The Quest for Variance : PCA, MDS and ISOMAP

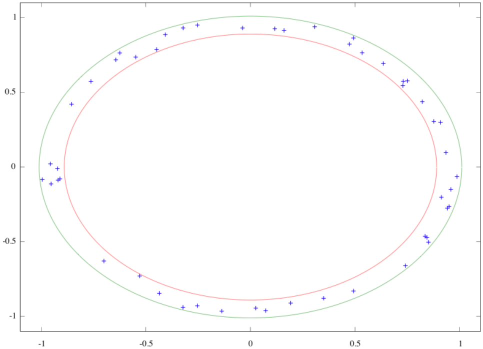
Tyrus Berry
George Mason University

June 26, 2017

The Curse of Dimensionality

Too Much Space, Too Little Data

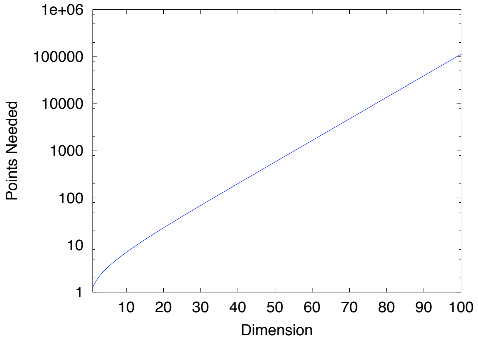
- How many points do we need to be 95% confident we have a hole of radius $r \leq .9$?



Number of points needed vs. Dimension: $N \cong 1.1^n$

- Volume of n-ball: $V_n(r) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} r^n$
- Probability of a uniform random point having $r < 0.9$ is the volume ratio
- Percent of volume with $r < 0.9$ is $V_n(0.9)/V_n(1) = 0.9^n$
- Probability of N points randomly falling in the outer shell: $P(r_1, ..., r_N \in [0.9, 1]) = (1 - 0.9^n)^N$
- We are 95% certain there is a hole if $(1 - 0.9^n)^N < 0.05$
- We need $N > \frac{\log(0.05)}{\log(1-0.9^n)} \approx \frac{3}{0.9^n} \propto 1.1^n$

Number of points needed vs. Dimension: $N \cong 1.1^n$



Dimensionality Reduction Goals

- Find new coordinates in Lower Dimensional Space
- Preserve Desired Features of Data:
  - Variances and Distances
  - Topology
  - Geometry
- Minimize Reconstruction Error

Dimensionality Reduction Goals

- Reduce redundancy in the data
- In general: $0 = f(x_1, x_2, ..., x_n)$ is redundant
- More simple: $x_1 = f(x_2, ..., x_n)$
- Even simpler: $x_1 = a_2 x_2 + a_3 x_3 + \cdots + a_n x_n + c$
- How can we detected redundant variables?
- Simple method: Covariance detects *linear* redundancy

Covariance

- Let $\{x(i)\}_{i=1}^N \subset \mathbb{R}^m$ be data points
- Let X by an $m \times N$ matrix with $x(i)$ as the i-th column
- So $X_{ji} = x(i)_j$ is the j-th variable of the i-th data point
- Let $\mu_j = \frac{1}{N}\sum_{i=1}^N X_{ji}$ be the mean
- The covariance of the j-th and k-th variables is

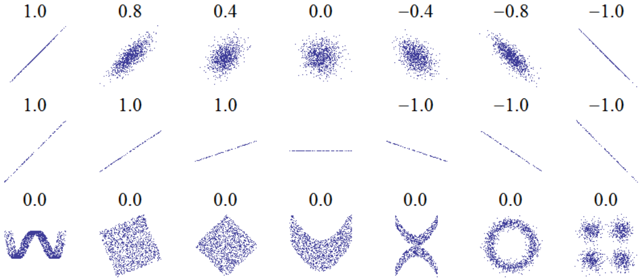
$$S_{jk} \equiv \frac{1}{N}\sum_{i=1}^{N}(X_{ji} - \mu_j)(X_{ki} - \mu_k)$$

- If we redefine X by subtracting μ from each column:

$$S_{jk} = \frac{1}{N}(XX^\top)_{jk}$$

Covariance

- When $S_{jk} = 0$ the j-th and k-th variables are uncorrelated
- When S is diagonal the data are uncorrelated
- Warning: Uncorrelated does not imply independent:



