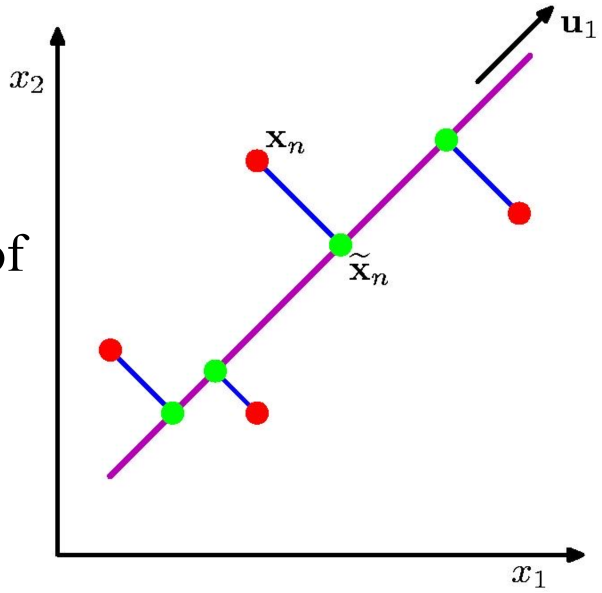
Dimension Reduction

Wei-Ta Chu

1.1 Principal Component Analysis (PCA)

- Widely used in dimensionality reduction, lossy data compression, feature extraction, and data visualization
- Also known as Karhunen-Loeve transform
- Two commonly-used definitions
  - Orthogonal projection of the data onto a lower dimensional linear space such that the variance of the projected data is maximized.
  - Linear projection that minimizes the average projection cost



C.M. Bishop, Chapter 12 of Pattern Recognition and Machine Learning, Springer, 2006.

Maximum Variance Formulation

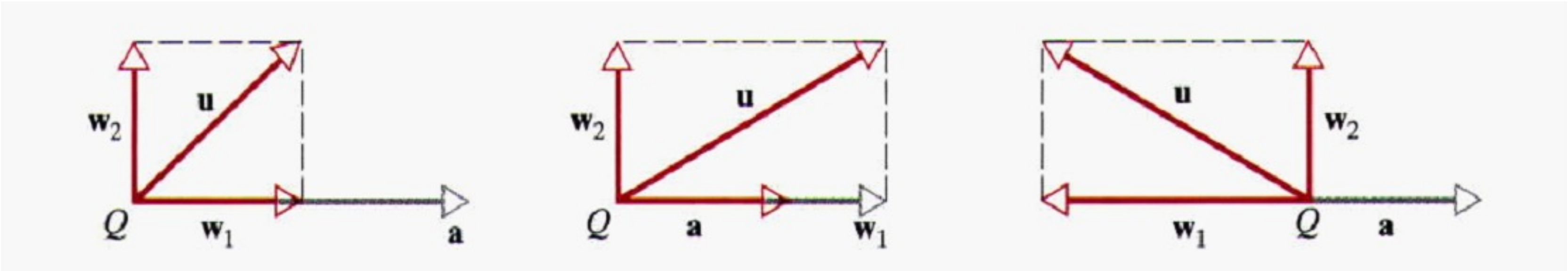
- Data set of observation $\{\boldsymbol{x}_n\}$ with dimensionality D.
- Goal: project the data onto a space having dimensionality $M < D$ with maximizing the variance of the projected data. Assume the value of M is given.
- Begin with M=1. Data are projected onto a line in a D-dimensional space. The direction of the line is denoted by a D-dimensional vector $\boldsymbol{u}_1$.
- Each data point $\boldsymbol{x}_n$ is then projected onto a scalar value $\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{x}_n$.

LA Recap: Orthogonal Projection

$$\text{proj}_{\mathbf{a}}\,\mathbf{u} = \frac{\mathbf{u}\cdot\mathbf{a}}{\|\mathbf{a}\|^2}\mathbf{a} \qquad \text{(vector component of } \mathbf{u} \text{ along } \mathbf{a}\text{)}$$

$$\mathbf{u} - \text{proj}_{\mathbf{a}}\,\mathbf{u} = \mathbf{u} - \frac{\mathbf{u}\cdot\mathbf{a}}{\|\mathbf{a}\|^2}\mathbf{a} \qquad \text{(vector component of } \mathbf{u} \text{ orthogonal to } \mathbf{a}\text{)}$$

$$\|\text{proj}_{\mathbf{a}}\,\mathbf{u}\| = \frac{|\mathbf{u}\cdot\mathbf{a}|}{\|\mathbf{a}\|} = \|\mathbf{u}\|\,|\cos\theta|$$



Maximum Variance Formulation

- The mean of the projected data is $\boldsymbol{u}_1^T \bar{\boldsymbol{x}}$

$$\bar{\boldsymbol{x}} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n$$

- The variance of the projected data is given by

$$\frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{u}_1^T \boldsymbol{x}_n - \boldsymbol{u}_1^T \bar{\boldsymbol{x}} \right)^2 = \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1$$

