Trading Spaces: Measures of Document Proximity and Methods for Embedding Them

Michael W. Trosset

Department of Mathematics

College of William & Mary

Thanks to Stephanie Valentino and Renée Welch!

Document Proximity

The concept of *proximity* is intrinsic to most text mining activities. For example, what is a *cluster* of documents? What does it mean for subsets of a corpus to exhibit *internal cohesion* and *external isolation*?

The phrase *proximity* comprehends both similarity and dissimilarity:

- A *dissimilarity* measure δ is symmetric ($\delta(i,j) = \delta(j,i)$), nonnegative ($\delta(i,j) \geq 0$), and hollow ($\delta(i,i) = 0$). The interpretation of dissimilarity demands the following monotonicity property: pair (i,j) is more dissimilar than pair (r,s) if and only if $\delta(i,j) > \delta(r,s)$.

- A *similarity* measure γ is symmetric, nonnegative, and satisfies $\gamma(i,i) \geq \gamma(i,j)$. The interpretation of similarity demands the following monotonicity property: pair (i,j) is more similar than pair (r,s) if and only if $\gamma(i,j) > \gamma(r,s)$.

In theory, document proximities might be obtained by direct comparison of actual documents; more commonly, attributes of each document are quantified, then proximities are computed from a mediating vector space model (VSM).

Binary VSM

Suppose that the corpus comprises n documents, and that q terms are of interest. To compare documents i and j, we construct a 2×2 contingency table:

	present in j	absent in j
present in i	a	b
absent in i	c	d

Two natural measures of similarity are the simple matching coefficient and Jaccard's matching coefficient:

$$\gamma_{ij} = \frac{a+d}{a+b+c+d} \qquad \gamma_{ij} = \frac{a}{a+b+c}$$

In both cases,

- an intuitive measure of dissimilarity is $\delta_{ij} = 1 - \gamma_{ij}$;
- the $n \times n$ similarity matrix $\Gamma = [\gamma_{ij}]$ is positive semidefinite (psd).

Quantitative VSM

Let o_{ik} denote the presence ($o_{ik} = 1$) or absence ($o_{ik} = 0$) of term k in document i. Let m_{ik} denote the number of occurences of term k in document i.

To construct an $n \times q$ data matrix $Y = [y_{ik}]$, we might quantify the importance of term k on document i in various ways, e.g.,

$$\begin{aligned} y_{ik} &= m_{ik}, \\ y_{ik} &= (1 + m_{ik}) \log_2 \left(\frac{n}{o_{+k}} \right), \\ y_{ik} &= \log \left(\frac{m_{ik}/m_{i+}}{m_{+k}/m_{++}} \right). \end{aligned}$$

Two natural measures of dissimilarity between documents i and j are weighted Minkowski (L^p) distance and a measure proposed by Lance & Williams (1966):

$$\delta_{ij}^p = \sum_{k=1}^{q} w_k \left| y_{ik} - y_{jk} \right|^p \qquad \delta_{ij} = \sum_{k=1}^{q} \frac{|y_{ik} - y_{jk}|}{|y_{ik} + y_{jk}|}$$

Angles

Let $y_1, \ldots, y_n$ denote the feature vectors in the VSM, i.e., the rows of the data matrix $Y = [y_{ik}]$. For $\tilde{y}_i, \tilde{y}_j \in \Re^q$, the quantity

$$r_{ij} = \frac{\langle \tilde{y}_i, \tilde{y}_j \rangle}{\|\tilde{y}_i\| \ \|\tilde{y}_j\|} = \left\langle \frac{\tilde{y}_i}{\|\tilde{y}_i\|}, \frac{\tilde{y}_j}{\|\tilde{y}_j\|} \right\rangle,$$

is the cosine of the angle between $\tilde{y}_i$ and $\tilde{y}_j$. Other than $\tilde{Y}_0 = Y$, one might construct

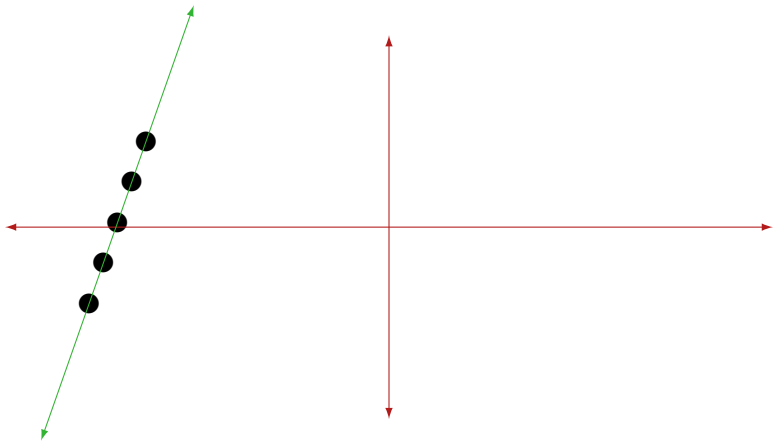
$$\tilde{Y}_{\text{col}} = \left(I_n - \frac{1}{n} e_n e_n^t \right) Y \quad \text{or} \quad \tilde{Y}_{\text{row}} = Y \left(I_q - \frac{1}{q} e_q e_q^t \right).$$

The former translates the y_i so that their centroid lies at the origin.

The latter projects the feature vectors into the linear subspace $e_q^{\perp}$, in which case r_{ij} is Pearson's product-moment correlation coefficient.

Euclidean Subspaces

Suppose that we want to represent the documents as $x_1, \ldots, x_n \in \Re^d$, $d \ll q$. If $y_1, \ldots, y_n \in \Re^q$, i.e., if $\delta_{ij} = \|y_i - y_j\|$, then a natural approach is to project the y_i into the (affine) linear subspace that minimizes squared residual error.



Latent semantic indexing finds the best linear subspace; principal component analysis finds the best affine linear subspace.

To find the best affine linear subspace, first translate the y_i so that their centroid lies at the origin, then find the best linear subspace.

Principal Component Analysis

$\tilde{Y} = (I - ee^t/n)Y$ is the $n \times q$ centered data matrix;
$\tilde{Y}^t\tilde{Y}$ is the $q \times q$ matrix of inner products between terms (variables);
$\tilde{Y}\tilde{Y}^t$ is the $n \times n$ matrix of inner products between documents (objects).

Let $\tilde{Y} = U \begin{bmatrix} \Sigma \\ \hline 0 \end{bmatrix} V^t$ denote the singular value decomposition of $\tilde{Y}$. Then

$$\tilde{Y}^t\tilde{Y} = V\Sigma^2 V^t \quad \text{and} \quad \tilde{Y}\tilde{Y}^t = U \left[\begin{array}{c|c} \Sigma^2 & 0 \\ \hline 0 & 0 \end{array}\right] U^t.$$

Given $d < q$, let $U = \left[\begin{array}{c|c} U_d & \cdot \end{array}\right]$, $V = \left[\begin{array}{c|c} V_d & \cdot \end{array}\right]$, and $\Sigma_d = \text{diag}\,(\sigma_1, \ldots, \sigma_d)$.
Then the d-dimensional PC representation of Y is

$$\begin{aligned} \tilde{Y}V_d &= U \begin{bmatrix} \Sigma \\ \hline 0 \end{bmatrix} \begin{bmatrix} v_1^t \\ \vdots \\ v_q^t \end{bmatrix} [v_1 \cdots v_d] \\ &= U \begin{bmatrix} \Sigma \\ \hline 0 \end{bmatrix} \begin{bmatrix} I \\ \hline 0 \end{bmatrix} = U \begin{bmatrix} \Sigma_p \\ \hline 0 \end{bmatrix} = U_d\Sigma_d. \end{aligned}$$

Embedding in Euclidean Space

Suppose that δ is non-Euclidean, but that we want to represent the documents as $x_1, \ldots, x_n \in \Re^d$.