Covariance

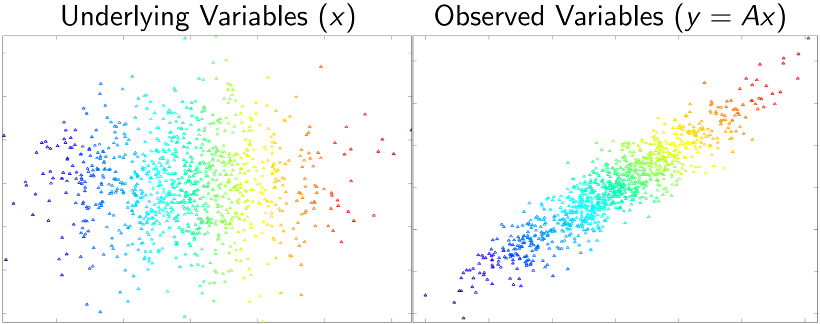
- When $S_{jk} = 0$ the j-th and k-th variables are uncorrelated
- When $S = \frac{1}{N} XX^\top$ is diagonal the data are uncorrelated
- If the data is uncorrelated and $S_{jj} \neq 0$ there are no *linear* redundancies:
  - A linear redundancy says $a_1 x_1 + \cdots + a_n x_n = 0$
  - In terms of X this says that $\vec{a}^\top X = a_1 X_{1i} + \cdots + a_n X_{ni} = 0$
  - This implies $\vec{a}^\top XX^\top \vec{a} = 0$ and $\vec{a}^\top S \vec{a} = 0$
  - Since S is diagonal we have $0 = \vec{a}^\top S \vec{a} = \sum_j S_{jj} a_j^2$
  - Since $S_{jj} > 0$ we must have $a_j = 0$.

Linear Model

PCA assumes a Linear Model:

- Underlying Variables $x \in \mathbb{R}^m$ are mean zero, uncorrelated
- Observed Variables $y \in \mathbb{R}^n$ are given by $y = Ax$
- Assume $n > m$ but $Rank(A) = m$ is unknown.

PCA Schematic



PCA is based on Linear Correlation

- Let X by an $m \times N$ matrix with $x(i)$ as the i-th column
- Let $Y = AX$ be the $n \times N$ matrix with $y(i)$ as the i-th column
- We are only given Y, these are *observed* data points
- Since the coordinates of X are uncorrelated, $\frac{1}{N}XX^\top = S$ where S is diagonal with

$$S_{jj} = \mathrm{var}(x_j) \approx \frac{1}{N}\sum_i X_{ji}^2$$

- Thus, $\frac{1}{N}YY^\top = \frac{1}{N}AXX^\top A^\top = ASA^\top$

PCA assumes latent variables are uncorrelated

- We can compute: $\frac{1}{N}YY^\top = ASA^\top$
- Note that $\frac{1}{N}YY^\top$ is symmetric and positive semi-definite
- So it has an eigen-decomposition $\frac{1}{N}YY^\top = U\Lambda U^\top$

1. PCA: Assume that A is orthogonal, so that $A = U$ and $S = \Lambda$.
   - We can recover X by computing $U^\top Y = U^\top AX$
   - The entries of Λ tell us the variance of the coordinates of x.

PCA assumes latent variables are uncorrelated

- We can compute: $\frac{1}{N}YY^\top = ASA^\top$
- Note that $\frac{1}{N}YY^\top$ is symmetric and positive semi-definite
- So it has an eigen-decomposition $\frac{1}{N}YY^\top = U\Lambda U^\top$

1. PCA: Assume that A is orthogonal, so that $A = U$ and $S = \Lambda$.
   - We can recover X by computing $U^\top Y = U^\top AX$
   - The entries of Λ tell us the variance of the coordinates of x.

PCA Algorithm

- Inputs: Observed data matrix Y and number of PCA modes k
- Output: Recovered intrinsic variables X and reconstructed $\tilde{Y}$
- Step 1: Compute the mean $\mu_j = \frac{1}{N}\sum_{i=1}^{N} Y_{ji}$
- Step 2: Center the data: Subtract μ from each column of Y
- Step 3: Compute the singular value decomposition (SVD) of Y: $Y = USV^\top$ (note: $YY^\top = US^2U^\top$)
- Step 4: Select the top k singular vectors $U = U(:, 1:k)$
- Step 5: Project onto the principal components $X = U^\top Y$
- Step 6: Reconstruct $\tilde{Y} = UX + \mu$ (add μ to each column)

Linear Model Example: Noise Reduction

- PCA projects onto the largest linear component(s):

(Loading Video...)

