Program

~30min	**1. Introduction** Carole Lartizien
~75min	**2. Supervised learning** Rémi Emonet + Carole Lartizien
~75min	**3. Unsupervised learning** Nicolas Duchateau + Rémi Emonet
~30min	**4. Methods evaluation** Carole Lartizien + Rémi Emonet + Nicolas Duchateau
	5. Conclusions / to go further

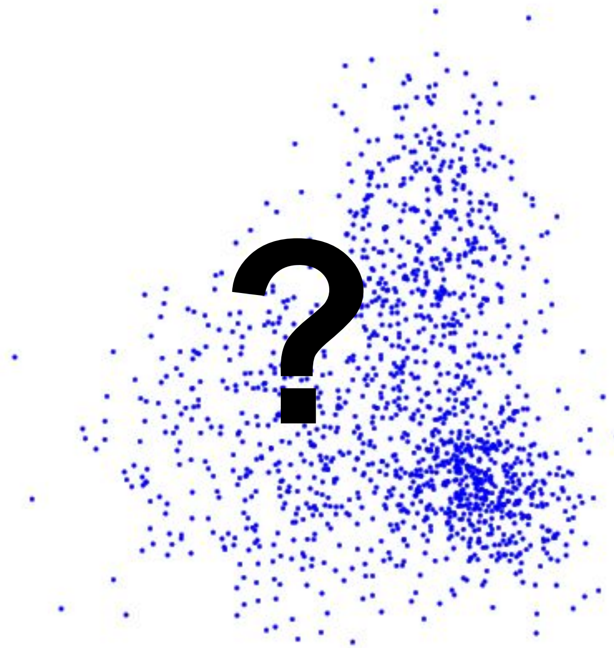
3. Unsupervised learning

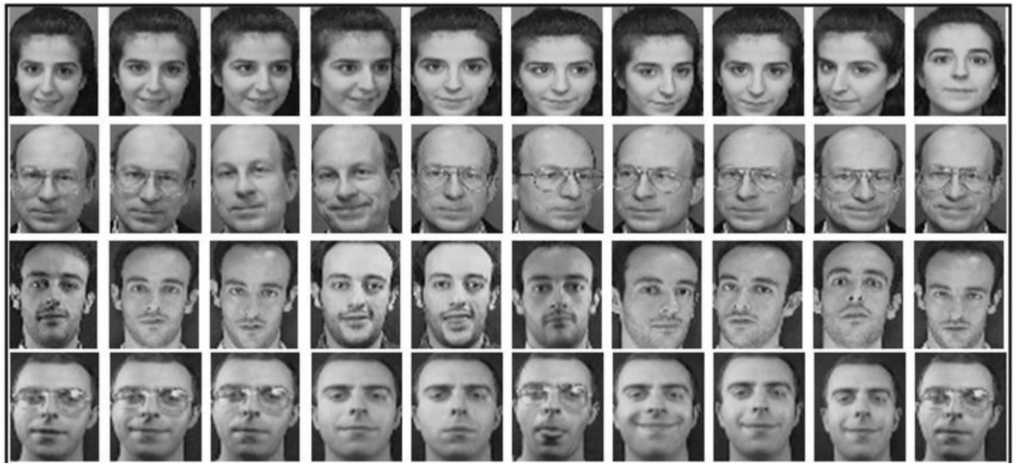
Why not using labels?

3. Unsupervised learning

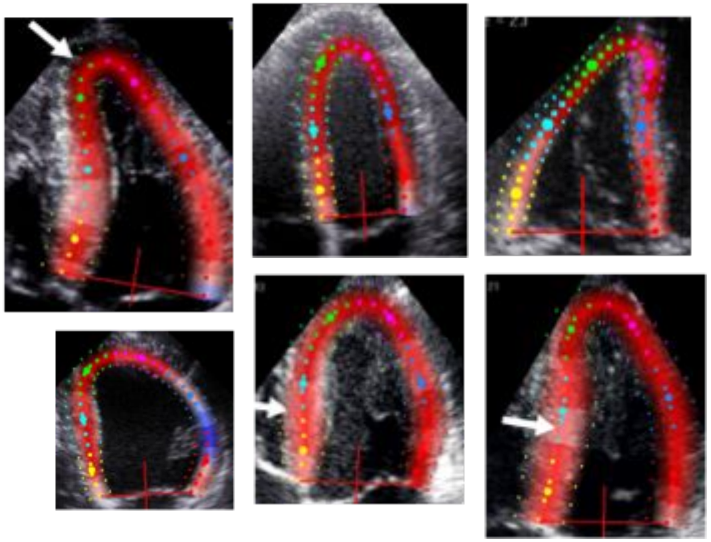
Why not using labels?

→ No labels available





ORL database



Myocardial strain patterns

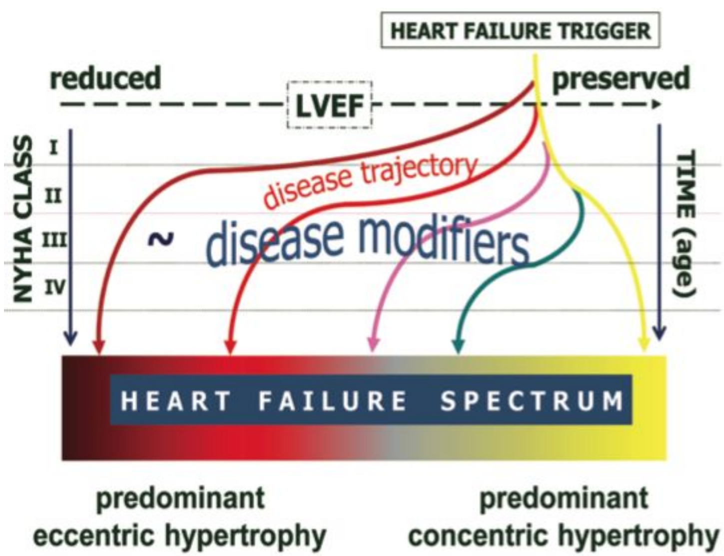
3. Unsupervised learning

Why not using labels?

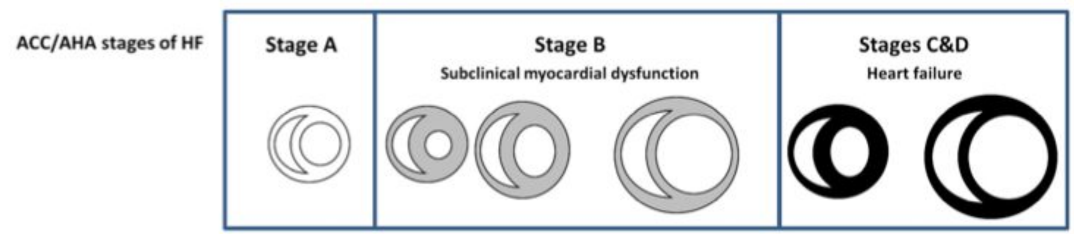
➔ No labels available

➔ **Low relevance of labels**

Continuum of disease



De Keulenaer et al. *Circulation* 2011



Geometrical Description	Normal	Concentric remodeling	Concentric hypertrophy	Eccentric hypertrophy	Concentric hypertrophy	Eccentric hypertrophy
Pathophysiological Description	Risk factors (e.g. DM/HTN)	Myocardial dysfunction with pEF		Myocardial dysfunction with rEF	HFpEF	HFrEF
Deposition (collagen, fibrosis, etc.)	↑	↑	↑	↑↑	↑↑	↑↑↑
Dimensions	n	n/↓	n	↑	n	[illegible]
Thickness						
LV mass	n	n/↑	↑	↑	n	[illegible]
EF	n	n		↓	n	↓↓
Wall stress	n	n/↑		↑	↑↑	↑↑↑
GLS	↓	↓		↓↓	↓↓	↓↓↓
GCS	n	n/↑		↓	↑	↓↓
GRS	n	n/↓		↓	↓	↓↓

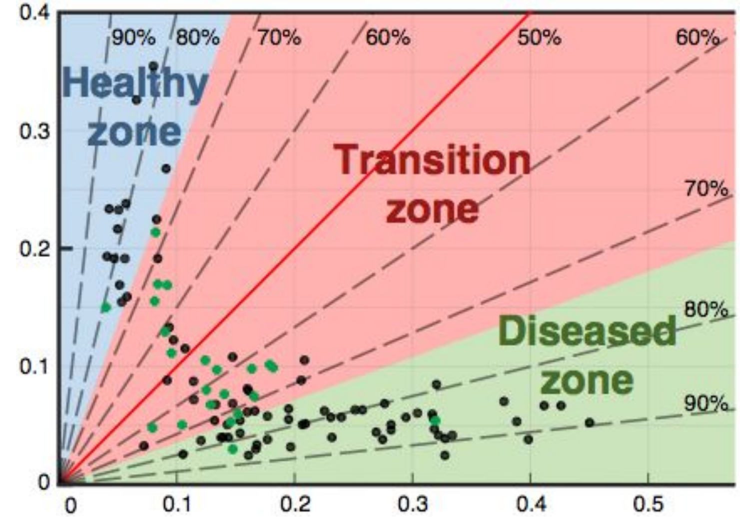
Omar et al. *Circ Res* 2016

3. Unsupervised learning

Why not using labels?

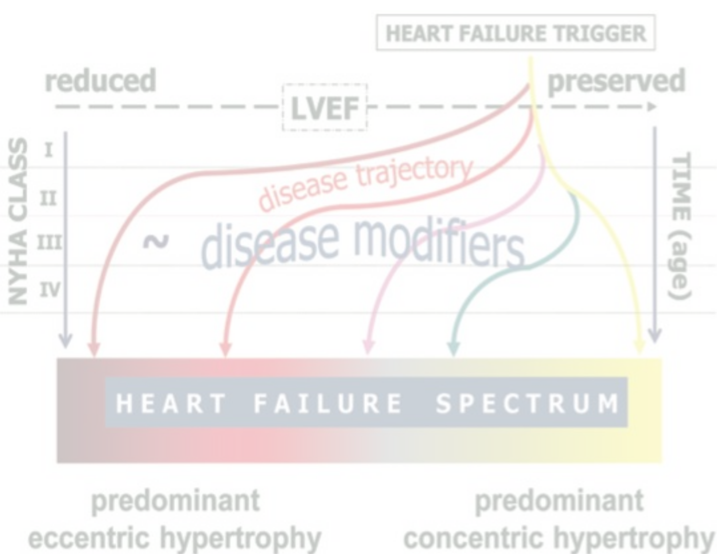
➔ No labels available
➔ **Low relevance of labels**

ex: heart failure preserved ejection

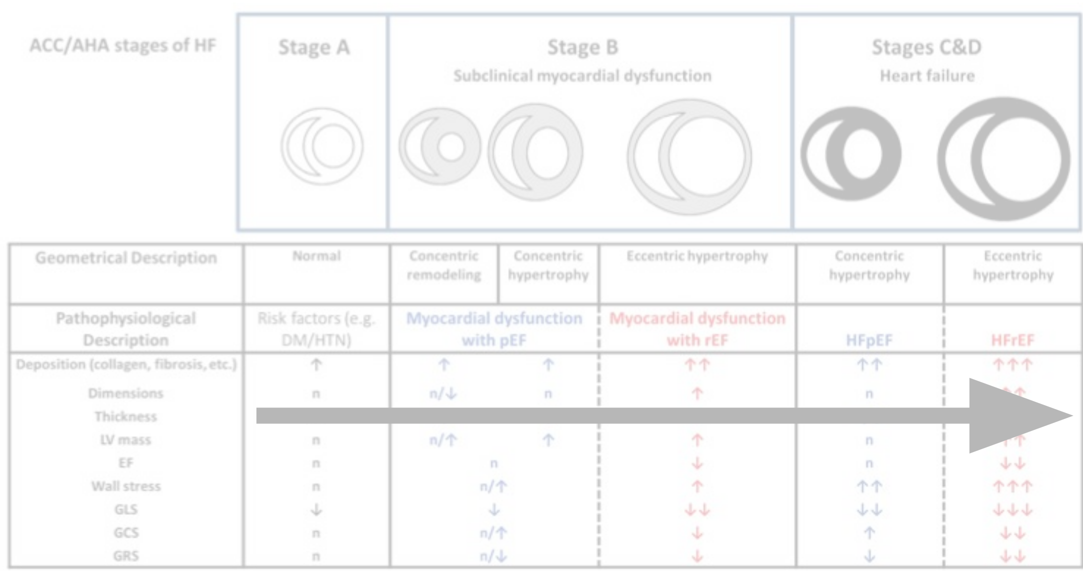


Sanchez-Martinez et al. *Circ Cardiovasc Imag* 2018

Continuum of disease



De Keulenaer et al. *Circulation* 2011



Omar et al. *Circ Res* 2016

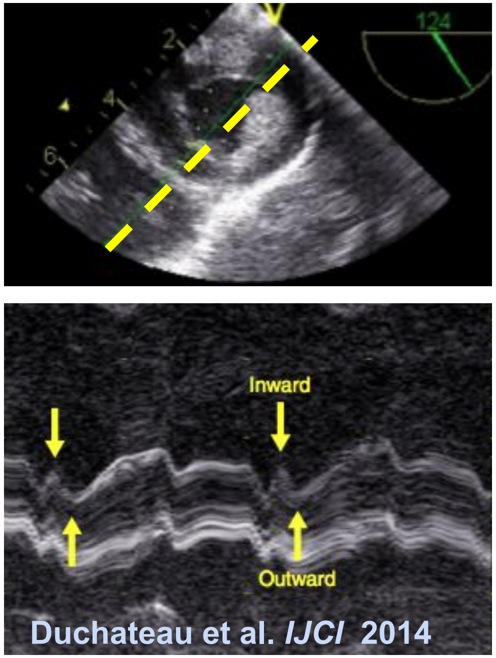
3. Unsupervised learning

Why not using labels?

➔ No labels available
➔ Low relevance of labels
➔ **Limits of supervised approaches**

ex: Cardiac Resynchronization Therapy (CRT)

- 30% of non-responders
- cases selected in a supervised way



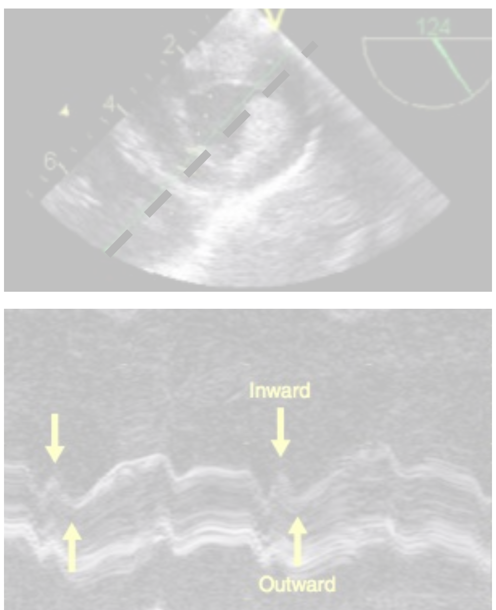
Duchateau et al. *IJCI* 2014

Why not using labels?

- ➔ No labels available
- ➔ Low relevance of labels
- ➔ **Limits of supervised approaches**

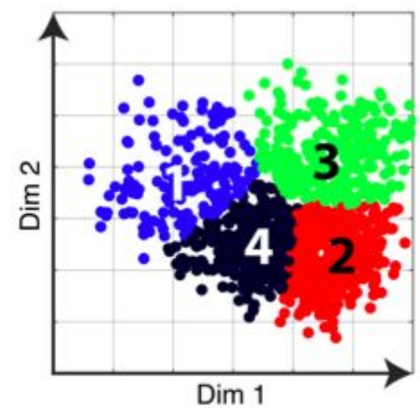
ex: Cardiac Resynchronization Therapy (CRT)

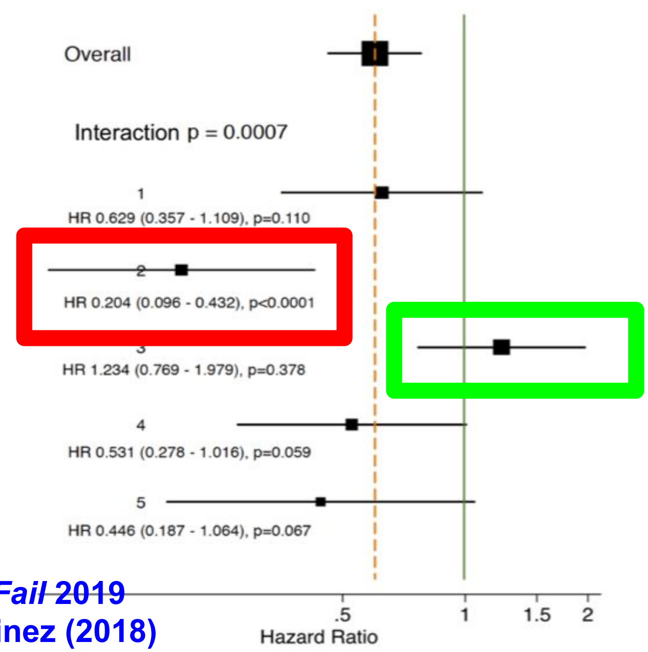
- 30% of non-responders
- cases selected in a supervised way



Duchateau et al. *IJCI* 2014

High-risk and low-risk clusters
identified by unsupervised learning





Cikes et al. *Eur J Heart Fail* 2019
PhD of S. Sanchez-Martinez (2018)

3. Unsupervised learning

Why not using labels?

