

Manifold Learning for Signal and Visual Processing

Lecture 1: Introduction to spectral and graph-based methods

Radu Horaud

INRIA Grenoble Rhone-Alpes, France

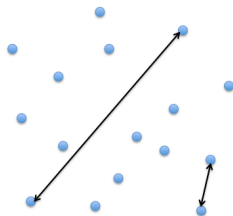
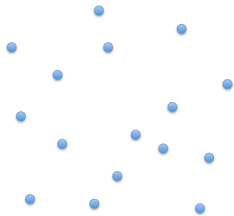
Radu.Horaud@inrialpes.fr

<http://perception.inrialpes.fr/>

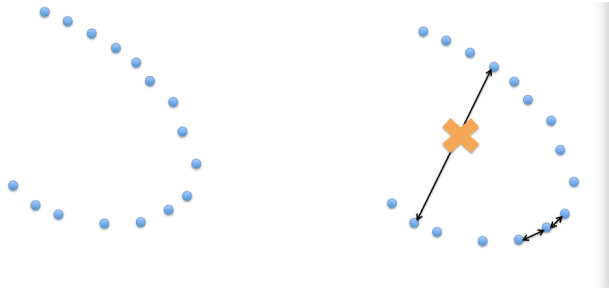
Introduction

- I do not have a formal definition of **manifold learning**
- The general philosophy of what we want to study:
 - *Input*: An unorganized cloud of points in \mathbb{R}^D , where D (the dimension of the observation space) may be arbitrarily large.
 - *Output*: An intrinsic representation (parameterization) of the linear or non-linear subspace that best characterizes the data.
- Linear dimensionality reduction: find a subspace $\mathbb{R}^d \subset \mathbb{R}^D$ with $d < D$, possibly $d \ll D$.
- Non-linear dimensionality reduction: find a manifold $\mathcal{M} \subset \mathbb{R}^d$ and a **global** parameterization of that manifold.

Metric spaces



Manifolds



Some definitions

- **Metric space:** one can compute the *distance* between any two points, e.g., Euclidean distances and Euclidean spaces.
- **Manifold:** every point has a neighborhood that is *homeomorphic* to an open subset of an Euclidean space.
- The dimension of a manifold is equal to the dimension of this Euclidean space
- One may say that a manifold is *locally* Euclidean while *globally* its structure is more complex.
- A *Riemannian manifold* is differentiable; the tangent space at each point on the manifold is an Euclidean space. The dimension of the tangent space is equal to the dimension of the manifold.

Discrete-data analysis and manifolds

- The theoretical properties of continuous spaces/manifolds do not easily extend to point clouds.
- Ideally, one would like to deal with dense data that are uniformly sampled from a linear or a non-linear space.
- Of course, this is rarely the case and one is left with the difficult task of analysing sparse and/or non-uniform sampled data.
- The representation of choice is an *undirected graph*:
 - Linear case: it is a complete (fully connected) graph – easy case.
 - Non-linear case: it is a sparse (locally connected) graph – difficult case.

Methods for linear dimensionality reduction

Preamble: The space spanned by the data is linear and not the method itself!

- **Principal component analysis (PCA):** It represents the data using the directions of maximum variance; it boils down to compute the principal eigenvectors of the *covariance* matrix of the data.
- **Multidimensional scaling (MDS):** It is a distance preserving method. It first computes a matrix whose entries are the pairwise dot-products between the data points and then it represents the data using the principal vectors of this *Gram* matrix.
- It can be shown that PCA and MDS are somehow equivalent:
 - PCA needs the point coordinates
 - MDS only needs the pairwise dot-products

Methods for non-linear dimensionality reduction

- **Graph-based methods:** The first step is to build a sparse graph with nodes representing data points and edges representing neighborhood relations. The second step is to build a graph matrix. The third step is to compute the principal eigenvectors of this matrix.
- **Kernel-based methods:** They use a kernel function to evaluate the dot-product and to construct a Gram matrix. They may be seen as a generalization of MDS. They can also be referred to as graph-based kernel methods (more on this later).
- Many other methods can be found in the literature and in textbooks.