Distance methods embed documents by approximating document dissimilarities, δ_{ij}, with Euclidean distances, $\|x_i - x_j\|$, e.g., by minimizing the *raw stress criterion*,

$$\sum_{i,j} \left(\|x_i - x_j\| - \delta_{ij} \right)^2 .$$

Inner product methods embed documents as follows:

1. Estimate document inner products, b_{ij}, e.g., from document dissimilarities.

2. Approximate the b_{ij} with Euclidean inner products, $\langle x_i, x_j \rangle$, e.g., by minimizing the *strain criterion*,
$$\sum_{i,j} \left(\langle x_i, x_j \rangle - b_{ij} \right)^2 .$$

Euclidean Distance Geometry

A dissimilarity matrix $\Delta_2 = [\delta_{ij}^2]$ is a *Euclidean distance matrix* (EDM) iff there exist $x_1, \ldots, x_n \in \Re^p$ such that $\delta_{ij}^2 = \|x_i - x_j\|^2$. The smallest such p is the *embedding dimension* of the EDM.

Let $P = I - ee^t/n$. Notice that P is symmetric and idempotent; Pv is the projection of $v \in \Re^n$ into $e^\perp$, $P\Delta P$ is the "double centering" of Δ, and $P\Delta Pe = 0$.

Theorem: A dissimilarity matrix Δ_2 is an EDM with embedding dimension p iff

$$\tau(\Delta_2) = -\frac{1}{2} P \Delta_2 P$$

is psd and has rank p. Furthermore, if $\Delta_2 = [\delta_{ij}^2]$ is an EDM and

$$\tau(\Delta_2) = \begin{bmatrix} x_1^t \\ \vdots \\ x_n^t \end{bmatrix} [x_1 \cdots x_n],$$

then $\delta_{ij}^2 = \|x_i - x_j\|^2$.

If $x_1, \ldots, x_n \in R^p$ have inner product matrix B, then they have EDM

$$\kappa(B) = \mathrm{diag}(B)ee^t - 2B + ee^t\mathrm{diag}(B).$$

Notice that interpoint distances do not depend on where configurations are centered, whereas inner products do. To remove this indeterminancy, we center each configuration at the origin, i.e., we require $X^te = 0$. If a configuration is centered at the origin, then the corresponding inner product matrix must satisfy $Be = XX^te = 0$. Conversely, if $B = XX^t$ and $Be = 0$, then $0 = e^tBe = (X^te)^t(X^te)$ implies that $X^te = 0$.

Upon restricting attention to symmetric psd B that satisfy $Be = 0$, the linear transformations κ and τ are mutually inverse:

- κ converts centered Euclidean inner products to squared Euclidean distances;
- τ converts squared Euclidean distances to centered Euclidean inner products.

Classical Multidimensional Scaling

To embed fallible squared dissimilarities in $\Re^d$, we first convert the fallible squared dissimilarities to fallible centered inner products, $B = \tau(\Delta_2)$, then replace B with $\bar{B}$, the nearest symmetric psd matrix with rank $\leq d$.

Let $\lambda_1 \geq \cdots \geq \lambda_n$ denote the eigenvalues of B and let $v_1, \ldots, v_n$ denote the corresponding eigenvectors. Let $\sigma_i^2 = \max(\lambda_i, 0)$ for $i = 1, \ldots, d$. Then

$$\bar{B} = \sum_{i=1}^{d} \sigma_i^2 v_i v_i^t = [\sigma_1 v_1 \cdots \sigma_d v_d] \begin{bmatrix} \sigma_1 v_1^t \\ \vdots \\ \sigma_d v_d^t \end{bmatrix}$$

produces a d-dimensional configuration of points whose principal components are eigenvectors of $\tau(\Delta_2)$.

Example 1: Principal Component Analysis

If $\delta_{ij}^2 = \|y_i - y_j\|^2$, then $B = \tilde{Y}\tilde{Y}^t$ and CMDS is equivalent to PCA.

Example 2: Isomap

Suppose that $y_1, \ldots, y_n \in \Re^q$ lie on a nonlinear manifold and fix $\epsilon > 0$.
To construct the ϵ-Isomap embedding of $y_1, \ldots, y_n$ as $x_1, \ldots, x_n \in \Re^d$:

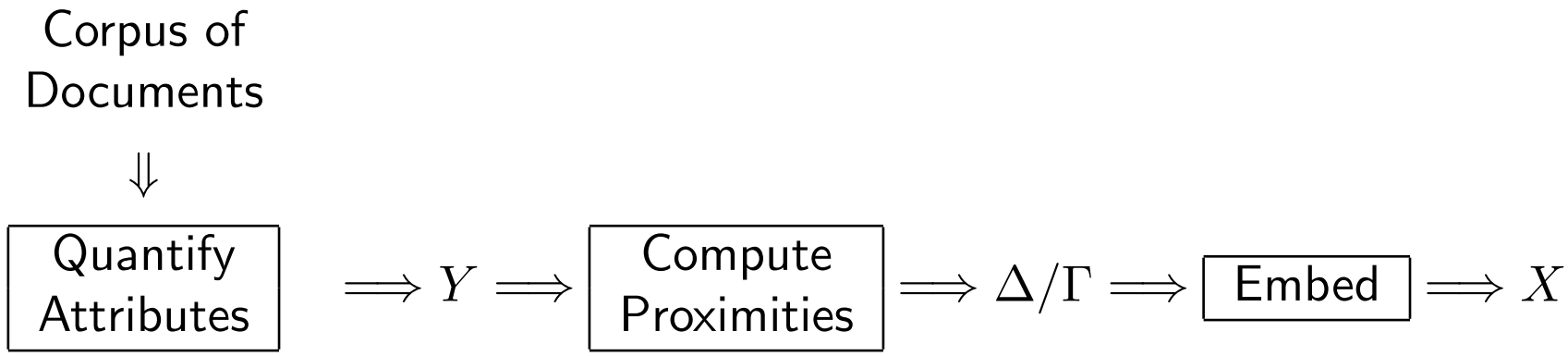
1. Let $w_{ij} = 1$ if $\|y_i - y_j\| < \epsilon$ and $w_{ij} = 0$ otherwise. Define a weighted undirected graph G with n vertices as follows: vertices i and j are connected iff $w_{ij} > 0$, in which case edge $i \sim j$ is weighted by w_{ij}.

2. Let δ_{ij} denote the shortest path distance in G between vertices i and j.

3. Embed $\Delta = [\delta_{ij}]$ in $\Re^d$ by CMDS.

Document Space

Our conception of document space depends on decoupling the act of embedding from the proximities that are embedded.



Y contains feature vectors in the q-dimensional VSM;
X contains vectors in the representational *document space*, typically $\Re^d$.

Information retrieval? Supervised learning?
Embed similarities? Other embedding methods?

Information Retrieval

What is a query? One possibility: a *query* is a linear combination of terms.

For a quantitative VSM, let f_k denote the feature vector in which term k has unit weight and other terms have zero weight. Given coefficients α_k, let

$$y_\alpha(\mu) = \mu \sum_{k=1}^{q} \alpha_k f_k,$$

then embed the linear trajectory $\{y_\alpha(\mu)\}$ in the VSM as a (possibly nonlinear) trajectory $x_\alpha(\mu)$ in the document space $\Re^d$.

We might answer a query by returning documents with *small* values of

$$D_{\text{abs}}(x_i) = \min_\mu \|x_i - x_\alpha(\mu)\| \quad \text{or} \quad D_{\text{rel}}(x_i) = D_{\text{abs}}(x_i) \,/\, \|x_i - x_\alpha(0)\| .$$

If $\{x_\alpha(\mu)\}$ is a line, then the latter is equivalent to returning documents with *large* values of

$$C(x_i) = \frac{\langle x_i - x_\alpha(0), x_\alpha(1) - x_\alpha(0)\rangle}{\|x_i - x_\alpha(0)\| \; \|x_\alpha(1) - x_\alpha(0)\|}.$$

Example 3: Euclidean Biplots

If Δ_2 is an EDM with embedding dimension p, then $y_1, \ldots, y_n, y_\alpha(\mu)$ can be exactly embedded in $\Re^{p+1}$ as $\hat{y}_1, \ldots, \hat{y}_n, \hat{y}_\alpha(\mu)$.

