Seminar on Dimensionality Reduction

Locally Linear Embedding (LLE)

Paul von Bünau | TU Berlin

Introduction

Dimensionality Reduction: Given the data $x_1, \ldots, x_n \in \mathbb{R}^d$ in observation space, find corresponding coordinates $y_1, \ldots, y_n \in \mathbb{R}^{d'}$ in lower dimensional embedding space ($d' < d$) by $some\ criterion$.

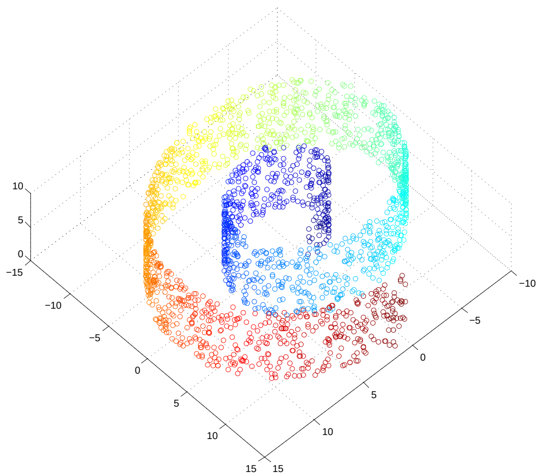
The PCA criterion: Find projection on linear subspace that minimizes the reconstruction error (alternatively: linear subspace that explains most of the variance).

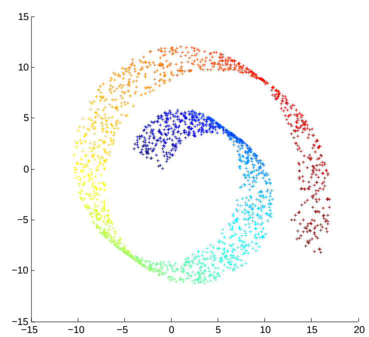
The Isomap criterion: Preserve pairwise distances between the observed data measured as shortest paths along a graph which is expected to follow the data manifold.

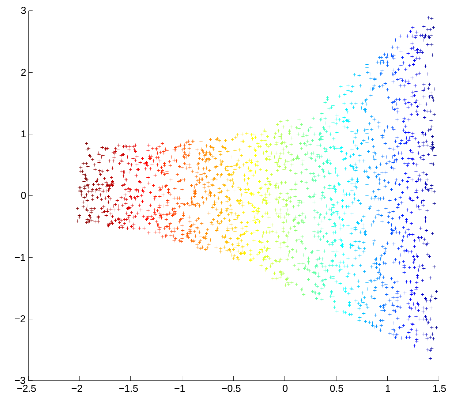
The LLE criterion: Preserve geometric properties of the local neighbourhood of each observed data point.

Why LLE? (1)