Graph-based and kernel methods

- Kernel PCA
- ISOMAP
- Laplacian eigenmaps (LE)
- Locally linear embedding (LLE)
- Hessian eigenmaps (HE)
- Diffusion maps
- Heat-kernel embedding (HKE)
- Maximum variance unfolding
- ...

Other methods

- Principal curves and surfaces
- Curvature component analysis (CCA)
- Manifold charting
- Local tangent-space alignment (LTSA)
- Unsupervised kernel regression
- ...

Where to read about manifold learning?

There are numerous classical and recent textbooks that address linear/non-linear dimensionality reduction. Manifold learning is a more recent term. There are several tens of papers in the machine learning and statistics literature: NIPS, JML, JMLR, NECO, PAMI, etc. These books are interesting:

- C. Bishop. Pattern Recognition and Machine Learning (in particular chapter 12).
- J. Shawe-Taylor & N. Cristianini. Kernel Methods in Pattern Analysis (chapters 3, 5 & 6).
- J. A. Lee & M. Verleysen. Nonlinear Dimensionality Reduction.
- A. J. Izenman. Modern Multivariate Statistical Learning Techniques.

Mathematical notations

- Scalars: $a, A, \alpha, \lambda \dots$
- Vectors: $\mathbf{u} \in \mathbb{R}^M$ is a column vector while its transpose \mathbf{u}^\top is a row vector:

$$\mathbf{u}^\top = (u_1 \dots u_m \dots u_M)$$

- $\mathbf{1}$ denotes a column vectors of 1's.
- Matrices: \mathbf{U} and its transpose \mathbf{U}^\top

$$\mathbf{U} = [\mathbf{u}_1 \dots \mathbf{u}_N] = \begin{bmatrix} u_{11} & \dots & u_{N1} \\ \vdots & & \vdots \\ u_{1M} & \dots & u_{NM} \end{bmatrix}$$

- \mathbf{I}_N is the identity matrix of size $N \times N$.
- $\mathbf{I}_{m \times N}, m < N$ is a matrix formed with the top m rows of \mathbf{I}_N .

Dot-products, norms, distances, etc.

- Dot-product: $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_k x_{ik}x_{jk} = \mathbf{x}_i^\top \mathbf{x}_j$
- Vector norm: $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^\top \mathbf{x}$
- Euclidean distance:
$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle$$
- Matrix norm: $\|\mathbf{A}\|_F^2 = \sum_i \sum_j A_{ij}^2 = \text{tr}(\mathbf{A}^\top \mathbf{A})$. (This norm is known as the Frobenius norm and it is the most used matrix norm.)

An Intuitive Introduction to PCA and MDS

- Let's start with a few more notations:
- The input (observation) space: $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n \dots \mathbf{x}_N]$,
 $\mathbf{x}_n \in \mathbb{R}^D$
- The output (latent) space: $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_n \dots \mathbf{y}_N]$, $\mathbf{y}_n \in \mathbb{R}^d$
- **Projection:** $\mathbf{Y} = \mathbf{Q}^\top \mathbf{X}$ with \mathbf{Q}^\top a $d \times D$ matrix.
- **Reconstruction:** $\mathbf{X} = \mathbf{Q} \mathbf{Y}$ with \mathbf{Q} a $D \times d$ matrix.
- $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_d$
- *Reconstruction* will be useful for building a generative model – probabilistic PCA.