If document space was constructed by CMDS, then the desired trajectory is obtained by projecting $\hat{y}_\alpha(\mu) \in \Re^{p+1}$ into $\Re^d$, obtaining the line $\{x_\alpha(\mu)\}$.

More generally, a *biplot* displays information about terms in the constructed document space. Constructing a biplot requires: (1) a VSM; (2) a measure of document dissimilarity; and (3) an embedding method.

Quantitative terms are represented by continuous curves; categorical terms are represented by simplices. These curves and simplices constitute a reference system analogous to conventional orthogonal coordinate axes.

Supervised Learning

Linear discriminant analysis (LDA) requires preliminary construction of a document space, which may be facilitated by including unlabelled documents. However, the discriminant coordinates may differ from the principal components.

Suppose that a corpus contains 3 classes, each of which contains 10 documents. The matrix $\tau(\Delta_2)$ has 16 positive eigenvalues:

$$\begin{array}{cccccccc} 24.94 & 13.21 & 4.70 & 2.90 & 2.17 & 1.56 & 1.44 & 1.30 \\ 0.99 & 0.67 & 0.61 & 0.42 & 0.28 & 0.25 & 0.09 & 0.04 \end{array}$$

We construct 2 discriminant coordinates using various choices of d:

d	F_1	F_2
2	3.26	1.91
3	4.33	2.74
8	20.24	8.25
16	171.15	62.93

Similarities and Inner Products

Given (possibly fallible) inner products, $B = [b_{ij}]$, one might choose X to minimize $\|B - XX^t\|^2$. This is equivalent to computing $\Delta_2 = \kappa(B)$, then performing CMDS.

The "standard" transformation from similarity to dissimilarity interprets γ as an inner product, then computes

$$\Delta_2 = \kappa(\Gamma).$$

The standard transformation threatens the monotonicity of γ; however, if each $\gamma_{ii} = 1$, then it yields

$$\delta_{ij}^2 = 1 - \gamma_{ij}.$$

Example 4: Matching Coefficients

For a binary VSM, let γ_{ij} be a matching coefficient and embed by applying CMDS to $\Delta_2 = \kappa(\Gamma)$.

Because $\gamma_{ii} = 1$, $\delta_{ij} = \sqrt{1 - \gamma_{ij}}$, as opposed to the more intuitive nonmatching coefficient, $\delta_{ij} = 1 - \gamma_{ij}$.

Because Γ is psd, Δ_2 is an EDM.

Example 5: Cosine Measures

For a quantitative VSM $\sim \Re^q$, we compute the cosine measure,

$$r_{ij} = \frac{\langle \tilde{y}_i, \tilde{y}_j \rangle}{\|\tilde{y}_i\| \ \|\tilde{y}_j\|} = \left\langle \frac{\tilde{y}_i}{\|\tilde{y}_i\|}, \frac{\tilde{y}_j}{\|\tilde{y}_j\|} \right\rangle,$$

then embed by applying CMDS to $\Delta_2 = \kappa(R)$. This procedure is equivalent to first scaling each $\tilde{y}_i$ to lie on the unit sphere in $\Re^q$, then performing PCA.

Example 6: Laplacian Eigenmaps

Let G be a weighted undirected graph with a vertex for each document in the corpus and edge weights γ_{ij}.

The graph Laplacian is the $n \times n$ matrix $L = \text{diag}(\Gamma e) - \Gamma$.
L is symmetric, psd, and $Le = 0$.

Let $0 < \sigma_1^2 \leq \cdots \leq \sigma_r^2$ denote the strictly positive eigenvalues of L, let $v_1, \ldots, v_r$ denote the corresponding eigenvectors, and let

$$X = \left[\begin{array}{c|c|c} \frac{v_1}{\sigma_1} & \cdots & \frac{v_d}{\sigma_d} \end{array} \right].$$

If Δ_2 is such that $\tau(\Delta_2) = L^\dagger$, then X is constructed from Δ_2 by CMDS.

What is $\Delta_2 = \kappa(L^\dagger)$, i.e., what is the transformation from similarity to dissimilarity?

Because L is psd, so is $L^\dagger$; hence, $\Delta_2 = \kappa(L^\dagger)$ is an EDM.

Two interpretations of dissimilarities implicit in Laplacian eigenmaps:

1. G is an electrical circuit. Vertices are terminals. Each edge is a resistor with conductance γ_{ij}. Δ_2 is the *effective resistance* of G, i.e., δ_{ij}^2 is the potential difference between terminals i and j when a unit current source is applied.

   The resistance, δ_{ij}^2, is small when there are many paths with high conductance between terminals i and j. The Euclidean distance δ_{ij} is the *resistance distance*.

2. G is a Markov chain. Vertices are states and the transition probabilities are $p_{ij} = \gamma_{ij}/\gamma_{i+}$. Let $t(j|i)$ denote the expected number of transitions to get from state i to state j for the first time. Then

$$\delta_{ij}^2 = \frac{t(j|i) + t(i|j)}{\gamma_{++}}.$$

   The expected commute time, $\gamma_{++}\delta_{ij}^2$, is small when there are many paths with high probability between states i and j.

Comparison to Isomap:

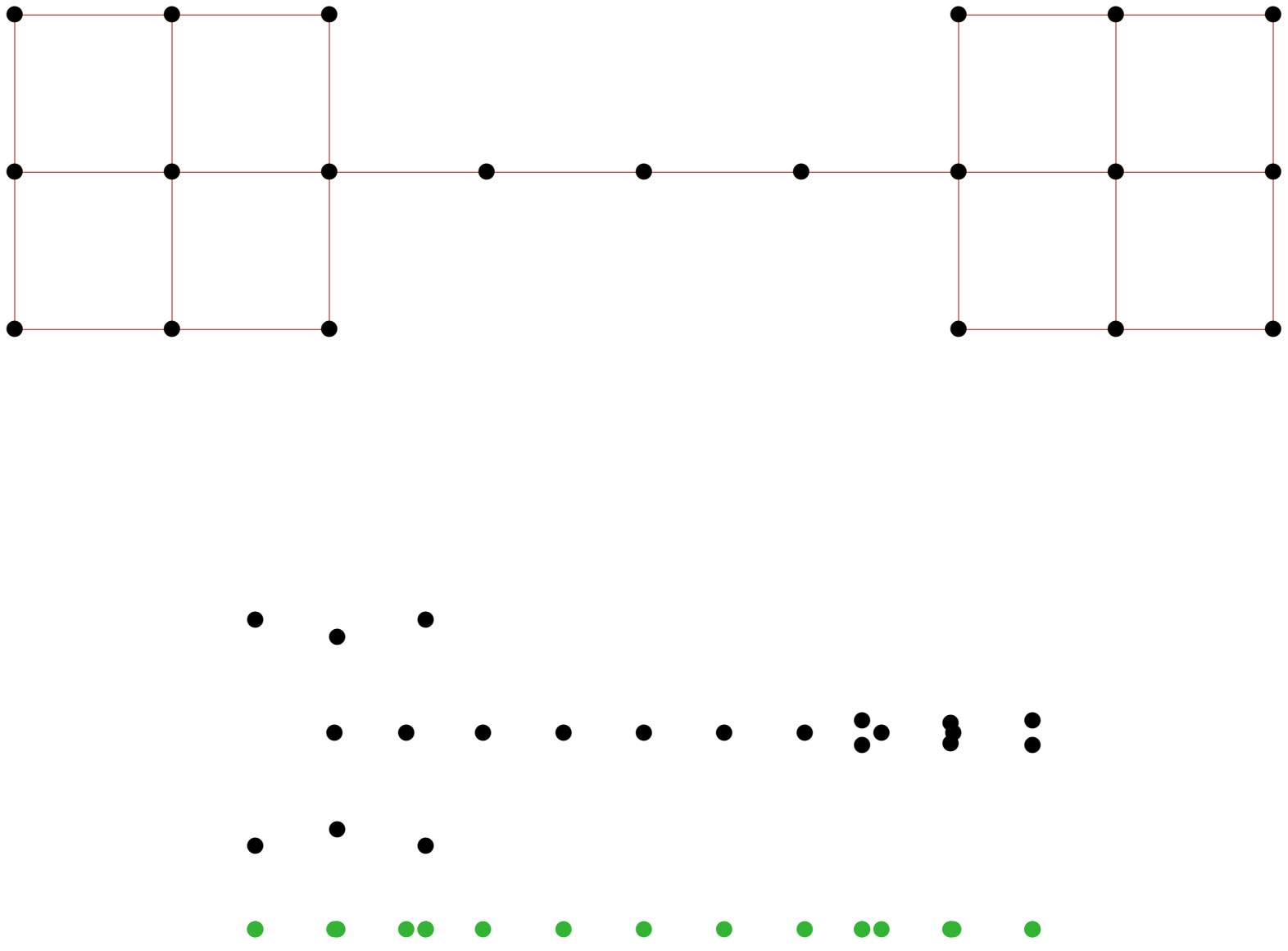
Suppose that $y_1, \ldots, y_n \in \Re^q$ lie on a nonlinear manifold and fix $\epsilon > 0$.
To construct an embedding of $y_1, \ldots, y_n$ as $x_1, \ldots, x_n \in \Re^d$:

