i > j} \langle z_i, z_i \rangle + \langle z_j, z_j \rangle - 2\langle z_i, z_j \rangle
\]

\[
= (N - 1) \sum_{i=1}^{N} \langle z_i, z_i \rangle - \left( \sum_{i, j=1}^{N} \langle z_i, z_j \rangle - \sum_{i=1}^{N} \langle z_i, z_i \rangle \right)
\]

\[
= N \sum_{i=1}^{N} \langle z_i, z_i \rangle - \sum_{i, j=1}^{N} \langle z_i, z_j \rangle
\]
Term 1

\[ \sum_{i>j} d_{ij}^2(Z) = N \sum_{i=1}^{N} \langle z_i, z_i \rangle - \sum_{i,j=1}^{N} \langle z_i, z_j \rangle \]

\[ = \text{trace}(ZZ^\top) \]

\[ = \text{trace}(VZZ^\top) \]

\[ = \text{trace}(ZVZ^\top) \]

where \( V \) is an \( N \times N \) matrix with elements

\[ v_{ij} = \begin{cases} 
-1 & i \neq j \\
 N - 1 & i = j 
\end{cases} \]
Term 2

\[
\sum_{i>j} d_{ij}(Z) d_X(x_i, x_j) = \sum_{i>j} d_X(x_i, x_j) \left( \sum_{k=1}^{m} (z_i^k - z_j^k)^2 \right)^{1/2}
\]

\[
= \sum_{i>j} d_X(x_i, x_j) d_{ij}^{-1}(Z) \sum_{k=1}^{m} (z_i^k - z_j^k)^2
\]

For \( i \neq j \) and zero otherwise

\[
= \sum_{i>j} d_X(x_i, x_j) d_{ij}^{-1}(Z) \left( \langle z_i, z_i \rangle + \langle z_j, z_j \rangle - 2 \langle z_i, z_j \rangle \right)
\]

\[
= \sum_{i,j=1}^{N} d_X(x_i, x_j) d_{ij}^{-1}(Z) \left( \langle z_i, z_i \rangle - \langle z_i, z_j \rangle \right)
\]
Term 2

$$\sum_{i>j} d_{ij}(Z) d_X(x_i, x_j) = \sum_{i,j=1}^N d_X(x_i, x_j) d_{ij}^{-1}(Z)(\langle z_i, z_i \rangle - \langle z_i, z_j \rangle)$$

$$= \text{trace}(B(Z)ZZ^\top) = \text{trace}(Z^\top B(Z)Z)$$

where $B(Z)$ is and $N \times N$ matrix-valued function with elements

$$b_{ij}(Z) = \begin{cases} 
-\frac{d_X(x_i, x_j)}{d_{ij}(Z)} & i \neq j, \ d_{ij}(Z) \neq 0 \\
0 & i \neq j, \ d_{ij}(Z) = 0 \\
-\gamma_{k \neq i} \ b_{ij}(Z) & i = j \end{cases}$$
LS-MDS

Minimization of $L_2$-stress is called **least squares MDS (LS-MDS)**

$$
\min_{Z \in \mathbb{R}^{N \times m}} \sum_{i > j} d_{i,j}^2(Z) - 2d_{i,j}(Z)d_X(x_i, x_j) + d_X^2(x_i, x_j) =
$$

$$
\min_{Z \in \mathbb{R}^{N \times m}} \text{tr}(Z^TVZ) - 2\text{tr}(Z^TB(Z)Z) + \sum_{i > j} d_X^2(x_i, x_j)
$$

- $Nm$ variables
- **Non-linear**
- **Non-convex** (this liable to local convergence)
- Optimum defined **up to Euclidean transformation**
Gradient of $L_2$-stress

\[
\nabla \sigma_2(Z) = \nabla \left( \text{tr}(Z^TVZ) - 2\text{tr}(Z^TB(Z)Z) + \sum_{i>j} d^2_{X}(x_i, x_j) \right)
\]