- Where $\boldsymbol{S}$ is the covariance matrix defined by

$$\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^T$$

Maximum Variance Formulation

- Maximize the projected variance $\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1$ with respect to $\boldsymbol{u}_1$

  $$\|\boldsymbol{u}_1\| = \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

- Introduce a Lagrange multiplier denoted by λ_1

  $$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1)$$

- By setting the derivative with respect to $\boldsymbol{u}_1$ equal to zero, we see that this quantity will have a stationary point when

  $$\boldsymbol{S} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1$$

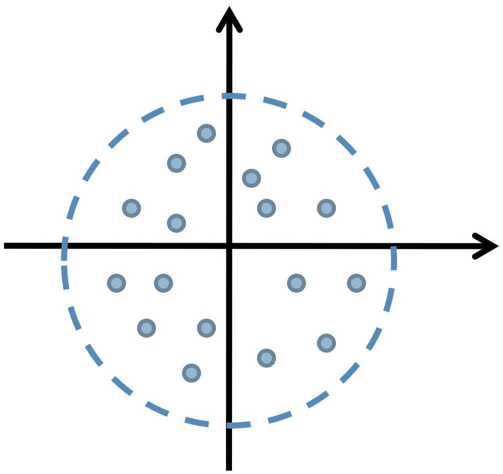
  - $\boldsymbol{u}_1$ must be an eigenvector of $\boldsymbol{S}$
  - The variance will be a maximum when we set $\boldsymbol{u}_1$ equal to the eigenvector having the largest eigenvalue λ_1

Maximum Variance Formulation

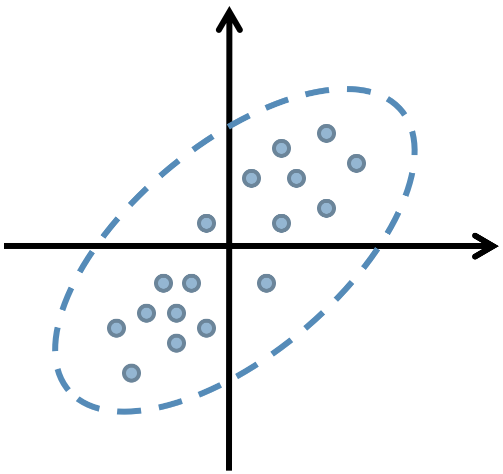
- The optimal linear projection for which the variance of the projected data is maximized is now defined by the M eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ of the data covariance matrix $\boldsymbol{S}$ corresponding to the M largest eigenvalues $\lambda_1,\ldots,\lambda_M$

- Principal component analysis involves evaluating the mean and the covariance matrix of the data set and then finding the M eigenvectors of $\boldsymbol{S}$ corresponding the M largest eigenvalues.

Covariance



High variance, low covariance
→ No inter-dimension dependency



High variance, high covariance
→ inter-dimension dependency

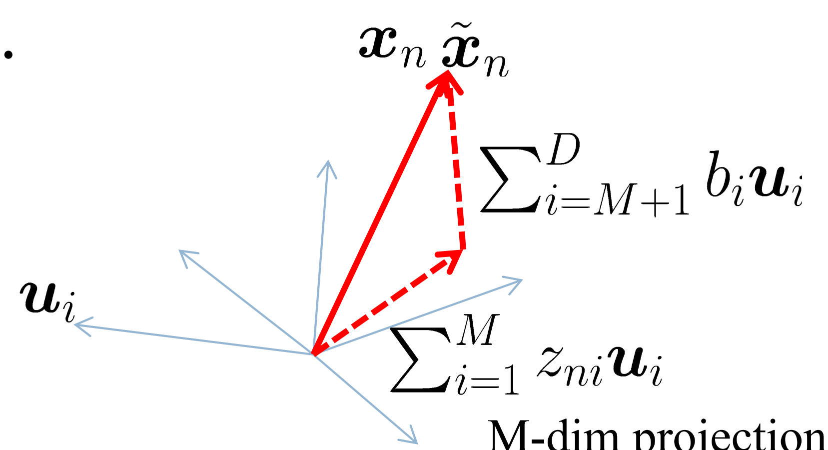
Minimum Error Formulation

- Each data point can be represented by a linear combination of the basis vectors

$$\boldsymbol{x}_n = \sum_{i=1}^{D} \alpha_{ni} \boldsymbol{u}_i \quad \Rightarrow \quad \boldsymbol{x}_n = \sum_{i=1}^{D} (\boldsymbol{x}^T \boldsymbol{u}_i) \boldsymbol{u}_i$$

- Our goal is to approximate this data point using a representation involving a restricted number M < D of variables corresponding to a projection onto a lower-dimensional subspace.

$$\tilde{\boldsymbol{x}}_n = \sum_{i=1}^{M} z_{ni} \boldsymbol{u}_i + \sum_{i=M+1}^{D} b_i \boldsymbol{u}_i$$



Minimum Error Formulation

- Minimize approximation error

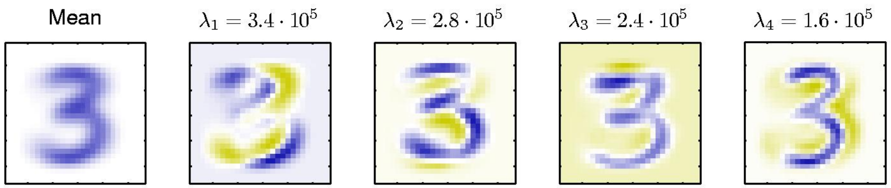
$$J = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 \quad \Rightarrow \quad J = \sum_{i=M+1}^{D} \lambda_i$$

- Obtaining the minimum value of *J* by selecting eigenvectors to those having the *D-M* smallest eigenvalues, and hence the eigenvectors defining the principal subspace are those corresponding to the *M* largest eigenvalues.