Nonlinear Model Example: Noise Reduction

- PCA can only make linear projections:

(Loading Video...)

Gram matrices

- Let G be an $N \times N$ symmetric positive semi-definite matrix
- Then $G = V\Lambda_{\text{MDS}}V^\top$ with m positive eigenvalues
- Let $X = I_{m\times N}\Lambda_{\text{MDS}}^{1/2}V^\top$ so X is $m \times N$
- Let $x(i) \in \mathbb{R}^m$ be the i-th column of X
- Notice that $G_{ij} = (X^\top X)_{ij} = \sum_{l=1}^m X_{li}X_{lj} = x(i)\cdot x(j)$
- We say that G is the **Gram matrix** of a data set $\{x(i)\}$ if the entries of G are the pairwise inner products of the data points

Theorem: For any symmetric positive semi-definite $N \times N$ matrix, there exists an uncorrelated data set $\{x(i)\}_{i=1}^N \subset \mathbb{R}^m$ where $m = \text{rank}(G)$ such that G is the Gram matrix of $\{x(i)\}$. We call x the coordinates of G, notice that $XX^\top$ is diagonal.

MDS preserves inner products

- Same context as PCA: $Y = AX$
- Instead of correlations, compute the Gram matrix $G = Y^\top Y$
- If A is orthogonal, the $G = Y^\top Y = X^\top A^\top AX = X^\top X$
- Compute the eigen-decompositon of $G = V \Lambda_{\text{MDS}} V^\top$
- **Dimensionality Reduction:** Set $\tilde{X} = I_{p\times N}\Lambda_{\text{MDS}}^{1/2} V^\top$
- $\tilde{X}$ are the p-dimensional coordinates with the closest Gram matrix to X, minimizes the residual R (Frobenius norm):

$$G = X^\top X = \tilde{X}^\top \tilde{X} + \sum_{j=p+1}^{N} (\lambda_{\text{MDS}})_j v(j) v(j)^\top = \tilde{G} + R$$

Equivalence of MDS and PCA

- PCA: $\frac{1}{N}YY^\top = U\Lambda_{\text{PCA}}U^\top$ set $X_{\text{PCA}} = I_{p\times N}U^\top Y$
- MDS: $Y^\top Y = V\Lambda_{\text{MDS}}V^\top$, set $X_{\text{MDS}} = I_{p\times N}\Lambda_{\text{MDS}}^{1/2}V^\top$
- Singular value decomposition: $Y = USV^\top$, $S = \Lambda_{\text{MDS}}^{1/2}$

$$X_{\text{PCA}} = I_{p\times N}U^\top Y = I_{p\times N}U^\top U\Lambda_{\text{MDS}}^{1/2}V^\top = X_{\text{MDS}}$$

- PCA/MDS preserve variance (maximal variance projection), inner products, and Euclidean distances:

$$||x(i)-x(j)||^2 = x(i)\cdot x(i)+x(j)\cdot x(j)-2x(i)\cdot x(j) = G_{ii}+G_{jj}-2G_{ij}$$

Why do we need MDS?

- PCA needs the coordinates of Y to compute correlations
- MDS appears to need the coordinates of Y to compute the Gram matrix
- Actually, Gram matrix can be reconstructed from pairwise distances
- This means we can start with a collection of distances
- These distances don't need to be Euclidean!

Double Centering

- Double centering recovers the Gram matrix from the matrix of pairwise distances
- Let $D_{ij} = ||x(i) - x(j)||^2 = x(i) \cdot x(i) + x(j) \cdot x(j) - 2x(i) \cdot x(j)$
- Assume $\frac{1}{N}\sum_i x(i) = 0$ and $\frac{1}{N}\sum_i x(i) \cdot x(i) = \sigma^2$
- Then $\frac{1}{N}\sum_i D_{ij} = \sigma^2 + x(j) \cdot x(j)$ and $\frac{1}{N^2}\sum_{i,j} D_{i,j} = 2\sigma^2$ so

$$-\frac{1}{2}\left(D_{ij} - \frac{1}{N}\sum_i D_{ij} - \frac{1}{N}\sum_j D_{ij} + \frac{1}{N^2}\sum_{i,j} D_{ij}\right) = x(i) \cdot x(j) = G_{ij}$$