- ➔ No labels available
- ➔ Low relevance of labels
- ➔ Limits of supervised approaches

➔ **Or simply a different point of view?**

Which **end point** for the application?

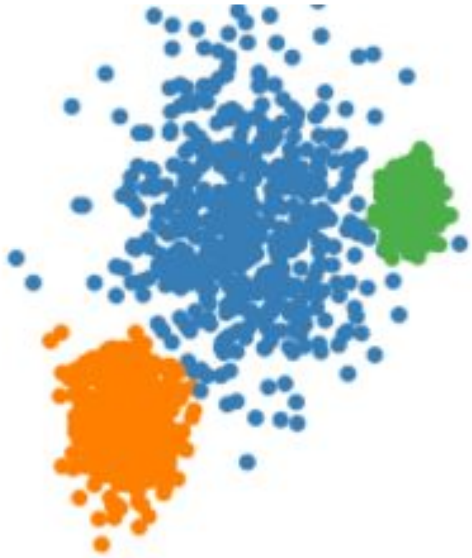
3. Unsupervised learning

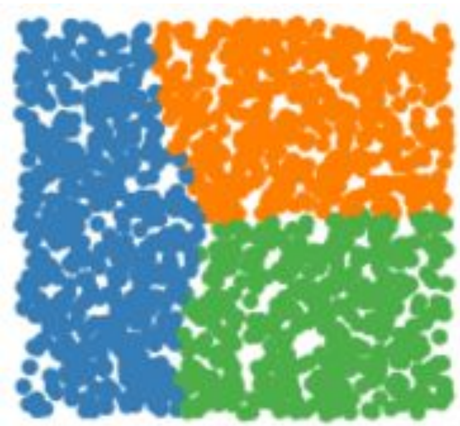
Which end point?

➔	Subgroups identification / similar trends	Clustering
➔	Detect novelty / unexpected values	Outliers detection
➔	Understand the data space	(low-dimensional) embedding
➔	Statistical distances	Manifold learning
➔	Sampling/generate new cases	Reconstruction

3. Unsupervised learning

Clustering





Clustering

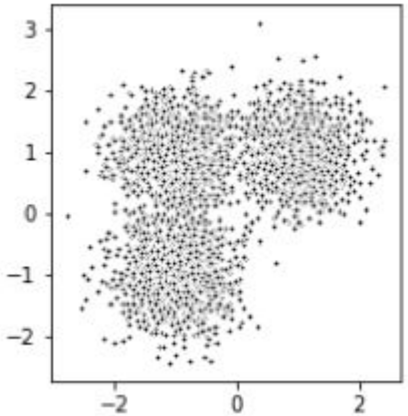
➔ ***K*-means**

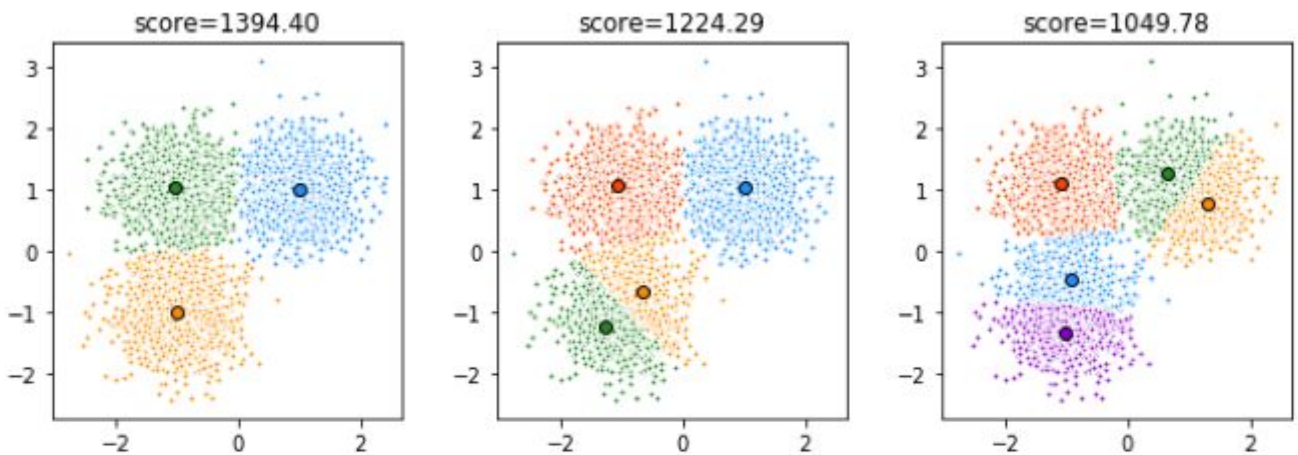
Parameter = *K* (number of clusters) $\mathbf{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots \mathcal{S}_K\}$

Idea = minimize the within-cluster variance

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{\mathbf{v} \in \mathcal{S}_k} \|\mathbf{v} - \mu_k\|^2$$

(distance to each centroid)



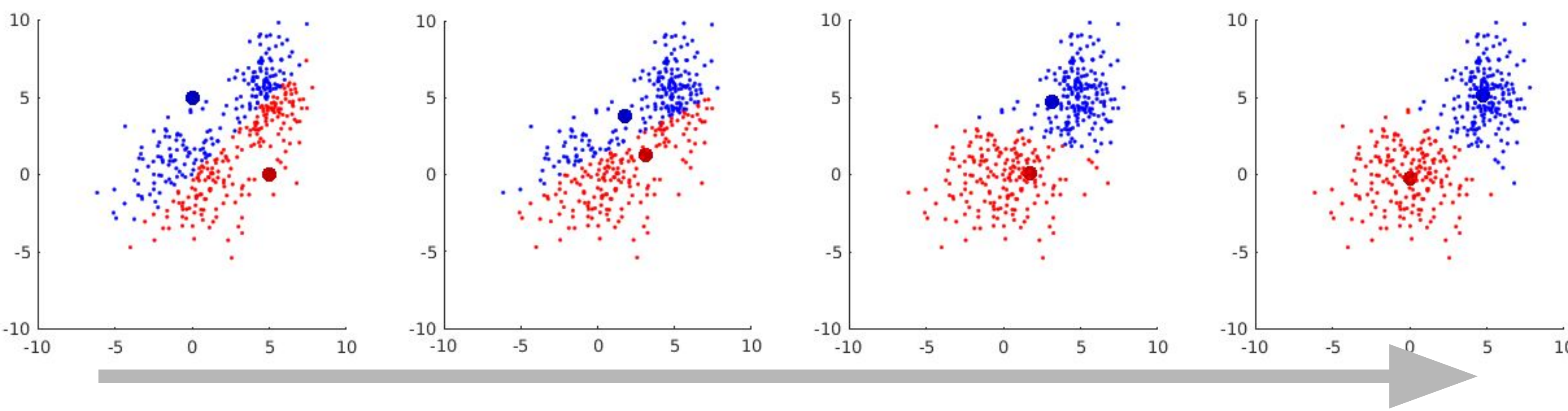


Clustering

➔ ***K*-means**

Lloyd's algorithm:

1. **Initialize** centroids (e.g. *K* random samples)
2. Assign each sample to its **nearest centroid**
3. **Update centroids** as average of assigned samples



Clustering

➔ *K*-means

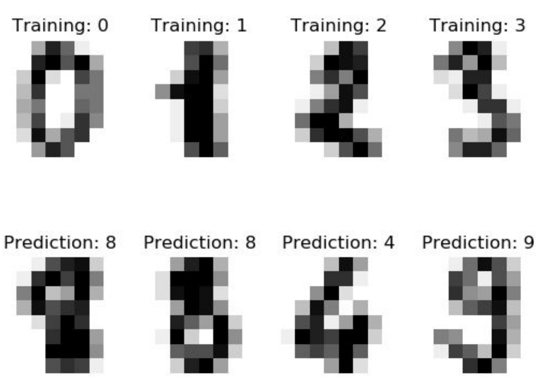
Lloyd's algorithm:

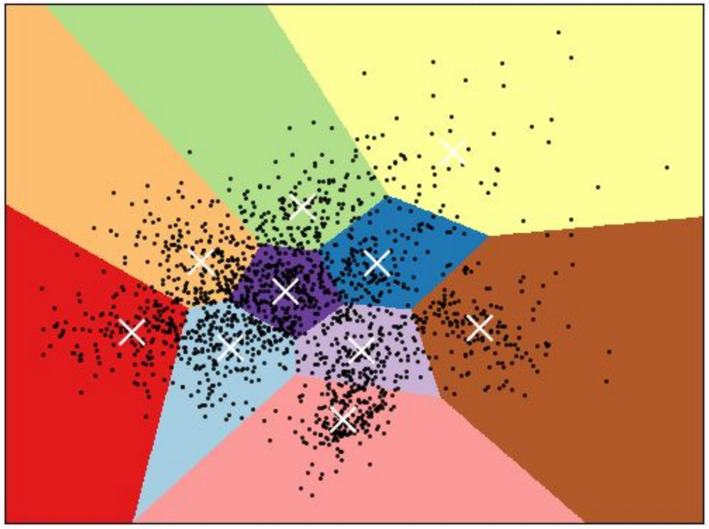
1. **Initialize** centroids (e.g. *K* random samples)
2. Assign each sample to its **nearest centroid**
3. **Update centroids** as average of assigned samples

Comments:

- converges to **local minimum**: needs several restarts
- simple **cluster boundaries**
- not applicable in **high dimension**: reduce dimensionality first

8x8 images, *K*=10





scikit learn

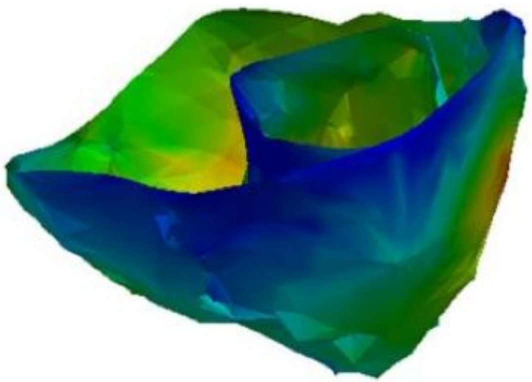
Clustering

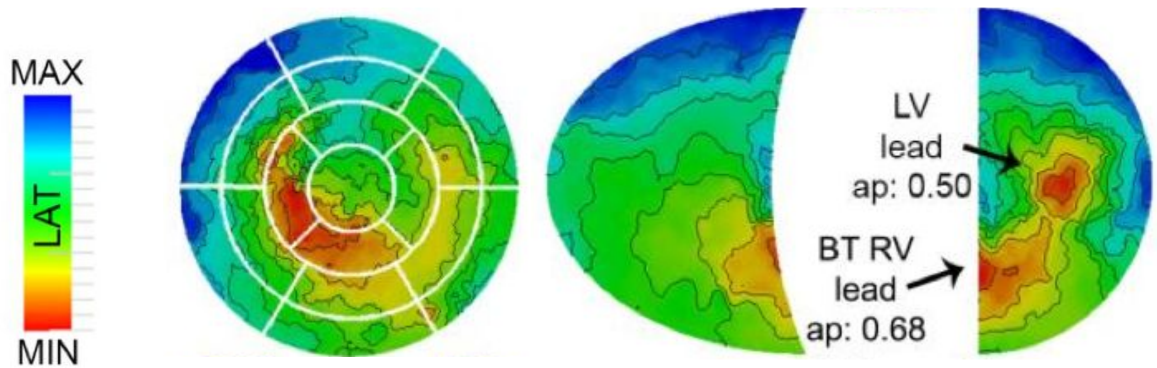
ex: cardiac resynchronization therapy

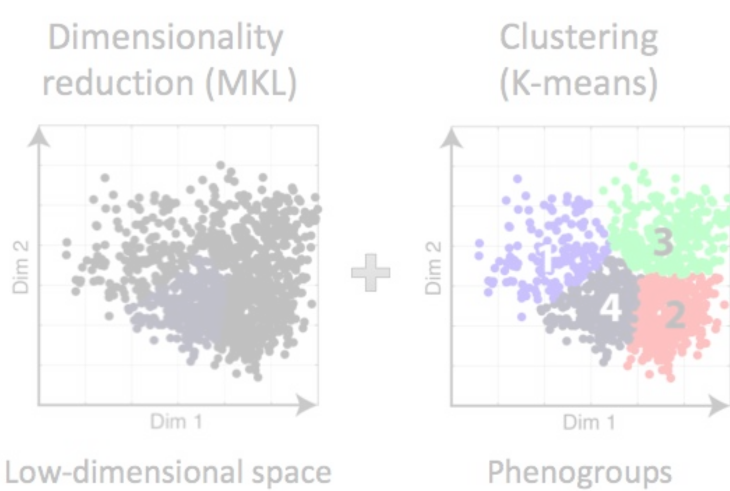
➔ ***K*-means**

Lead location from electroanatomical activation maps

Soto-Iglesis et al. *IEEE J Transl Eng Health Med* 2017







High-risk and low-risk clusters
identified by <u>unsupervised</u> learning

Cikes et al. *Eur J Heart Fail* 2019
PhD of S. Sanchez-Martinez (2018)