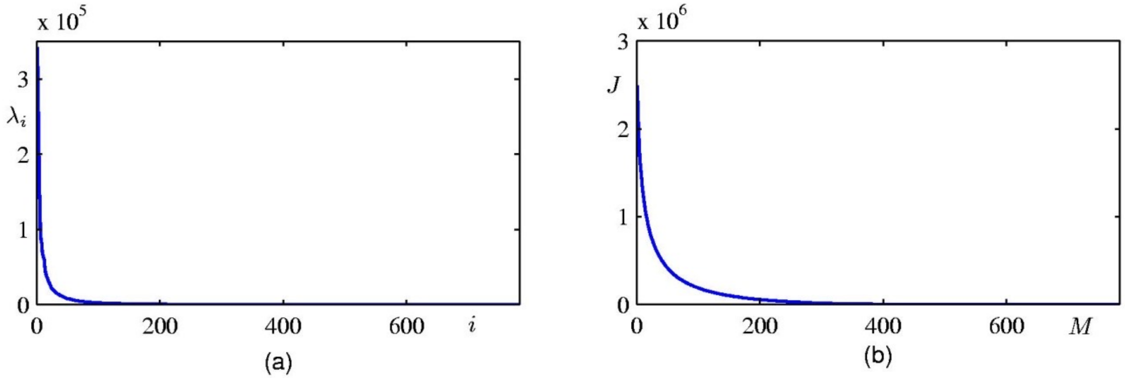
L.I. Smith, "A tutorial on Principal Component Analysis," http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

J. Shlens, "A tutorial on Principal Component Analysis," http://www.cs.cmu.edu/~elaw/papers/pca.pdf

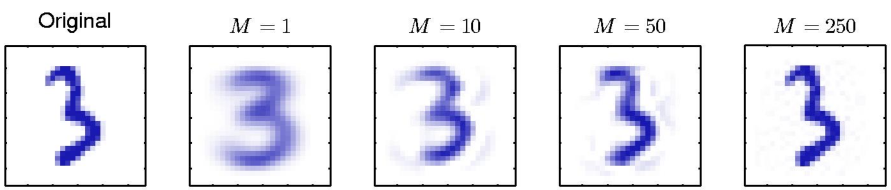
Applications of PCA



Mean vector and the first four PCA eigenvectors for the off-line digits data set



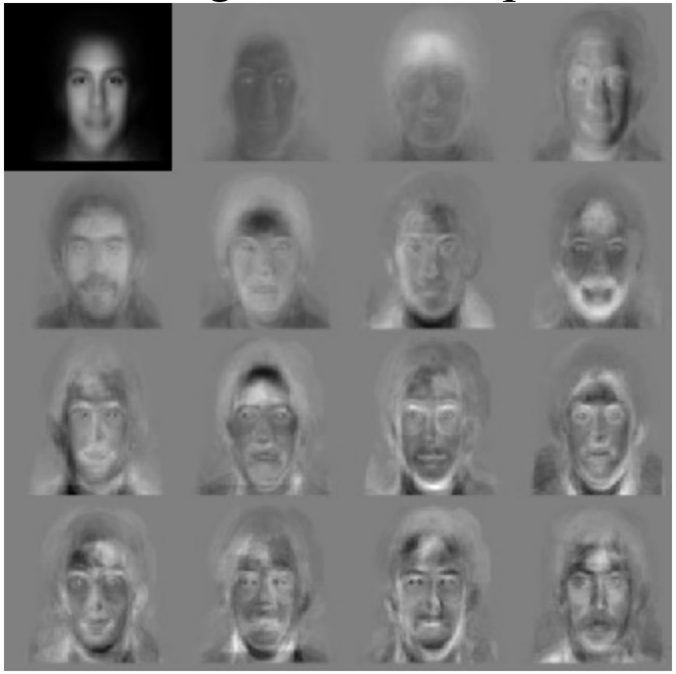
Eigenvalue spectrum and the sum of the discard eigenvalues



An original example together with its PCA reconstructions obtained by retaining M principal components

Eigenfaces

- Eigenfaces for face recognition is a famous application of PCA
  - Eigenfaces capture the majority of variance in face data
  - Project a face on those eigenfaces to represent face features



M. Turk and A.P. Pentland, “Face recognition using eigenfaces,” Proc. of CVPR, pp. 586-591, 1991.

1.2 Singular Value Decomposition (SVD)

- SVD works directly on data
  - PCA works on covariance matrix of data
  - The SVD technique examines the entire set of data and rotates the axis to maximize variance along the first few dimensions.
- Problem:
  - #1: Find concepts in text
  - #2: Reduce dimensionality

term / document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

http://www.cs.cmu.edu/~guestrin/Class/10701-S06/Handouts/recitations/recitation-pca_svd.ppt

SVD - Definition

$$\mathbf{A}_{[\mathbf{n\ x\ m}]} = \mathbf{U}_{[\mathbf{n\ x\ r}]}\ \mathbf{L}_{[\mathbf{\ r\ x\ r}]}\ (\mathbf{V}_{[\mathbf{m\ x\ r}]})^{T}$$

- **A**: *n x m* matrix (e.g., *n* documents, *m* terms)
- **U**: *n x r* matrix (*n* documents, *r* concepts)
- **L**: *r x r* diagonal matrix (strength of each 'concept') (*r*: rank of the matrix)
- **V**: *m* x *r* matrix (*m* terms, *r* concepts)

SVD - Properties

‘spectral decomposition’ of the matrix:

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix}
| & | \\
u_1 & u_2 \\
| & |
\end{bmatrix}
\text{ x }
\begin{bmatrix}
l_1 & \varnothing \\
\varnothing & l_2
\end{bmatrix}
\text{ x }
\begin{bmatrix}
\text{———} & v_1 & \text{———} \\
\text{———} & v_2 & \text{———}
\end{bmatrix}
$$