Quadratic in $Z$  Nonlinear in $Z$  Constant in $Z$

\[
= 2VZ - 2B(Z)Z
\]

Recall exercise in optimization
Complexity

- Stress computation complexity: $\mathcal{O}(mN^2)$
  
  $$\sigma_2(Z) = \text{trace}(Z^T V Z) - 2\text{trace}(Z^T B(Z) Z) + \sum_{i>j} d_{x_i x_j}^2$$

- Stress gradient computation complexity: $\mathcal{O}(mN^2)$
  
  $$\nabla \sigma_2(Z) = 2VZ - 2B(Z)Z$$

- Steepest descent
  
  $$Z^{(k+1)} = Z^{(k)} - \alpha^{(k)} \nabla \sigma_2(Z^{(k)})$$
  
  $$= Z^{(k)} - 2\alpha^{(k)} \left( VZ^{(k)} - B(Z^{(k)})Z^{(k)} \right)$$

  with line search may be prohibitively expensive (requires multiple evaluations of the stress and the gradient)
Observation

By Cauchy-Schwarz inequality

\[
\sum_{k=1}^{m} (z^k_i - z^k_j)(q^k_i - q^k_j) \leq \left( \sum_{k=1}^{m} (z^k_i - z^k_j)^2 \right)^{\frac{1}{2}} \left( \sum_{k=1}^{m} (q^k_i - q^k_j)^2 \right)^{\frac{1}{2}}
\]

\[
\langle z_i - z_j, q_i - q_j \rangle \\
|| z_i - z_j || \\
|| q_i - q_j || \\
= d_{ij}(Z)d_{ij}(Q)
\]

for all \( Q \in \mathbb{R}^{N \times m} \) and \( i, j = 1, \ldots, N \)

Equality is achieved for \( Q = Z \)

Write it differently: \( d_{ij}(Z) \geq \sum_{k=1}^{m} (z^k_i - z^k_j)(q^k_i - q^k_j)d_{ij}^{-1}(Q) \)
Observation

Consider the nonlinear term in the stress

\[
\text{trace}(Z^T B(Z)Z) = \sum_{i>j} d_{ij}(Z) d_X(x_i, x_j)
\]

\[
\geq \sum_{i>j} \sum_{k=1}^{m} (z_i^k - z_j^k)(q_i^k - q_j^k)d_{ij}^{-1}(Q)d_X(x_i, x_j)
\]

\[
= \text{trace}(Z^T B(Q)Q)
\]
Majorizing inequality

\[ \sigma_2(Z) = \text{trace}(Z^T V Z) - 2\text{trace}(Z^T B(Z) Z) + \sum_{i > j} d_X^2(x_i, x_j) \]

\[ \leq \text{trace}(Z^T V Z) - 2\text{trace}(Z^T B(Q) Q) + \sum_{i > j} d_X^2(x_i, x_j) \]

\[ = h(Z, Q) \]

Equality is achieved for \( Q = Z \)

[de Leeuw, 1977]

We have a quadratic majorizing function to use in iterative majorization algorithm!
Iterative majorization

Construct a majorizing function $h(Z, Q)$ satisfying

- $h(Q, Q) = \sigma_2(Q)$
- Majorizing inequality: $\sigma_2(Z) \leq h(Z, Q)$ for all $Z$
- $h(Z, Q)$ is convex
Iterative majorization

- Start with some $Q^{(0)}$
- Find $Z^{(k+1)}$ such that
  \[ Z^{(k+1)} = \arg\min_{Z \in \mathbb{R}^{N \times m}} h(Z, Q^{(k)}) \]
- Update iterate
  \[ Q^{(k+1)} = Z^{(k)} \]
- Increment iteration counter $k \leftarrow k + 1$
- Solution $Z^* \approx Q^{(k)}$

Until convergence
Iterative majorization

The majorizing function

\[ h(Z, Q) = \text{trace}(Z^T V Z) - 2\text{trace}(Z^T B(Q) Q) + \sum_{i > j} d_X^2(x_i, x_j) \]

is quadratic!