3. Unsupervised learning

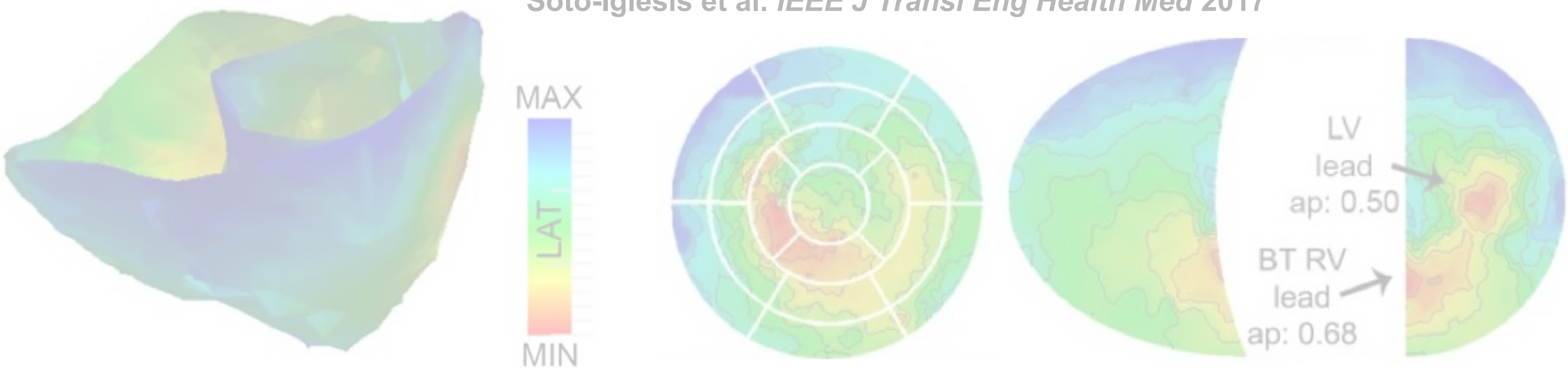
Clustering

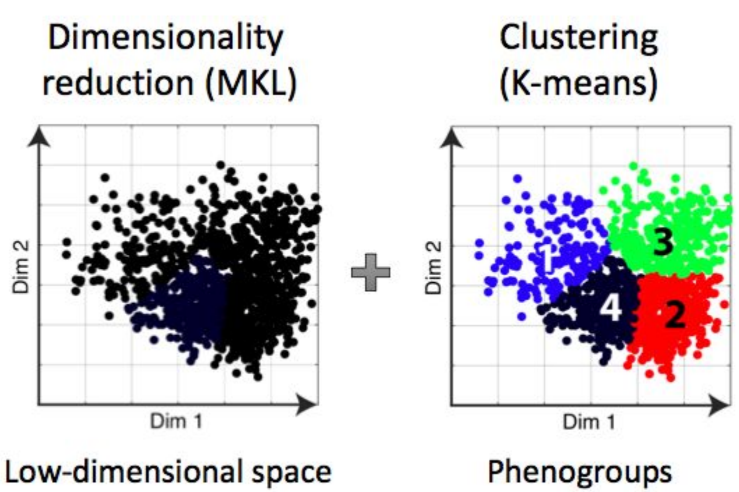
ex: cardiac resynchronization therapy

➔ ***K*-means**

Lead location from electroanatomical activation maps

Soto-Iglesis et al. *IEEE J Transl Eng Health Med* 2017



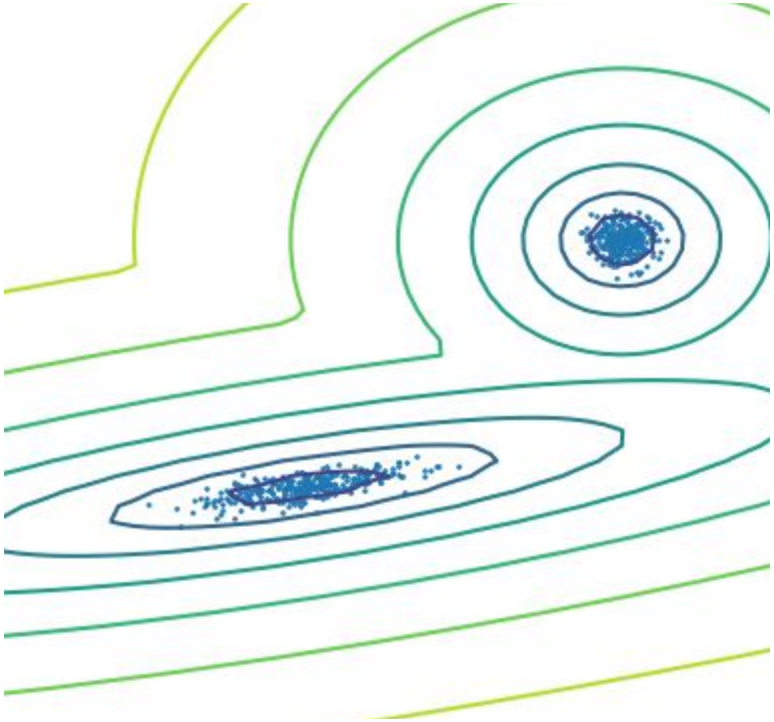


High-risk and low-risk clusters
identified by <u>unsupervised</u> learning

Cikes et al. *Eur J Heart Fail* 2019
PhD of S. Sanchez-Martinez (2018)

Clustering

➔ **Gaussian mixture models**



Clustering

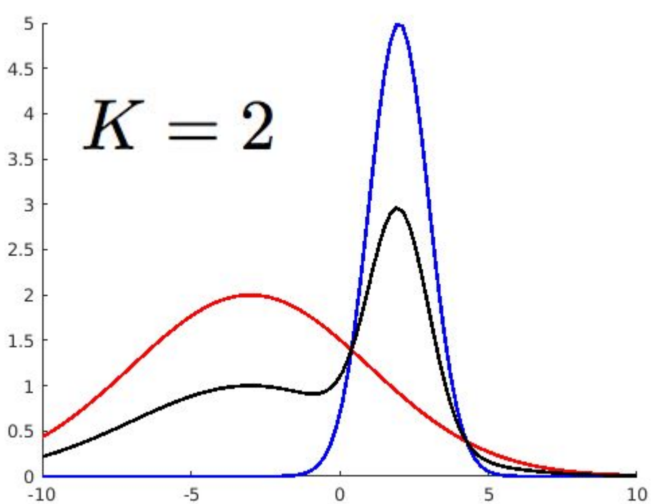
➔ Gaussian mixture models

<u>Parameters</u> = weights + mean + covariance of each Gaussian

<u>Idea</u> = samples generated from a mixture of *K* Gaussian ~ generalization of *K*-means

$$p(\mathbf{v}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{v}|\mu_k, \Sigma_k)$$

$$p(\mathbf{z}_k) \qquad p(\mathbf{v}|\mathbf{z}_k)$$



Clustering

➔ Gaussian mixture models

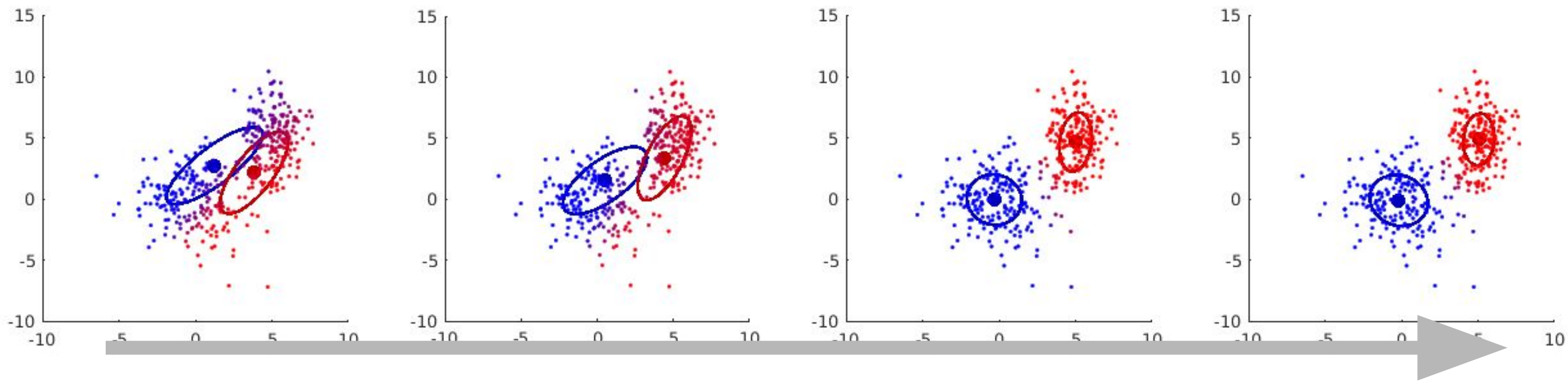
Algorithm = Expectation-Maximization (EM)

1. **Initial** random components (e.g. around *K*-means centroids)
2. Compute **probability at each point** $\gamma_k^{(i)} = p(\mathbf{z}_k | \mathbf{v}^{(i)})$
3. **Maximize likelihood** / update parameters

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_k^{(i)}$$

$$\mu_k = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \gamma_k^{(i)} \mathbf{v}^{(i)}$$

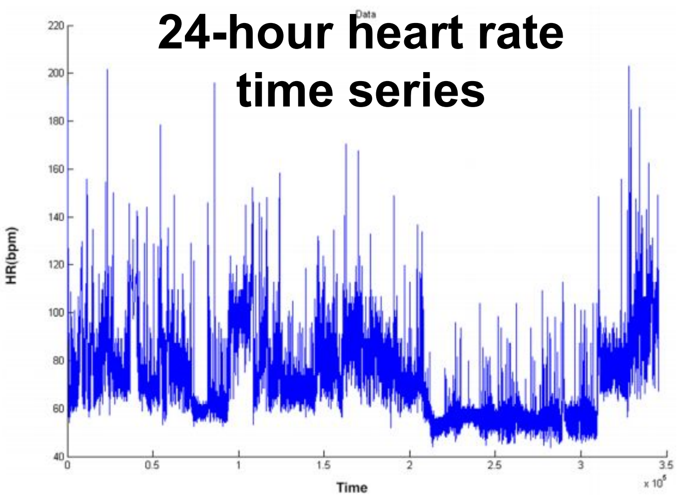
$$\Sigma_k = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \gamma_k^{(i)} (\mathbf{v}^{(i)} - \mu_k)(\mathbf{v}^{(i)} - \mu_k)^T$$



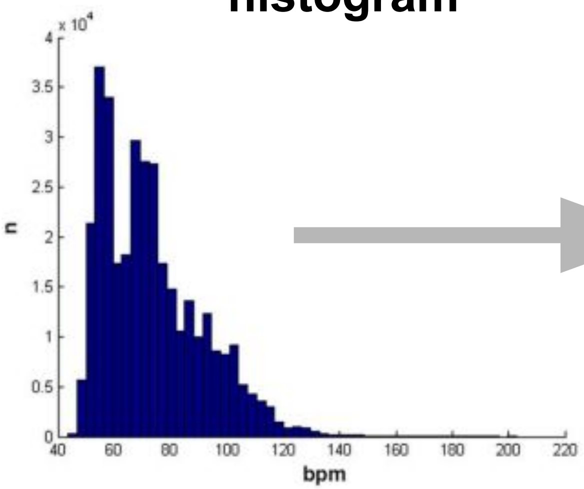
Clustering

➔ **Gaussian mixture models**

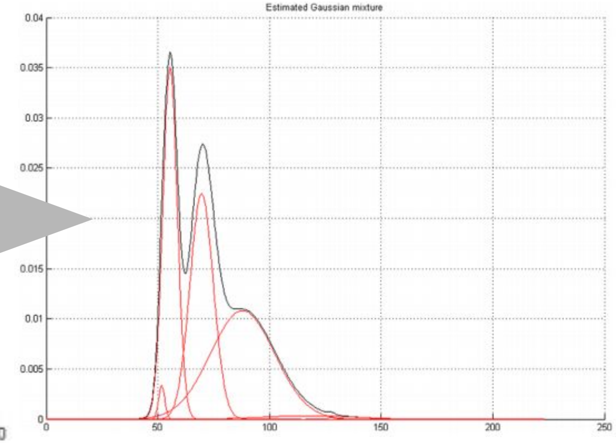
24-hour heart rate time series



histogram



Gaussian mixture



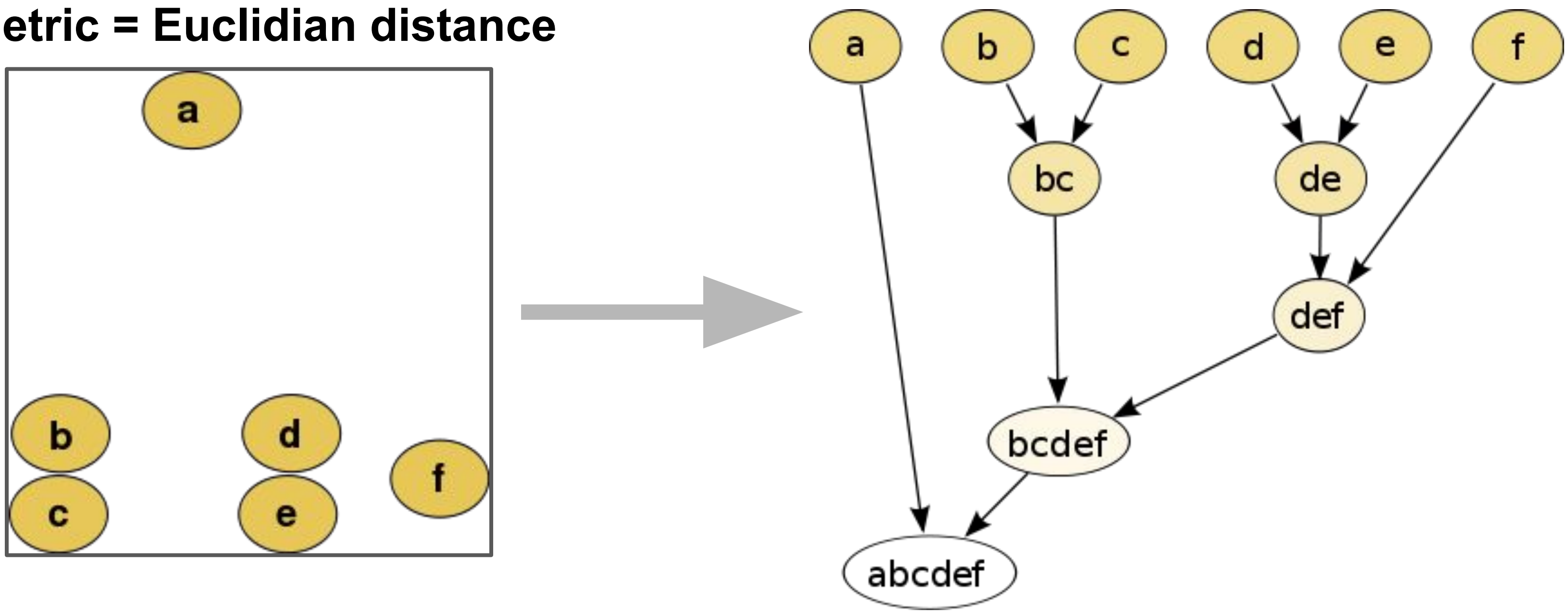
Costa et al. *Plos One* 2012

Clustering

➔ Hierarchical clustering

- **<u>Agglomerate</u>** = 1 cluster for each sample + merging across the hierarchy
- **<u>Divise</u>** = 1 single cluster + divise across the hierarchy