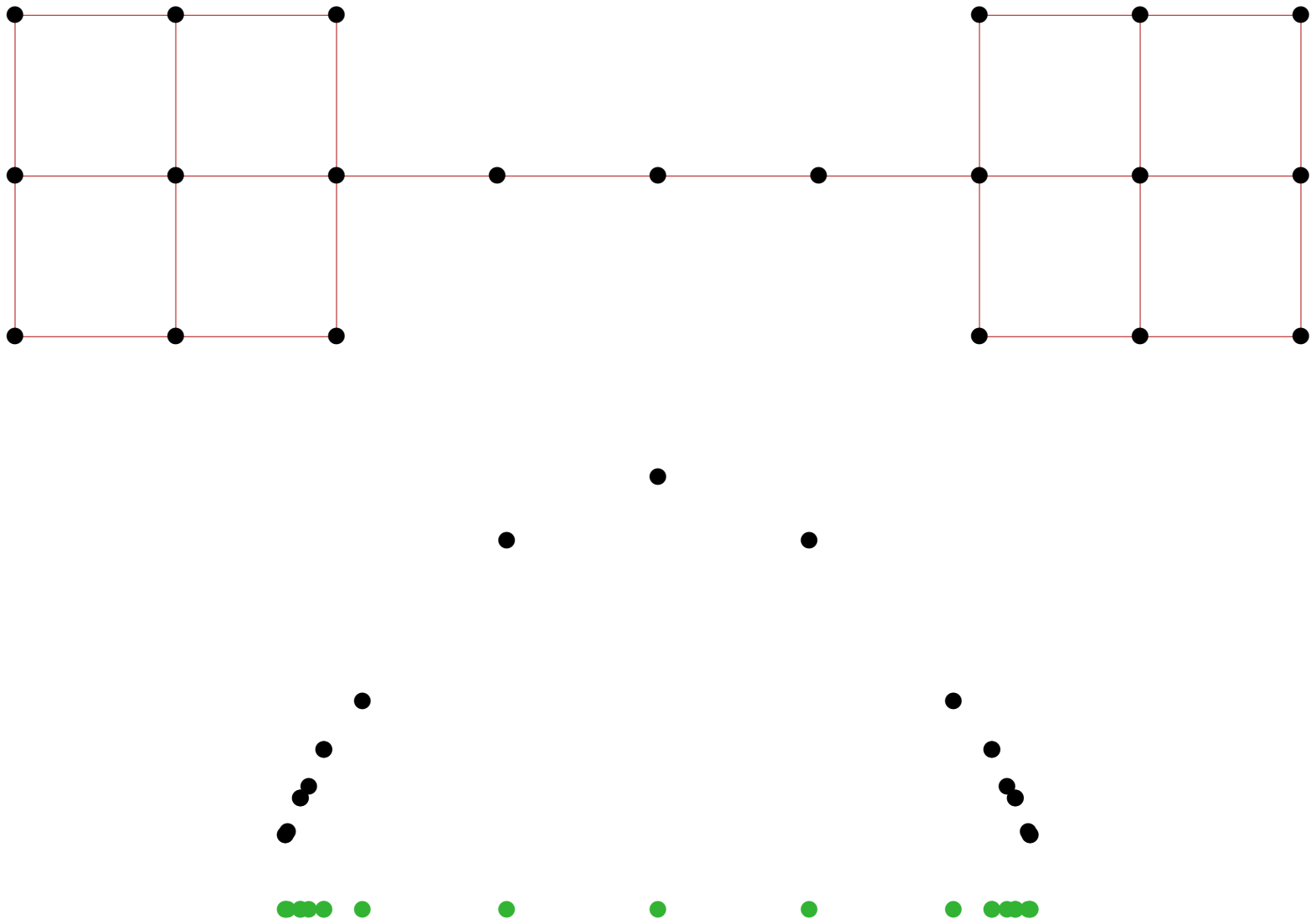
1. Let $w_{ij} = 1$ if $\|y_i - y_j\| < \epsilon$ and $w_{ij} = 0$ otherwise. Define a weighted undirected graph G with n vertices as follows: vertices i and j are connected iff $w_{ij} > 0$, in which case edge $i \sim j$ is weighted by w_{ij}.

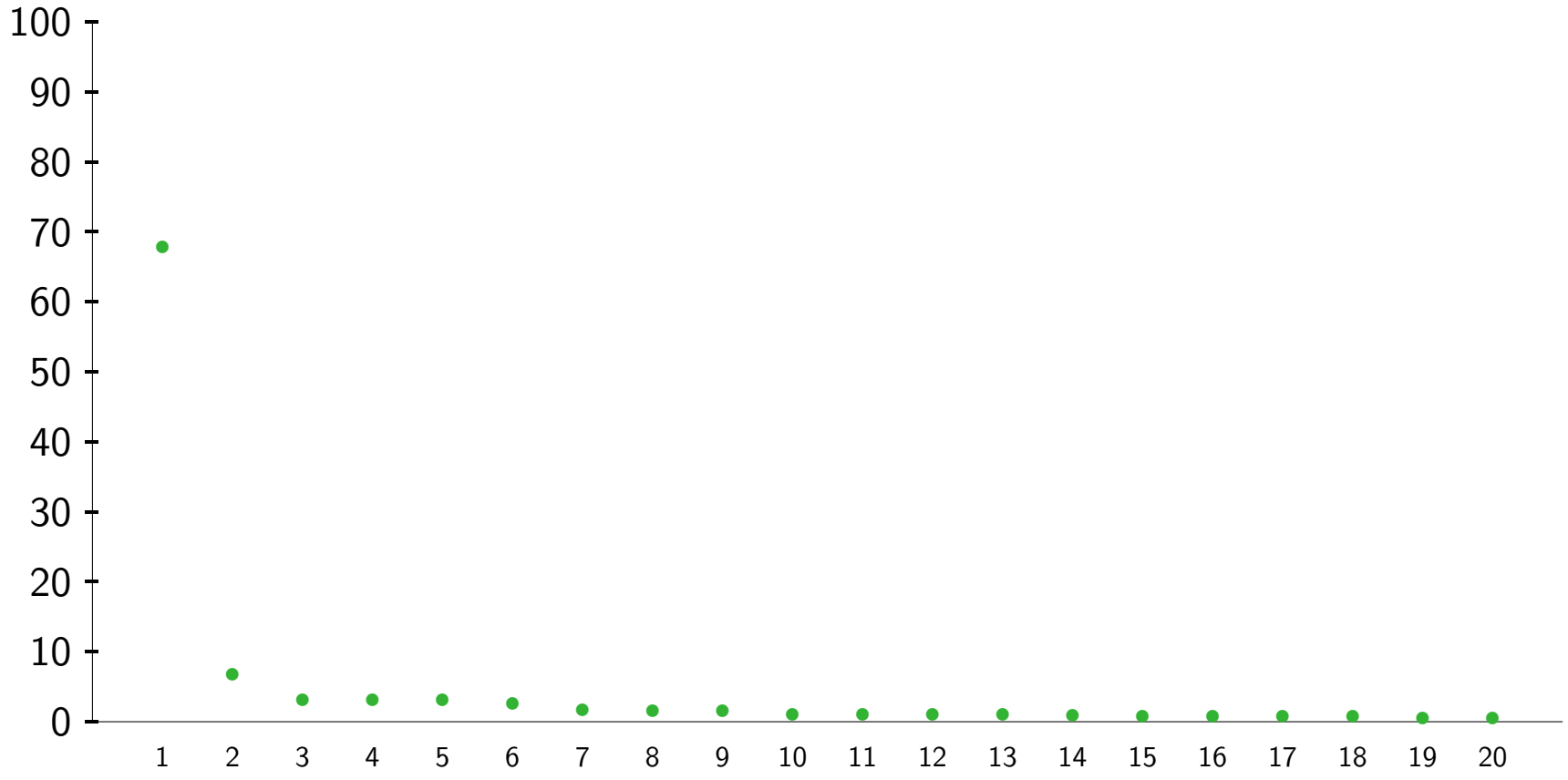
2. Let δ_{ij} denote the distance in G between vertices i and j.
   Isomap uses shortest path distance; Laplacian eigenmaps use resistance distance.