SVD - Interpretation

'documents', 'terms' and 'concepts':

- **U**: document-to-concept similarity matrix
- **V**: term-to-concept similarity matrix
- **L**: its diagonal elements: 'strength' of each concept

Projection:

- best axis to project on: ('best' = min sum of squares of projection errors)

SVD - Example

- $\mathbf{A} = \mathbf{U}\ \mathbf{L}\ \mathbf{V}^{T}$ - example:

doc-to-concept similarity matrix

CS-concept

MD-concept

$$
\begin{array}{c}
 \\
\text{CS} \\
 \\
 \\
\text{MD} \\

\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

- $\mathbf{A} = \mathbf{U}\ \mathbf{L}\ \mathbf{V}^T$ - example:

'strength' of CS-concept

$$
\begin{array}{c|c}
 & \text{data} \quad \text{inf.} \quad \text{retrieval} \quad \text{brain} \quad \text{lung} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}
$$

SVD - Example

- $\mathbf{A} = \mathbf{U}\ \mathbf{L}\ \mathbf{V}^T$ - example:

term-to-concept similarity matrix

CS-concept

$$\begin{array}{c c}
 & \begin{array}{ccccc}\text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung}\end{array} \\
\begin{array}{c} \\ \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \end{array} &
\left[\begin{array}{ccccc}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}
=
\begin{bmatrix}
0.18 & 0 \\
0.36 & 0 \\
0.18 & 0 \\
0.90 & 0 \\
0 & 0.53 \\
0 & 0.80 \\
0 & 0.27
\end{bmatrix}
\times
\begin{bmatrix}
9.64 & 0 \\
0 & 5.29
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
0 & 0 & 0 & 0.71 & 0.71
\end{bmatrix}$$

SVD – Dimensionality reduction

- Q: how exactly is dim. reduction done?
- A: set the smallest singular values to zero:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & \cancel{0} \\ 0.36 & \cancel{0} \\ 0.18 & \cancel{0} \\ 0.90 & \cancel{0} \\ 0 & \cancel{0.53} \\ 0 & \cancel{0.80} \\ 0 & \cancel{0.27} \end{bmatrix} \text{x} \begin{bmatrix} 9.64 & 0 \\ 0 & \cancel{5.29} \end{bmatrix} \text{x} \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{0} & \cancel{0.71} & \cancel{0.71} \end{bmatrix}$$

SVD - Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ x } \begin{bmatrix} 9.64 \end{bmatrix} \text{ x } \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

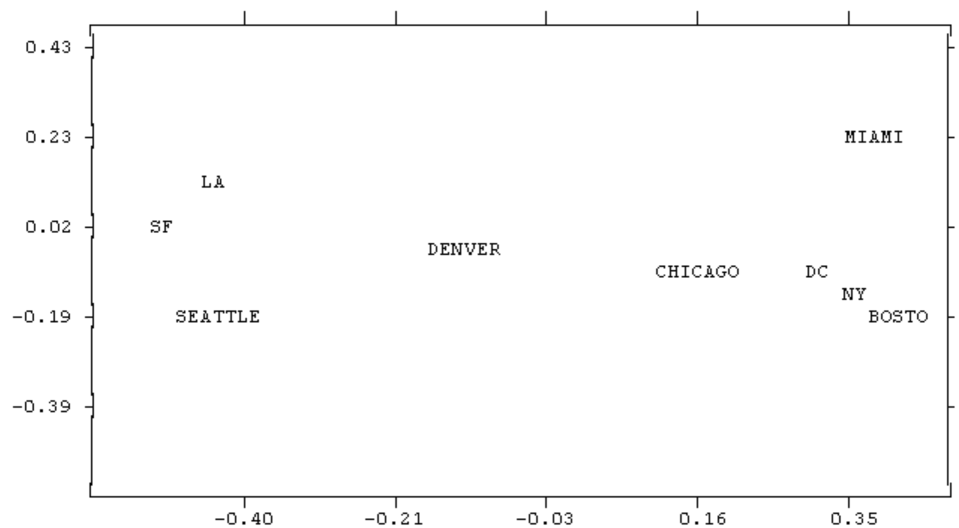
SVD - Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2.1 Multidimensional Scaling (MDS)

- Goal: represent data points in some lower-dimensional space such that the distances between points in that space correspond to the distance between points in the original space

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0



http://www.analytictech.com/networks/mds.htm

Multidimensional Scaling (MDS)

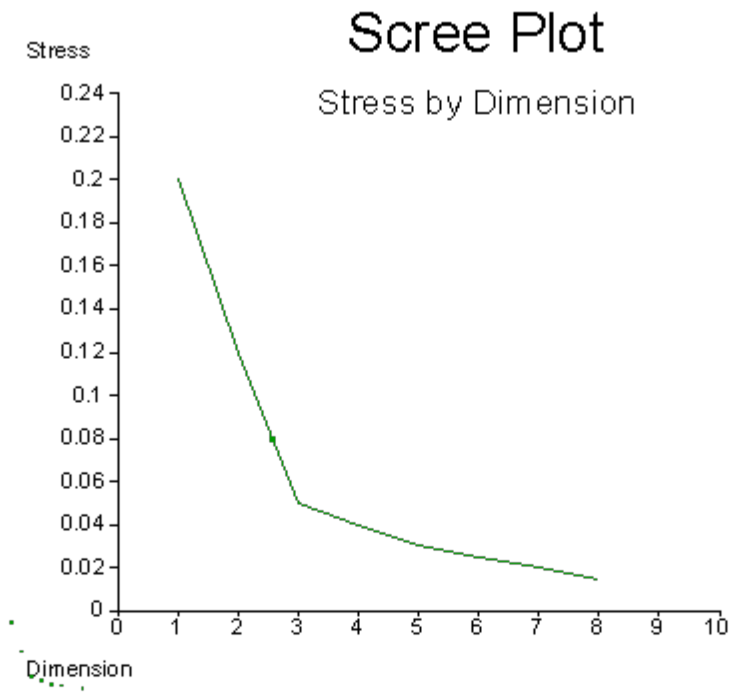
- What MDS does is to find a set of vectors in *p*-dimensional space such that the matrix of Euclidean distances among them corresponds as closely as possible to some function of the input matrix according to a criterion function called *stress*.
- Stress: the degree of correspondence between the distances among points implied by MDS map and the input matrix.

