

Introduction, Difficulties and Perspectives

Tutorial on Manifold Learning with Medical Images



HelmholtzZentrum

Diana Mateus

CAMP (Computed Aided Medical Procedures)
TUM (Technische Universität München)

&

Helmholtz Zentrum

September 22, 2011

Outline

Manifold Learning

- Three seminal algorithms

Common Practical Problems

- Breaking the implicit assumptions

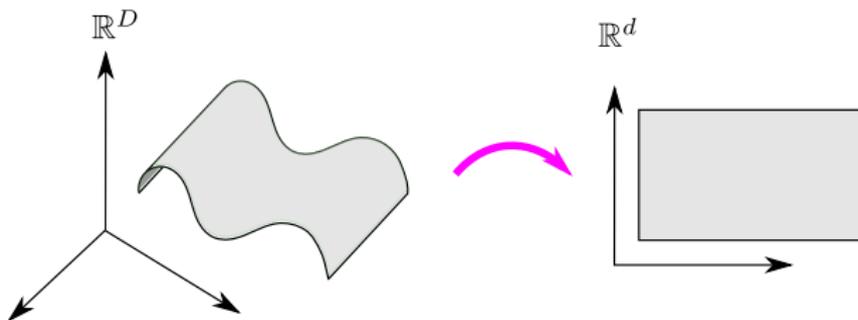
- Determining the parameters

- Mapping new points

- Large data-sets

Conclusions

Manifold learning



GOAL: dimensionality reduction

- For data lying on (or close to) a **manifold** ...
- find a **new representation** ...
- that is **low dimensional** ...
- allowing a **more efficient** processing of the data.

Three seminal algorithms

- Isomap

[▷ Joshua Tenenbaum, Vin de Silva, John Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science*, 2000.]

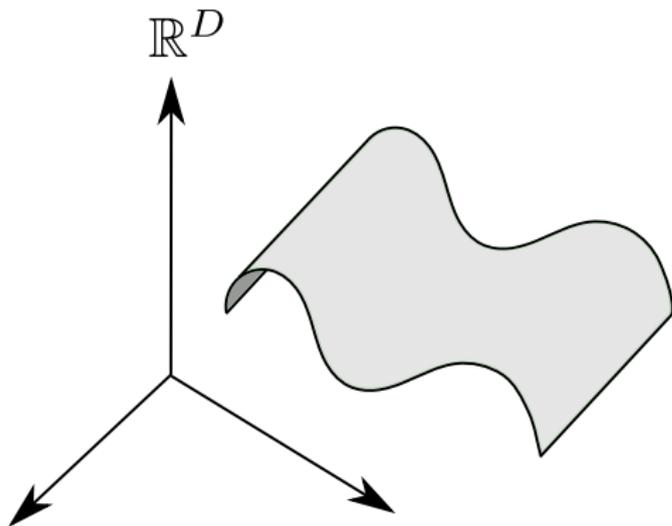
- Locally Linear Embedding (LLE)

[▷ Sam Roweis, Lawrence Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, 2000.]

- Laplacian Eigenmaps (LapEigs)

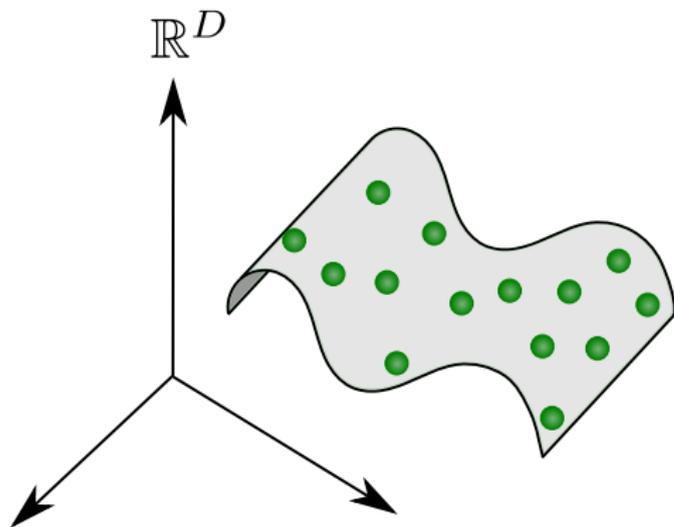
[▷ M. Belkin, P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, June 2003; 15 (6):1373-1396.]