3. Embed $\Delta = [\delta_{ij}]$ in $\Re^d$ by CMDS.

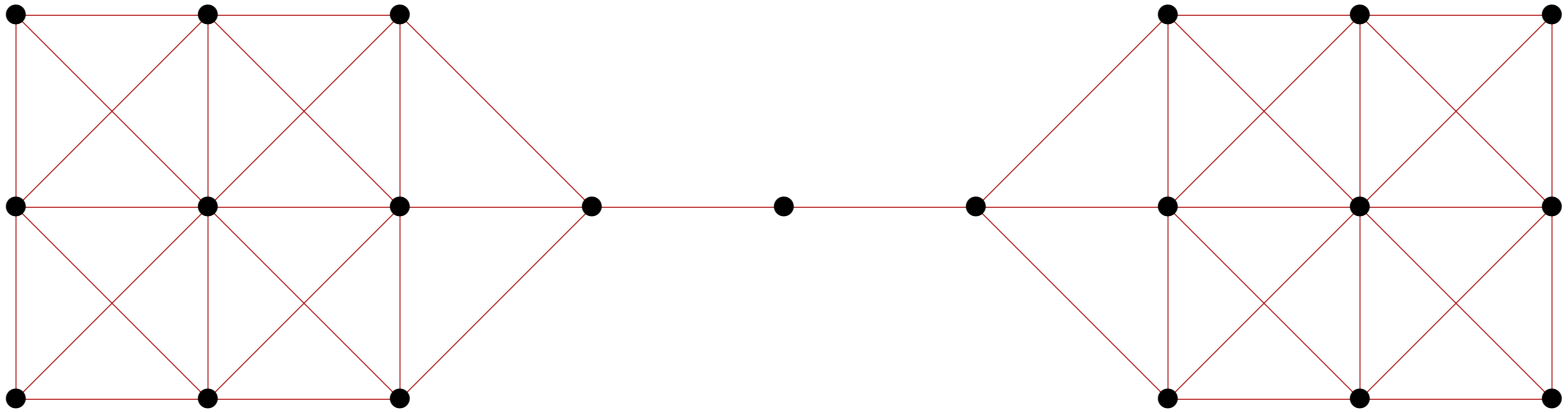
Remark: *Fast Iterative Denoising* uses k approximately nearest neighbors (instead of ϵ-neighborhoods) and resistance distance.