Metric = Euclidian distance



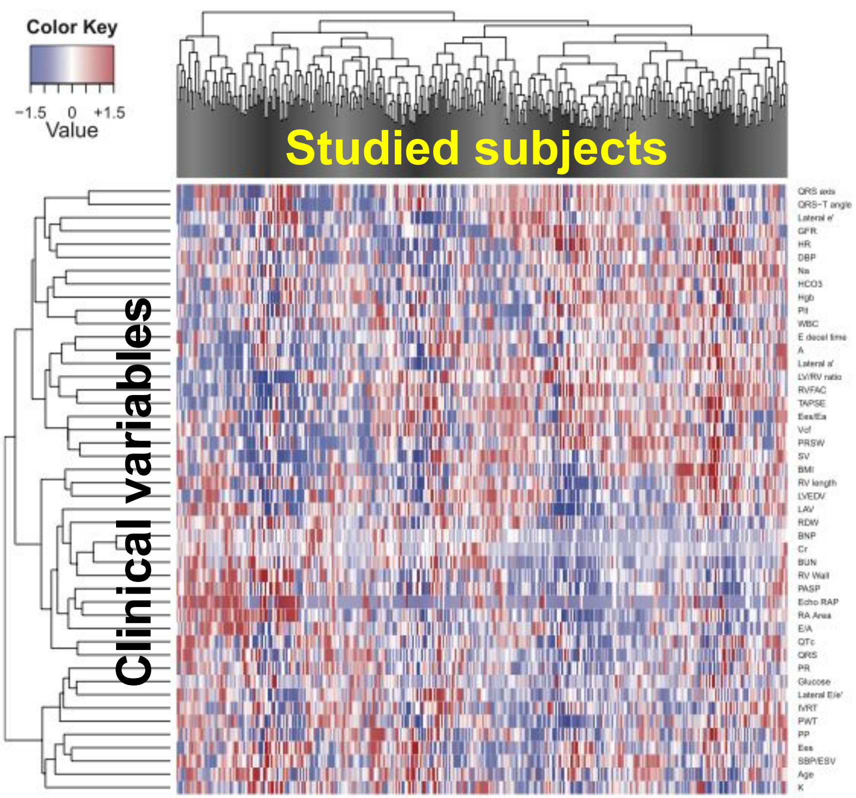
www.wikipedia.org

Clustering

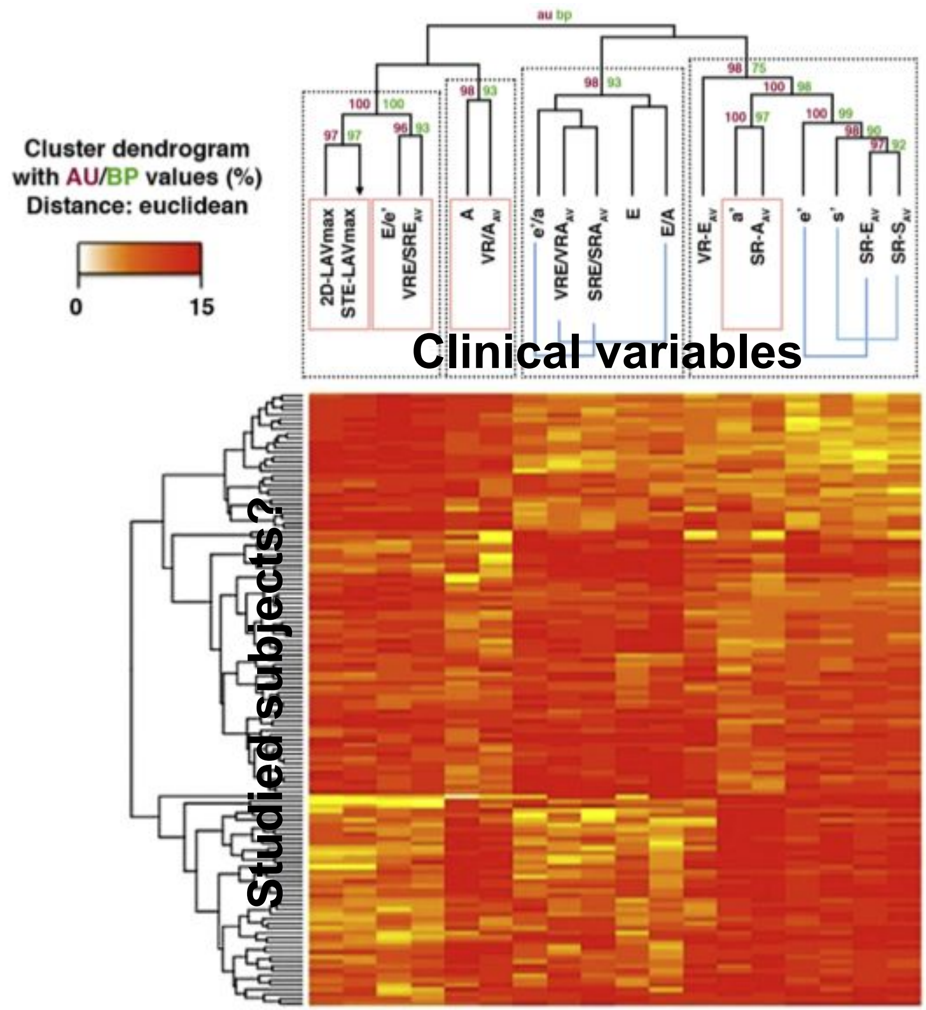
Phenotyping

ex: heart failure reduced / preserved ejection

➔ **Hierarchical clustering**



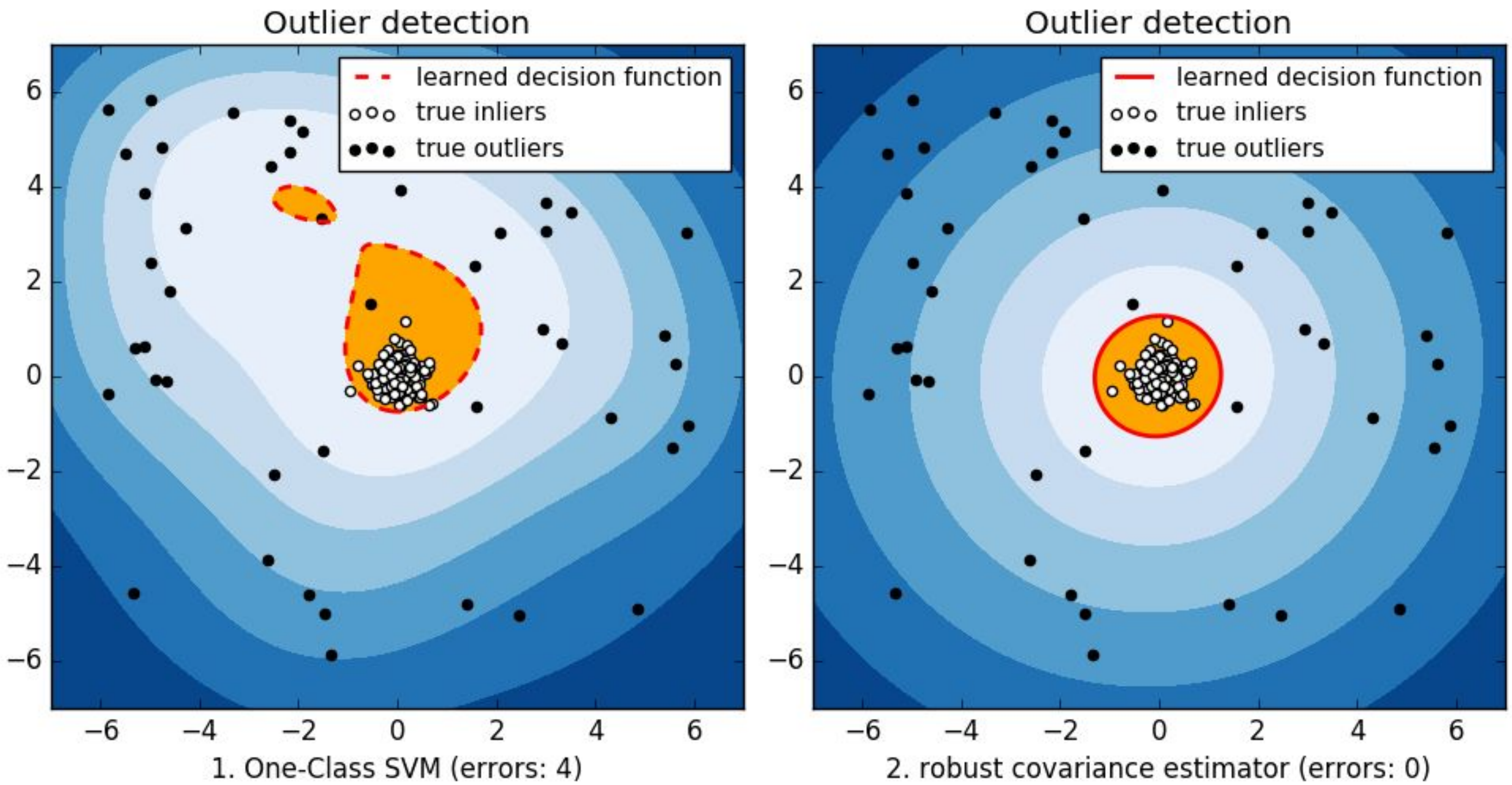
Shah et al. *Circulation* 2015



Omar et al. *JACC Imaging* 2017

3. Unsupervised learning

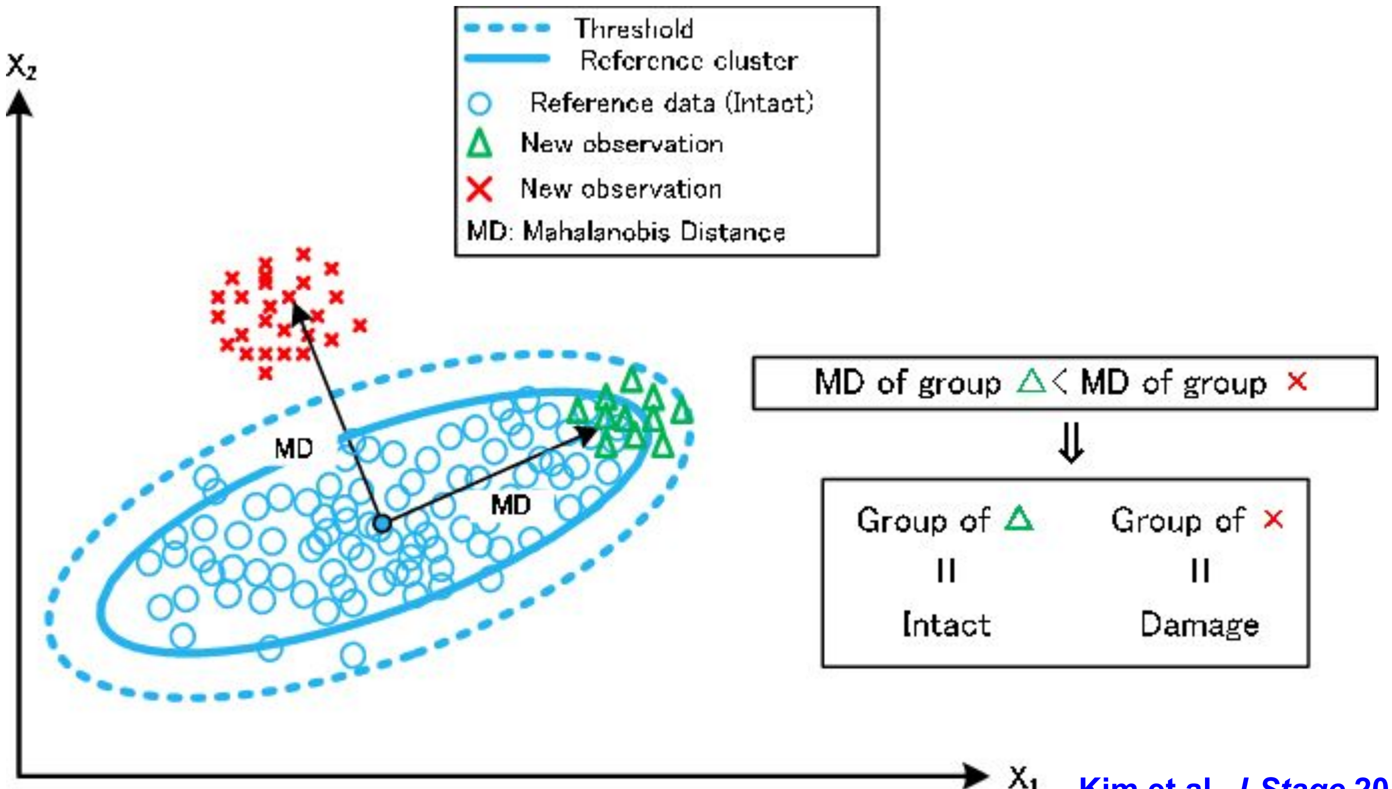
Outliers detection



Outliers detection

➔ **Distribution fit + decision function**

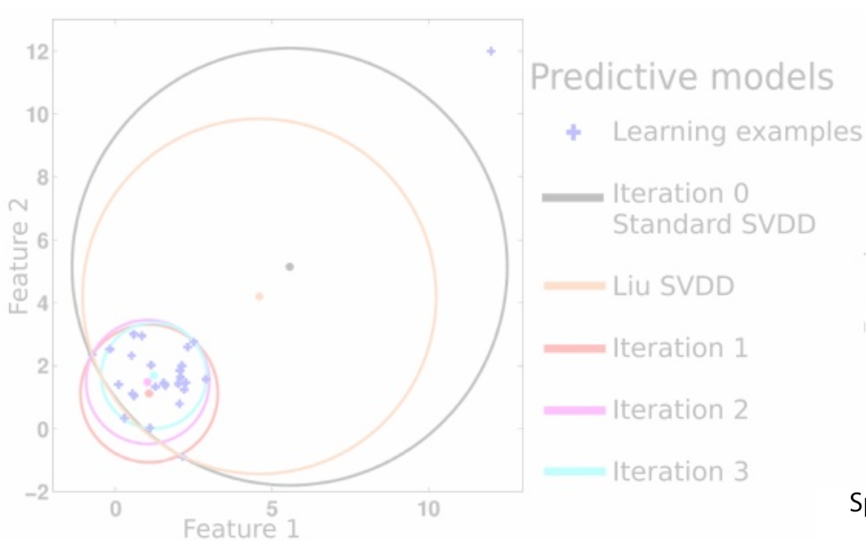
ex: Mahalanobis distance $$D_M(\mathbf{x}) = \sqrt{(\mathbf{x}-\mu_M)^T \Sigma_M^{-1} (\mathbf{x}-\mu_M)}$$



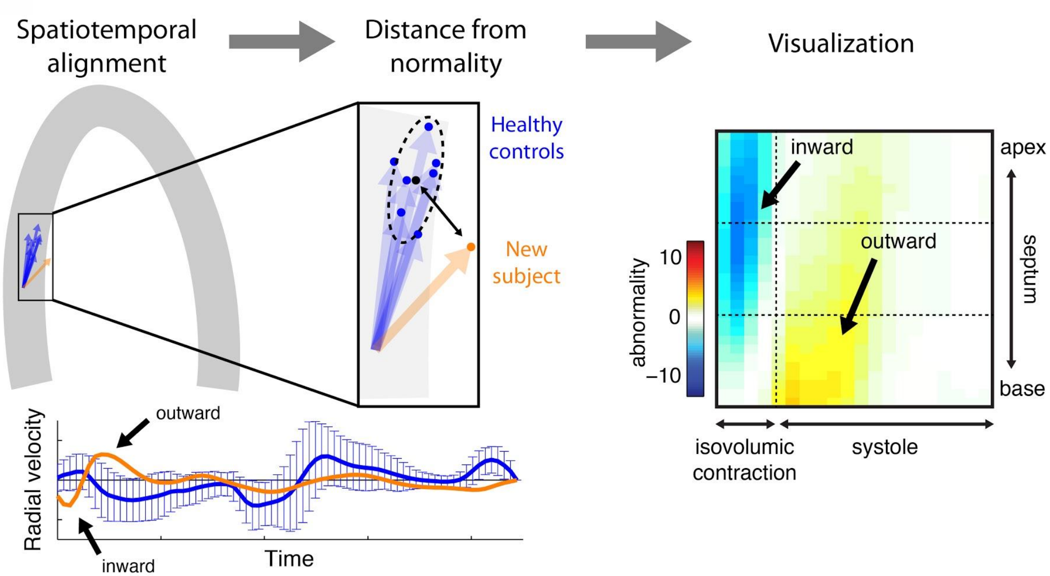
Kim et al. *J-Stage* 2013

Outliers detection

➔ Distribution fit + decision function



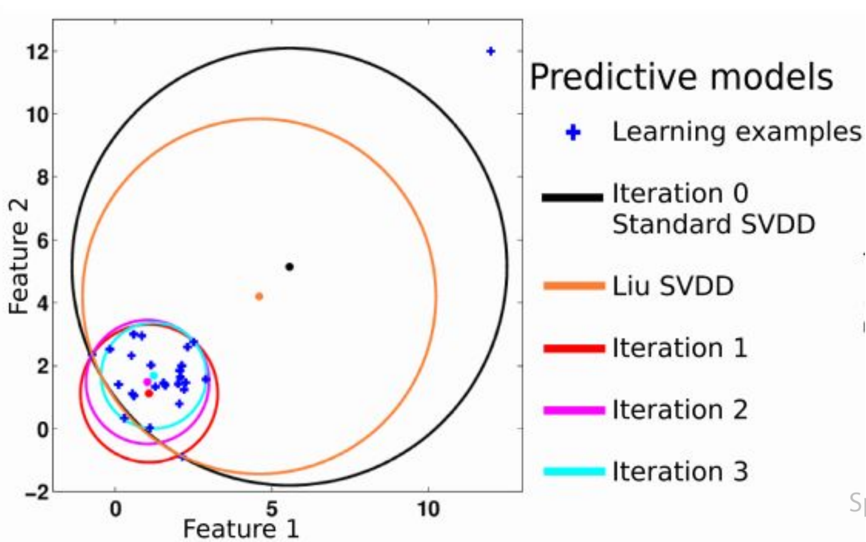
Robustness of outliers detection
ex: detection of epileptogenic foci / MRI
El Azami et al. *ESANN* 2014



Atlas of "normal" motion
ex: abnormalities vs. CRT response
Duchateau et al. *Med Image Anal* 2011

Outliers detection

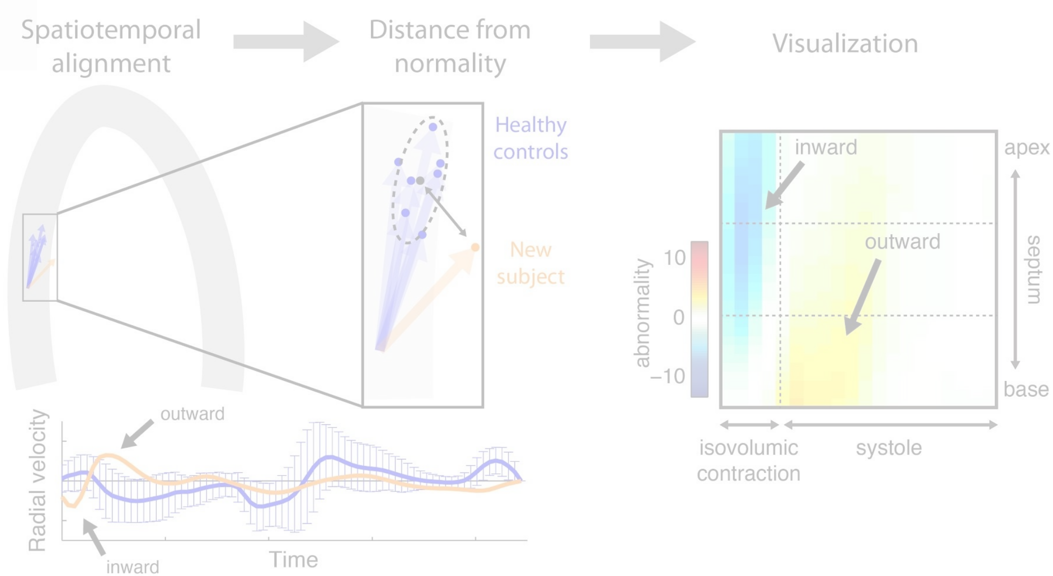
➔ Distribution fit + decision function



Robustness of outliers detection

ex: detection of epileptogenic foci / MRI

El Azami et al. *ESANN* 2014



Atlas of "normal" motion

ex: abnormalities vs. CRT response

Duchateau et al. *Med Image Anal* 2011

3. Unsupervised learning

Which end point?