Three seminal algorithms



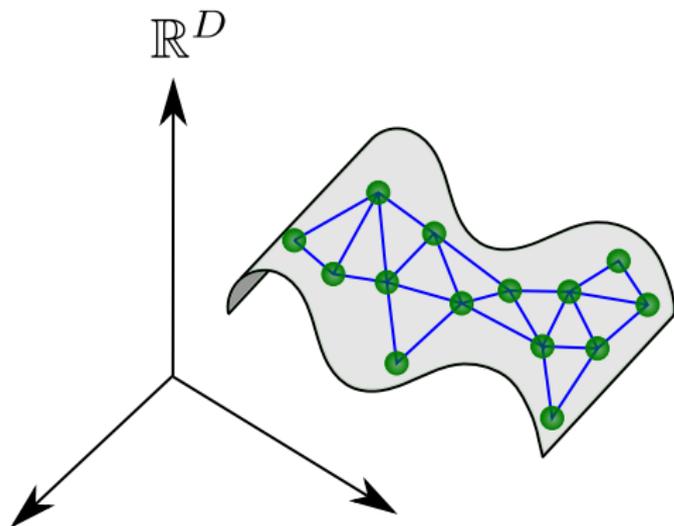
The data is **assumed** to lie on or close to a **Manifold**

Three seminal algorithms



Access is given only to a number of **samples** of the Manifold (data points)

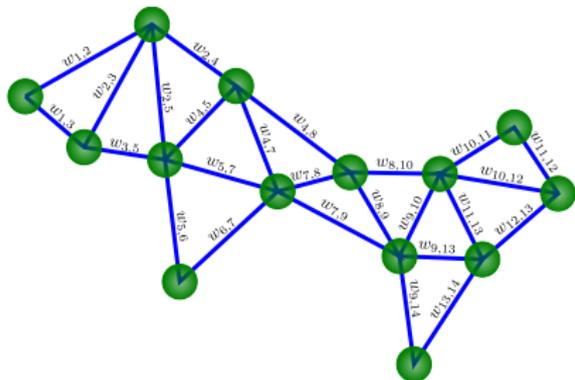
Three seminal algorithms



Build a **neighborhood graph** using the samples.

The graph **approximates** the manifold

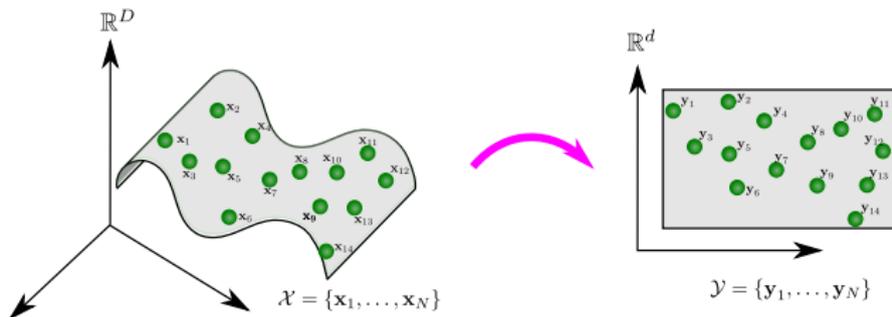
Three seminal algorithms



To complete the graph by determining **weights** on the edges (between every pair of neighbor nodes). The graph can then be expressed in a **matrix form**:

$$W = \begin{bmatrix} 0 & w_{1,2} & w_{1,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{1,2} & 0 & w_{2,3} & w_{2,4} & w_{2,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_{1,3} & w_{2,3} & 0 & 0 & w_{3,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{2,4} & 0 & 0 & w_{4,5} & 0 & w_{4,7} & w_{4,8} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{2,5} & w_{3,5} & w_{4,5} & 0 & w_{5,6} & w_{5,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{5,6} & 0 & w_{6,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{4,7} & w_{5,7} & w_{6,7} & 0 & w_{7,8} & w_{7,9} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{4,8} & 0 & 0 & 0 & w_{7,8} & 0 & w_{8,9} & w_{8,10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{8,9} & 0 & w_{9,10} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{8,10} & w_{9,10} & 0 & 0 & w_{9,13} & w_{9,14} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{10,11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{10,11} & w_{10,12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{10,12} & w_{11,12} & 0 & w_{11,13} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{9,13} & 0 & w_{11,12} & 0 & w_{12,13} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{9,13} & 0 & w_{11,13} & 0 & w_{12,13} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_{9,14} & 0 & 0 & 0 & w_{13,14} & 0 \end{bmatrix}$$

Three seminal algorithms



Find \mathcal{Y}^* by optimizing some cost function \mathcal{J}

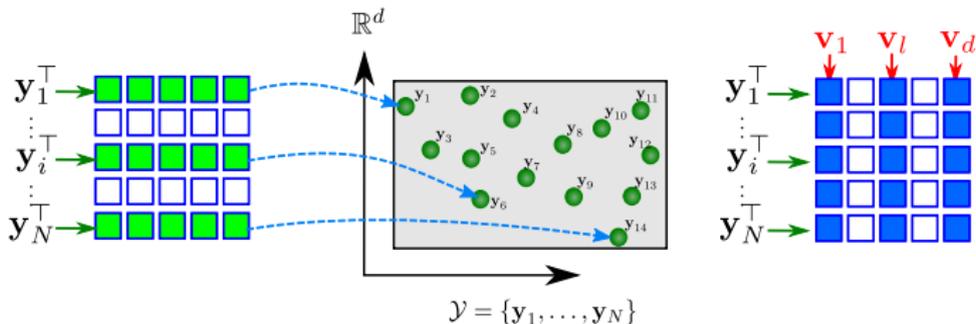
$$\mathcal{Y}^* = \min_{\mathcal{Y}} \mathcal{J}(\mathcal{T}(\mathbf{W}), \mathcal{Y})$$