**Analytic expression** for the minimim

\[ \nabla_Z h(Z, Q) = 2VZ - 2B(Q)Q = 0 \]

\[ \Rightarrow Z = V^\dagger B(Q)Q \]

\[ = \frac{1}{N} B(Q)Q \]

Analytic expression for pseudoinverse:

\[ V^\dagger = \frac{1}{N} \left( I - \frac{1}{N} 1_{N \times N} \right) \]
SMACOF algorithm

- Start with some $Z^{(0)}$
- Update iterate

$$Z^{(k+1)} = \frac{1}{N} B(Z^{(k)}) Z^{(k)}$$

- Increment iteration counter $k \leftarrow k + 1$
- Solution $Z^* \approx Q^{(k)}$

The algorithm is called **SMACOF (Scaling by Minimizing A CONvex Function)**
SMACOF algorithm vs steepest descent

\[ Z^{(k+1)} = V^\dagger B(Z^{(k)})Z^{(k)} \]

\[ = Z^{(k)} - Z^{(k)} + V^\dagger B(Z^{(k)})Z^{(k)} \]

\[ = Z^{(k)} - \frac{1}{2} V^\dagger \left( 2VZ^{(k)} - 2B(Z^{(k)})Z^{(k)} \right) \]

\[ = Z^{(k)} - \frac{1}{2N} 2VZ^{(k)} - \frac{1}{2N} 2B(Z^{(k)})Z^{(k)} \]

\[ = Z^{(k)} - \frac{1}{2N} \nabla \sigma_2(Z^{(k)}) \]

**SMACOF is a constant-step gradient descent!**

- Majorization guarantees **monotonically decreasing** sequence of stress values (untypical behavior for constant-step steepest descent)
- **No guarantee** of global convergence
- Iteration cost: \( \mathcal{O}(mN^2) \)
MATLAB® intermezzo

SMACOF algorithm

Canonical forms
Examples of canonical forms

Near-isometric deformations of a shape

Canonical forms
Examples of canonical forms

Visualization of intrinsic similarity
Weighted stress

\[ \sigma_w(Z; D_X) = \sum_{i>j} w_{ij} |d_{ij}(Z) - d_X(x_i, x_j)|^2 \]

where \( w_{ij} \geq 0 \) are some fixed weights

Matrix expression

\[ \sigma_w(Z) = \text{trace}(Z^T V Z) - 2\text{trace}(Z^T B(Z) Z) + \sum_{i>j} d_X^2(x_i, x_j) \]

where \( V \) is and \( N \times N \) matrix with elements

\[ v_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k \neq i} w_{ij} & i = j \end{cases} \]

**SMACOF algorithm** can be used to minimize weighted stress!

\[ Z^{(k+1)} = V^\dagger B(Z^{(k)}) Z^{(k)} \]
Numerical geometry of non-rigid shapes  Multidimensional scaling

Variations on the stress theme

Generic form of the stress

\[
\sigma_\rho(Z; D_X) = \sum_{i>j} \rho(d_{ij}(Z) - d_X(x_i, x_j))
\]

where \( \rho(t) \) is some norm

For example, \( \rho(t) = |t|^p \) gives the \( L_p \)-stress

The necessary condition for \( Z^* \) to be the minimizer of \( \sigma_\rho \) is

\[
\nabla_Z \sigma_\rho(Z^*; D_X) = \sum_{i>j} \rho'(d_{ij}(Z^*) - d_X(x_i, x_j)) \nabla_Z d_{ij}(Z^*) = 0
\]
Variations on the stress theme

**Idea:** minimize weighted stress instead of the generic stress and choose the weights such that the two are equivalent

\[
\sum_{i > j} \rho'(d_{ij}(Z^*) - d_X(x_i, x_j)) \nabla_Z d_{ij}(Z^*) = \sum_{i > j} 2 w_{ij} (d_{ij}(Z^*) - d_X(x_i, x_j)) \nabla_Z d_{ij}(Z^*) = \nabla_Z \sigma_w(Z^*; D_X)
\]