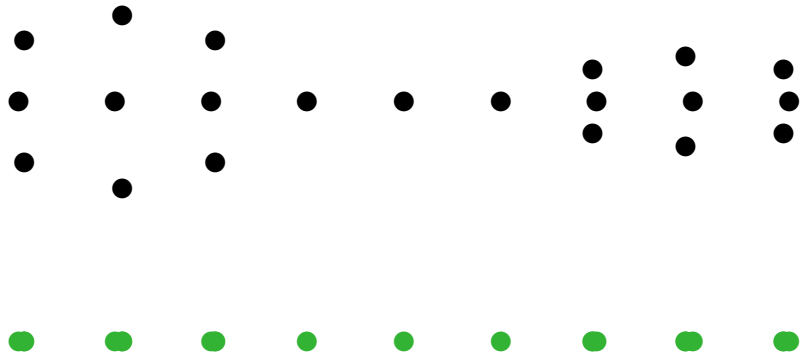


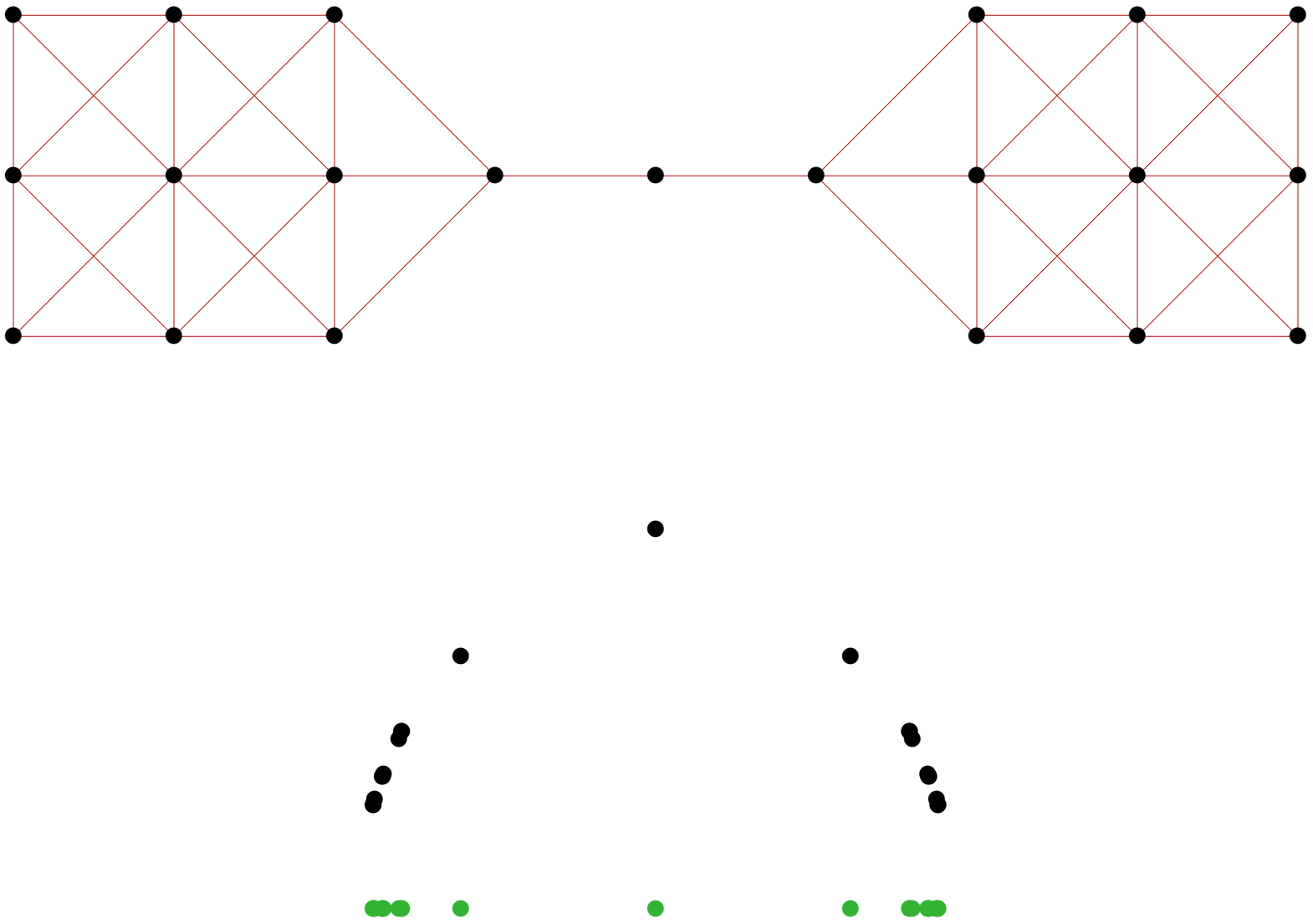


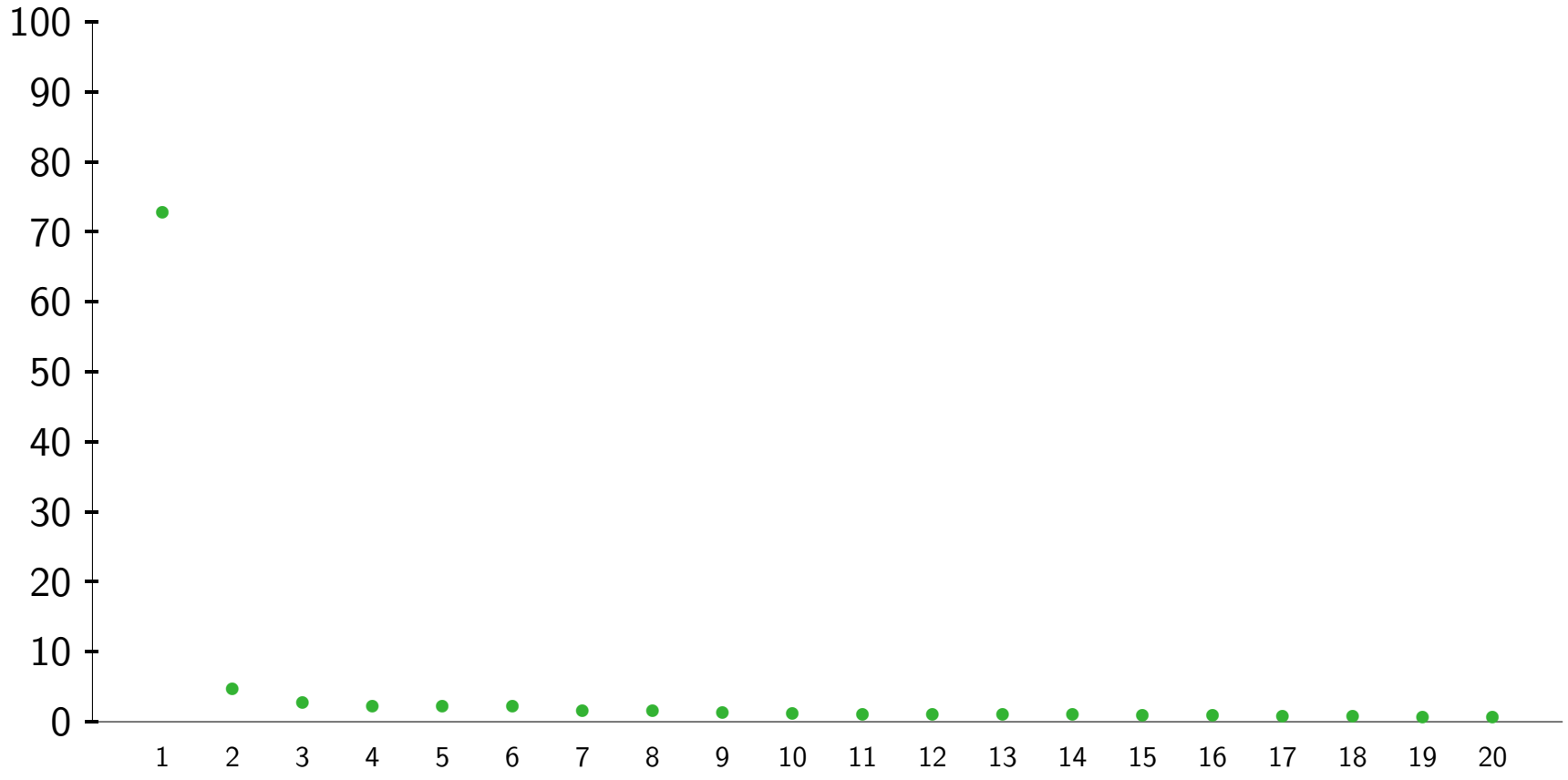


i	$1/\sigma_i^2$	%	cum%
1	21.778	67.75	67.75
2	2.144	6.67	74.43
3	1.000	3.11	77.54
4	1.000	3.11	80.65
5	1.000	3.11	83.76









i	$1/\sigma_i^2$	%	cum%
1	13.471	72.76	72.76
2	0.863	4.66	77.42
3	0.492	2.66	80.08
4	0.405	2.19	82.26
5	0.405	2.19	84.45

Scalable Distance Embedding

The weighted raw stress criterion is

$$\sigma(X) = \sum_{i<j} w_{ij} \left[d_{ij}(X) - \delta_{ij} \right]^2 ,$$

where $d_{ij} = \|x_i - x_j\|$ and $w_{ij} \geq 0$. A popular embedding method is to construct an initial configuration by CMDS/PCA, then minimize $\sigma()$ by the *Guttman majorization algorithm* (GMA), a fixed point method that converges to stationary X. Instead. . .

1. Construct an initial configuration by the *method of standards*:
   - Embed a fixed number of *anchor points*; then, individually position the remaining points in relation to the anchor points. This construction is $O(n)$.
   - Instead of minimizing a traditional error criterion, use a fast heuristic that solves $Ax = b_i$, where A is $d \times d$, for $b_1, \ldots, b_{n-d-1}$.

2. Decrease $\sigma()$ by several iterations of a new *diagonal majorization algorithm* (DMA). Use $O(n)$ dissimilarities and stop after a fixed number of iterations.

Guttman Majorization Algorithm

The stationary equation $\nabla\sigma(X) = 0$ can be written as $VX = B(X)X$, where

$$V = \sum_{i<j} w_{ij} (e_i - e_j) (e_i - e_j)^t$$

and $B(X)$ are $n \times n$ matrices. This suggests an iterative algorithm:
choose X_{k+1} to solve

$$VX = B(X_k) X_k, \quad \text{i.e., solve } d \text{ linear systems, } \quad Vx = b_k.$$

GMA has traditionally been written as

$$X_{k+1} = V^\dagger B(X_k) X_k = \Gamma(X_k),$$

where Γ is the Guttman transform. This representation creates the misleading impression that implementing GMA necessitates computing $V^\dagger$, which is generally expensive when n is large.

GMA: Equal Weights

If $w_{ij} = c$ for $i \neq j$, then

$$\begin{aligned} V &= c\sum_{i<j} (e_i - e_j)(e_i - e_j)^t = cn\left(I - \frac{ee^t}{n}\right), \\ V^\dagger &= \frac{1}{cn}\left(I - \frac{ee^t}{n}\right), \quad \text{and} \\ B(X)X &= c\sum_{i<j} \frac{\delta_{ij}}{d_{ij}(X)} (e_i - e_j)(e_i - e_j)^t X. \end{aligned}$$

$B(X)X$ is already centered, so $V^\dagger B(X)X$ is

$$\frac{1}{n}\sum_{i<j} \frac{\delta_{ij}}{d_{ij}(X)} \begin{pmatrix} y_{ij1}^t \\ \vdots \\ y_{ijn}^t \end{pmatrix}, \quad \text{where} \quad y_{ijs} = \begin{Bmatrix} x_i - x_j & s = i \\ x_j - x_i & s = j \\ 0 & s \neq i, j \end{Bmatrix}.$$

GMA: General Weights

Because col$(V) = e^{\perp}$, $\tilde{V} = V + ee^t > 0$.