➔ Subgroups identification / similar trends	Clustering
➔ Detect novelty / unexpected values	Outliers detection
➔ Understand the data space	(low-dimensional) embedding
➔ Statistical distances	Manifold learning
➔ Sampling/generate new cases	Encoding/decoding

3. Unsupervised learning

Which end point?

Key step = data representation

➔ Subgroups identification / similar trends	Clustering
➔ Detect novelty / unexpected values	Outliers detection
➔ Understand the data space	(low-dimensional) embedding
➔ Statistical distances	Manifold learning
➔ Sampling/generate new cases	Encoding/decoding

3. Unsupervised learning

Representation learning

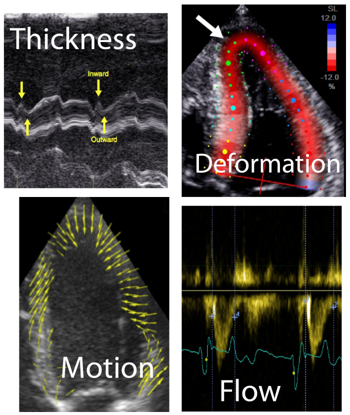
__Idea__ = better represent the data space

- lower dimensional space
- unsupervised

Representation learning

<u>Idea</u> = better represent the data space

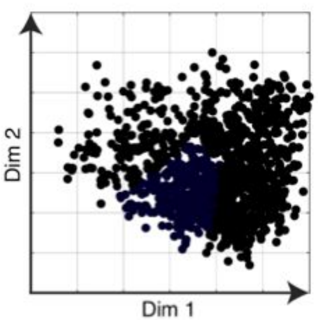
- lower dimensional space
- unsupervised



<u>Inputs</u> =

- Single **high-dimensional** descriptors
- **Multiple** scalars

...or **Multiple** **high-dimensional** descriptors



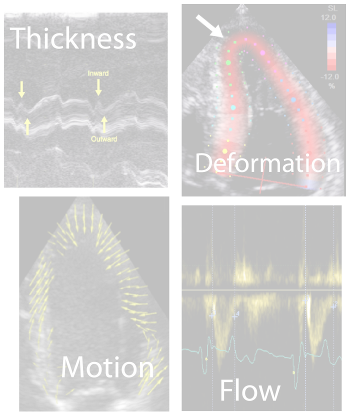
<u>Output</u> =

- **Low-dimensional** representation

Representation learning

<u>Idea</u> = **better represent the data space**

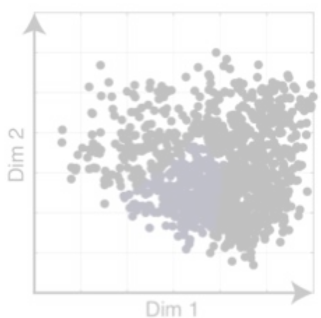
- lower dimensional space
- unsupervised



<u>Inputs</u> =
- Single **high-dimensional** descriptors
- **Multiple** scalars
  ...or **Multiple high-dimensional** descriptors

1) Embedding

3) Reconstruction



<u>Output</u> =
- **Low-dimensional** representation

2) Manifold / latent space

What for? = which space to work on?

Distances in **low dimension**? / Reconstructed cases in **high dimension**?

3. Unsupervised learning

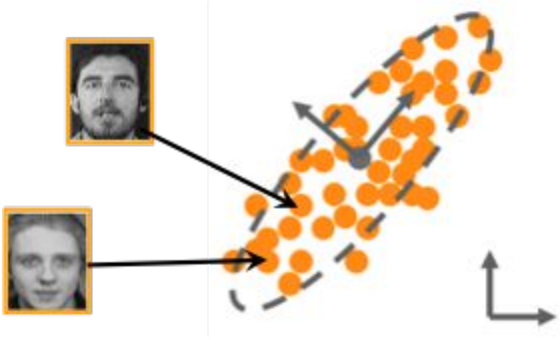
Representation learning

Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Linear ?



ORL database

Representation learning

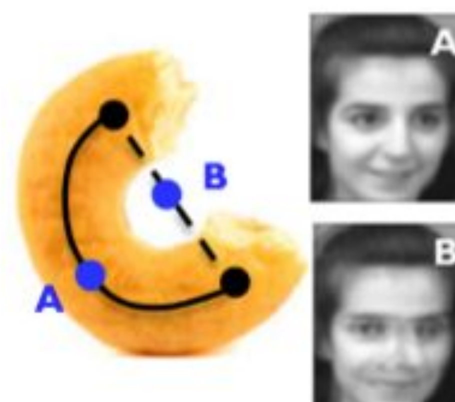
Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Linear ?

Non-linear !!!



ORL database

3. Unsupervised learning

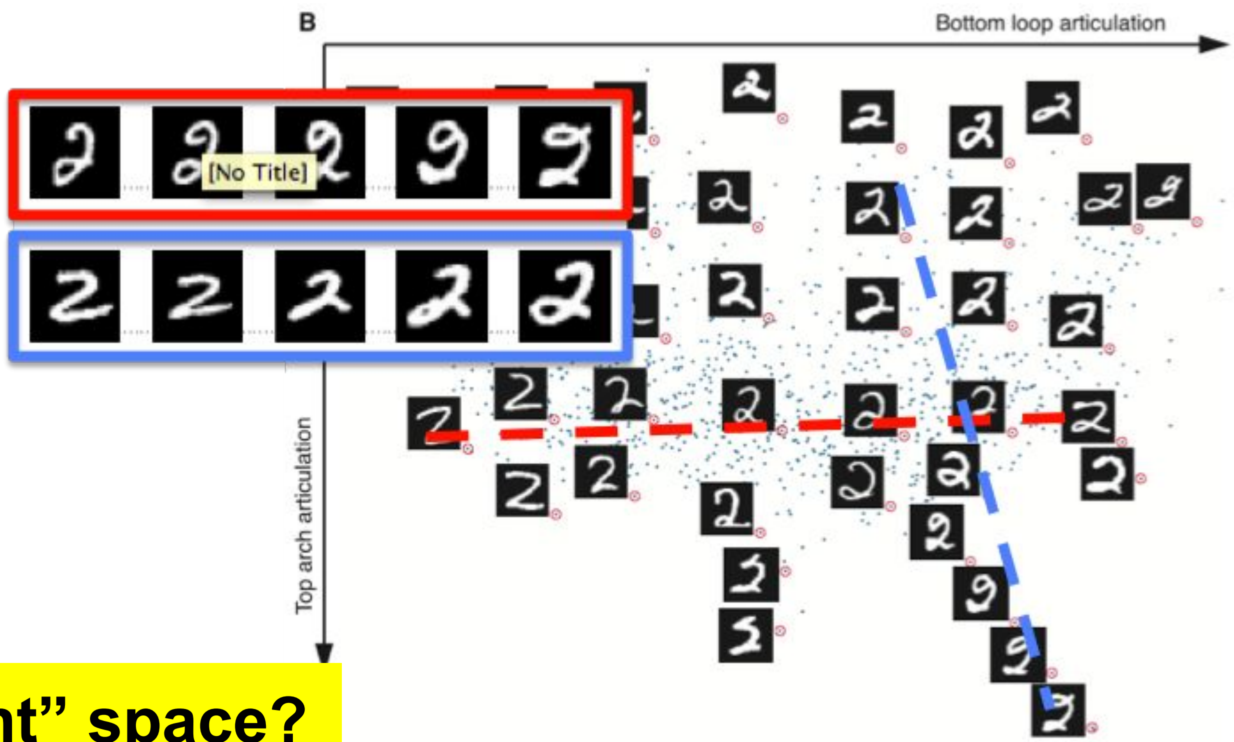
Representation learning

Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Low number of dimensions to encode high dimensional data variations



Exploitable "latent" space?

Tenenbaum et al. *Science* 2000

Representation learning

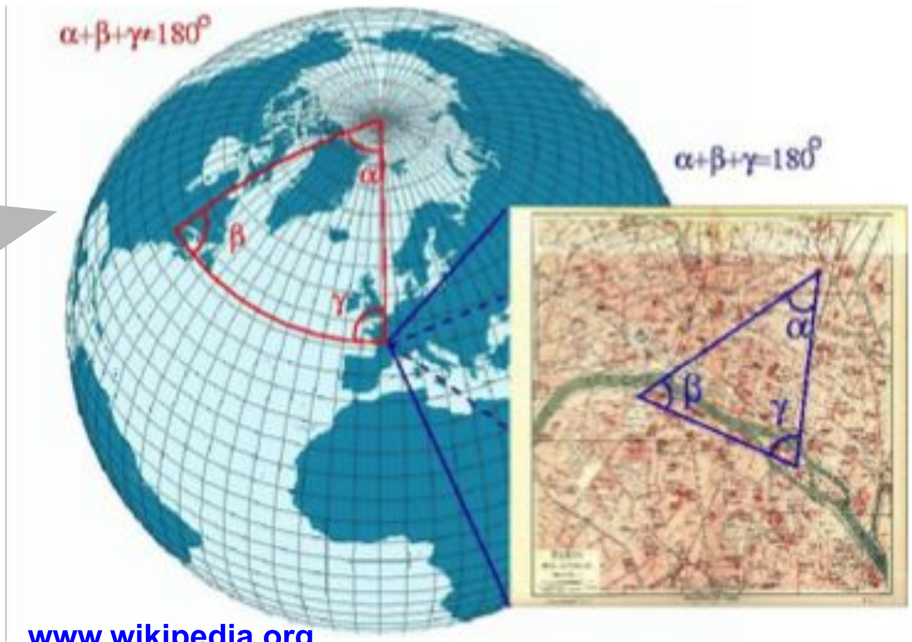
Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Manifold of dimension *N* = topological space that near each sample resembles (is homeomorphic to) a *N*-dimensional Euclidean space

ex:
- **lines and circles (*N*=1)**
- **plane, sphere, surfaces (*N*=2)**
- **brain images, cardiac shapes (*N*=?)**



www.wikipedia.org

3. Unsupervised learning

Representation learning

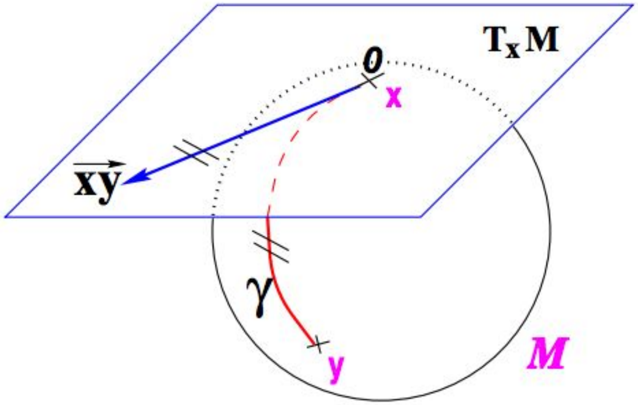
Idea = better represent the data space
- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Manifold of dimension *N*

◆ In some cases = **known structure**

Vector space	Riemannian manifold
$\overrightarrow{xy} = y - x$	$\overrightarrow{xy} = \log_x(y)$
$y = x + \overrightarrow{xy}$	$y = \exp_x(\overrightarrow{xy})$



log-exponential mapping
Pennec et al. *Int J Comput Vis* 2006

3. Unsupervised learning

Representation learning

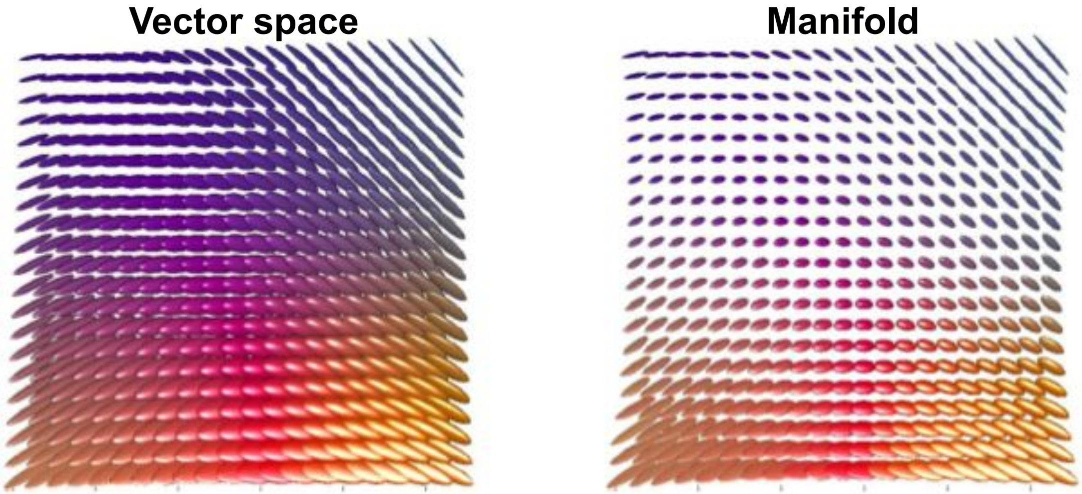
Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ “Structure” of the data space?

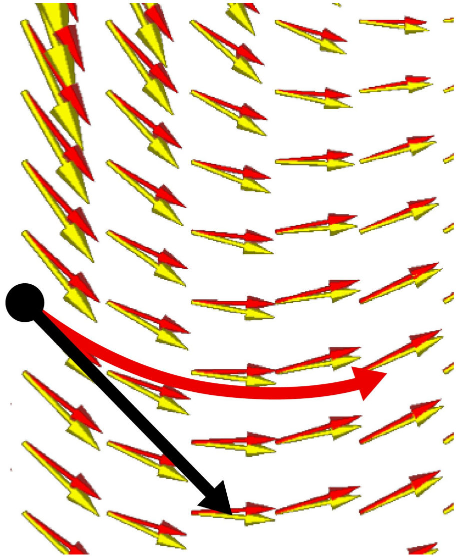
Manifold of dimension *N*

◆ In some cases = **known structure**



ex: Interpolation of tensors (diffusion, strain)

Pennec et al. *Int J Comput Vis* 2006



Diffeomorphic transformations (registration)

Arsigny et al. *MICCAI* 2006

3. Unsupervised learning

Representation learning

Idea = better represent the data space

- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Manifold of dimension *N*

- In some cases = **known structure**
- **Otherwise = learn it from data !**

3. Unsupervised learning

Representation learning

Idea = better represent the data space

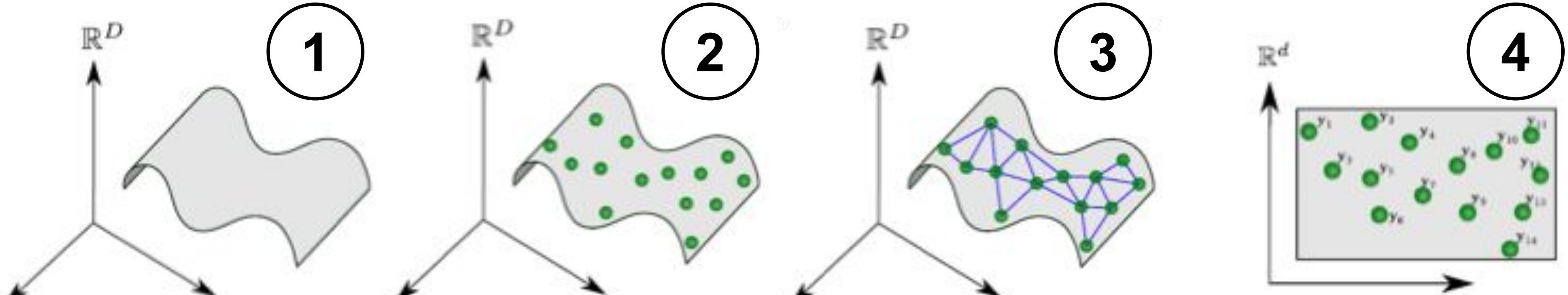
- lower dimensional space
- unsupervised

➔ "Structure" of the data space?