Three seminal algorithms

$$\mathcal{Y}^* = \min_{\mathcal{Y}} \mathcal{J}(\mathcal{T}(\mathbf{W}), \mathcal{Y})$$

In **spectral methods** the optimization of \mathcal{J} can be expressed in the form:

$$\min_{\mathbf{v}_l} \frac{\mathbf{v}_l^\top \mathcal{T}(\mathbf{W}) \mathbf{v}_l}{\mathbf{v}_l^\top \mathbf{v}_l} \quad \forall l \in \{1, \dots, d\} \quad (\text{Rayleigh Quotient})$$

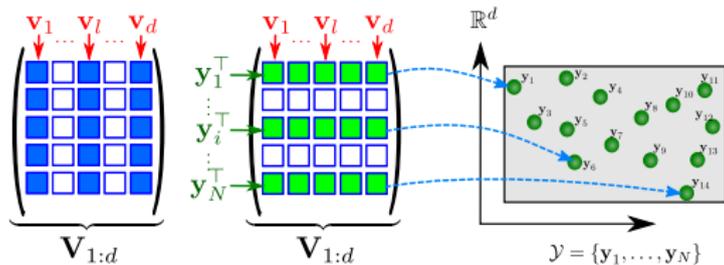


Three seminal algorithms

$$\min_{\mathbf{y}} \frac{\mathbf{v}_l^\top \mathcal{T}(\mathbf{W}) \mathbf{v}_l}{\mathbf{v}_l^\top \mathbf{v}_l} \quad + \quad \text{constraints} \\ \text{(Rayleigh Quotient)} \quad \quad \quad \text{(orthonormality, centering of } \mathcal{Y}, \dots)$$

Solved by a **spectral decomposition** of $\mathcal{T}(\mathbf{W})$.

$$\underbrace{\begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}}_{\mathcal{T}(\mathbf{W})} = \underbrace{\begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}}_{\mathbf{V}} \underbrace{\begin{pmatrix} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{pmatrix}}_{\Sigma} \underbrace{\begin{pmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}}_{\mathbf{V}^\top}$$



Three seminal algorithms

- Isomap

[▷ Joshua Tenenbaum, Vin de Silva, John Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science*, 2000.]

- Locally Linear Embedding (LLE)

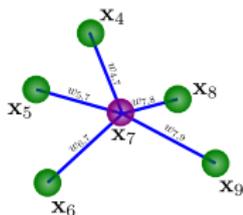
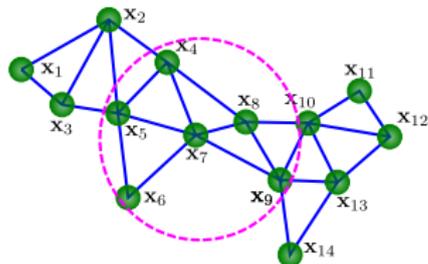
[▷ Sam Roweis, Lawrence Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, 2000.]

- Laplacian Eigenmaps (LapEigs)

[▷ M. Belkin, P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, June 2003; 15 (6):1373-1396.]

Algorithm: Locally Linear Embedding (LLE)

[▷ Sam Roweis & Lawrence Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, 2000.]



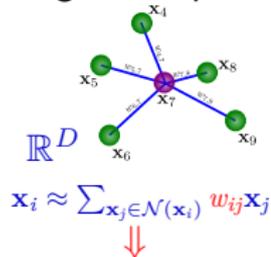
$$\mathbf{x}_i \approx \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} w_{ij} \mathbf{x}_j$$

Find the reconstructing weights

$$\mathcal{E}(\mathbf{W}) = \sum_{\mathbf{x}_i} \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} w_{ij} \mathbf{x}_j\|^2$$

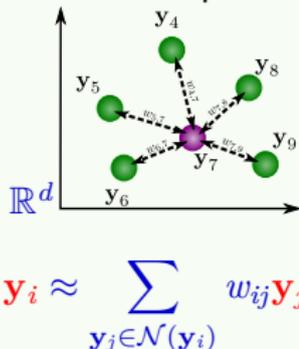
Algorithm: Locally Linear Embedding (LLE)

High-dim space



Preserve the weights

Low-dim space



Find the new coordinates that preserve the reconstructing weights

$$\mathcal{J}(\mathbf{W}, \mathcal{Y}) = \sum_{\mathbf{y}_i} \left\| \mathbf{y}_i - \sum_{\mathbf{y}_j \in \mathcal{N}(\mathbf{y}_i)} w_{ij} \mathbf{y}_j \right\|^2$$

Using the transformation:

$$\mathcal{T}(\mathbf{W}) = (\mathbf{I} - \mathbf{W})^\top (\mathbf{I} - \mathbf{W})$$

The solution is given by the $d + 1$ eigenvectors of $\mathcal{T}(\mathbf{W})$ corresponding to the smallest eigenvalues.