Instead of computing $V^\dagger$ and $V^\dagger B(X_k)X_k$, we can obtain X_{k+1} by solving

$$\tilde{V}X = B\left(X_k\right)X_k = B_k.$$

To compute K iterations, one must solve $\tilde{V}x = b$ with Kd choices of b. Because $\tilde{V} > 0$, we can do so as follows:

1. Compute the Cholesky decomposition $\tilde{V} = LL^t$.
   This requires approximately $n^3/6$ multiplications.

2. To solve each $LL^t x = b$,

   (a) Backsolve the triangular system $Ly = b$ to obtain $\tilde{y}$; then
   (b) Backsolve the triangular system $L^t x = \tilde{y}$ to obtain $\tilde{x}$.

   This requires approximately dn^2 multiplications per iteration.

Diagonal Majorization Algorithm

Because $e^t X = 0$, an iteration of GMA is

$$\begin{aligned} X_{k+1} &= X_k - X_k + X_{k+1} = X_k - V^\dagger V X_k + V^\dagger B\left(X_k\right) X_k \\ &= X_k - \frac{1}{2} V^\dagger \, 2\left[V X_k - B\left(X_k\right) X_k\right] = X_k - \frac{1}{2} V^\dagger \, \nabla \sigma\left(X_k\right). \end{aligned}$$

DMA replaces V with $2\,\mathrm{diag}(V)$: an iteration of DMA is

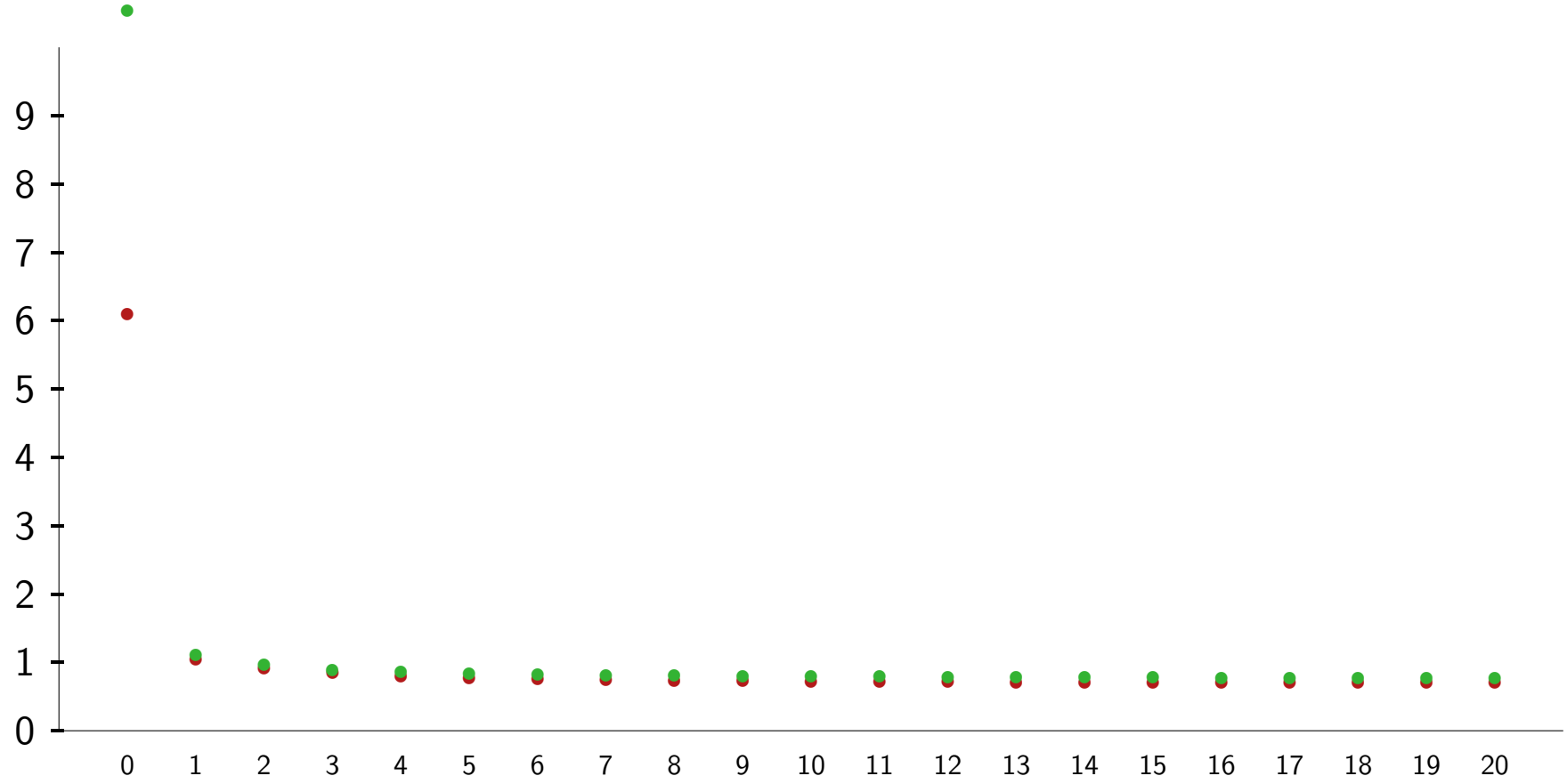
$$X_{k+1} = X_k - \frac{1}{2}\left[2\,\mathrm{diag}(V)\right]^{-1} \nabla \sigma\left(X_k\right) = X_k + \frac{1}{2}\,\mathrm{diag}(V)^{-1}\left[B\left(X_k\right) - V\right] X_k.$$

After computing $[B(X_k) - V]X_k$, DMA requires $2pn$ additional multiplications per iteration. In contrast, after computing $B(X_k)X_k$, GMA with general w_{ij} requires pn^2 additional multiplications per iteration, plus an initial $n^3/6$ multiplications to compute a Cholesky factor.

Experiment 1

$n = 2818$ documents, $d = 5$, equal weights.

Raw Stress Criterion, 20 iterations of PCA-GMA v LIN-GMA,



How much does this cost?

Computational expense (in seconds):

	PCA-GMA	LIN-GMA
Data input	45.48	45.08
Preprocessing	0.16	0.16
Initial configuration	9.11	$<$0.01
Recover dissimilarities	0.32	
Initial stress evaluation	0.42	0.46
20 iterations w/ stress evaluation	24.28	24.16