$$\sqrt{\frac{\sum\sum(d_{ij}-z_{ij})^2}{\sum\sum z_{ij}^2}}$$

d_{ij} refers to the distance between points *i* and *j* in the original space
z_{ij} refers to the distance between points *i* and *j* on the map

Multidimensional Scaling (MDS)

- The true dimensionality of the data will be revealed by the rate of decline of stress as dimensionality increases.



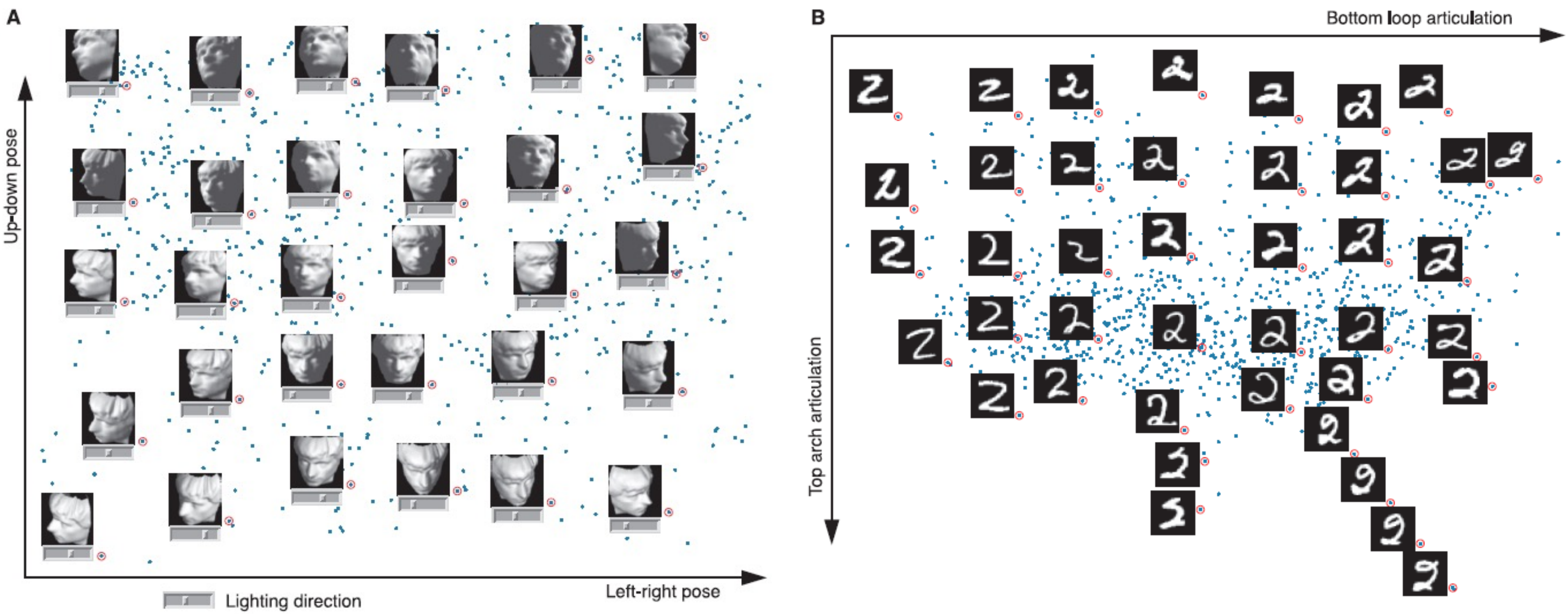
Multidimensional Scaling (MDS)

- Algorithm
  - Assign points to arbitrary coordinates in *p*-dimensional space
  - Compute Euclidean distances among all pairs of points to form a $\hat{D}$ matrix
  - Compare the matrix with the input matrix by evaluating the stress function. The smaller the value, the greater the correspondence between the two.
  - Adjust coordinates of each point in the direction that best maximally stress
  - Repeat steps 2 through 4 until stress won't get any lower

T.F. Cox and M.A.A. Cox, Multidimensional Scaling, Chapman & Hall/CRC; 2 edition, 2000