Computing the Spread of the Data

- We start with N scalars $x_1 \dots x_N$; the mean and the variance are given by:

$$\bar{x} = \frac{1}{N} \sum_n x_n \quad \sigma_x = \frac{1}{N} \sum_n (x_n - \bar{x})^2 = \frac{1}{N} \sum_n x_n^2 - \bar{x}^2$$

- More generally, for the data set \mathbf{X} :
- The mean: $\bar{\mathbf{x}} = \frac{1}{N} \sum_n \mathbf{x}_n$
- The covariance matrix is of dimension $D \times D$:

$$\mathbf{C}_X = \frac{1}{N} \sum_n (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top = \frac{1}{N} \mathbf{X}\mathbf{X}^\top - \bar{\mathbf{x}} \bar{\mathbf{x}}^\top$$

The Gram matrix

- The Gram matrix: Consider N data points $\mathbf{x}_1 \dots \mathbf{x}_N$ with mean $\bar{\mathbf{x}}$. The (i, j) entry of the associated *centred* Gram matrix is the dot-product of two centred data points:

$$G_{ij} = \langle \mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{x}_j - \bar{\mathbf{x}} \rangle$$

- The centred Gram matrix writes:

$$\mathbf{G}_x = \left(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}^\top \right)^\top \left(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}^\top \right) = \mathbf{J}\mathbf{X}^\top \mathbf{X}\mathbf{J}$$

with: $\mathbf{J} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. \mathbf{G} is an $N \times N$ positive semi-definite symmetric matrix.

- Note that its dimension corresponds to the number of data points and not to the dimension of the underlying space.

The Covariance and Gram Matrices, Side by Side

- For the same **centred data set** we have:
- A $D \times D$ covariance matrix: $\mathbf{C}_X = \frac{1}{N} \mathbf{X} \mathbf{X}^\top$
- A $N \times N$ Gram matrix: $\mathbf{G}_X = \mathbf{X}^\top \mathbf{X}$
- Let $\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^\top$ be the singular value decomposition (SVD) of the data set.
- We obtain for our matrices

$$\mathbf{C}_X = \frac{1}{N} \mathbf{U} \mathbf{S}^2 \mathbf{U}^\top \text{ and } \mathbf{G}_X = \mathbf{V} \mathbf{S}^2 \mathbf{V}^\top$$

Eigenvalues and Eigenvectors

- Let $\{\mathbf{u}_i, \lambda_i\}$ be the eigenvector-eigenvalue pairs of \mathbf{C}_X .
- Let $\{\mathbf{v}_i, \gamma_i\}$ be the eigenvector-eigenvalue pairs of \mathbf{G}_X .
- From the SVD of the data matrix \mathbf{X} we have:

$$\lambda_i = N^{-1}\gamma_i$$

- Let's establish a relationship between the eigenvectors. Starting with the Gram matrix, we successively obtain:

$$\begin{aligned}\mathbf{X}^\top \mathbf{X} \mathbf{v} &= \gamma \mathbf{v} \\ \mathbf{X} \mathbf{X}^\top (\mathbf{X} \mathbf{v}) &= \gamma (\mathbf{X} \mathbf{v}) \\ \mathbf{C}_X (\mathbf{X} \mathbf{v}) &= N^{-1} \gamma (\mathbf{X} \mathbf{v})\end{aligned}$$

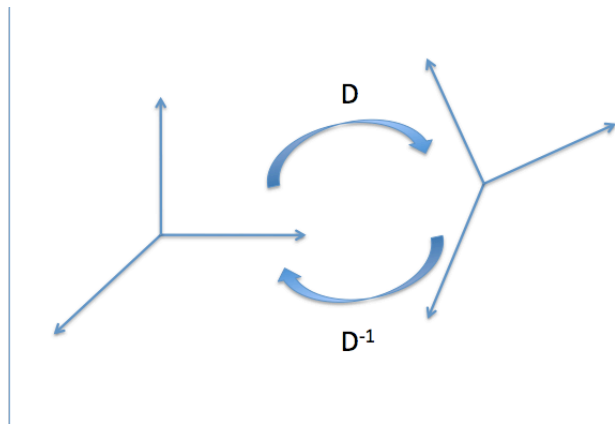
- Notice that: $\|\mathbf{X} \mathbf{v}\|^2 = \gamma$ (with $\mathbf{v}^\top \mathbf{v} = 1$), hence:

$$\mathbf{u} = \frac{\mathbf{X} \mathbf{v}}{\|\mathbf{X} \mathbf{v}\|} = \gamma^{-\frac{1}{2}} \mathbf{X} \mathbf{v}$$

Which Matrix to Use?