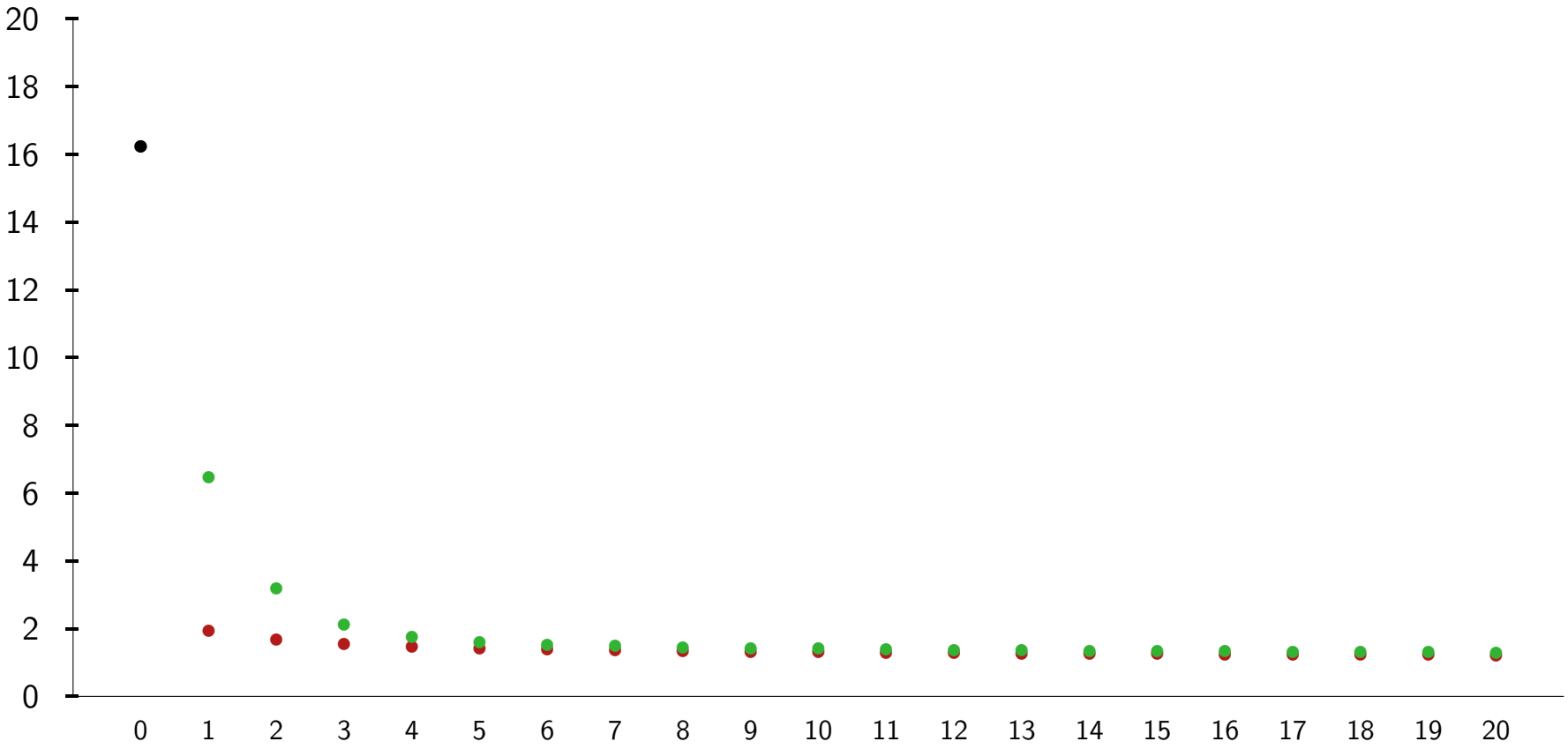
Notice that. . .

- PCA is surprisingly affordable, but orders of magnitude more expensive than LIN. LIN-GMA with one iteration is substantially less expensive and produces a substantially better configuration than PCA.

- Stress evaluation is expensive. Unlike general methods for numerical optimization, GMA does not require evaluation of the objective function—we only computed values in order to monitor progress. Eliminating this extravagance substantially decreases the expense of GMA.

Now let $w_{ij} = 0.4/(0.4 + \delta_{ij})$ when $\delta_{ij} < 0.6$ and $w_{ij} = 0$ otherwise, resulting in 58% of the pairs having zero weight.

Weighted Raw Stress Criterion, 20 iterations of LIN-GMA v LIN-DMA



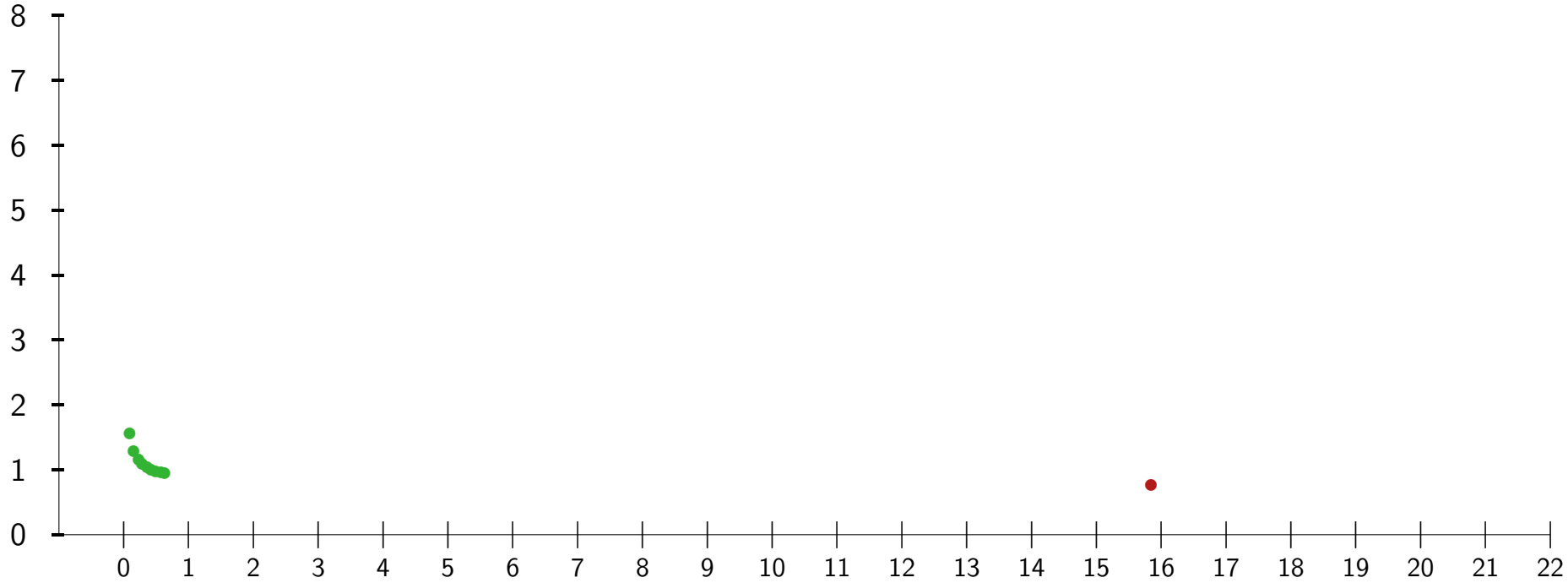
How much does this cost?

Computational expense (in seconds):

	LIN-GMA	LIN-DMA
Data input	45.23	45.54
Preprocessing	0.30	0.33
Initial configuration	0.01	0.01
Initial stress evaluation	0.26	0.28
Cholesky factorization	47.96	
20 iterations w/ stress evaluation	23.01	14.07

Finally, we set $w_{ij} = 1$ for $6k$ "cycles" of δ_{ij}, e.g., $i \leftrightarrow i \pm 1$, $i \leftrightarrow i \pm 2$, etc. and $w_{ij} = 0$ otherwise. (Thus, for $k = 7$, DMA uses $\approx 3\%$ of the 3969153 δ_{ij}.)

We compare the cpu-stress tradeoff for 20 iterations of LIN-GMA with all $w_{ij} = 1$ versus LIN-DMA with $k = 1 : 9$.



For d and k fixed, a fixed number of iterations of LIN-DMA requires $O(n)$ operations. We expect the tradeoff to increasingly favor LIN-DMA as n increases.

Experiment 2

$n = 17,000$ objects, $n(n-1)/2 = 144,491,500$ pairwise dissimilarities.

$y_1, \ldots, y_n \in \Re^5$

$\delta_{ij} = \|y_i - y_j\| \cdot \exp(Z/100)$, where $Z \sim \text{Normal}(0,1)$

Embed $\Delta = [\delta_{ij}]$ in $\Re^5$ using LIN-DMA with 54 cycles.

Initial raw stress criterion:	3,058,174
After 200 DMA iterations:	57,921

2 stress evaluations:	86.5 seconds
Total embedding time:	66.0 seconds

For more information, manuscripts, and source code, please visit

`http://www.math.wm.edu/~trosset/X-MDS`