2.2 Isometric Feature Mapping (Isomap)

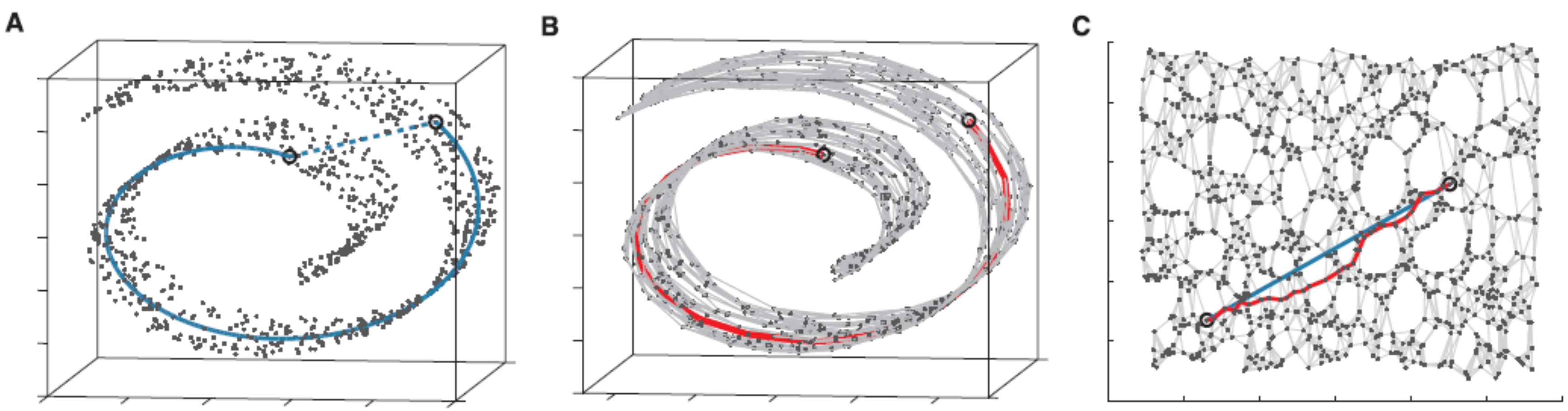
- Examples



J.B. Tenenbaum, V. de Silva, and J.C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” Science, vol. 290, pp. 2319-2323, 2000.

Isometric Feature Mapping (Isomap)

- Estimate the geodesic distance between far away points, given only input-space distances.
  - Adding up a sequence of "short hops" between neighboring points



Isometric Feature Mapping (Isomap)

- Algorithm
  - Step 1: construct neighborhood graph
    - Determines which points are neighbors on the manifold
    - Connect each point to all points within some fixed radius ε, or to its K nearest neighbors
  - Step 2: compute shortest paths
    - Estimate the geodesic distance between all pairs of points on the manifold by computing their shortest path in the graph
  - Step 3: construct d-dimensional embedding
    - Apply MDS to the matrix of graph distances constructing an embedding of the data

Isometric Feature Mapping (Isomap)