Three seminal algorithms

- Isomap

[▷ Joshua Tenenbaum, Vin de Silva, John Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science*, 2000.]

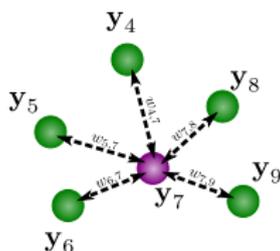
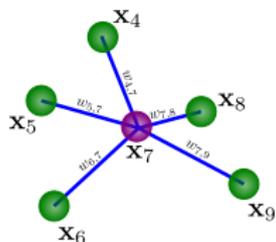
- Locally Linear Embedding (LLE)

[▷ Sam Roweis, Lawrence Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, 2000.]

- Laplacian Eigenmaps (LapEigs)

[▷ M. Belkin, P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, June 2003; 15 (6):1373-1396.]

Algorithm: Laplacian Eigenmaps (LapEig)



1. Neighborhood graph with similarity values w_{ij} (binary or Gaussian $w_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2})$)
2. Define a cost that preserves neighborhood relations: if $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \rightarrow \mathbf{y}_j \in \mathcal{N}(\mathbf{y}_i)$.

$$\mathcal{J}(\mathbf{W}, \mathcal{Y}) = \sum_i \sum_j w_{ij} (\mathbf{y}_i - \mathbf{y}_j)^2$$

3. Define $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$, where $d_i = \sum_j w_{ij}$ and the Laplacian

$$\mathbf{L} = \mathcal{T}(\mathbf{W}) = \mathbf{D} - \mathbf{W}$$

4. The min of $\mathcal{J}(\mathcal{Y})$ is given by the eigenvectors (corresp. to the smallest eigenvalues) of \mathbf{L} .

Three seminal algorithms

- Isomap
- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LapEigs)

Common Points and Differences

- **Non-linear:** build a non-linear mapping $\mathbb{R}^D \mapsto \mathbb{R}^d$
- **Graph-based:** Use a neighborhood graph to approx the manifold.
- Impose a **preservation criteria**:
 - **Isomap:** geodesic distances (the metric structure of the manifold).
 - **LLE:** the local reconstruction weights.
 - **LapEig:** the neighborhood relations.
- **Closed form** solution obtained through the spectral decomposition of a p.s.d. matrix (**spectral methods**).
- **Global vs. Local:**
 - **Isomap** : **global**, eigendecomposition of a **full** matrix.
 - **LLE, LapEig** : **local**, eigendecomposition of a **sparse** matrix.

Other popular algorithms

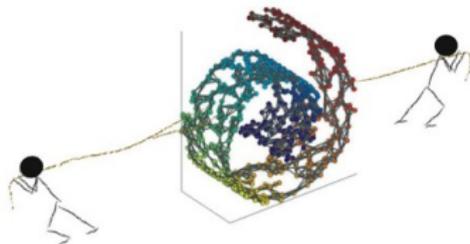
- Kernel PCA

[▷ B. Schölkopf, A. Smola, and K.R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*. 10, 5 (July 1998), 1299-1319.]

- LTSA (Local Tangent Space Alignment),

[▷ Z. Zhang and H. Zha. 2005. Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. *SIAM J. Sci. Comput.* 26, 1 (January 2005), 313-338.]

- MVU (Maximum Variance Unfolding or Semidefinite Embedding)



[▷ Weinberger, K. Q. Saul, L. K. An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding *National Conf. on Artificial Intelligence (AAAI)*. 2006.]

Other popular **probabilistic** algorithms

- GTM (Generative Topographic Mapping),

[▷ C. M. Bishop, M. Svensén and C. K. I. Williams, GTM: The Generative Topographic Mapping, *Neural Computation*, 1998, 10:1, 215-234.]

- GPLVM (Gaussian Process Latent Variable Models)

[▷ N. Lawrence, Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models, *Journal of Machine Learning Research* 6(Nov):1783–1816, 2005.]

- Diffusion maps

[▷ R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 2006]

- t-SNE (t-Distributed Stochastic Neighbor Embedding)

[▷ L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*. 9(Nov):2579-2605, 2008.]

Code

Laurens van der Maaten released a Matlab toolbox featuring 25 dimensionality reduction techniques

http://ticc.uvt.nl/~lvdrmaaten/Laurens_van_der_Maaten/Home.html

Outline

Manifold Learning

Three seminal algorithms

Common Practical Problems

Breaking the implicit assumptions

Determining the parameters

Mapping new points

Large data-sets

Conclusions

Breaking implicit assumptions

- The manifold assumption
- The sampling assumption

The manifold assumption

The high-dimensional input data lies on or close to a lower-dimensional manifold.