Manifold of dimension *N*

- In some cases = **known structure**
- **Otherwise = learn it from data !**

1. Assumption = data **lies on / close to a manifold**
2. **Few samples** (on the manifold) available
3. **Neighborhood graph** = approximation of the manifold
4. Dimensionality reduction = **spectral decomposition** of…?

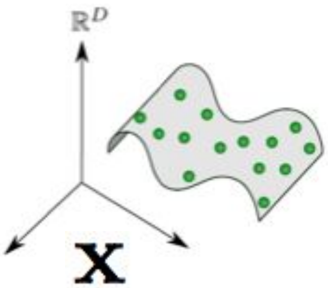


Atasoy & Mateus *MICCAI tutorial* 2011

Representation learning

Idea = better represent the data space

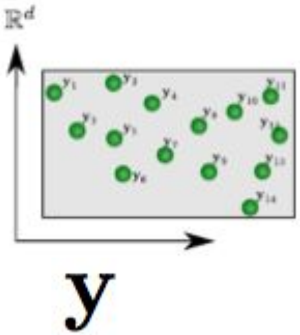
- lower dimensional space
- unsupervised



Inputs =
- Single **high-dimensional** descriptors
- **Multiple** scalars

...or **Multiple high-dimensional** descriptors

1) Embedding



Output =
- **Low-dimensional** representation

2) Manifold / latent space

$$D \gg d \qquad f : \mathbf{x} \in \mathbb{R}^D \mapsto \mathbf{y} \in \mathbb{R}^d$$

3. Unsupervised learning

(low-dimensional) embedding: linear

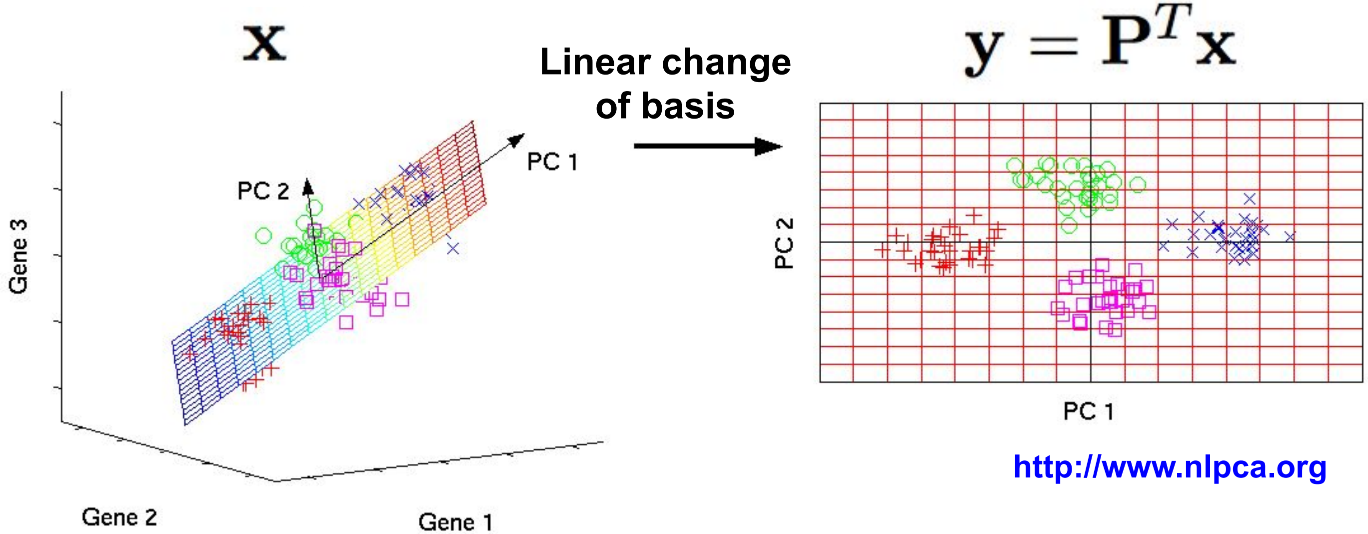
3. Unsupervised learning

(low-dimensional) embedding: linear

➔ **PCA = Principal Component Analysis**

<u>Idea</u> = principal directions of variance ➔ diagonalize the covariance matrix

$$\Sigma = \mathbf{P}\mathbf{D}\mathbf{P}^T$$

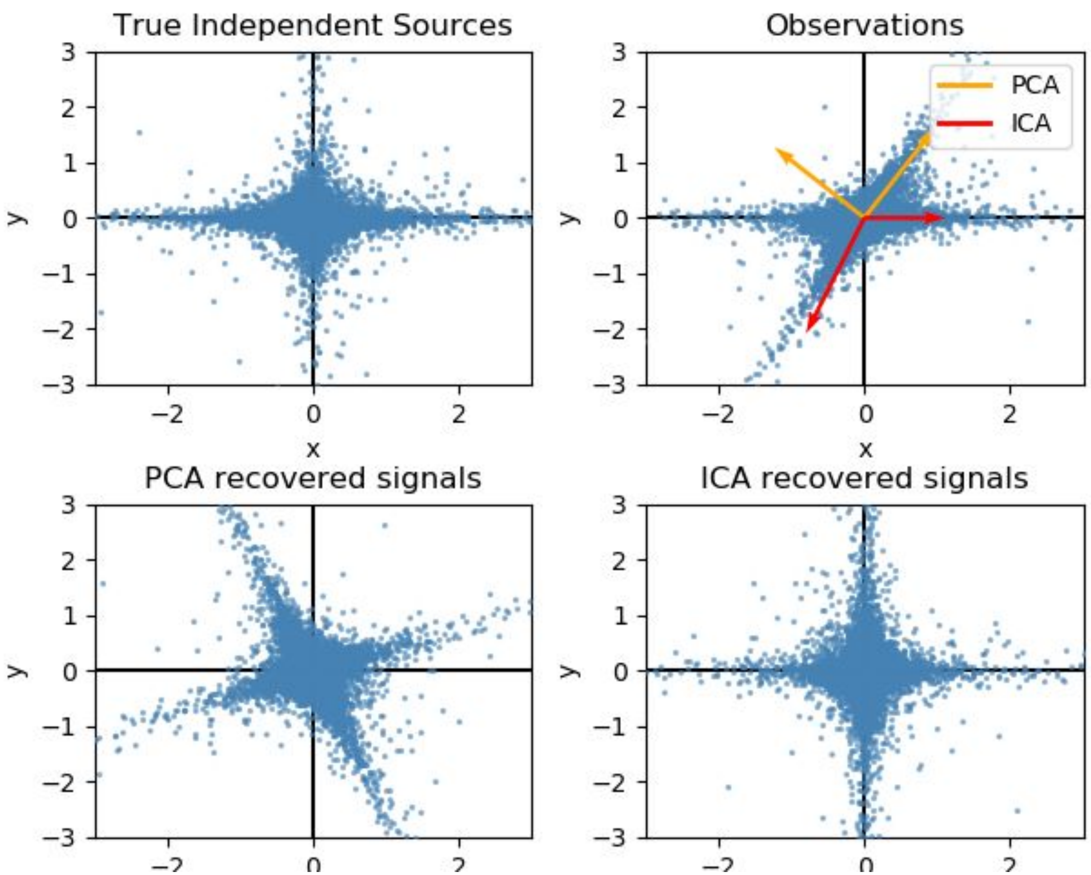


http://www.nlpca.org

(low-dimensional) embedding: linear

➔ ICA = Independent Component Analysis

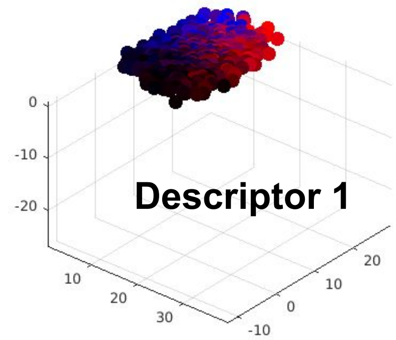
Idea = Independent, non-Gaussian variables

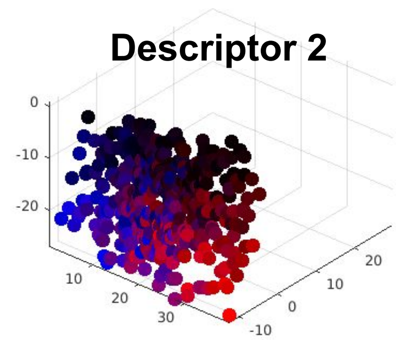


3. Unsupervised learning

(low-dimensional) embedding: linear

To go further...		Nb descriptors	Maximizes...	
Principal Component Analysis	**PCA**	1	Variance	
Partial Least Squares	**PLS**	2+	Covariance	Wold et al. *Chemo* 1984
Canonical Correlation Analysis	**CCA**	2+	Correlation	Hotelling *Biometr* 1936

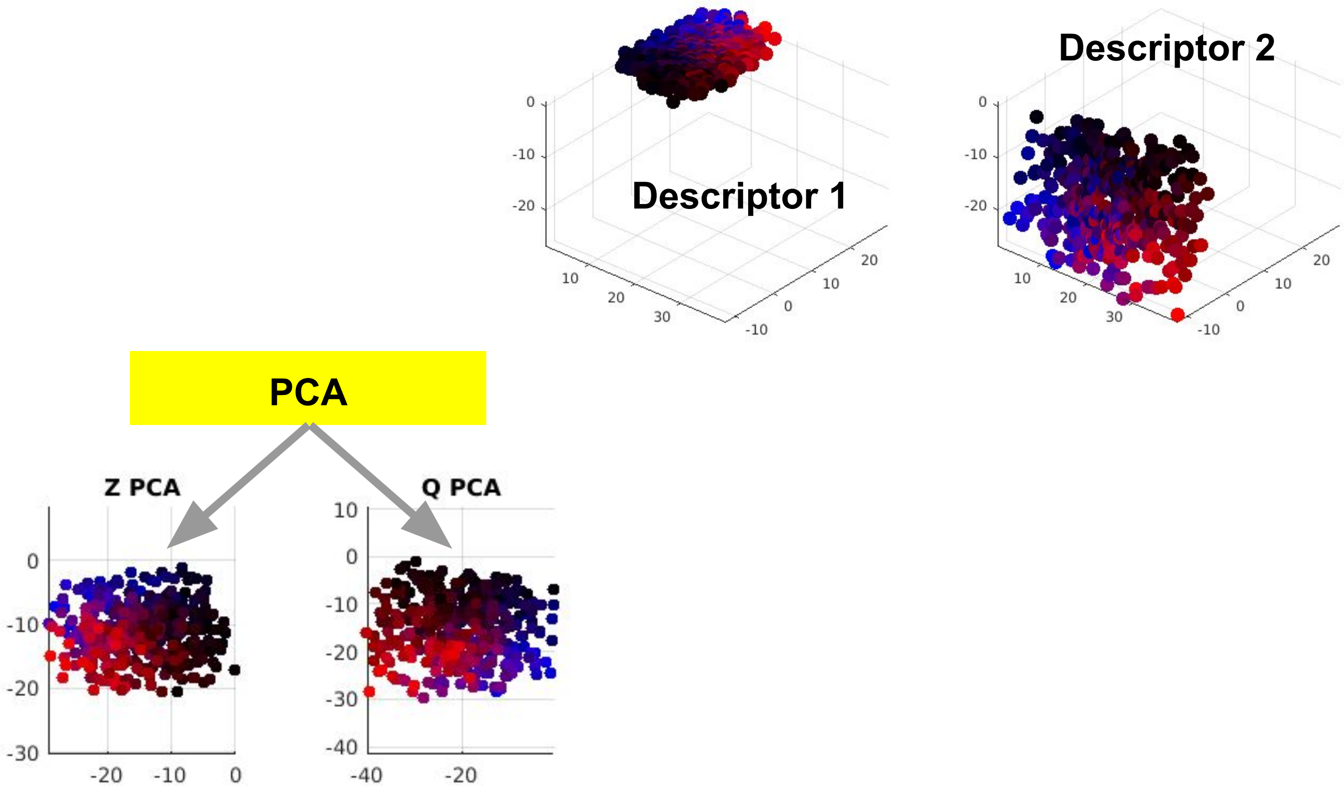




3. Unsupervised learning

(low-dimensional) embedding: linear

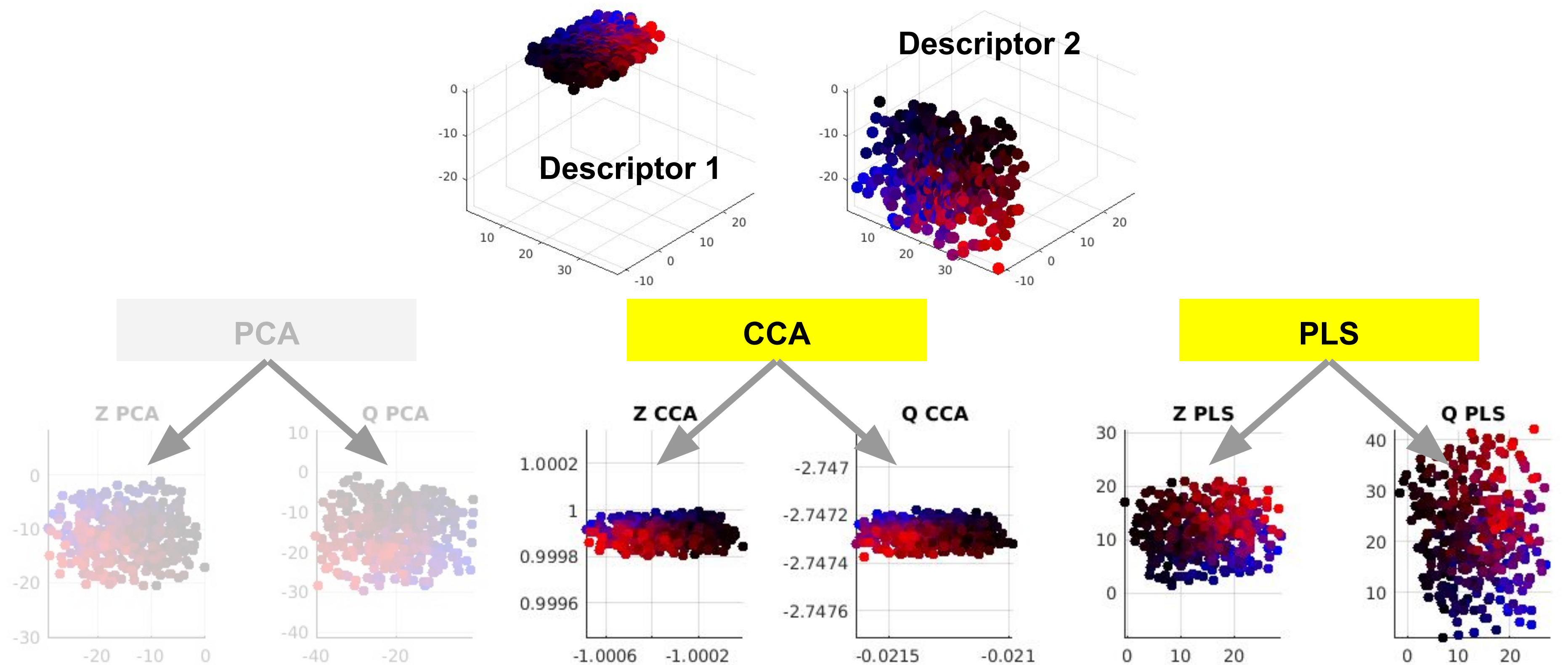
To go further...		Nb descriptors	Maximizes...	
Principal Component Analysis	**PCA**	1	Variance	
Partial Least Squares	**PLS**	2+	Covariance	Wold et al. *Chemo* 1984
Canonical Correlation Analysis	**CCA**	2+	Correlation	Hotelling *Biometr* 1936



