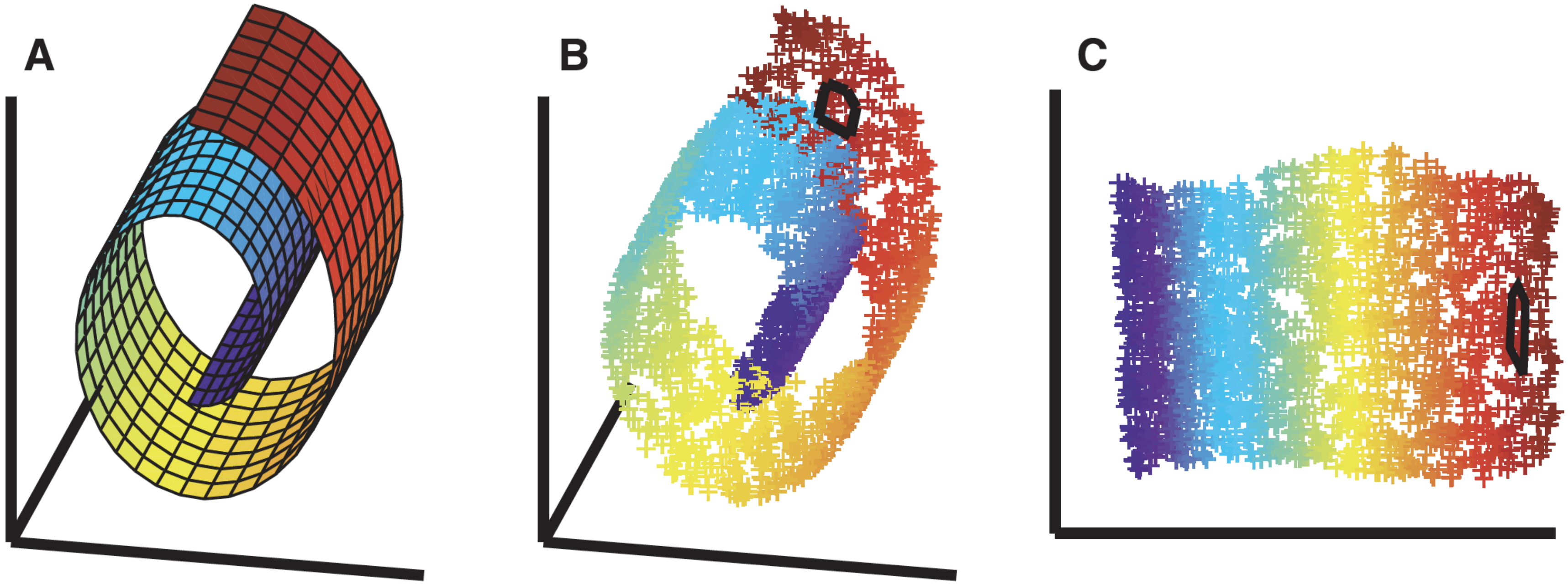
Lecture 1

Low-Dimensional Embedding 1 (LLE)

Klaus-Robert Müller & Wojciech Samek & Grégoire Montavon

Today

Local Linear Embedding (LLE)



[from Roweis & Saul Science 290, p 2323 (2000)]

Dimensionality Reduction

In many applications, we have

- high-dimensional data
- reason to believe they lie close to a lower dimensional subspace

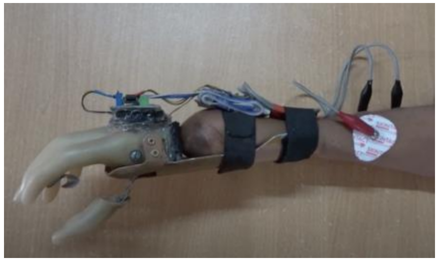
$\rightarrow$ Fewer parameters needed to account for the data properties
hidden causes or *latent variables*

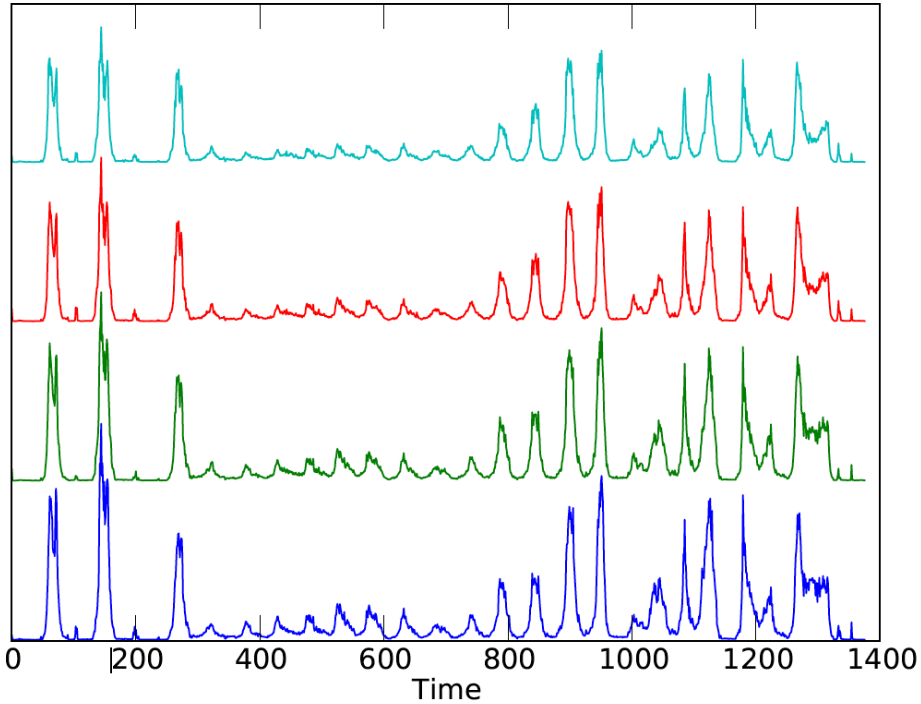
Examples:

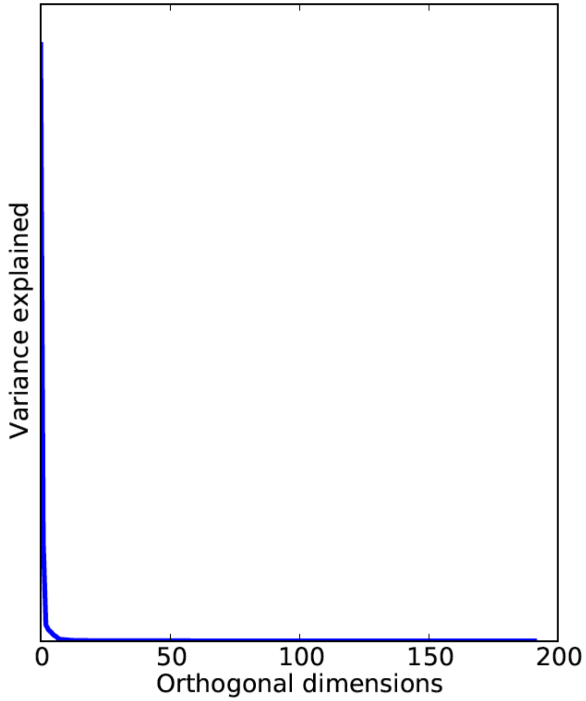
- you want to classify high resolution images
- you want to make a predictive model based on hundreds of customer attributes
- you want to analyse high dimensional neural data
- you want to detect trends in news data

Dimensionality Reduction

Electromyographic (EMG) Signal

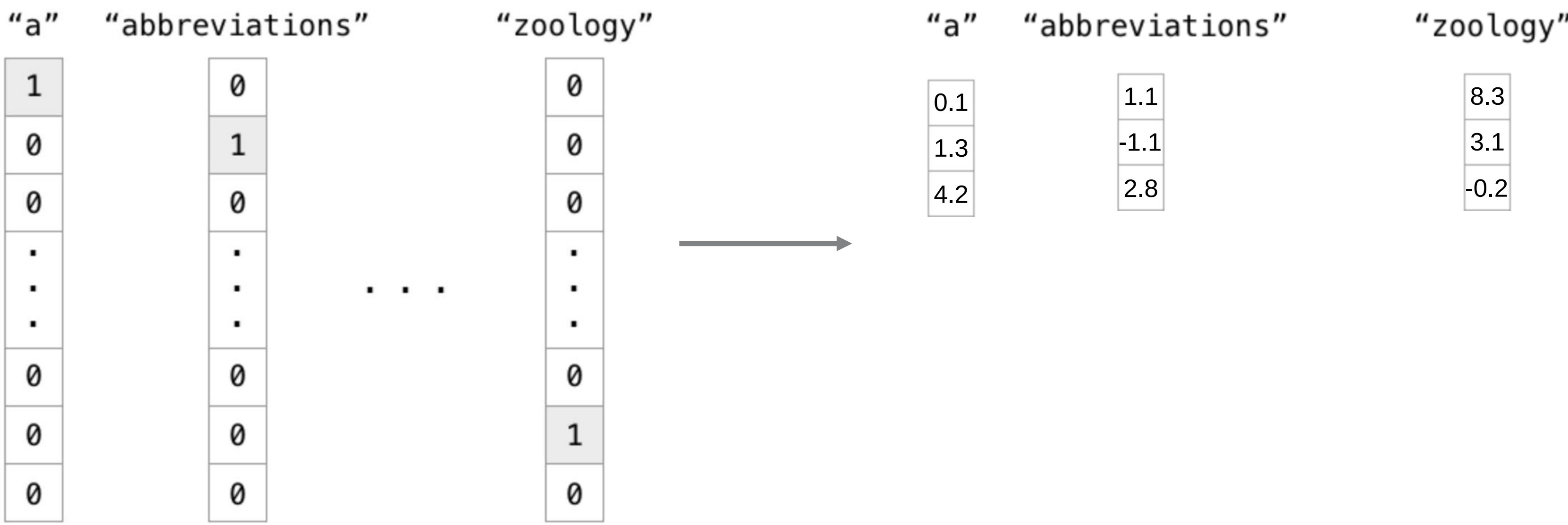






Dimensionality Reduction

Word2Vec Embedding



Word2Vec embedding encodes semantic information
(more than simple dimensionality reduction)

Dimensionality Reduction

Why dimensionality reduction / embeddings ?

- **Visualization**:
  Insights into high-dimensional structures in the data
- **Better Generalization**:
  Fewer dimensions $\rightarrow$ less chances of overfitting / better representation
- **Speeding up** learning algorithms:
  Most algorithms scale badly with increasing data dimensionality
- **Data compression**:
  Less storage requirements

Recap: PCA

We obtained some data $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$
PCA finds a direction $\mathbf{w} \in \mathbb{R}^D$ such that the variance of the projected data $\mathbf{w}^\top X$ is maximal

$$
\begin{aligned}
\mathrm{Var}(\mathbf{w}^T X) &= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{x}_n - \mathrm{E}(\mathbf{w}^\top \mathbf{x}))^2 \\
&= \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^\top \mathbf{x}_n - \mathbf{w}^\top \mathrm{E}(\mathbf{x}))^2 = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{w}^\top (\mathbf{x}_n - \mathrm{E}(\mathbf{x})))^2 \\
&= \frac{1}{N} \sum_{n=1}^{N} \mathbf{w}^\top (\mathbf{x}_n - \mathrm{E}(\mathbf{x})) \cdot (\mathbf{x}_n - \mathrm{E}(\mathbf{x}))^\top \mathbf{w} \\
&= \mathbf{w}^\top \underbrace{\left(\frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}_n - \mathrm{E}(\mathbf{x})) \cdot (\mathbf{x}_n - \mathrm{E}(\mathbf{x}))^\top \right)}_{\text{Covariance matrix S}} \mathbf{w}
\end{aligned}
$$

Recap: PCA

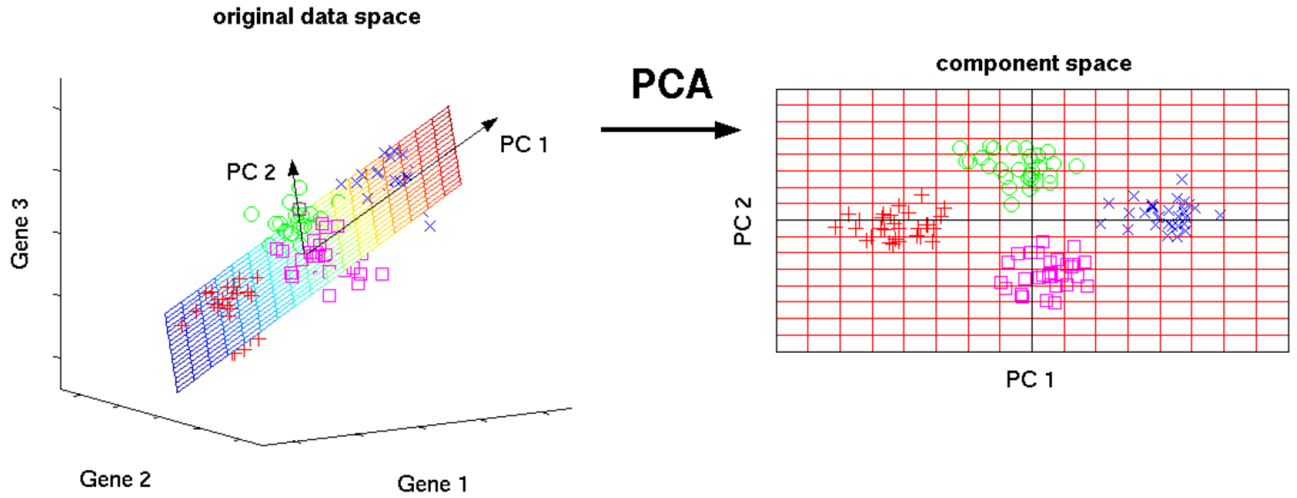
For $S\mathbf{w} = \lambda\mathbf{w}$, we see that the variance in direction $\mathbf{w}$ is given by:

$$\underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w}^\top S\mathbf{w}}{\mathbf{w}^\top \mathbf{w}} = \frac{\mathbf{w}^\top \lambda\mathbf{w}}{\mathbf{w}^\top \mathbf{w}} = \lambda$$