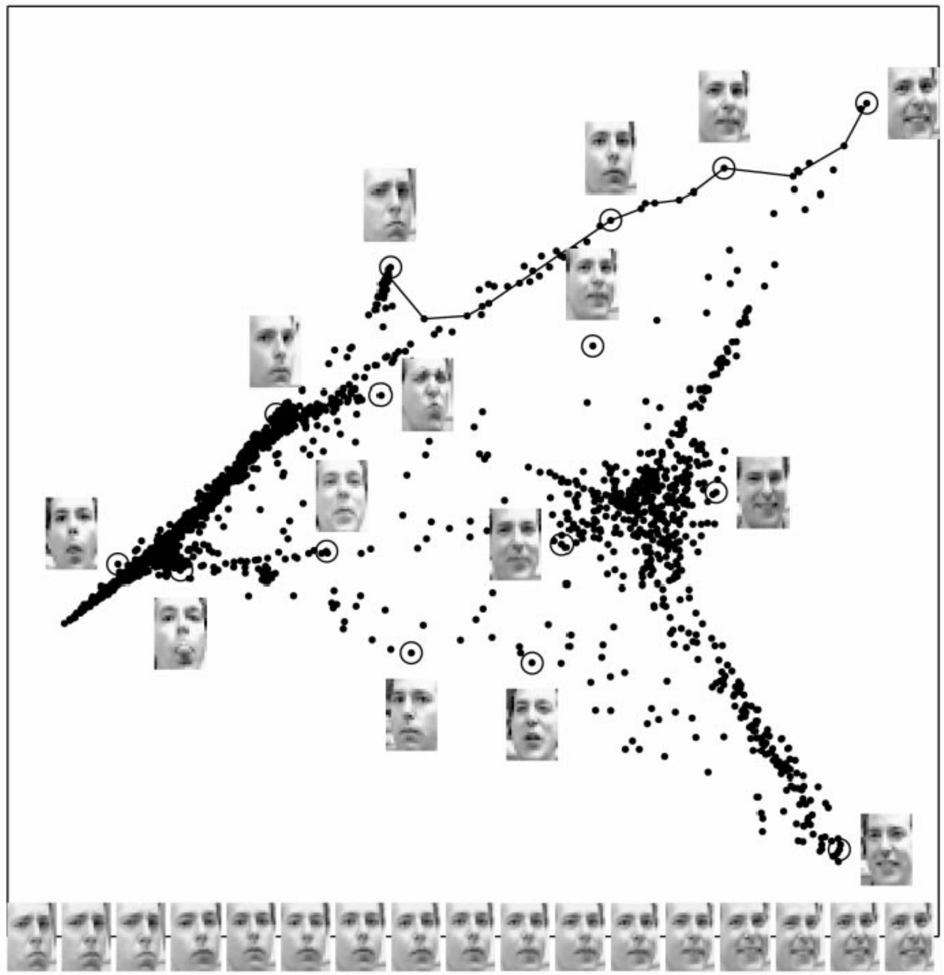
There exist toy-datasets where linear methods are bound to fail.



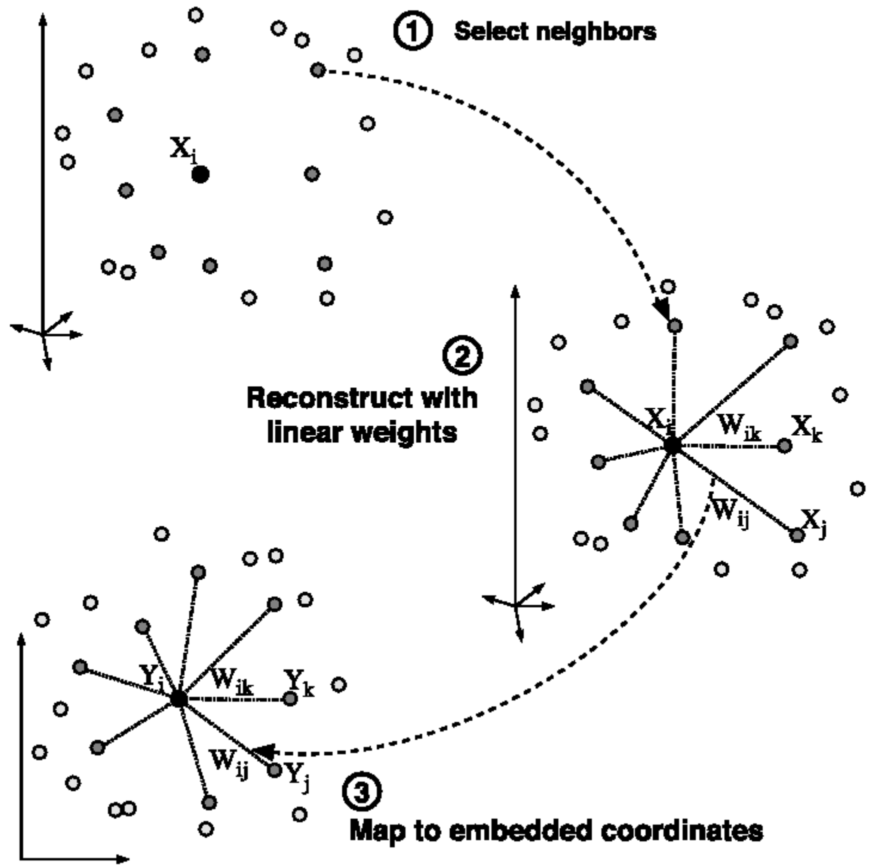




Why LLE? (2)



Outline of the Algorithm (pictorial)



Outline of the Algorithm (formally)

1. For each datapoint x_i, select its neighbours $x_{\eta_{i1}}, \ldots, x_{\eta_{ik_i}}$.

2. Find matrix of weights $W^* \in \mathbb{R}^{n \times n}$ that minimizes the reconstruction error of a datapoint from its neighbours

$$W^* = \operatorname*{argmin}_W \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{n} W_{ij} x_j \right\|^2$$

where $W_{ij} = 0$ if x_j is not a neighbour x_i and $\sum_j W_{ij} = 1$ for all rows i.