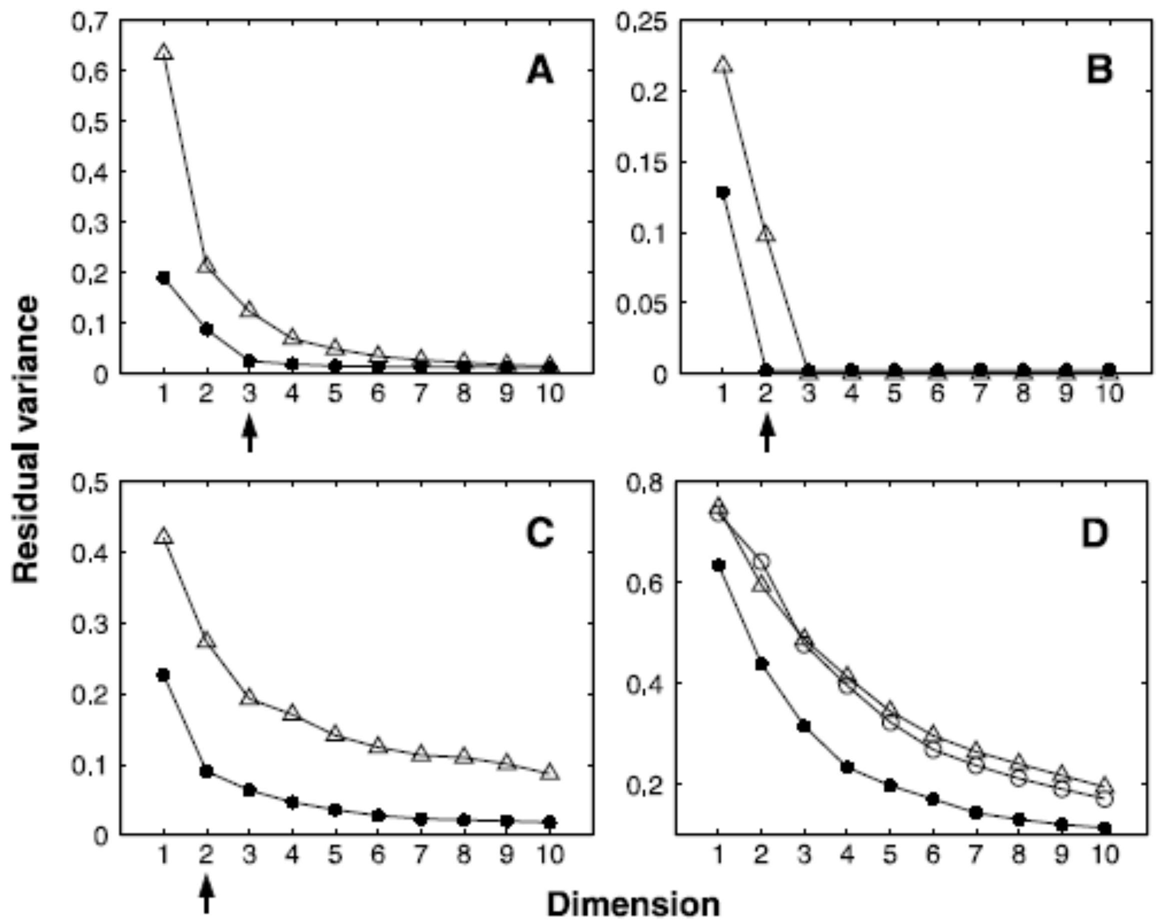


Fig. 2. The residual variance of PCA (open triangles), MDS [open triangles in (A) through (C); open circles in (D)], and Isomap (filled circles) on four data sets (*42*). (**A**) Face images varying in pose and illumination (Fig. 1A). (**B**) Swiss roll data (Fig. 3). (**C**) Hand images varying in finger extension and wrist rotation (*20*). (**D**) Handwritten "2"s (Fig. 1B). In all cases, residual variance decreases as the dimensionality *d* is increased. The intrinsic dimensionality of the data can be estimated by looking for the "elbow" at which this curve ceases to decrease significantly with added dimensions. Arrows mark the true or approximate dimensionality, when known. Note the tendency of PCA and MDS to overestimate the dimensionality, in contrast to Isomap.

2.3 Locally Linear Embedding (LLE)

- Eliminate the need to estimate pairwise distances between widely separated data points. LLE recovers global nonlinear structure from locally linear fits.

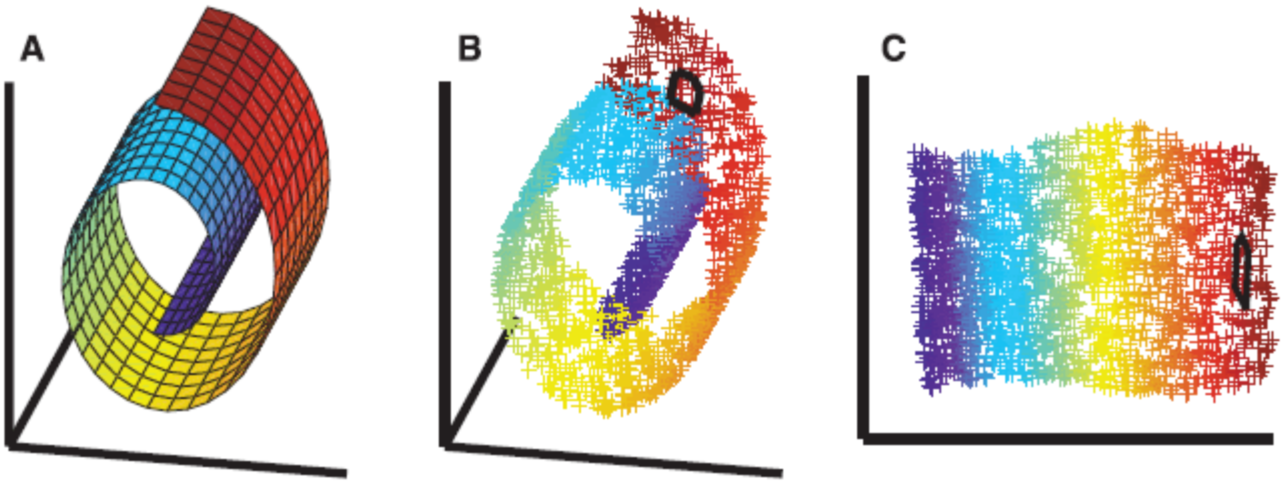


Fig. 1. The problem of nonlinear dimensionality reduction, as illustrated (*10*) for three-dimensional data (B) sampled from a two-dimensional manifold (**A**). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The color coding illustrates the neighborhood-preserving mapping discovered by LLE; black outlines in (**B**) and (**C**) show the neighborhood of a single point. Unlike LLE, projections of the data by principal component analysis (PCA) (*28*) or classical MDS (*2*) map faraway data points to nearby points in the plane, failing to identify the underlying structure of the manifold. Note that mixture models for local dimensionality reduction (*29*), which cluster the data and perform PCA within each cluster, do not address the problem considered here: namely, how to map high-dimensional data into a single global coordinate system of lower dimensionality.

S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," Science, vol. 290, pp. 2323-2326, 2000 http://www.cs.toronto.edu/~roweis/lle/publications.html