The variance of the projected data in an eigendirection $\mathbf{w}$ is given by the corresponding eigenvalue!

The direction of maximal variance in the data is equal to the eigenvector having the largest eigenvalue.

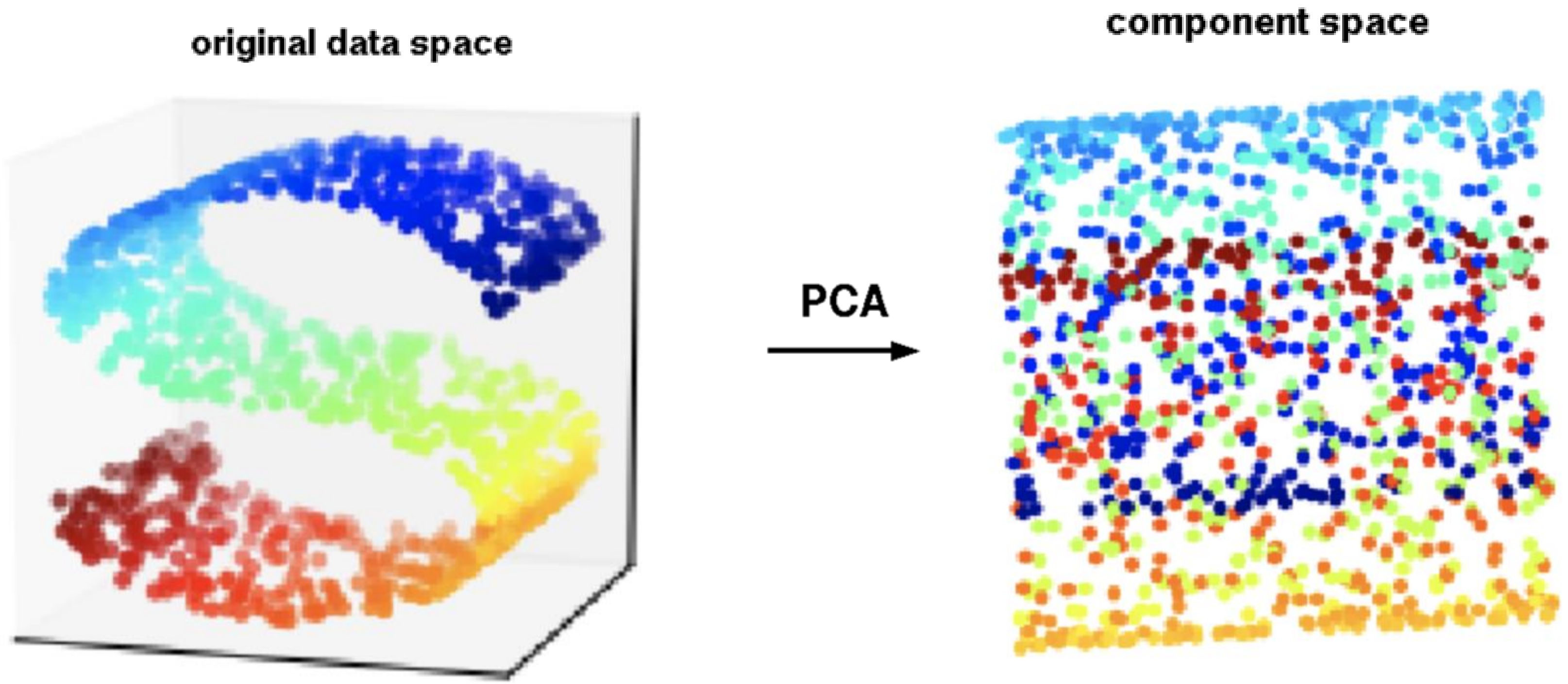
Recap: PCA



Now that we have $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_k] \in \mathbb{R}^{D \times k}$, we project each data point $\mathbf{x}$ onto W

$$H = \begin{bmatrix} \mathbf{w}_1^T \mathbf{x} \\ \vdots \\ \mathbf{w}_k^T \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_k^T \end{bmatrix} \mathbf{x} = W^T \cdot \mathbf{x}$$

Recap: PCA



[from Ornek]

Recap: PCA

PCA is a linear dimensionality reduction technique

If data lies on a non-linear manifold, PCA may not be able to capture its structure.

Many non-linear dimension reduction techniques exist, e.g.,
- Kernel PCA
- ISOMAP
- Locally linear embedding
- Hessian eigenmaps
- Diffusion maps
- Maximum variance unfolding
- ...

Recap: PCA

Idea: To make PCA non-linear, we *implicitly* map the data to a higher dimensional space and perform PCA there ("kernel trick").