- The number of points N is larger than the dimension of the input space D : $\text{rank}(\mathbf{C}_X) \leq D$.
- The number of points is smaller than the dimension of the input space: $\text{rank}(\mathbf{C}_X) \leq N - 1$.
- If $N \geq D$ extract the eigenvectors and eigenvalues of \mathbf{C}_X .
- If $N < D$ extract instead the eigenvectors and eigenvalues of \mathbf{G}_X . The remaining $D - N + 1$ eigenvalues of \mathbf{C}_X are of value zero.

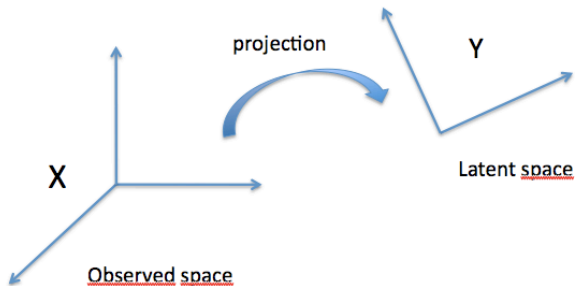
Changing the Coordinate Frame



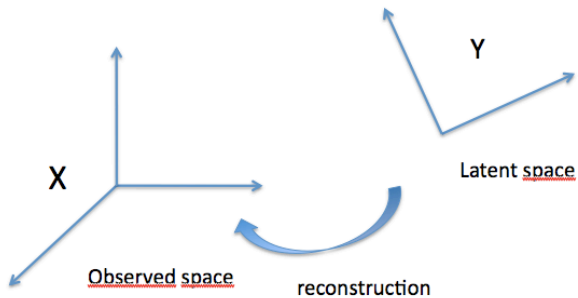
Orthogonal Transformations

- If we ignore the translation between two frames, \mathbf{D} is reduced to a $D \times D$ orthonormal matrix \mathbf{R} :
- $\mathbf{R}\mathbf{R}^\top = \mathbf{I}_D$
- The rows are mutually orthogonal, the columns are mutually orthogonal, the norm of each row- and column-vector is equal to 1.
- $\det(\mathbf{R}) = \pm 1$
- These matrices belong to O_D which is a notation for the *orthogonal group* of dimension D
- The *special orthogonal group* SO_D is characterized by $\det(\mathbf{R}) = +1$

Projecting the Data



"Reconstructing" the Data



Projection Versus Reconstruction

- Projection of \mathbb{R}^D onto \mathbb{R}^d : Remove $D - d$ rows of \mathbf{R}^\top to obtain a $d \times D$ row-orthogonal matrix \mathbf{Q}^\top .
- Reconstruction of \mathbb{R}^D from \mathbb{R}^d : Remove $D - d$ columns of \mathbf{R} to obtain a $D \times d$ column-orthogonal matrix \mathbf{Q}
- $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_d$ but $\mathbf{Q} \mathbf{Q}^\top \neq \mathbf{I}_D$!!
- Questions:
 - How to choose the low-dimensional reference frame?
 - How to choose d ?
 - How to select d **principal** directions?
- Both PCA and MDS attempt to answer these questions.