Analytic solution (matrix inversion)!

3. Find embedding coordinates $y_1^*, \ldots, y_n^*$ that minimize the embedding error

$$\{y_1^*, \ldots, y_n^*\} = \underset{\{y_1, \ldots, y_n\}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} W_{ij}^* y_j \right\|^2$$

using the weights W^* obtained in the observation space (step 2).

Analytic solution (eigenvalue problem)!

Step 1: Finding Neighbours

- Use the k-nearest-neighbour rule: the neighbours of x_i are the k closest datapoints $x_{\eta_{i1}}, \ldots, x_{\eta_{ik}}$ with respect to the Euclidean norm.

- The ϵ-ball rule is also applicable but choosing an appropriate ϵ is even more difficult than choosing k.

Step 2: Finding the Weights (1) – Invariants

Let us first note some desirable properties.

The aim is to determine W^* that minimizes the reconstruction error in observation space

$$\mathcal{E}(W) = \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{n} W_{ij} x_j \right\|^2$$

subject to $W_{ij} = 0$ if x_j is not a neighbour x_i and $\sum_j W_{ij} = 1$ for all rows i.

The minimizer W^* of $\mathcal{E}(W)$ is invariant to rescaling, translation and rotation of the observed data. This is what we want since W^* is should capture the local geometry of the data.

Step 2: Finding the Weights (2) – Rescaling Invariance

Quick verification

Rescaling the data by $\alpha \in \mathbb{R}$:

$$\sum_{i=1}^{n} \left\| \alpha x_i - \sum_{j=1}^{n} W_{ij} \alpha x_j \right\|^2 = \alpha^2 \mathcal{E}(W)$$

Step 2: Finding the Weights (3) – Translation Invariance

Quick verification

Translating the data by $t \in \mathbb{R}^d$:

$$\begin{aligned}\sum_{i=1}^{n}\left\|x_i + t - \sum_{j=1}^{n} W_{ij}(x_j + t)\right\|^2 &= \sum_{i=1}^{n}\left\|x_i + t - \sum_{j=1}^{n} W_{ij}x_j - \underbrace{\sum_{j=1}^{n} W_{ij}t}_{=t}\right\|^2 \\ &= \mathcal{E}(W)\end{aligned}$$

Step 2: Finding the Weights (4) – Rotation Invariance

Quick verification

Rotating the data by $R \in \mathbb{R}^{d\times d}$:

$$\begin{aligned}\sum_{i=1}^{n}\left\|Rx_i-\sum_{j=1}^{n}W_{ij}Rx_j\right\|^2 &= \sum_{i=1}^{n}\left(x_i-\sum_{j=1}^{n}W_{ij}x_j\right)^\top \underbrace{R^\top R}_{=I}\left(x_i-\sum_{j=1}^{n}W_{ij}x_j\right)\\ &= \mathcal{E}(W)\end{aligned}$$

Step 2: Finding the Weights (5) – Reshaping the Problem

We want to find the minimizer

$$W^* = \underset{W}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x_i - \sum_{j=1}^{n} W_{ij} x_j \right\|^2$$

subject to $W_{ij} = 0$ if x_j is not a neighbour x_i and $\sum_j W_{ij} = 1$ for all rows i.