Manifold definition

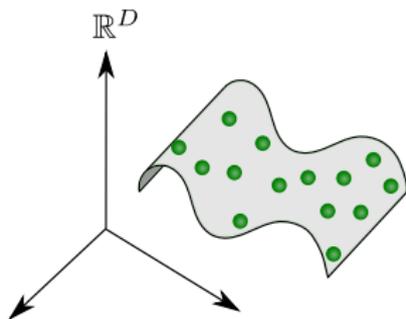
A manifold is a **topological space** that is **locally Euclidean**. [\[▷ Wolfram MathWorld\]](#)

- Does a low-dimensional manifold really exist?
- Does the data form clusters in low dimensional subspaces?
- How much noise can be tolerated?

Problem In many cases we do not know in advance.

The sampling assumption

*It is possible to acquire **enough** data samples with **uniform density** from the manifold*



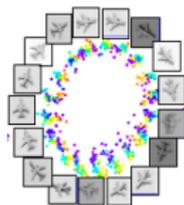
How many samples are really enough?

Breaking the assumptions

- Manifold assumption: Clusters!!!
 - Sampling assumption: Low number of samples w.r.t. the complexity of the manifold
 - Images of objects with deformations of high-order.
- Disconnected components (good for clustering).
- Unexpected Results.
- Not better than PCA or linear methods.

Good data examples

- **Rhythmic motions:** motion capture walk, breathing/cardiac motions, musical pieces
- Images/motions with a reduced number of **deformation “modes”**: MNIST dataset, population studies of a rigid organ.
- Images with **smooth changes** in viewpoint/lightning.



(walk.mp4)

Outline

Manifold Learning

Three seminal algorithms

Common Practical Problems

Breaking the implicit assumptions

Determining the parameters

Mapping new points

Large data-sets

Conclusions

Determining the parameters: Neighborhood Size

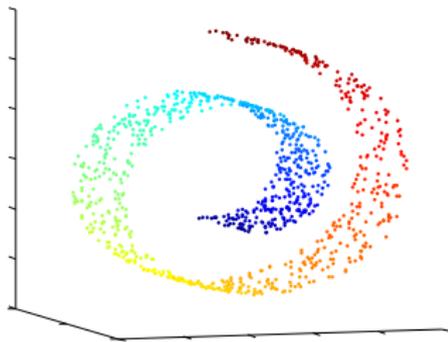
So far no standard principled way to automatically estimate the size of the neighborhood:

- Usually k-NN ϵ -ball neighborhoods are used.
- Attempts to make it **adaptive to the local density** of the points.
 - ▷ Jing Wang, Zhenyue Zhang. Adaptive Manifold Learning. *NIPS*. 2004
 - ▷ Zhang, Z.; Wang, J.; Zha, H.; *PAMI*. 2011

Determining the parameters: Dimensionality

What is the dimension of the manifold that best captures the structure of the data set?

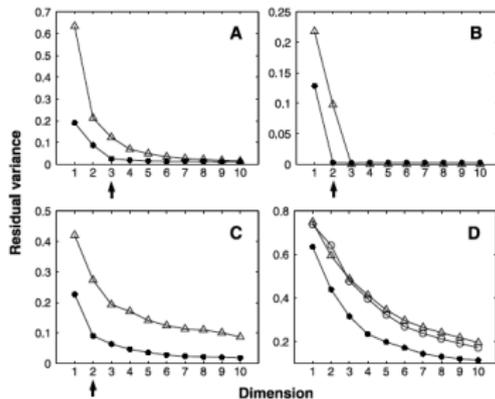
Intuitively, the **intrinsic dimensionality** of the manifold is the **number of independent parameters** needed to pick out a unique point inside the manifold. Examples $d = 2$



(manifold-walk.mp4)

Determining the parameters: Dimensionality

- in **PCA**, and **Isomap** the dimensionality chosen by imposing a threshold over the **residual variance** (e.g. 95%).
- in general the error of minimized cost function \mathcal{J} **does not decrease with increasing number of dimensions d** .



Decrease in error as the dimensionality d of \mathcal{Y} is increased in PCA and Isomap

[▷ Tenenbaum et.al., Science, 2000]

Determining the parameters: Dimensionality

Techniques for intrinsic dimensionality estimation.

- Eigenvalue-based estimation (like in PCA)

- Maximum Likelihood Estimator (MLE)

[▷ E. Levina and P. J. Bickel, Maximum Likelihood Estimation of Intrinsic Dimension, *Advances in Neural Information Processing Systems (NIPS)*, 777-784, 2005.]

- Correlation dimension (CorrDim)

[▷ P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9:189-208, 1983.]