If the weights are selected as

\[
w_{ij} = \frac{\rho'(d_{ij}(Z^*) - d_X(x_i, x_j))}{2(d_{ij}(Z^*) - d_X(x_i, x_j))}
\]

the minimizers of \(\sigma_\rho\) and \(\sigma_w\) will coincide

Since \(Z^*\) is unknown in advance, iterate!
Iteratively reweighted LS

- Start with some $Z^{(0)}$ and $w_{ij} = 1$
- Find

$$Z^{(k+1)} = \arg \min_{Z} \sigma_{w_{(k)}}(Z; D_X)$$

using $Z^{(k)}$ as an initialization
- Update weights

$$w_{ij}^{(k+1)} = \frac{\rho'(d_{ij}(Z^{(k+1)}) - d_X(x_i, x_j))}{2(d_{ij}(Z^{(k+1)}) - d_X(x_i, x_j))}$$

- Increment iteration counter $k \leftarrow k + 1$

The algorithm is called Iteratively Reweighted Least Squares (IRLS)
L_∞-stress

\[ \sigma_\infty(Z; D_X) = \max_{i,j=1,...,N} |d_{ij}(Z) - d_X(x_i, x_j)|. \]

Optimization problem

\[ Z^*_\infty = \arg\min_Z \max_{i,j=1,...,N} |d_{ij}(Z) - d_X(x_i, x_j)|. \]

can be written equivalently as a constrained problem

\[ Z^*_\infty = \arg\min_{Z, \tau} \tau \quad \text{s.t.} \quad |d_{ij}(Z) - d_X(x_i, x_j)| \leq \tau \]

\[ = \arg\min_{Z, \tau} \tau \quad \text{s.t.} \quad \begin{cases} d_{ij}(Z) - d_X(x_i, x_j) - \tau \leq 0, \\ -d_{ij}(Z) + d_X(x_i, x_j) - \tau \leq 0, \end{cases} \]

\( \tau \) is called an artificial variable
Complexity, *bis*

N=1000 points

MDS

N=200 points

MDS

x5 less points

x25 lower complexity
Multiresolution MDS

- **Bottom-up** approach: solve coarse level MDS problem to initialize fine level problem
- **Reduce complexity** (less fine-level iterations)
- Reduce the risk of local convergence (**good initialization**)
- Can be performed on multiple resolution levels
Two-resolution MDS

- Grid \( \Omega_0 = \{1, ..., N\} \) \quad \Omega_1 \subset \Omega_0
- Data \( D_0 = D_X \) \quad D_1 = (d_X(x_i, x_j) |_{ij \in \Omega_1})
- Interpolation operator \( P_1^0 : \mathbb{R}^{|\Omega_1| \times m} \rightarrow \mathbb{R}^{N \times m} \) to transfer solution from coarse level to fine level

Can be represented as an \( N \times |\Omega_1| \) sparse matrix

\[
Z_0^{(0)} = P_1^0 Z_1^* \quad \text{Solve fine level problem} \quad Z^*
\]

\[
\text{Interpolate}
\]

\[
Z_1^{(0)} \quad \text{Solve coarse level problem} \quad Z_1^*
\]
Multiresolution MDS

\[ Z^{(0)} = P_0^1 Z_1^* \]

Solve finest level problem \( Z^* \)

\[ Z_{L-1}^{(0)} = P_{L-1}^L Z_L^* \]

Solve \((L - 1)\)-st level problem \( Z_{L-1}^* \)

Interpolate

\[ Z_L^{(0)} \]

Solve coarsest level problem \( Z_L^* \)

