Beyond projections

PCA and SVD find informative linear projections. Given a data set in \( \mathbb{R}^d \), and a number \( k < d \), they:

- Find orthogonal directions \( u_1, \ldots, u_k \in \mathbb{R}^d \)
- Approximate points in \( \mathbb{R}^d \) by their projection into the subspace spanned by these directions

What if the data lies on (or near) a nonlinear surface?

Embeddings and low-dimensional manifolds

CSE 250B

Low dimensional manifolds

Sometimes data in a high-dimensional space \( \mathbb{R}^d \) in fact lies close to a \( k \)-dimensional manifold, for \( k \ll d \)

1. Motion capture
   - \( M \) markers on a human body yields data in \( \mathbb{R}^{3M} \)
2. Speech signals
   - Representation can be made arbitrarily high dimensional by applying more filters to each window of the time series

This whole area: “Manifold learning”

The ISOMAP algorithm

ISOMAP (Tenenbaum et al, 1999): given data \( x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d \),

1. Estimate geodesic distances between the data points: that is, distances along the manifold.
2. Then embed these points into Euclidean space so as to (approximately) match these distances.
Estimating geodesic distances

Key idea: for nearby pairs of points, Euclidean distance and geodesic distance are approximately the same.

1. Construct the neighborhood graph.
   Given data \( x^{(1)}, \ldots, x^{(n)} \), construct a graph \( G = (V, E) \) with
   - Nodes \( V = \{1, 2, \ldots, n\} \) (one per data point)
   - Edges \((i, j) \in E\) whenever \( x^{(i)} \) and \( x^{(j)} \) are close together

2. Compute distances in this graph.
   Set length of any \((i, j) \in E\) to \( \|x^{(i)} - x^{(j)}\| \).
   Compute all pairwise distances between nodes.

Distance-preserving embeddings

The algorithmic task:
- Input: An \( n \times n \) matrix of pairwise distances \( D_{ij} = \text{desired distance between points } i \text{ and } j \), as well as an integer \( k \).
- Output: an embedding \( z^{(1)}, \ldots, z^{(n)} \in \mathbb{R}^k \) that realizes these distances as closely as possible.

Most widely-used algorithm: classical multidimensional scaling.
- \( D \in \mathbb{R}^{n \times n} \): matrix of desired squared interpoint distances.
- Schoenberg (1938): \( D \) can be realized in Euclidean space if and only if \( B = -\frac{1}{2}HDH \) is p.s.d., where \( H = I_n - \frac{1}{n}11^T \).
- In fact, looking at this matrix \( B \) suggests an embedding even if it is not positive semidefinite.

The Gram matrix

In Euclidean space, write squared distances using dot products:
\[
\|x - x'\|^2 = \|x\|^2 + \|x'\|^2 - 2x \cdot x' = x \cdot x + x' \cdot x' - 2x \cdot x'.
\]

What about expressing dot products in terms of distances?

Let \( z^{(1)}, \ldots, z^{(n)} \) be points in Euclidean space.
- Let \( D_{ij} = \|z^{(i)} - z^{(j)}\|^2 \) be the squared interpoint distances.
- Let \( B_{ij} = z^{(i)} \cdot z^{(j)} \) be the dot products. “Gram matrix”

Moving between \( D \) and \( B \):
- We’ve seen: \( D_{ij} = B_{ii} + B_{jj} - B_{ij} - B_{ji} \). So \( D \) is linear in \( B \).
- For \( H = I_n - \frac{1}{n}11^T \), we have \( B = -\frac{1}{2}HDH \).

And we can read off an embedding from the Gram matrix.

Question

Consider the two points \( x_1 = (1, 0) \) and \( x_2 = (-1, 0) \) in \( \mathbb{R}^2 \).

1. What is the matrix \( D \) of squared interpoint distances?
2. Write down \( H = I_n - \frac{1}{n}11^T \).
3. Compute \( B = -\frac{1}{2} HDH \). Is this the correct Gram matrix?
Recovering an embedding from interpoint distances

\[ D \in \mathbb{R}^{n \times n}; \] matrix of desired squared interpoint distances.

Suppose these are realizable in Euclidean space: that is, there exist vectors \( z^{(1)}, \ldots, z^{(n)} \) such that \( D_{ij} = \| z^{(i)} - z^{(j)} \|^2 \).

We can easily obtain the Gram matrix, \( B_{ij} = z^{(i)} \cdot z^{(j)} \).

- \( B \) is p.s.d. (why?). Thus its eigenvalues are nonnegative.
- Compute spectral decomposition:
  \[ B = U \Lambda U^T = YY^T, \]
  where \( \Lambda \) is the diagonal matrix of eigenvalues and \( Y = U \Lambda^{1/2} \).
- Denote the rows of \( Y \) by \( y^{(1)}, \ldots, y^{(n)} \). Then
  \[ y^{(i)} \cdot y^{(j)} = (YY^T)_{ij} = B_{ij} = z^{(i)} \cdot z^{(j)}. \]

If dot products are preserved, so are distances.

Result: an embedding \( y^{(1)}, \ldots, y^{(n)} \) that exactly replicates \( D \).

What is the dimensionality of this embedding?

Classical multidimensional scaling

A slight generalization works even when the distances cannot necessarily be realized in Euclidean space.

Given \( n \times n \) matrix \( D \) of squared distances, target dimension \( k \):

1. Compute \( B = -\frac{1}{2} HDH \).
2. Compute the spectral decomposition \( B = U \Lambda U^T \), where the eigenvalues in \( \Lambda \) are arranged in decreasing order.
3. Zero out any negative entries of \( \Lambda \) to get \( \Lambda_+ \).
4. Set \( Y = U \Lambda_+^{1/2} \).
5. Set \( Y_k \) to the first \( k \) columns of \( Y \).
6. Let the embedding of the \( n \) points be given by the rows of \( Y_k \).

More manifold learning

1. Other good algorithms, such as
   - Locally linear embedding
   - Laplacian eigenmaps
   - Maximum variance unfolding
2. Notions of intrinsic dimensionality
3. Statistical rates of convergence for data lying on manifolds
4. Capturing other kinds of topological structure