- Nearest neighbor evaluation (NearNb)

[▷ Costa, J.A.; Girotra, A.; Hero, A.O.; Estimating Local Intrinsic Dimension with k-Nearest Neighbor Graphs. *IEEE Workshop on Statistical Signal Processing*, 2005]

- Packing numbers (PackingNumbers)

[▷ B. Kégl. Intrinsic dimension estimation using packing numbers. *Advances in Neural Information Processing Systems (NIPS)*. 2002]

- Geodesic minimum spanning tree (GMST)

[▷ J. Costa, A. Hero, Manifold Learning with Geodesic Minimal Spanning Trees . *Computing Research Repository - CORR*. 2003]

▷ J. Theiler. Statistical precision of dimension estimators. *Physical Review A*, 41(6):3038-3051, 1990.

▷ F. Camastra. Data dimensionality estimation methods: a survey. *Pattern Recognition*, 36:2945- 2954, 2003.

Outline

Manifold Learning

Three seminal algorithms

Common Practical Problems

Breaking the implicit assumptions

Determining the parameters

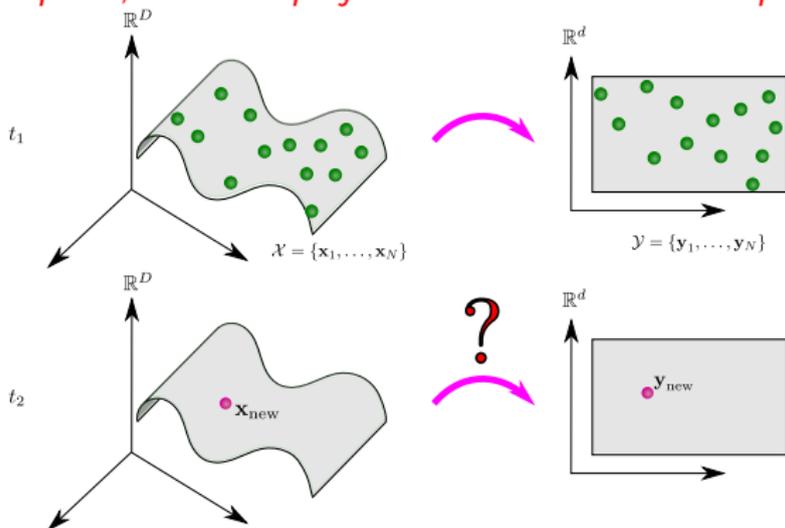
Mapping new points

Large data-sets

Conclusions

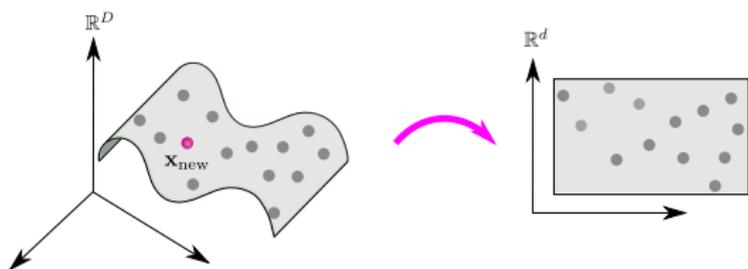
Mapping new points

I have a new point, how do I project it onto the low-dim representation?

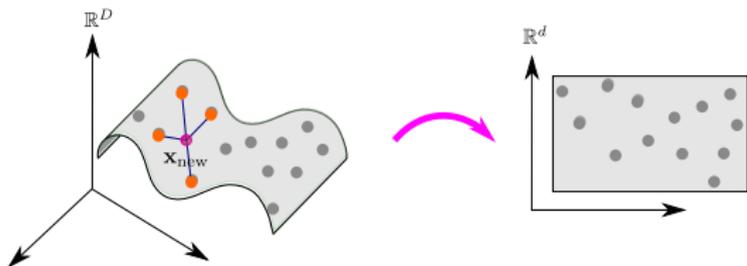


- There is no linear projection like in PCA, $\mathbf{y}_{\text{new}} = \mathbf{P}\mathbf{x}_{\text{new}}$.
- Most methods work with **batch data** and find directly (**without a projection step**) the coordinates of each data point in the low-dimensional space.

Mapping new points: Kernel Regression



Mapping new points: Kernel Regression



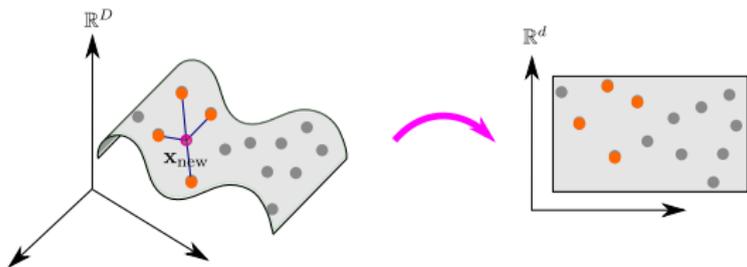
Find the neighborhood points of \mathbf{x}_{new}

$$\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_{\text{new}}) \quad \mathbf{x}_j \in \mathcal{X}$$