3. Unsupervised learning

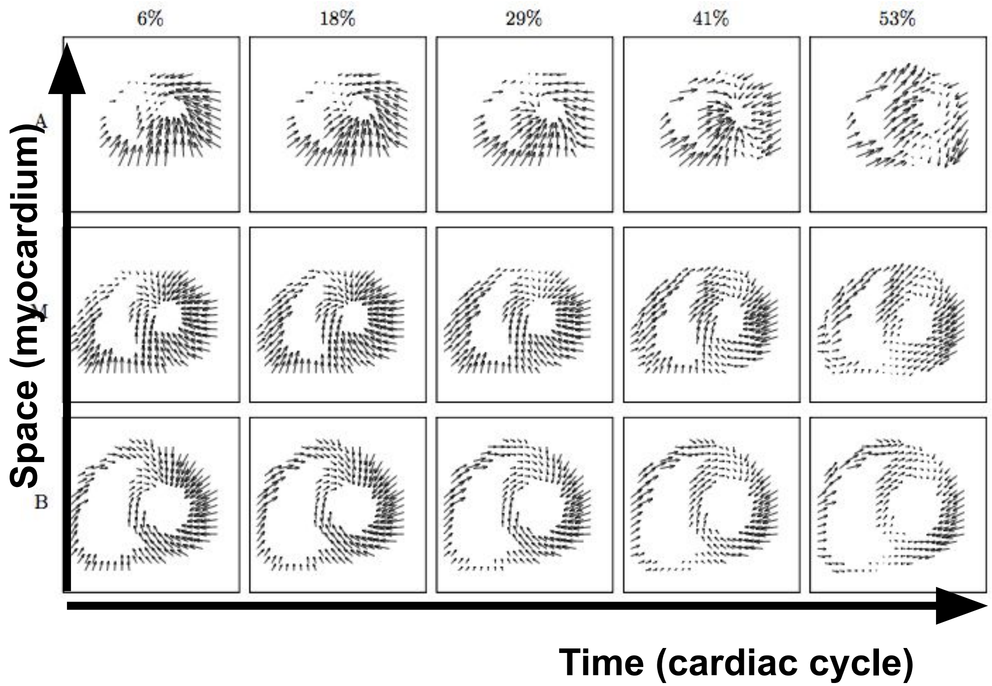
(low-dimensional) embedding: linear

To go further...		Nb descriptors	Maximizes...	
Principal Component Analysis	**PCA**	1	Variance	
Partial Least Squares	**PLS**	2+	Covariance	Wold et al. *Chemo* 1984
Canonical Correlation Analysis	**CCA**	2+	Correlation	Hotelling *Biometr* 1936



3. Unsupervised learning

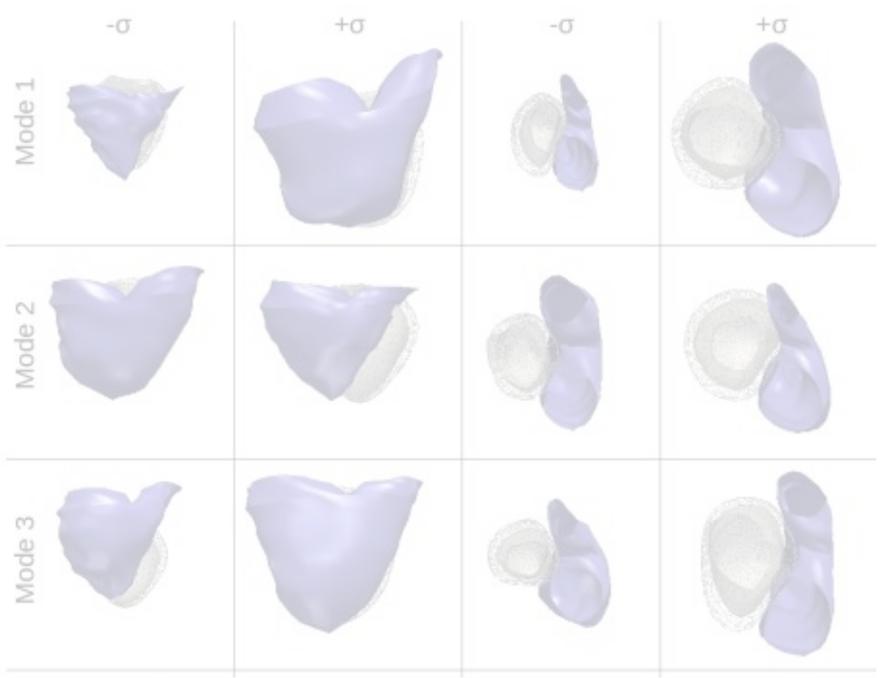
(low-dimensional) embedding: linear



Atlas of "normal" motion

(PCA on myocardial velocities)

Rougon et al. *SPIE* 2004



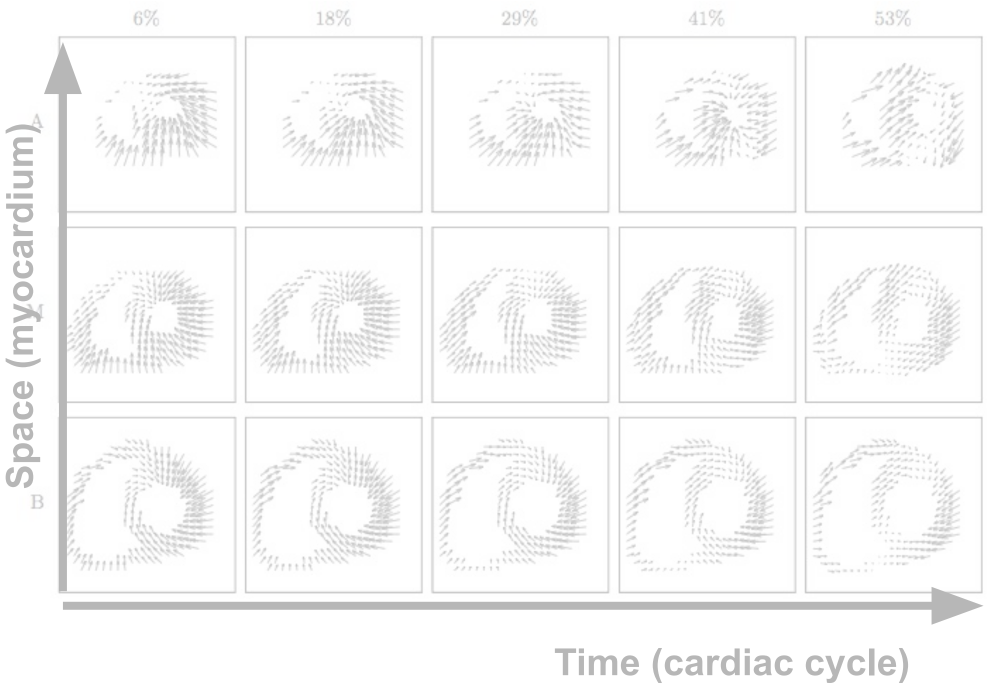
Variations in pathological shapes

(PCA on velocity fields from registration)

McLeod et al. *MCBB* 2013

3. Unsupervised learning

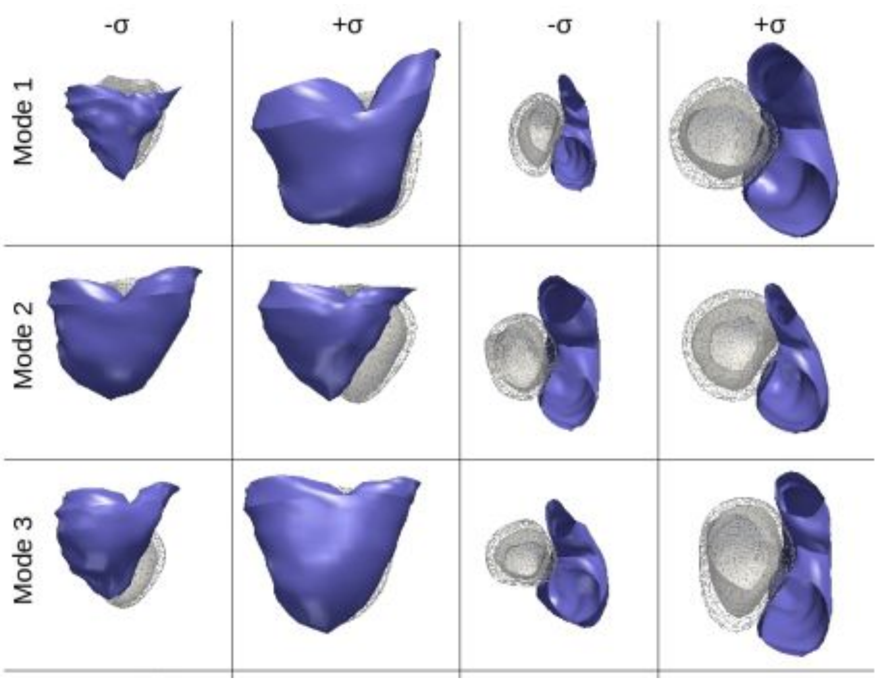
(low-dimensional) embedding: linear



Atlas of "normal" motion

(PCA on myocardial velocities)

Rougon et al. *SPIE* 2004



Variations in pathological shapes

(PCA on velocity fields from registration)

McLeod et al. *MCBB* 2013

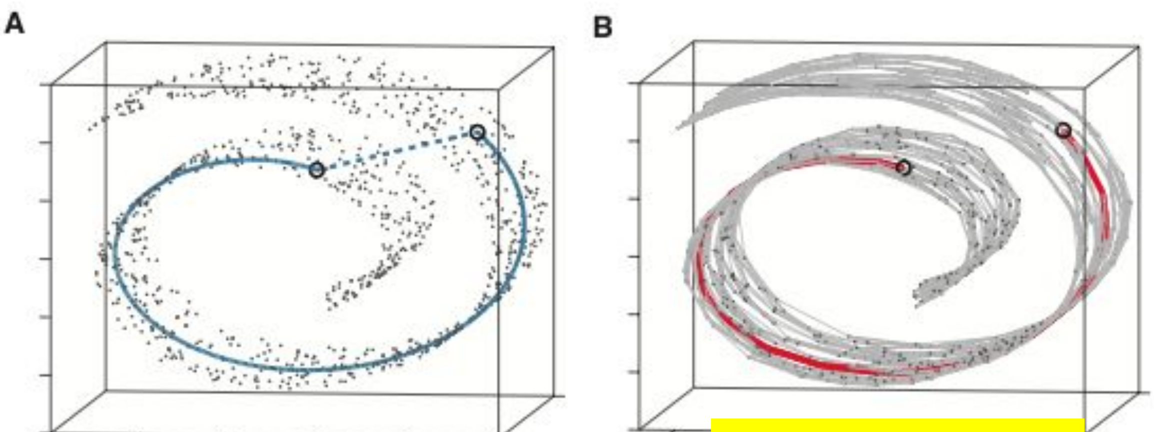
3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

(low-dimensional) embedding: ~~linear~~ non-linear

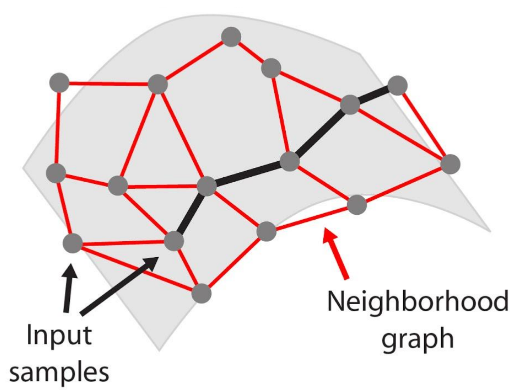
➔ **Isomap** **Tenenbaum et al. *Science* 2002**

<u>Idea</u> = approximate **geodesic distances** / shortest path along the graph



Tenenbaum et al. *Science* 2002

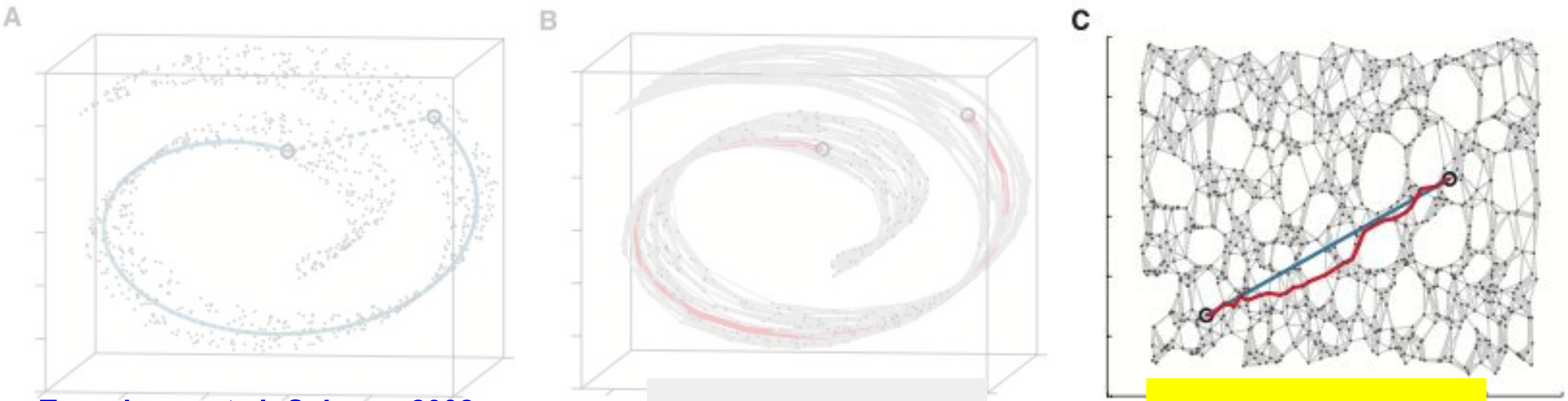
Input space



(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Isomap** — **Tenenbaum et al. *Science* 2002**

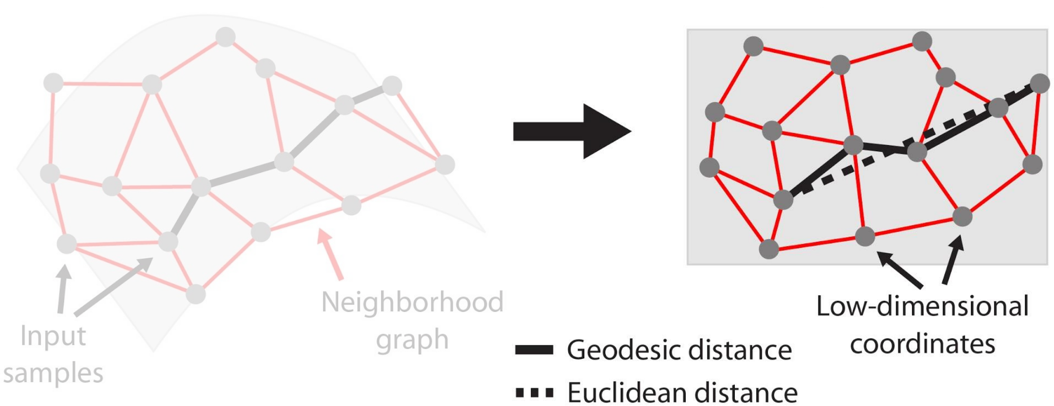
<u>Idea</u> = approximate **geodesic distances** / shortest path along the graph



Tenenbaum et al. *Science* 2002

Input space

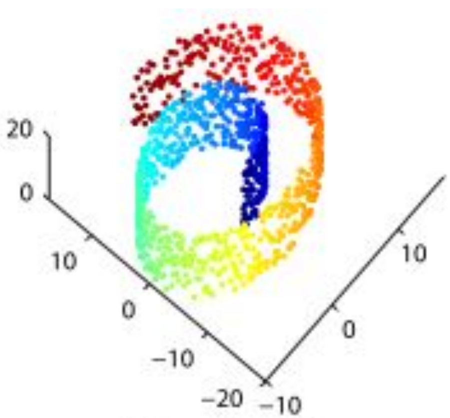
Output space



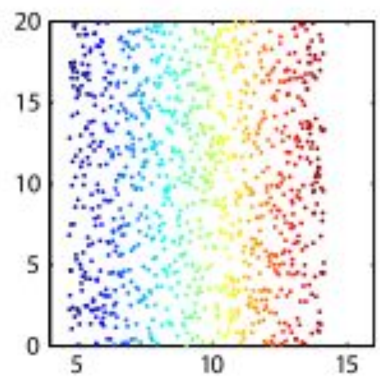
(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Isomap** Tenenbaum et al. *Science* 2002