Minimize the reconstruction error of each x_i separately. Denote by w^i the vector of coefficients in the i-th row of W that correspond to its neighbours,

$$w^i = \left[W_{i\eta_{i1}}, \ldots, W_{i\eta_{ik_i}} \right] \in \mathbb{R}^{k_i}.$$

Step 2: Finding the Weights (6) – Reshaping the Problem

Thus we find the minimizer w^{i*} for each datapoint x_i

$$
\begin{aligned}
w^{i*} &= \underset{w^i}{\operatorname{argmin}} \left\| x_i - \sum_{j=1}^{k_i} w^i_j x_{\eta_{ij}} \right\|^2 = \underset{w^i}{\operatorname{argmin}} \left\| \sum_{j=1}^{k_i} w^i_j (x_i - x_{\eta_{ij}}) \right\|^2 \\
&= \underset{w^i}{\operatorname{argmin}} \left[\sum_{j=1}^{k_i} w^i_j (x_i - x_{\eta_{ij}}) \right]^\top \left[\sum_{j=1}^{k_i} w^i_j (x_i - x_{\eta_{ij}}) \right] \\
&= \underset{w^i}{\operatorname{argmin}} \sum_{j=1}^{k_i} \sum_{l=1}^{k_i} w^i_j w^i_l \underbrace{(x_i - x_{\eta_{ij}})^\top (x_i - x_{\eta_{il}})}_{=:C^i_{jl}} = \underset{w^i}{\operatorname{argmin}}\, w^{i\top} C^i w^i
\end{aligned}
$$

subject to $\sum_{j=1}^{k_i} w^i_j = 1$ with the local covariance matrix $C^i \in \mathbb{R}^{k_i \times k_i}$ as defined above.

Step 2: Finding the Weights (7) – The Minimization

We can readily solve

$$w^{i*} = \underset{w^i}{\operatorname{argmin}}\, {w^i}^\top C^i w^i$$

subject to $\sum_{j=1}^{k_i} w_j^i = {w^i}^\top \mathbf{1} = 1$ using the Lagrangian formulation

$$L(w^i, \lambda) = {w^i}^\top C^i w^i + \lambda({w^i}^\top \mathbf{1} - 1)$$

which yields

$$\partial L(w^i, \lambda)/\partial w^i = 2C^i w^i + \lambda \mathbf{1} \quad \text{and} \quad \partial L(w^i, \lambda)/\partial \lambda = {w^i}^\top \mathbf{1} - 1.$$

By setting $C^i w^i + \lambda \mathbf{1} \stackrel{!}{=} 0$ and ${w^i}^\top \mathbf{1} - 1 \stackrel{!}{=} 0$ we arrive at

$$w^i = \frac{(C^i)^{-1}\mathbf{1}}{\mathbf{1}^\top (C^i)^{-1}\mathbf{1}}.$$

Step 2: Finding the Weights (8) – Wrap-up

Let's look back:

For each datapoint x_i, we can compute its optimal reconstruction weights w^{i*} by solving the linear system

$$C^i w^i = \mathbf{1}$$

and subsequent rescaling of w^i to enforce the constraint $w^{i*\top}\mathbf{1} = 1$.

Thus we can assemble the weight matrix $W^* \in \mathbb{R}^{n \times n}$ from $w^{1*}, \dots, w^{n*}$.

The next step:

Find lower dimensional embedding coordinates $y_1, \dots, y_n \in \mathbb{R}^{d'}$ that best fit the reconstruction weights W^* obtained in observation space.

Step 3: Finding the Coordinates (1) – The Cost Function

We want to find coordinates $y_1^*, \dots y_n^* \in \mathbb{R}^{d'}$ that minimize the reconstruction error in embedding space using the fixed weights W^* obtained from the observed data in the previous step,

$$\{y_1^*, \dots, y_n^*\} = \underset{\{y_1,\dots,y_n\}}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} W_{ij}^* y_j \right\|^2}_{=:\Phi(\{y_i\}_1^n)}$$

subject to $\sum_{i=1}^{n} y_i = \mathbf{0}$ (centering) and $\frac{1}{n} \sum_{i=1}^{n} y_i^\top y_i = \mathbf{I}$ (unit covariance).

The objective function $\Phi(\{y_i\}_1^n)$ can be rewritten to yield an analytic solution.