Interpolate

Interpolate
Multiresolution MDS algorithm

- Hierarchy of grids \( \Omega_L \subset \Omega_{L-1} \subset \ldots \Omega_0 = \{1, \ldots, N\} \)
- Hierarchy of data \( D_L, \ldots, D_0 = D_X; \quad D_l = (d_X(x_i, x_j)|_{i,j \in \Omega_l}) \)
- Interpolation operators \( P_l^{l-1} : \mathbb{R}^{|\Omega_l| \times m} \rightarrow \mathbb{R}^{|\Omega_{l-1}| \times m} \)
- Start with coarsest-level initialization \( Z^{(0)}_L \)
- Solve the \( l \)-th level MDS problem

\[
Z^*_l = \arg\min_{Z_l \in \mathbb{R}^{|\Omega_l| \times m}} \sigma(Z_l; D_l)
\]

using \( Z^{(0)}_l \) as initialization

- Interpolate to next level \( Z^{(0)}_{l-1} = P_l^{l-1} Z^*_l \)
- Solve finest level problem

\[
Z^* = \arg\min_{Z \in \mathbb{R}^{N \times m}} \sigma(Z; D_X)
\]

using \( Z^{(0)}_0 \) as initialization
Top-down approach

- Start with a fine-level initialization $Z_0^{(0)}$
- **Decimate** fine-level initialization to the coarse grid

\[ Z_1^{(0)} = P_0^1 Z_0^{(0)} \]

- Solve the coarse-level MDS problem

\[ Z_1^* = \arg\min_{Z_1 \in \mathbb{R}^{|\Omega_1| \times m}} \sigma(Z_1; D_1) \]

using $Z_1^{(0)}$ as initialization

- Improve the fine-level solution propagating the coarse-level error

\[ Z_0^{(1)} = Z_0^{(0)} + P_1^0 (Z_1^* - Z_1^{(0)}) \]
\[ = Z_0^{(0)} + P_1^0 (Z_1^* - P_0^1 Z_0^{(0)}) \]

Typically
\[ P_0^1 = (P_1^0)^T \]
A problem

\[ \nabla_{Z_0} \sigma(Z_0^*) = 0 \]

Fine level

\[ \sigma(Z_1) \]

Coarse level

\[ \nabla_{Z_1} \sigma(P_0^1Z_0^*) = T_1 \neq 0 \]
Correction

The fine- and the coarse-level minimizers do not coincide!

\[ \nabla_{Z_1} \sigma(P_0^1 Z_0^*) = T_1 \neq 0 \]

\( T_1 \) is called a residual

Force consistency of the coarse- and fine-level problems by introducing a correction term into the stress

\[ \sigma(Z_1; D_1) - \langle T_1, Z_1 \rangle = \sigma(Z_1; D_1) - \text{trace}(Z_1^T T_1) \]

which gives the gradient

\[ \nabla_{Z_1} \left( \sigma(Z_1) - \langle T_1, Z_1 \rangle \right) \bigg|_{Z_1=P_1^0 Z_0^*} = \nabla_{Z_1} \sigma(Z_1) \bigg|_{Z_1=P_1^0 Z_0^*} - T_1 = 0 \]
Correction

\[ \sigma(Z_0) \]

\[ \sigma(Z_0 + P_1^0(Z_1 - P_0^1 Z_0); D_0) \]

Fine level

\[ \sigma(Z_1; D_1) - \text{trace}(Z_1^T T_1) \]

Coarse level
Correction

In order to guarantee consistency, \( T_1 \) must satisfy

\[
\nabla_{Z_1} \sigma(Z_1; D_1) - T_1 = \nabla_{Z_1} \sigma \left(Z_0 + P_1^0(Z_1 - P_0^1 Z_0); D_0\right)
\]

at \( Z_1 = P_0^1 Z_0 \)

\[
= (P_1^0)^T \nabla_{Z_0} \sigma(Z_0; D_0)
\]

which gives the correction term

\[
T_1 = \nabla_{Z_1} \sigma(P_0^1 Z_0; D_1) - P_0^1 \nabla_{Z_0} \sigma(Z_0; D_0)
\]