Using a kernel function

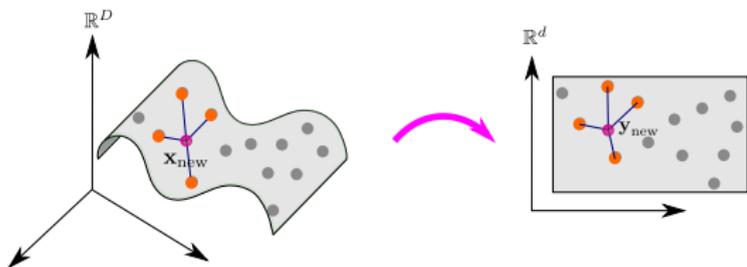
$$k(\mathbf{x}_{\text{new}}, \mathbf{x}_j)$$

Mapping new points: Kernel Regression



$$\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_{\text{new}}) \mapsto \mathbf{y}_j$$

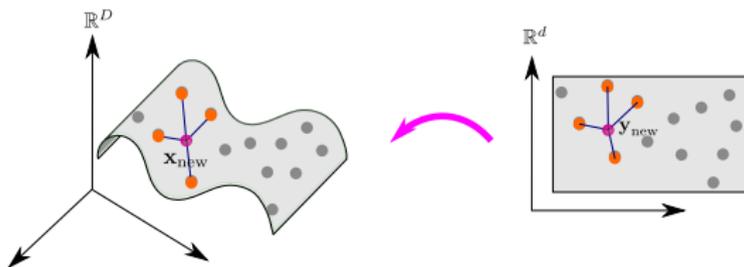
Mapping new points: Kernel Regression



$$y_{\text{new}} = \frac{\sum_{\{j|\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_{\text{new}})\}} k(\mathbf{x}_{\text{new}}, \mathbf{x}_j) y_j}{\sum_{\{j|\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_{\text{new}})\}} k(\mathbf{x}_{\text{new}}, \mathbf{x}_j)}$$

Backprojection

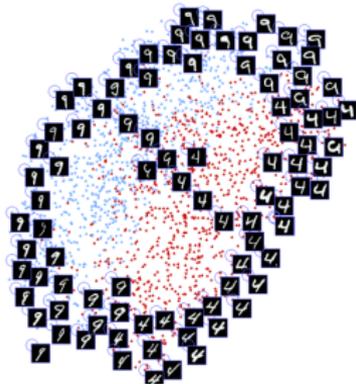
Back-projection is only implemented for some techniques, e.g. GPLVM



Similarly, kernel regression can be used to find the map.

Learn mappings along

- GPLVM: iterative, slow convergence.
- DRLIM [▷ Dimensionality Reduction by Learning an Invariant Mapping, CVPR 2006]
 - Uses contrastive learning
 - Defines a parametric function to map from $\mathbb{R}^D \mapsto \mathbb{R}^d$
 - Convolutional neural networks.
- P-DRUR [▷ Parametric Dimensionality Reduction by Unsupervised Regression, CVPR, 2010]
 - Learns both mappings $\mathbb{R}^D \mapsto \mathbb{R}^d$ and $\mathbb{R}^d \mapsto \mathbb{R}^D$
 - Mappings are radial basis function expansions
 - Variational problem
- Kernel Map [▷ Gerber, S.; Tasdizen, T.; Whitaker, R.; Dimensionality reduction and principal surfaces via Kernel Map Manifolds, ICCV, 2011.]



DRLIM

Outline

Manifold Learning

Three seminal algorithms

Common Practical Problems

Breaking the implicit assumptions

Determining the parameters

Mapping new points

Large data-sets

Conclusions

Large data-sets: the problem

For N samples:

- Find **nearest neighbors**: $O(N^2)$
- **Spectral decomposition** of $\mathcal{T}(W)$ (symmetric positive semi-definite matrix) : $O(N^3)$.

$$\mathcal{T}(W)\lambda = \nu\lambda$$

Iterative methods (Jacobi, Arnoldi, Hebbian), but

- Need matrix-vector products and several passes over data
- Not suitable for large dense matrices (Isomap), better for sparse matrices (LapEig).

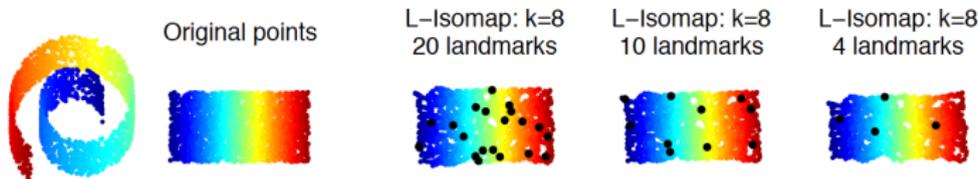
Large data-sets: some solutions

1. Hashing-based nearest neighbors

[> W. Liu, J. Wang, S. Kumar and S.F. Chang, Hashing with Graphs, *ICML*, 2011]

2. Use landmarks

- **Randomly** chosen and linear reconstruction for the remaining points.