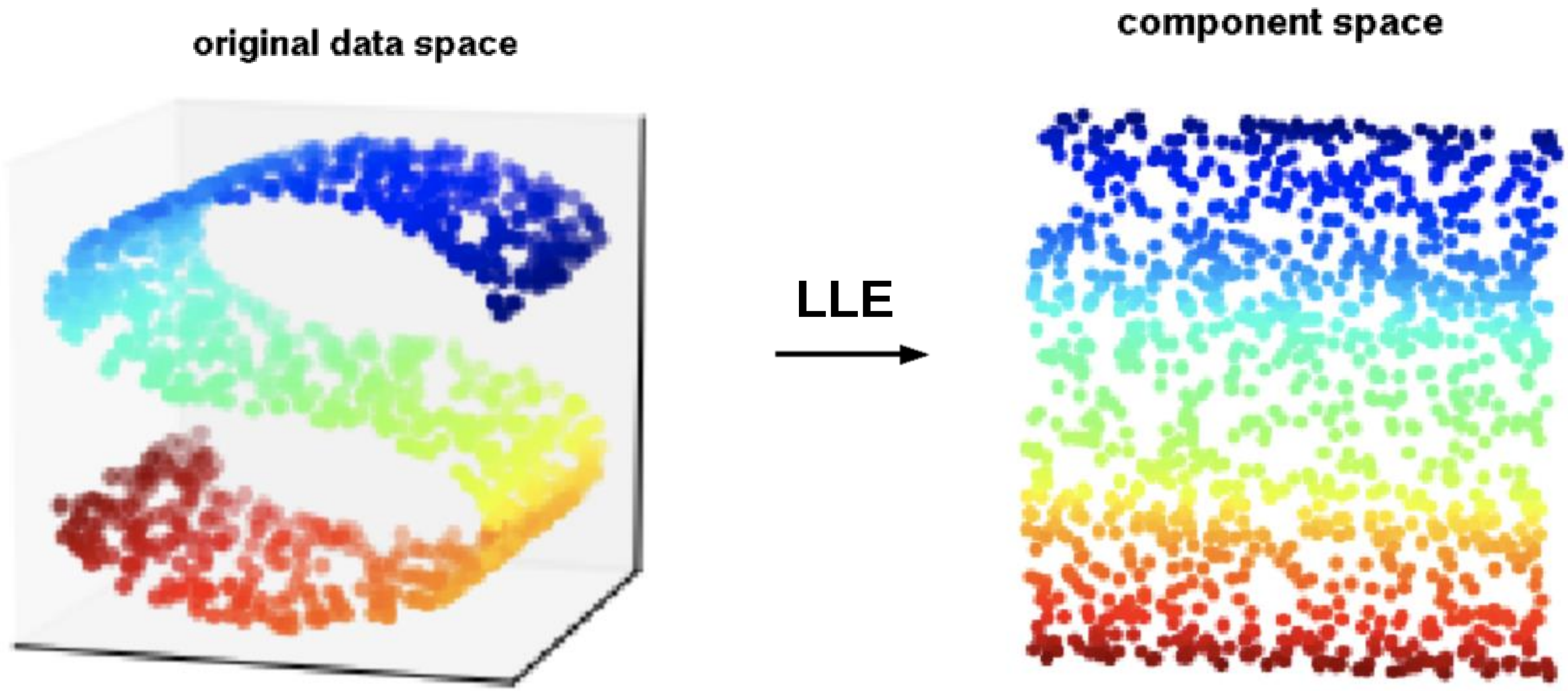
Solving PCA via $X^\top X$ instead of $XX^\top$ is called **linear kernel PCA**

This eigendecomposition only depends on inner products:
$(XX^T)_{ik} = \langle x_i, x_k \rangle$

We can replace this with a kernel matrix
$K(i,k) = \langle \Phi(x_i), \Phi(x_k) \rangle$.

Several non-linear dimensionality reduction methods can be viewed as kernel PCA with kernels learned from the data (see Ham et al. 2003).

Local Linear Embedding (LLE)

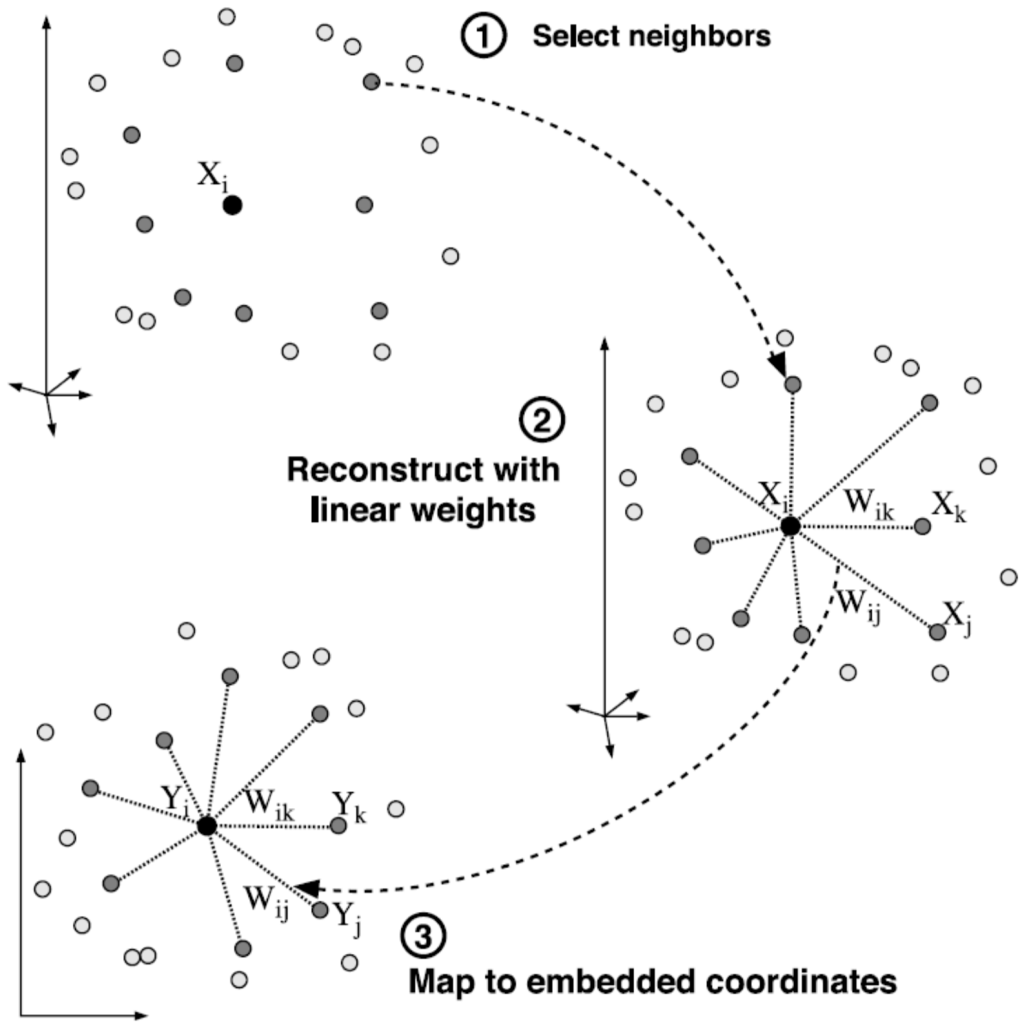


Idea: Find a low-dimensional representation that preserves neighborhood relations.