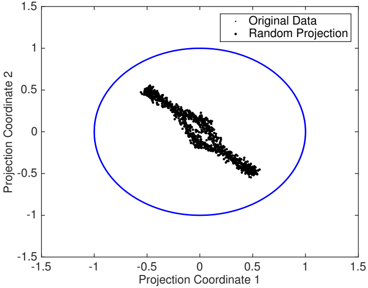
Double Centering: Let $\mathbb{1}$ be the $N \times N$ matrix of all 1's, then

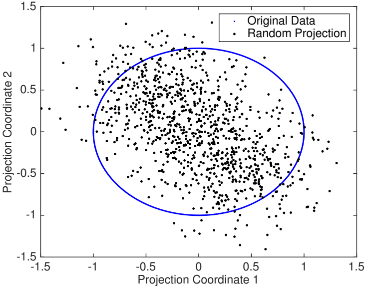
$$G = -\frac{1}{2}\left(D - D\mathbb{1}/N - \mathbb{1}D/N + \mathbb{1}D\mathbb{1}/N^2\right) = -\frac{1}{2}(\mathrm{Id} - \mathbb{1})D(\mathrm{Id} - \mathbb{1})$$

The Geometric Prior

- Assume data are sampled from a compact Riemannian manifold embedded in $\mathbb{R}^n$
- Example: Generate 1000 data points $(x_i, y_i)^\top$ on a unit circle in $\mathbb{R}^2$ let X be the 2×1000 matrix containing this data.
- Embed the circle into $\mathbb{R}^{10}$ using a random orthogonal matrix U ($U^\top U = I$) which is 10×2 so that $Y = UX$ is 10×1000.
- Also consider the more complex embedding $Y = [X \ \ (UX)^3]$ (where U is 8×2 and the cube is entrywise).
- Add some 10-dimensional Gaussian noise to Y.

The Geometric Prior

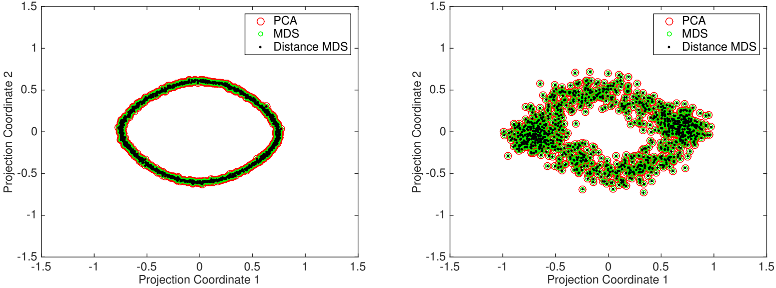




PCA for Nonlinear Dimensionality Reduction

- Assume data lies on a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^n$ with $n >> d$.
- Sard's Lemma: A randomly chosen linear projection from $\mathbb{R}^n$ to $\mathbb{R}^{2d+1}$, will preserve the topology of $\mathcal{M}$.
- PCA is Topology preserving
- Problem: What about the geometry of $\mathcal{M}$?
- Answer: PCA attempts to preserve Euclidean distances, long Euclidean distances do not respect the nonlinear structure, but short distances do (locally approximately linear).

PCA/MDS/Distance MDS for Nonlinear Data



Modified Distance MDS for Nonlinear Dimensionality Reduction

- PCA is Topology preserving
- Problem: What about the geometry of $\mathcal{M}$?
- Answer: PCA attempts to preserve Euclidean distances, long Euclidean distances do not respect the nonlinear structure, but short distances do (locally approximately linear).
- Distance MDS lets us play with the distances!
- Simple Idea: Very short distance = noise. Very long distance = Not meaningful. Weight distances by $e^{-(D-\mu)^2/\sigma}$.

ISOMAP

- PCA is Topology preserving
- Problem: What about the geometry of $\mathcal{M}$?
- Answer: ISOMAP replaces Euclidean distances with Graph Distances (shortest path in a kNN graph) which approximate Geodesic Distances.
- Geometry Preserving.
- Not very robust to noise.

Kernel PCA

- Forget the distances altogether!
- Define a kernel, such as $J(x, y) = e^{-||x-y||^2/\epsilon}$
- Evaluate kernel on all pairs of data points $J_{ij} = J(x_i, x_j)$.
- If matrix J is symmetric and positive definite it defines an embedding!
- Eigenvectors of matrix J give new coordinates for the data (MDS).
- We can interpret the kernel $J(x, y)$ as inner product

$$J(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{R}^m}$$