[> V.D. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, *Advances in neural information processing systems (NIPS)*, 2003.]

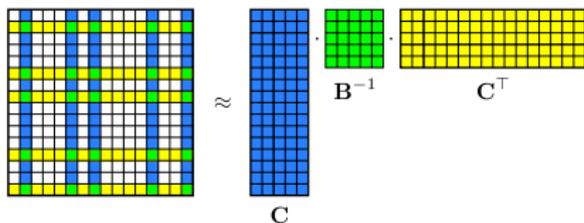
- Setting a **sparse regression problem** based on preserving the principal angles.

[> J Silva, J.S. Marques, J. Miranda Lemos. Selecting Landmark Points for Sparse Manifold Learning. *Neural Information Processing Systems (NIPS)*. 2005.]

Large data-sets: some solutions

3. Sampling-based approximation methods for the spectral decomposition.

- ▷ S. Kumar, M. Mohri, and A. Talwalkar. On sampling-based approximate spectral decomposition. *ICML*. 2009.
- ▷ K. Zhang, I. Tsang, J. Kwok. Improved Nyström Low Rank Approximation and Error Analysis. *ICML*. 2008.

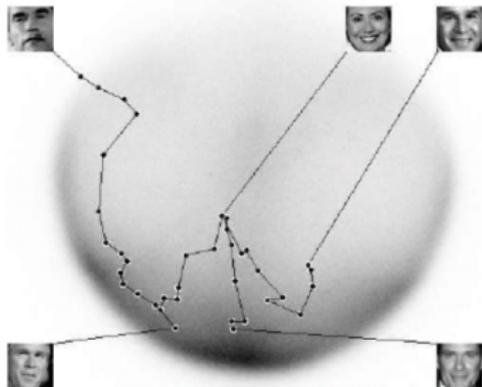


- **Nyström:** $\tilde{\mathbf{A}}_{\text{Nys}} = \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T \rightarrow O(l^3 + nld)$
- **Column-sampling:** $\tilde{\mathbf{A}}_{\text{col}} = \mathbf{C} \left(\left[\frac{l}{n} \mathbf{C}^T \mathbf{C} \right]^{1/2} \right)^{-1} \mathbf{C}^T \rightarrow O(nl^2)$

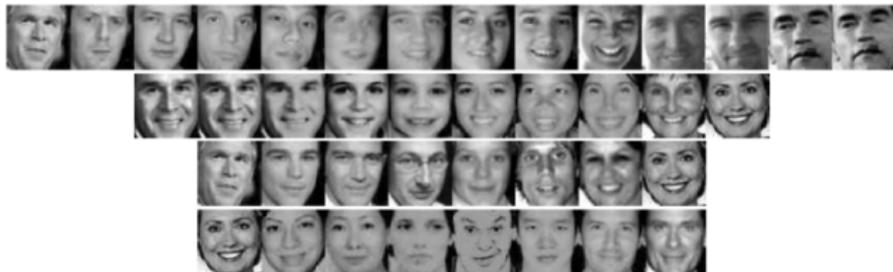
Different methods to sample: uniform, adaptive, ensemble, ...

Large data-sets: some solutions

Experiments on large databases **18M**.



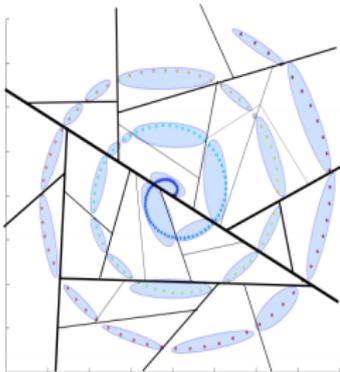
18M samples not enough!



Large data-sets: some solutions

4. Random Projections

Variant of the k-d tree which automatically adapts to intrinsic low dimensional structure in data.



- ▷ Y. Freund, S. Dasgupta, M. Kaba, N. Verma. Learning the structure of manifolds using random projections. *Neural Information Processing Systems (NIPS)*, 2007.
- ▷ C. Hegde, R. Baraniuk. Random Projections for Manifold Learning. *Neural Information Processing Systems (NIPS)*, 2007.
- ▷ Goldberg. Online semi-supervised learning. (*ECML*). 2008

Conclusions

Checklist to verify before using a manifold learning method

- Do I need a **non-linear** mapping?
- How likely is the data actually leaves close to a manifold?
- Can I acquire a **reasonable amount of samples** in accordance to the manifold complexity?
- Do I have any a-priori information on the **intrinsic dimensionality** of the manifold?
- Is all the data available at once or do **new points** need to be mapped?
- Is the optimization of the map (\mathcal{J}) coherent with my task?

Conclusions

Recall that some solutions exist in case of

- Need to map new points.
- Large data-sets.

Thanks for your attention!