Maximum Variance Formulation of PCA

- Let's center and project the data \mathbf{X} onto a line along a unit vector \mathbf{u} . The variance along this line writes:

$$\begin{aligned}\sigma_u &= \frac{1}{N} \sum_i (\mathbf{u}^\top (\mathbf{x}_n - \bar{\mathbf{x}}))^2 \\ &= \mathbf{u}^\top \left(\frac{1}{N} \sum_i (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top \right) \mathbf{u} \\ &= \mathbf{u}^\top \mathbf{C}_X \mathbf{u}\end{aligned}$$

- Maximizing the variance under the constraint that \mathbf{u} is a unit vector:

$$\mathbf{u}^* = \underset{\mathbf{u}}{\operatorname{argmax}} \left\{ \mathbf{u}^\top \mathbf{C}_X \mathbf{u} + \lambda(1 - \mathbf{u}^\top \mathbf{u}) \right\}$$

Maximum Variance Solution

- First note that the $D \times D$ covariance matrix is a symmetric semi-definite positive matrix. Therefore the quadratic form above is non-negative.
- Taking the derivative with respect to \mathbf{u} and setting the derivatives equal to 0, yields: $\mathbf{C}_X \mathbf{u} = \lambda \mathbf{u}$
- Making use of the fact that \mathbf{u} is a unit vector we obtain:
$$\sigma_{\mathbf{u}} = \lambda$$
- **Solution:** The *principal* or largest eigenvector–eigenvalue pair $(\mathbf{u}_{\max}, \lambda_{\max})$ of the covariance matrix.

Eigendecomposition of the Covariance Matrix

- Assume that the data are centred:

$$N\mathbf{C}_X = \mathbf{X}\mathbf{X}^\top = N\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$$

Where \mathbf{U} is a $D \times D$ orthogonal matrix and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues.

- If the data point lie on a lower dimensional space:

$$d = \text{rank}(\mathbf{X}) < D$$

and

$$\mathbf{\Lambda} = [\lambda_1 \dots \lambda_d \ 0 \dots 0]$$

$$\mathbf{C}_X = \tilde{\mathbf{U}}\mathbf{\Lambda}_d\tilde{\mathbf{U}}^\top$$

- $\tilde{\mathbf{U}} = \mathbf{U}\mathbf{I}_{D \times d}$ is a $D \times d$ column-orthogonal matrix (reconstruction).
- $\tilde{\mathbf{U}}^\top = \mathbf{I}_{d \times D}^\top \mathbf{U}^\top$ is a $d \times D$ row-orthogonal matrix (projection).

Data Projection on a Linear Subspace

- From $\mathbf{Y} = \mathbf{Q}^\top \mathbf{X}$ we have

$$\mathbf{Y}\mathbf{Y}^\top = \mathbf{Q}^\top \mathbf{X}\mathbf{X}^\top \mathbf{Q} = n\mathbf{Q}^\top \tilde{\mathbf{U}}\mathbf{\Lambda}_d\tilde{\mathbf{U}}^\top \mathbf{Q}$$

- 1 The projected data have a diagonal covariance matrix: $\frac{1}{N}\mathbf{Y}\mathbf{Y}^\top = \mathbf{\Lambda}_d$, by identification we obtain

$$\mathbf{Q}^\top = \tilde{\mathbf{U}}^\top$$

- 2 The projected data have an identity covariance matrix, this is called *whitening the data*: $\frac{1}{N}\mathbf{Y}\mathbf{Y}^\top = \mathbf{I}_d$

$$\mathbf{Q}^\top = \mathbf{\Lambda}_d^{-\frac{1}{2}}\tilde{\mathbf{U}}^\top$$

- Projection of the data points onto principal direction \mathbf{u}_i :

$$(y_{1i} \dots y_{Ni}) = \underbrace{\lambda_i^{-1/2}}_{\text{whitening}} \mathbf{u}_i^\top (\mathbf{x}_1 \dots \mathbf{x}_n \dots \mathbf{x}_N)$$