[from Ornek]

Local Linear Embedding (LLE)

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the K nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



[from Roweis]

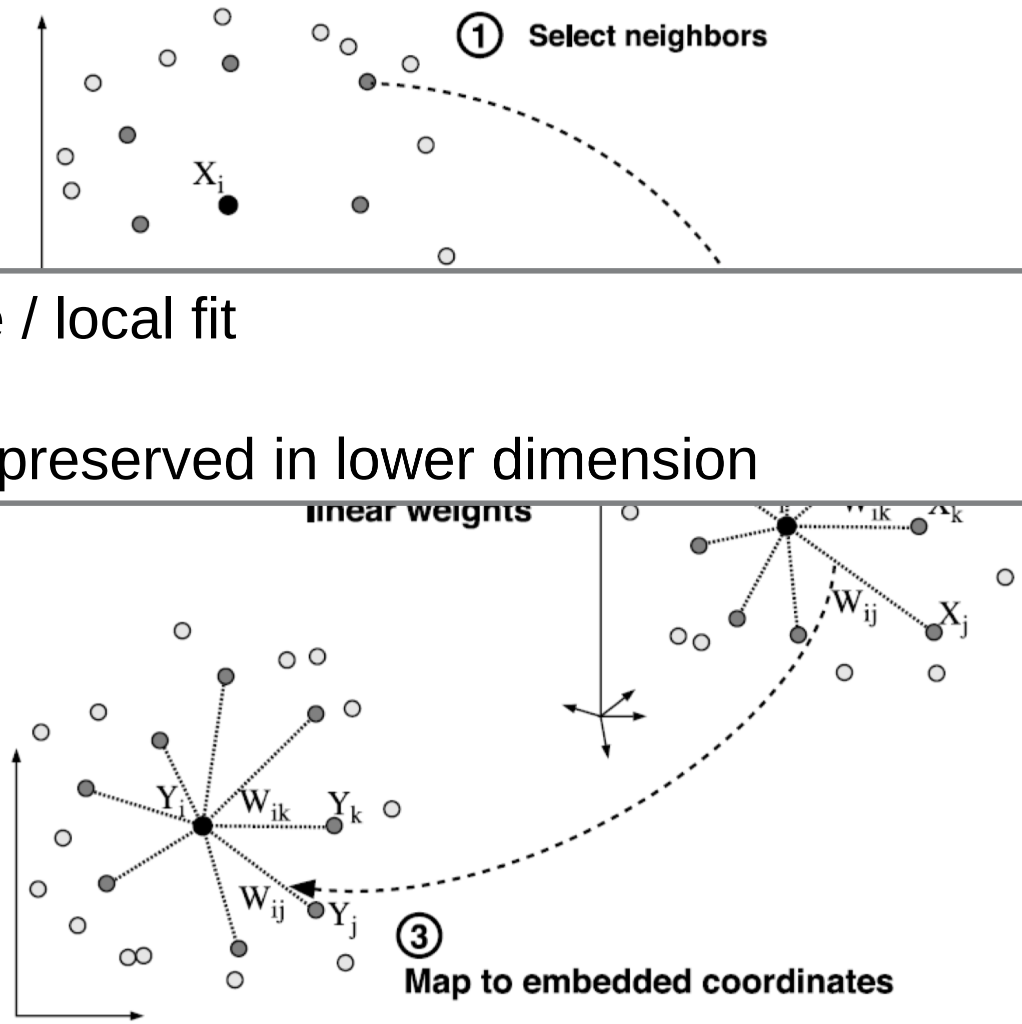
Local Linear Embedding (LLE)

Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the *K* nearest neighbors). (2) Compute the weights W_{ij} that best lin-

Extract local feature / local fit

Make sure that it is preserved in lower dimension

reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.



[from Roweis]

Local Linear Embedding (LLE)

LLE ALGORITHM

1. Compute the neighbors of each data point, $\mathbf{x}_i$.
2. Compute the weights w_{ij} that best reconstruct each data point $\mathbf{x}_i$ from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors $\mathbf{y}_i$ best reconstructed by the weights w_{ij} , minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

$$R(W) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij}\mathbf{x}_j\|^2 \qquad \textstyle\sum_{j \in N_i} w_{ij} = 1 \qquad (1)$$

$$\Phi(Y) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j \in N_i} w_{ij}\mathbf{y}_j\|^2 \qquad (2)$$

Local Linear Embedding (LLE)

1. Comp
2. Comp ... t $\mathbf{x}_i$ from
   its nei ... near fits.
3. Comp ... minimizing
   the qu ... ors.

Reconstruction errors obey an important symmetry: for any particular data point, they are *invariant* to rotations, rescalings, and translations of that data point and its neighbors.

--> Reconstruction weights characterize intrinsic geometric properties of each neighborhood (independent of frame of reference).

$$R(W) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij}\mathbf{x}_j\|^2 \qquad \sum_{j \in N_i} w_{ij} = 1 \tag{1}$$

$$\Phi(Y) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j \in N_i} w_{ij}\mathbf{y}_j\|^2 \tag{2}$$

Local Linear Embedding (LLE)

1. Comp

2. Comp ... t $\mathbf{X}_i$ from
its nei ... near fits.

3. Comp ... minimizing
the qu ... rs.

The reconstruction weights for each data point are computed from its local neighborhood - independent of the weights for other data points.

However, the embedding coordinates are computed by an N x N eigensolver, coupling all data points. This is how the algorithm leverages overlapping local information to discover global structure.

$$R(W) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij}\mathbf{x}_j\|^2 \qquad \textstyle\sum_{j \in N_i} w_{ij} = 1 \qquad (1)$$

$$\Phi(Y) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j \in N_i} w_{ij}\mathbf{y}_j\|^2 \qquad (2)$$

Local Linear Embedding (LLE)

Step 2

$$R(W) = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij} \mathbf{x}_j\|^2 = \sum_{i=1}^{n} R_i(\mathbf{x}_i, \mathbf{w}_i)$$

subject to $\sum_{j \in N_i} w_{ij} = 1 \qquad \forall\ i$

Minimize using Lagrange multipliers:

$$g(W, \lambda_1, \ldots, \lambda_n) = \sum_{i=1}^{n} g_i(\mathbf{w}_i, \lambda_i)$$

with $g_i(\mathbf{w}_i, \lambda_i) = R_i(\mathbf{x}_i, \mathbf{w}_i) - \lambda_i(\mathbf{1}^\top \mathbf{w}_i - 1)$

[derivation from Oldford]