Verify: \( P_0^1 Z_0^* \) is the minimizer of coarse-level problem with correction,

\[
P_0^1 Z_0^* = \arg\min_{Z_1 \in \mathbb{R}^{|\Omega_1| \times m}} \sigma(Z_1; D_1) - \text{trace}(Z_1^T T_1)
\]
Modified stress

Another problem: the coarse grid problem

$$\min_{Z_1 \in \mathbb{R}^{|\Omega_1| \times m}} \sigma(Z_1; D_1) - \text{trace}(Z_1^T T_1)$$

is unbounded for $T_1 \neq 0$

Fix the translation ambiguity by adding a **quadratic penalty** to the stress

$$\tilde{\sigma}(Z) = \sigma(Z) + \lambda \sum_{k=1}^{m} \left( \sum_{i=1}^{N} z_i^k \right)^2 = \sigma(Z) + \lambda \text{trace}(Z^T 1_{N \times N} Z)$$

$\tilde{\sigma}$ is called the **modified stress** [Bronstein et al., 05]

The modified coarse grid problem is bounded

$$\min_{Z_1 \in \mathbb{R}^{|\Omega_1| \times m}} \tilde{\sigma}(Z_1; D_1) - \text{trace}(Z_1^T T_1)$$
Two-grid MDS

- Iteratively improve the fine-level solution using coarse grid residual
- External iteration is called **cycle**

\[
Z_0^{(0)} \quad \text{Decimate} \quad Z_0^{(1)} = P_0^1 Z_0^{(0)} \quad \text{Solve corrected coarse level problem}
\]

\[
E_0 = P_1^0 E_1 \quad \text{Correct fine level solution} \quad Z_0^{(1)} = Z_0^{(0)} + \alpha_1 E_0
\]

\[
E_1 = (Z_1^* - P_0^1 Z_0^{(0)}) \quad \text{Interpolate}
\]
Two-grid MDS

- Perform a few optimization iterations at fine level between cycles (relaxation)

\[
\begin{align*}
Z_0^{(0)} & \quad \text{Relax} \quad Z_0^{(1)} \\
Z_1^{(1)} & = P_0^1 Z_0^{(1)} \quad \text{Decimate} \quad Z_1^{(1)} \\
E_0 & = P_1^0 E_1 \quad \text{Correct fine level solution} \quad Z_0^{(2)} = Z_0^{(1)} + \alpha_1 E_0 \\
E_1 & = (Z_1^* - P_0^1 Z_0^{(1)}) \quad \text{Interpolate} \quad Z_1^*
\end{align*}
\]
Two-grid MDS algorithm

- Start with fine-level initialization \( Z_0^{(0)} \)
- Produce an improved fine-level solution \( Z_0^{(1)} \) by making a few iterations on \( \sigma(Z_0) \) initialized with \( Z_0^{(0)} \)
- Decimate fine-level solution \( Z_1^{(1)} = P_0^1 Z_0^{(1)} \)
- Correction: \( T_1 = \nabla_{Z_1} \hat{\sigma}(Z_1^{(1)}; D_1) - P_0^1 \nabla_{Z_0} \hat{\sigma}(Z_0^{(1)}; D_0) \)
- Solve the coarse-level corrected MDS problem
  \[
  Z_1^* = \arg\min_{Z_1 \in \mathbb{R}^{|\Omega_1| \times m}} \hat{\sigma}(Z_1; D_1) - \text{trace}(Z_1^T T_1)
  \]
  using \( Z_1^{(1)} \) as initialization
- Transfer residual: \( Z_0^{(2)} = Z_0^{(1)} + \alpha_1 P_0^1 (Z_1^* - P_0^1 Z_0^{(1)}) \)
- Update iterate \( Z_0^{(0)} \leftarrow Z_0^{(2)} \)
- Solution \( Z^* = Z_0^{(0)} \)

Until convergence
Multigrid MDS

- Apply the two-grid algorithm recursively
- Different cycles possible, e.g. **V-cycle**
MATLAB® intermezzo

Multigrid MDS
Convergence example

Convergence of SMACOF and Multigrid MDS (N=2145)
How to accelerate convergence?