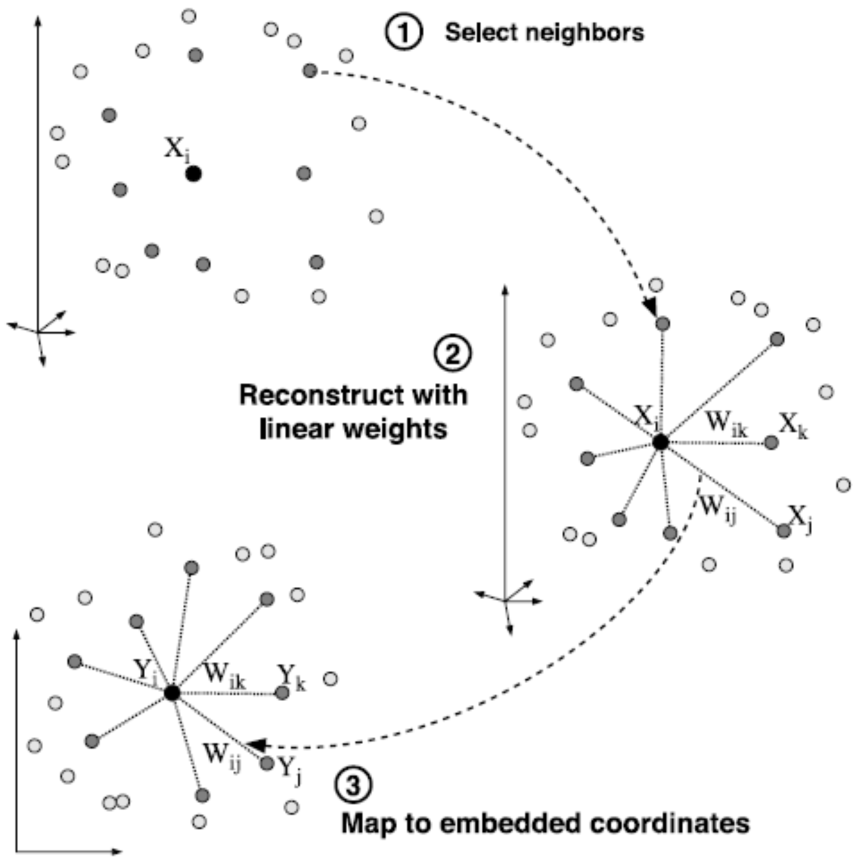
Locally Linear Embedding (LLE)

- Characterize the local geometry by linear coefficients that reconstruct each data point from its neighbors.
- Minimize the reconstruction errors

$$\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij}\vec{X}_j|^2$$

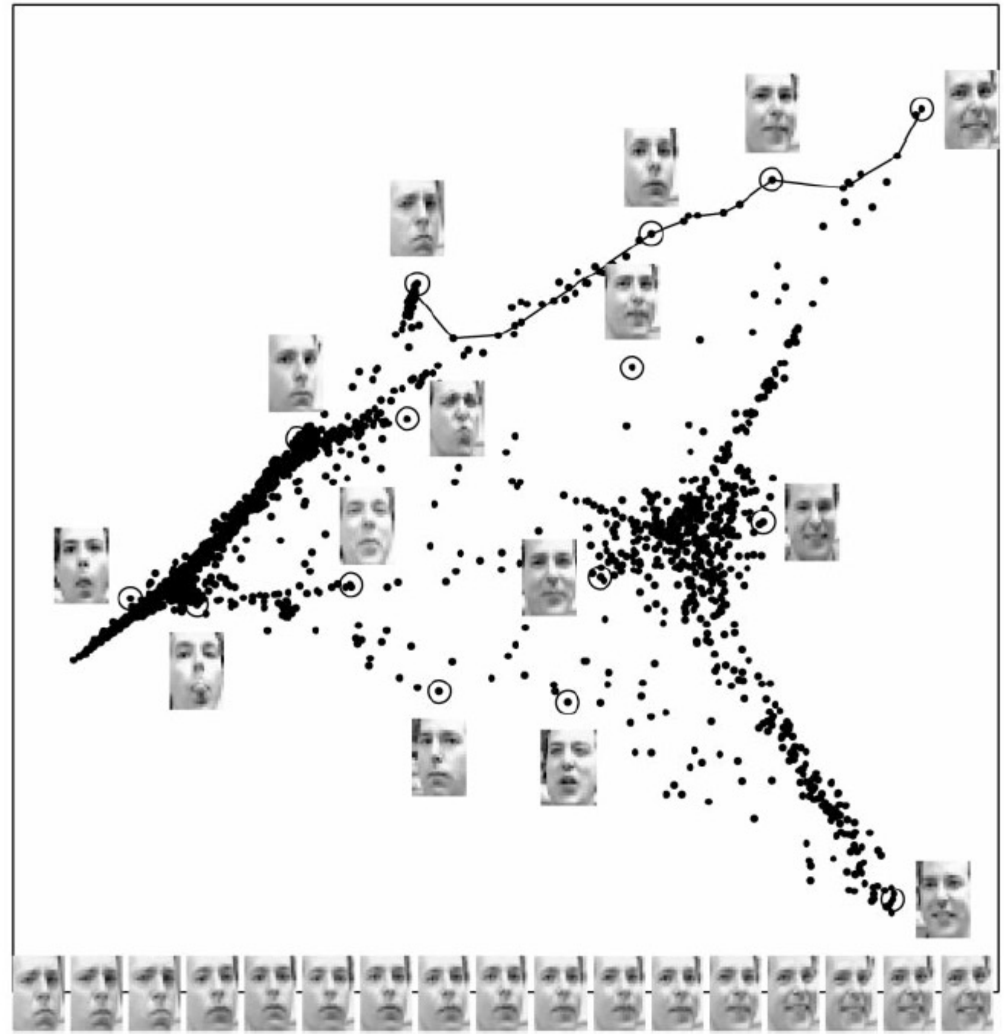
- Choosing d-dimensional coordinate Y_i to minimize the embedding cost function

$$\Phi(Y) = \sum_i |\vec{Y}_i - \sum_j W_{ij}\vec{Y}_j|^2$$



Example

- The bottom images correspond to points along the top-right path, illustrating one particular mode of variability in pose and expression.



References

- V. Castelli, "Multidimensional indexing structures for content-based retrieval," IBM Research Report, 2001.
- V. Gaede and O. Gunther, "Multidimensional access methods," ACM Computing Surveys, vol. 30, no. 2, pp. 170-231, 1998.
- L.I. Smith, A tutorial on Principal Component Analysis, http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- J. Shlens, "A tutorial on Principal Component Analysis," http://www.cs.cmu.edu/~elaw/papers/pca.pdf