Local Linear Embedding (LLE)

$$R_i(\mathbf{x}_i, \mathbf{w}_i) = \|\mathbf{x}_i - \sum_{j \in N_i} w_{ij}\mathbf{x}_j\|^2 = \|\mathbf{x}_i - \mathbf{X}_i{}^\top \mathbf{w}_i\|^2$$

$$\begin{aligned}
R_i(\mathbf{x}_i, \mathbf{w}_i) &= (\mathbf{x}_i - \mathbf{X}_i{}^\top \mathbf{w}_i)^\top (\mathbf{x}_i - \mathbf{X}_i^T \mathbf{w}_i) \\
&= \mathbf{x}_i{}^\top \mathbf{x}_i - \mathbf{x}_i{}^\top \mathbf{X}_i^T \mathbf{w}_i - \mathbf{w}_i^T \mathbf{X}_i \mathbf{x}_i + \mathbf{w}_i^T \mathbf{X}_i \mathbf{X}_i^T \mathbf{w}_i \\
&= \mathbf{w}_i^\top (\mathbf{1}\mathbf{x}_i^\top \mathbf{x}_i \mathbf{1}^\top - \mathbf{1}\mathbf{x}_i^\top \mathbf{X}_i^\top - \mathbf{X}_i \mathbf{x}_i \mathbf{1}^\top + \mathbf{X}_i \mathbf{X}_i^\top)\mathbf{w}_i \\
&= \mathbf{w}_i^\top C \mathbf{w}_i
\end{aligned}$$

C is a matrix of inner products with (j, k) element

$c_{jk} = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_k)$ for $j, k \in N_i$

[derivation from Oldford]

Local Linear Embedding (LLE)

$$\frac{\partial g_i(\mathbf{w}_i, \lambda_i)}{\partial \mathbf{w}_i} = 2C\mathbf{w}_i - \lambda_i \mathbf{1}$$

which when set to zero yields

$$\mathbf{w}_i \propto C^{-1}\mathbf{1}$$

Together with the constraint that $\mathbf{1}^\mathsf{T}\mathbf{w_i}$, this gives

$$\mathbf{w}_i = \frac{C^{-1}\mathbf{1}}{\mathbf{1}^\mathsf{T} C^{-1}\mathbf{1}}$$

If the covariance matrix is singular or nearly singular

$$C_{new} = C + \frac{\Delta^2}{k} I_k$$

[derivation from Oldford]

Local Linear Embedding (LLE)

Step 3

Find $n \times p$ matrix $Y = [\mathbf{y}_1 | \cdots | \mathbf{y}_n]^T$ which minimizes

$$\Phi(Y) = \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j \in N_i} w_{ij} \mathbf{y}_j\|^2$$

Let $\mathbf{w}_i^T = (w_{i1}, w_{i2}, \ldots, w_{in})$ where $w_{ij} = 0$ when j not $\in N_i$.

$$\Phi(Y) = \sum_{i=1}^{n} ||\mathbf{y}_i - Y^\top \mathbf{w}_i||^2$$

[derivation from Oldford]

Local Linear Embedding (LLE)

$$
\begin{aligned}
\Phi(Y) &= \sum_{i=1}^{n}(\mathbf{y}_i - Y^T\mathbf{w}_i)^T(\mathbf{y}_i - Y^T\mathbf{w}_i) \\
&= \sum_{i=1}^{n}(\mathbf{y}_i{}^\top\mathbf{y}_i - \mathbf{w}_i^T Y\mathbf{y}_i - \mathbf{y}_i{}^\top Y^\top\mathbf{w}_i + \mathbf{w}_i^T YY^\top\mathbf{w}_i) \\
&= (\sum_{i=1}^{n}\mathbf{y}_i{}^\top\mathbf{y}_i) - (\sum_{i=1}^{n}\mathbf{w}_i^T Y\mathbf{y}_i) - (\sum_{i=1}^{n}\mathbf{y}_i{}^\top Y^\top\mathbf{w}_i) + (\sum_{i=1}^{n}\mathbf{w}_i^\top YY^\top\mathbf{w}_i)
\end{aligned}
$$

Each term in the sum is a quadratic form $\displaystyle\sum_{i=1}^{n}\mathbf{a}_i^\top M\mathbf{b}_i$

[derivation from Oldford]

Local Linear Embedding (LLE)

So, for any matrices $A = [\ \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_n\ ]^T$ and $B = [\ \mathbf{b}_1 \mid \cdots \mid \mathbf{b}_n\ ]^T$ whose n rows are the $\mathbf{a}_i^T$s and $\mathbf{b}_i^T$s, we have:

$$\sum_{i=1}^{n} \mathbf{a}_i^T M \mathbf{b}_i = tr(AMB^T)$$

So letting $W = [\ \mathbf{w}_1 \mid \cdots \mid \mathbf{w}_n\ ]^T$, we can now write

$$\begin{aligned}
\Phi(Y) &= \textstyle\sum_{i=1}^{n} (\mathbf{y}_i^T \mathbf{y}_i - \mathbf{w}_i^T Y \mathbf{y}_i - \mathbf{y}_i^T Y^T \mathbf{w}_i + \mathbf{w}_i^T Y Y^T \mathbf{w}_i) \\
&= tr(YY^T - WYY^T - YY^T W^T + WYY^T W^T) \\
&= tr\left\{(Y - WY)(Y^T - Y^T W^T)\right\} \\
&= tr\left\{(I_n - W)YY^T(I_n - W)^T\right\} \\
&= tr\{(I_n - W)^\top (I_n - W) Y Y^\top\} \\
&= tr\{Y^\top (I_n - W)^\top (I_n - W) Y\}
\end{aligned}$$

[derivation from Oldford]

Local Linear Embedding (LLE)

$$\Phi(Y) = tr\{Y^\top M Y\} = \mathbf{Y_1^\top M Y_1} + \ldots + \mathbf{Y_p M Y_p}$$

$$\text{with } \boxed{\mathbf{Y}_i^\top \mathbf{Y}_i = 0 \text{ i} \neq \text{j}, \mathbf{Y}_i^\top \mathbf{Y}_i = 1}, \text{ and } \boxed{\mathbf{1}_n^\top \mathbf{Y}_j = 0}.$$

$$Y^T Y = I_p$$

$$(I_n - W)(\mathbf{Y}_j - a\mathbf{1}) = (I_n - W)\mathbf{Y}_j - \mathbf{0}$$

$$(I_n - W)\mathbf{1} = \mathbf{0}$$

The solution consists of the p eigenvectors of M orthogonal to **1** corresponding to the smallest eigenvalues.

The columns of Y are these eigenvectors and the new locations (in *p* dimensions) are the corresponding rows $\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top$.

[derivation from Oldford]

Local Linear Embedding (LLE)

LLE ALGORITHM

1. Compute the neighbors of each data point, $\mathbf{X}_i$.
2. Compute the weights W_{ij} that best reconstruct each data point $\mathbf{X}_i$ from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors $\mathbf{Y}_i$ best reconstructed by the weights W_{ij} , minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

Step 1: $O(dn^2)$ (with kd-trees often $O(n \log n)$)

Step 2: $O(dnk^3)$

Step 3: $O(dn^2)$ (methods from sparse eigenproblems can reduce it further)

LLE from Pairwise Distances

LLE can be applied to user input in the form of pairwise distances. In this case, nearest neighbors are identified by the smallest non-zero elements of each row in the distance matrix.