Let \( Z^{(0)}, Z^{(1)}, \ldots \) be a sequence of iterates produced by some optimization algorithms converging to \( Z^* \).

Denote by \( E^{(k)} = Z^{(k)} - Z^* \) the remained.

Construct a new sequence \( \hat{Z}^{(0)}, \hat{Z}^{(1)}, \ldots \) converging faster to \( Z^* \) in the sense

\[
\lim_{k \to \infty} \frac{\| E^{(k)} \|}{\| E^{(k)} \|} = \lim_{k \to \infty} \frac{\| \hat{Z}^{(k)} - Z^* \|}{\| Z^{(k)} - Z^* \|} = 0
\]
Vector extrapolation

Construct the new sequence as a transformation of iterates

\[ \hat{Z}^{(k)} = T(Z^{(k)}, \ldots, Z^{(k+K)}) \]

Particular choice: linear combination

\[ \hat{Z}^{(k)} = \gamma_0 Z^{(k)} + \ldots + \gamma_K Z^{(k+K)} = (\gamma_0 + \ldots + \gamma_K)Z^* + \gamma_0 E^{(k)} + \ldots + \gamma_K E^{(k+K)} \]

Question: how to choose the coefficients \( \gamma_0, \ldots, \gamma_K \)?

Ideally, \( \hat{Z}^{(k)} = Z^* \) hence

\[ \begin{align*}
\gamma_0 + \ldots + \gamma_K &= 1 \\
\gamma_0 E^{(k)} + \ldots \gamma_K E^{(k+K)} &= 0
\end{align*} \]
Reduced rank extrapolation (RRE)

Problem: \( E^{(k)} = Z^{(k)} - Z^* \) depends on unknown \( Z^* \)

Eliminate this dependence by considering the difference
\[
\Delta \hat{Z}^{(k)} = \hat{Z}^{(k+1)} - \hat{Z}^{(k)}
\]

which ideally should vanish.

Result: solve the constrained linear system
\[
\gamma_0 \Delta Z^{(k)} + \ldots + \gamma_K \Delta Z^{(k+K)} = 0 \quad \text{s.t.} \quad \gamma_0 + \ldots + \gamma_K = 1
\]
of \( Nm \) equations in \( K + 1 \) variables

Typically \( K \ll Nm \) hence the system is **over-determined** and must be solved approximately using least-squares
Optimization with vector extrapolation

- Start with some $Z^{(0)}$ and $k = 0$
- Generate iterates $Z^{(k+1)}, \ldots, Z^{(k+K)}$
  using optimization on $\sigma(Z)$ initialized by $Z^{(k)}$
- Extrapolate $\hat{Z}^{(k)}$ from $Z^{(k)}, \ldots, Z^{(k+K)}$
- Update $Z^{(k+K+1)} \leftarrow \hat{Z}^{(k)}$
- Increment iteration counter $k \leftarrow k + K + 1$

Until convergence
Safeguarded optimization with vector extrapolation

- Start with some $Z^{(0)}$ and $k = 0$
- Generate iterates $Z^{(k+1)}, \ldots, Z^{(k+K)}$
  - using optimization on $\sigma(Z)$ initialized by $Z^{(k)}$
- Extrapolate $\hat{Z}^{(k)}$ from $Z^{(k)}, \ldots, Z^{(k+K)}$
- If $\sigma(\hat{Z}^{(k)}) < \sigma(Z^{(k+K)})$
  - $Z^{(k+K+1)} \leftarrow \hat{Z}^{(k)}$
- else
  - $Z^{(k+K+1)} \leftarrow Z^{(k+K)}$
- Increment iteration counter $k \leftarrow k + K + 1$

Until convergence