Step 3: Finding the Coordinates (2) – Algebraic Massage

Rewriting the objective function:

$$\begin{aligned}
\Phi(\{y_i\}_1^n) &= \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} W_{ij}^* y_j \right\|^2 \\
&= \sum_{i=1}^{n} \left[y_i - \sum_{j=1}^{n} W_{ij}^* y_j \right]^\top \left[y_i - \sum_{j=1}^{n} W_{ij}^* y_j \right] \\
&= \sum_{i=1}^{n} \left[y_i^\top y_i - 2 \sum_{j=1}^{n} W_{ij}^* y_i^\top y_j + \sum_{j=1}^{n} \sum_{k=1}^{n} W_{ij}^* W_{ik}^* y_j^\top y_k \right]
\end{aligned}$$

Step 3: Finding the Coordinates (3) – Algebraic Massage

Rewriting the objective function (continued):

$$
\begin{aligned}
\Phi(\{y_i\}_1^n) &= \sum_{i=1}^{n} \left[\sum_{j=1}^{n} \delta_{ij} y_i^\top y_j - 2 \sum_{j=1}^{n} W_{ij}^* y_i^\top y_j + \sum_{j=1}^{n} \sum_{k=1}^{n} W_{ki}^* W_{kj}^* y_i^\top y_j \right] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} y_i^\top y_j
\end{aligned}
$$

where the Matrix $M \in \mathbb{R}^{n \times n}$ is defined as

$$
M_{ij} := \delta_{ij} - W_{ij}^* - W_{ji}^* + \sum_{k=1} W_{ki}^* W_{kj}^* = (\mathbf{I} - W^*)^\top (\mathbf{I} - W^*).
$$

Step 3: Finding the Coordinates (4) – Algebraic Massage

Let $y^i \in \mathbb{R}^n$ be the i-th embedding dimension, i.e. $y^i = [(y_1)_i, \ldots, (y_n)_i]$. Then we can write down the objective function as

$$\begin{aligned}\Phi(\{y_i\}_1^n) &= \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij} y_i^\top y_j \\ &= \sum_{k=1}^{d'}\sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}(y_i)_k(y_j)_k \\ &= \sum_{k=1}^{d'} (y^k)^\top M y^k\end{aligned}$$

which we want to minimize subject to $(y^i)^\top y^j = \delta_{ij}$ and $\sum_{j=1}^{n} y_j^i = 0$.

Step 3: Finding the Coordinates (5) – Minimization

Thus the embedding dimensions $y^{1*}, \ldots, y^{d'*} \in \mathbb{R}^n$ are the d' minimizers of

$$y^{k*} = \underset{y}{\operatorname{argmin}}\, y^\top M y$$

subject to $(y^i)^\top y^j = \delta_{ij}$ (unit covariance) and $\sum_{j=1}^n y_j^i = 0$ (centering).

By the Rayleigh-Ritz Theorem (M is Hermitian) we find that the embedding dimension y^{k*} is the $(k+1)$-th bottom eigenvector of M.

The bottom eigenvector is the constant $\mathbf{1}$. The constraints are fulfilled by virtue of orthogonality.

Hence we have found the embedding coordinates $y_1^*, \ldots, y_n^* \in \mathbb{R}^{d'}$. **That's it!**

Wrapping it all up

1. For each x_i, determine its k_i neighbours $x_{i\eta_{i1}}, \ldots, x_{i\eta_{ik_i}}$ using e.g. the k-nearest-neighbour rule.

2. For each x_i, compute its optimal reconstruction weights w^{i*} (from its neighbours) as $w^{i*} = (C^i)^{-1}\mathbf{1}/\mathbf{1}^\top(C^i)^{-1}\mathbf{1}$ where C^i is the local covariance matrix,

$$C^i_{jl} = (x_i - x_{\eta_{ij}})^\top (x_i - x_{\eta_{il}}).$$

3. Assemble the weight matrix W^* from $w^{1*}, \ldots, w^{n*}$.

4. Obtain the embedding dimensions $y^{1*}, \ldots, y^{d'*}$ as the bottom eigenvectors of

$$M = (\mathbf{I} - W^*)^\top (\mathbf{I} - W^*).$$

Conclusion

- LLE is a powerful nonlinear method.
- LLE is fast and soluble in closed form.

However:

- Sampling of the manifold needs to be very dense (worse than Isomap).
- LLE did not prove very useful in practice (cf. PCA).
- Parameter k, regularization of C^i and choice of eigensolver are weak spots.
- No built-in feature to determine embedding dimensionality.

Thank you for listening.