To derive the reconstruction weights for each data point, we need to compute the local covariance matrix between its nearest neighbors.

$$c_{jk} = \frac{1}{2}\left(D_j + D_k - D_{jk} - D_0\right)$$

where D_{jk} denotes the squared distance between the jth and kth neighbors, $D_\ell = \sum_z D_{\ell z}$ and $D_0 = \sum_{jk} D_{jk}$.

Kernel View on LLE

Coordinates of the eigenvectors 2, ..., p+1 provide the LLE embedding (see Ham et al. 2003).

$$K := (\lambda_{max} I - M)$$

with

$$M := (I - W)(I - W^T)$$

with weight matrix *W* whose ith row contains the linear coefficients that sum to unity and optimally reconstruct x_i from its *p* nearest neighbors.

Limitations of LLE

- Sensitivity to noise
- Sensitivity to non-uniform sampling of the manifold
- Does not provide a mapping (though one can be learned in a supervised fashion from the pairs $\{Xi, Yi\}$)
- Quadratic complexity on the training set size
- Unlike ISOMAP, no robust method to compute the intrinsic dimensionality, and
- No robust method to define the neighborhood size K

[from L26: Advanced dimensionality reduction]

Applications

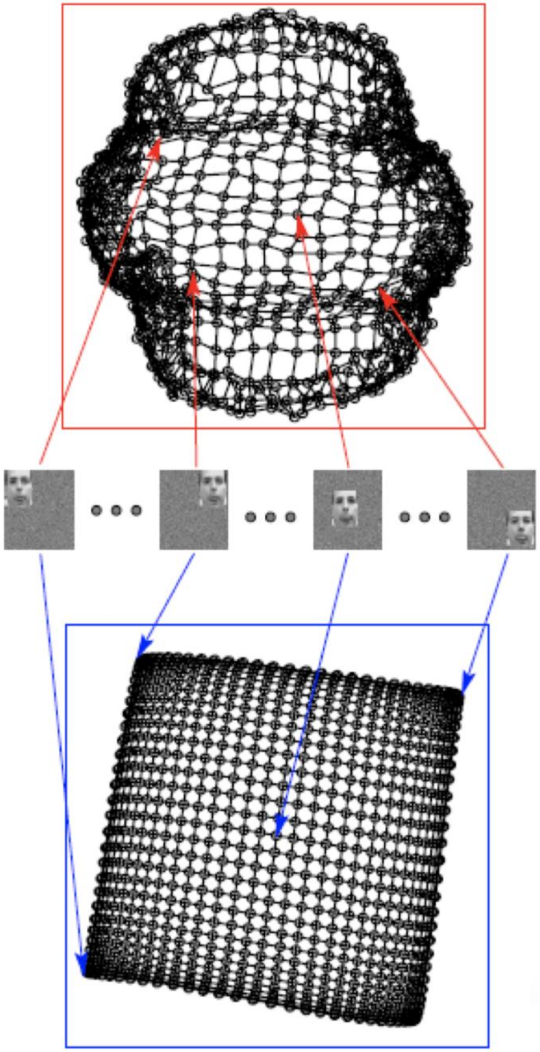


Figure 5: Successful recovery of a manifold of known structure. Shown are the results of PCA (top) and LLE (bottom), applied to $N = 961$ grayscale images of a single face translated across a two dimensional background of noise. Such images lie on an intrinsically two dimensional manifold, but have an extrinsic dimensionality equal to the number of pixels in each image ($D = 3009$). Note how LLE (using $K = 4$ nearest neighbors) maps the images with corner faces to the corners of its two dimensional embedding ($d = 2$), while PCA fails to preserve the neighborhood structure of nearby images.

[from Roweis]

Applications

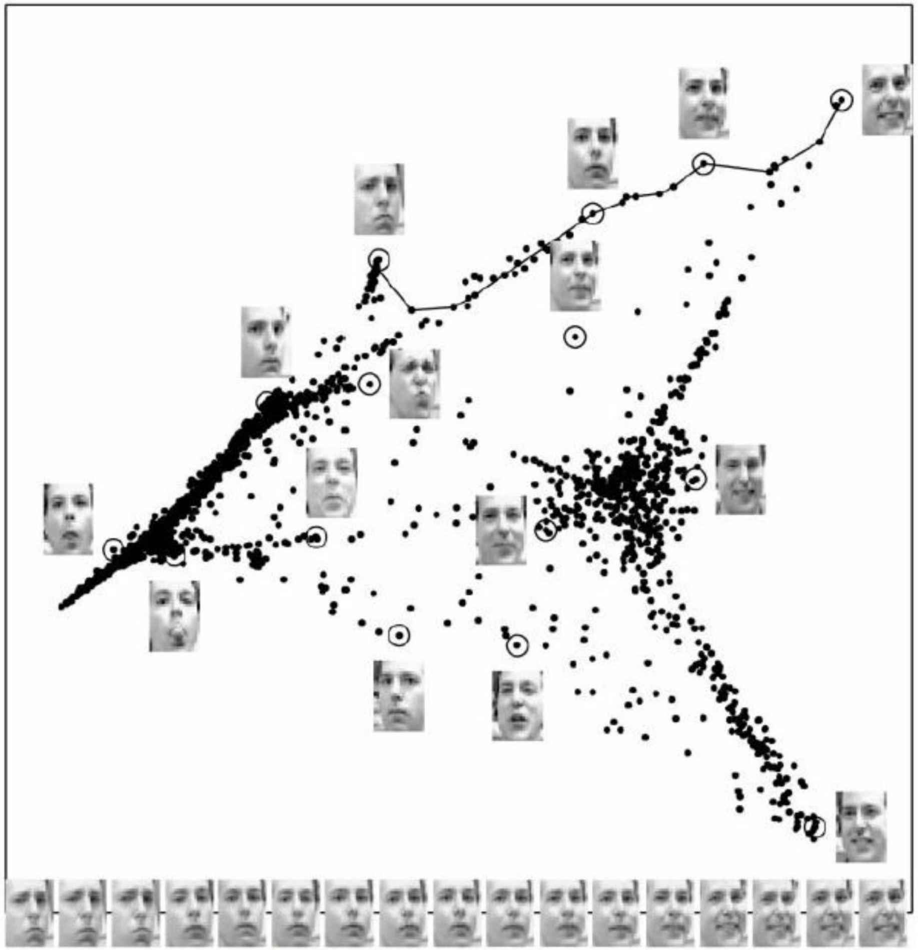


Fig. 3. Images of faces (11) mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points in different parts of the space. The bottom images correspond to points along the top-right path (linked by solid line), illustrating one particular mode of variability in pose and expression.