Illustration of PCA - The Input Data

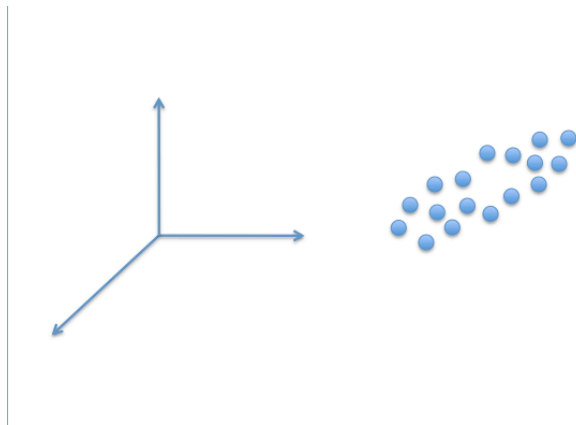


Illustration of PCA - Centering the Data

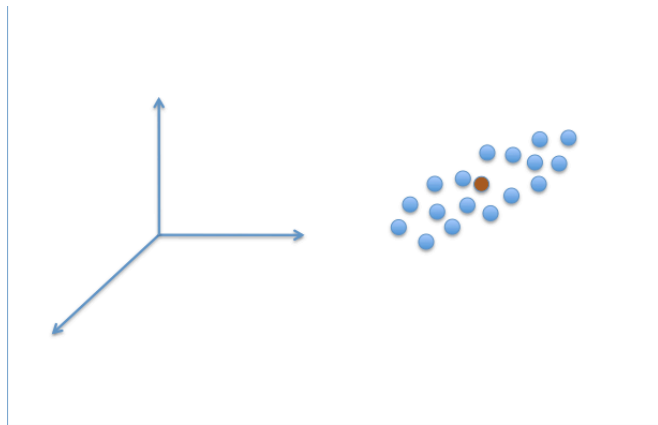
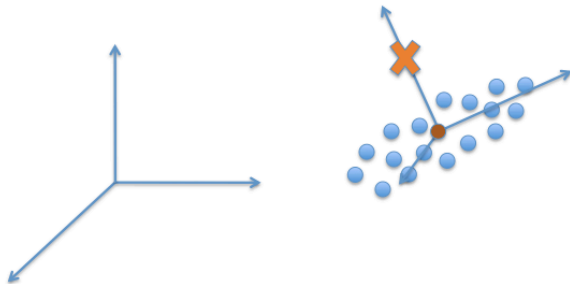


Illustration of PCA - Principal Eigenvectors of the Data



Metric MDS

- MDS uses the Gram matrix (dot-products). The data points \mathbf{X} are not explicitly required.
- Minimization criterion:

$$\min_{\mathbf{Y}} \|\mathbf{G}_X - \mathbf{Y}^\top \mathbf{Y}\|_F^2 \text{ with } \mathbf{G}_X = \tilde{\mathbf{V}} \mathbf{\Lambda}_d \tilde{\mathbf{V}}^\top$$

- Note that:

$$\|\mathbf{G}_X - \mathbf{Y}^\top \mathbf{Y}\|_F^2 = \text{tr}(\mathbf{G}_X^\top \mathbf{G}_X) + \text{tr}((\mathbf{Y}^\top \mathbf{Y})^2) - 2\text{tr}(\mathbf{G}_X \mathbf{Y}^\top \mathbf{Y})$$

- The criterion becomes:
 $\min_{\mathbf{Y}} \{\text{tr}((\mathbf{Y}^\top \mathbf{Y})^2) - 2\text{tr}(\mathbf{G}_X \mathbf{Y}^\top \mathbf{Y})\}$ and the solution and its covariance are:

$$\mathbf{Y} = \mathbf{\Lambda}_d^{\frac{1}{2}} \tilde{\mathbf{V}}^\top, \quad \text{NC}_Y = \mathbf{Y}\mathbf{Y}^\top = \mathbf{\Lambda}_d$$