[from Roweis]

Applications

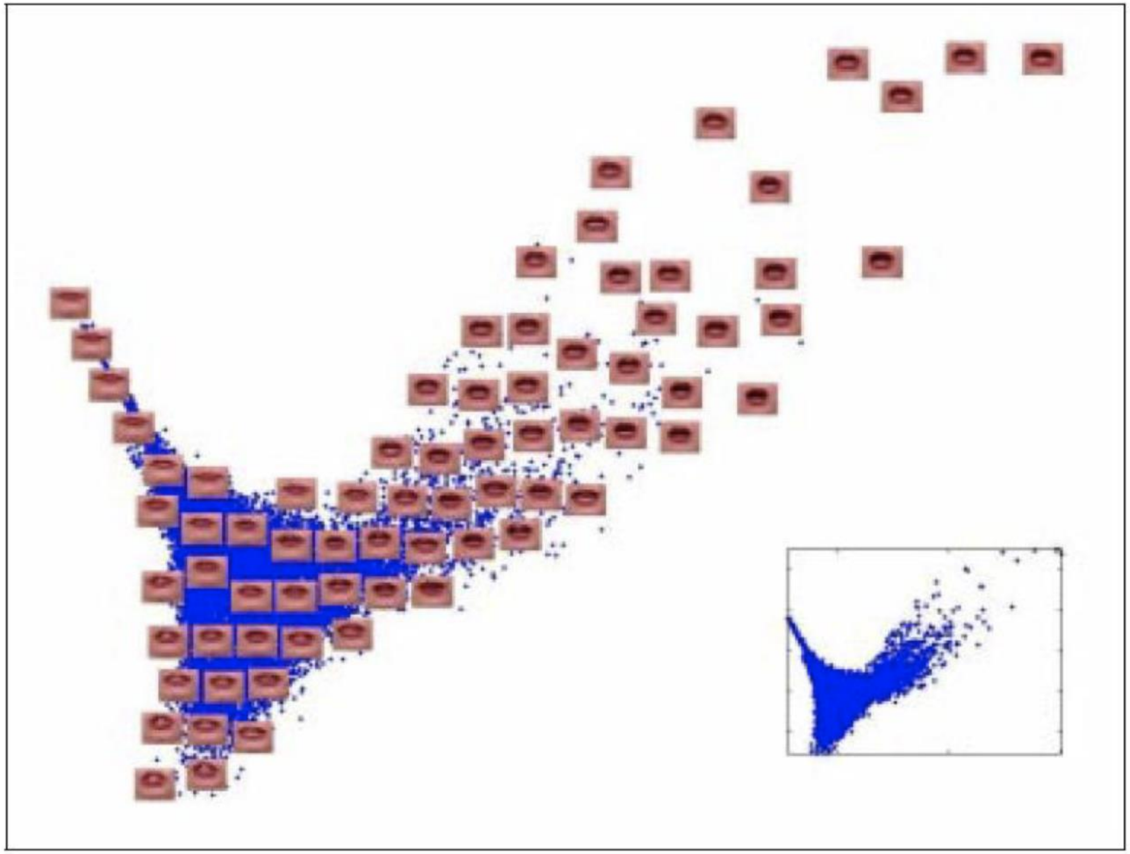
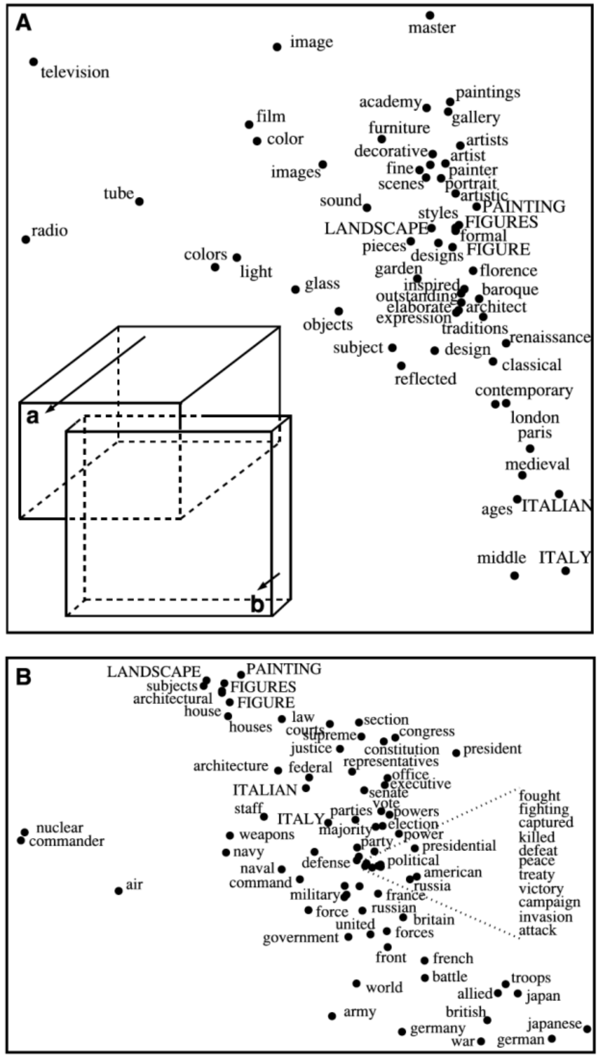


Figure 7: High resolution ($D = 65664$) images of lips, mapped into the embedding space discovered by the first two coordinates of LLE, using $K = 24$ nearest neighbors. Representative lips are shown at different points in the space. The inset shows the first two LLE coordinates for the entire data set ($N = 15960$) without any corresponding images.

[from Roweis]

Applications

Fig. 4. Arranging words in a continuous semantic space. Each word was initially represented by a high-dimensional vector that counted the number of times it appeared in different encyclopedia articles. LLE was applied to these word-document count vectors (*12*), resulting in an embedding location for each word. Shown are words from two different bounded regions (**A**) and (**B**) of the embedding space discovered by LLE. Each panel shows a two-dimensional projection onto the third and fourth coordinates of LLE; in these two dimensions, the regions (A) and (B) are highly overlapped. The inset in (A) shows a three-dimensional projection onto the third, fourth, and fifth coordinates, revealing an extra dimension along which regions (A) and (B) are more separated. Words that lie in the intersection of both regions are capitalized. Note how LLE colocates words with similar contexts in this continuous semantic space.



[from Roweis]

Conclusion

LLE illustrates a general principle of manifold learning, elucidated by Tenenbaum et al., that overlapping local neighborhoods—collectively analyzed—can provide information about global geometry.

As more dimensions are added to the embedding space,
the existing ones do not change.

A virtue of LLE is that it avoids the need to solve large dynamic programming problems.

Many more non-linear embedding techniques exist.

Next lecture: t-SNE