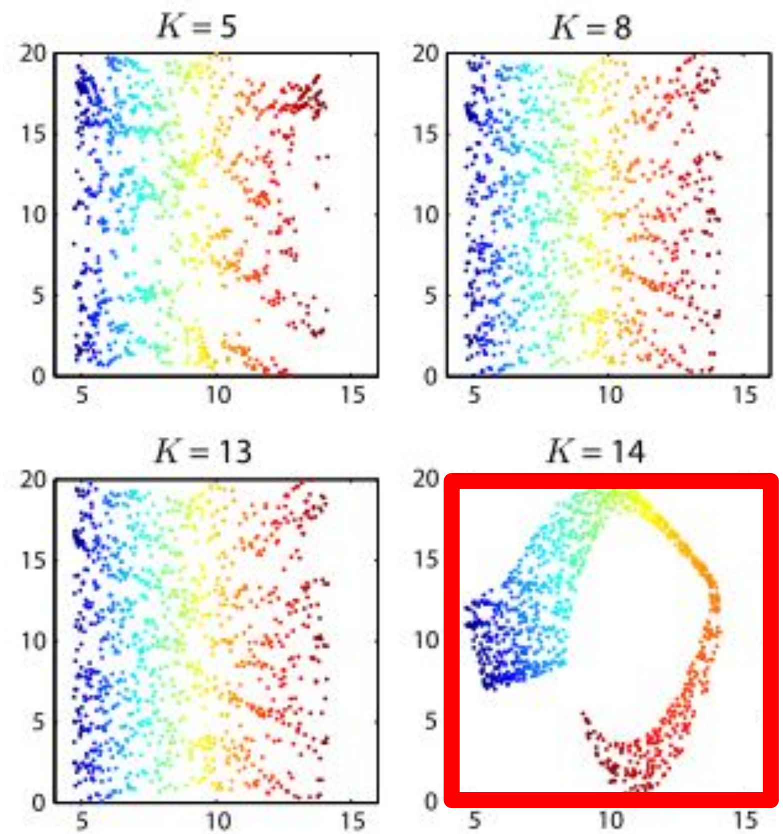
Parameter = number of neighbors *K* to build the graph



Input space



Ground truth embedding



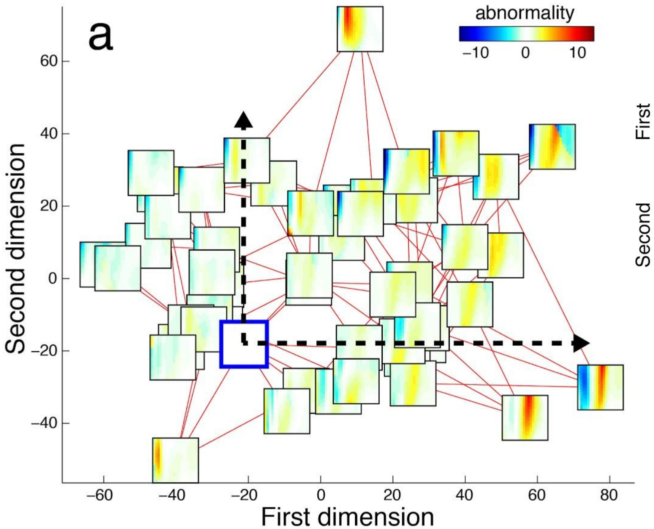
Estimated embeddings

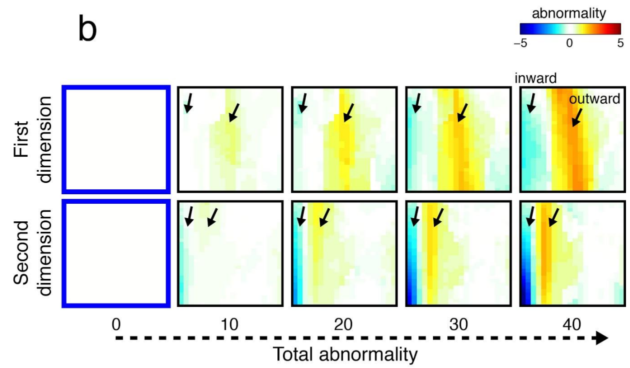
Duchateau et al. *Med Image Anal* 2012

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

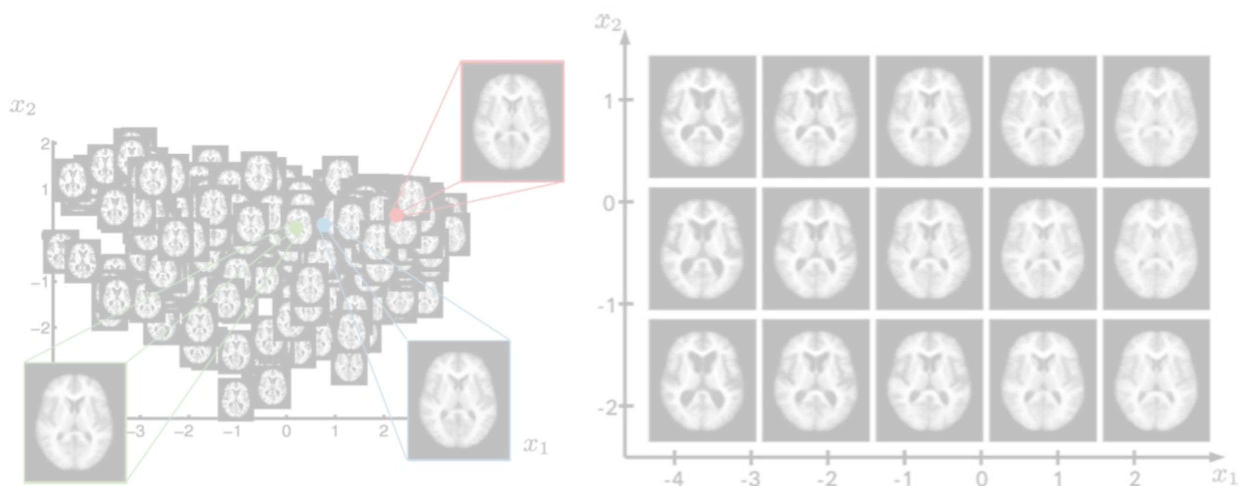
➔ **Isomap**





ex: disease evolution
(cardiac velocity patterns)

Duchateau et al. *Med Image Anal* 2012

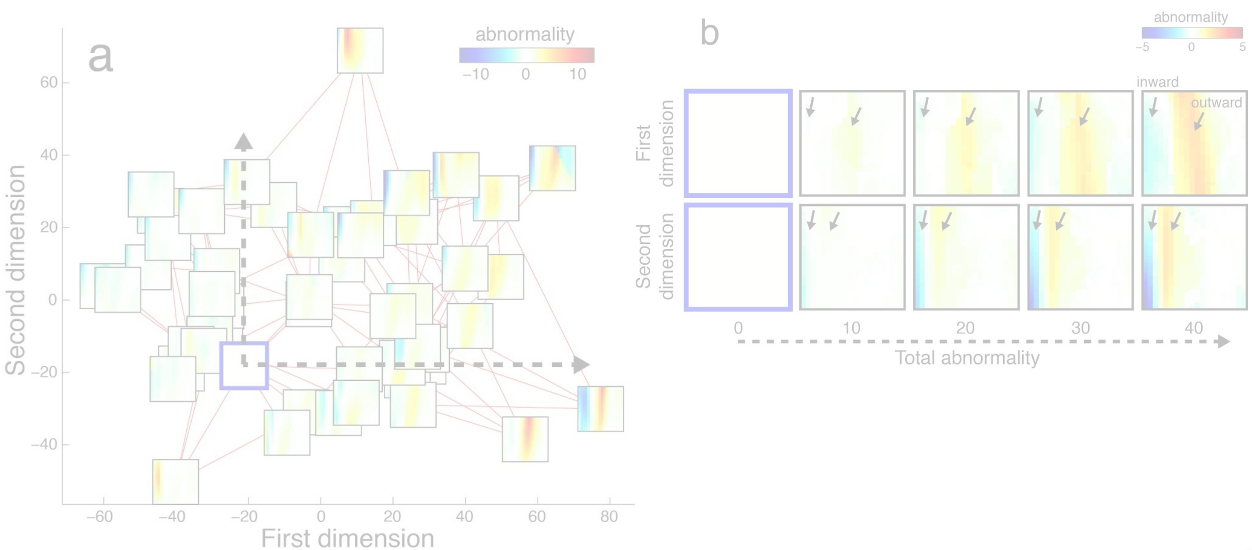


ex: preprocessing for regression (brain images)

Gerber et al. *Med Image Anal* 2010

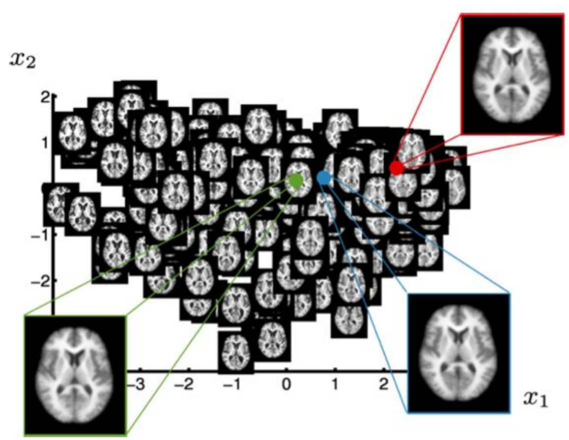
(low-dimensional) embedding: ~~linear~~ non-linear

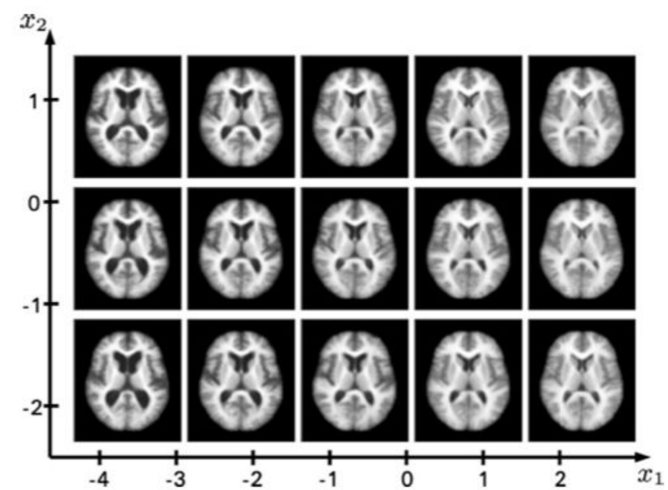
➔ **Isomap**



ex: disease evolution
(cardiac velocity patterns)

Duchateau et al. *Med Image Anal* 2012





ex: preprocessing for regression (brain images)

Gerber et al. *Med Image Anal* 2010

(low-dimensional) embedding: ~~linear~~ non-linear

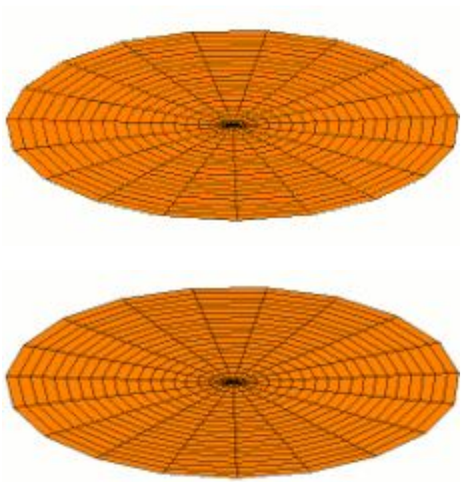
➔ **Laplacian eigenmaps** — **Belkin & Niyogi *Neur Comput* 2003**

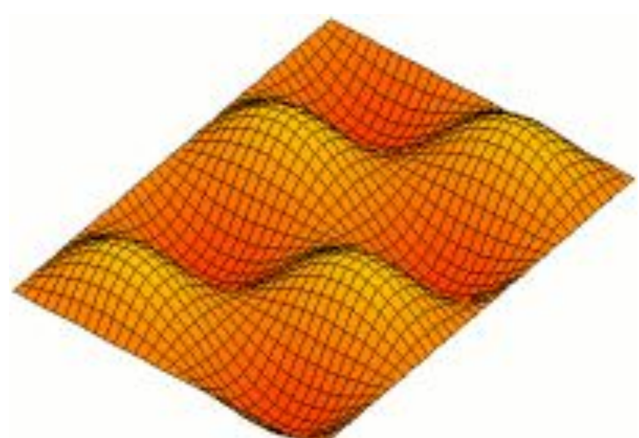
<u>**Idea**</u> = diagonalize the **graph Laplacian**

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 = \underset{\mathbf{Y}}{\operatorname{argmin}} \mathbf{Y}^T \mathbf{L} \mathbf{Y}$$

Constraint: $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = 1$

Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$





www.wikipedia.org

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Laplacian eigenmaps** **Belkin & Niyogi *Neur Comput* 2003**

Idea = diagonalize the **graph Laplacian**

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{i,j} w_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2 = \underset{\mathbf{Y}}{\operatorname{argmin}} \mathbf{Y}^T \mathbf{L} \mathbf{Y}$$

Constraint: $\mathbf{Y}^T \mathbf{D} \mathbf{Y} = 1$

Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$

Parameter = kernel bandwidth

$$w_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$$

- Close inputs ➔ $w_{ij} \approx 1$ ➔ close outputs
- Far inputs ➔ $w_{ij} \approx 0$ ➔ minor influence

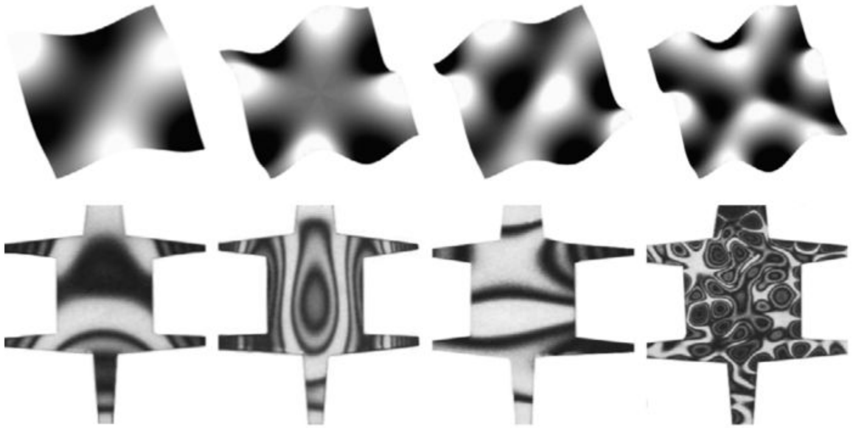
3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Laplacian eigenmaps**

Imaged vibrations (interferometry)

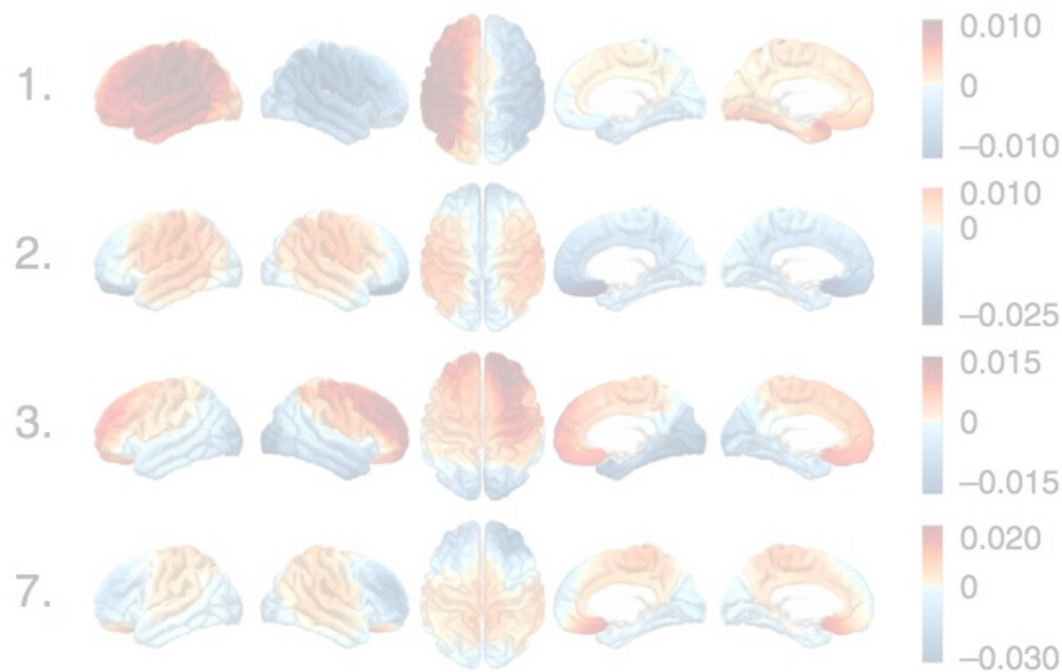
Xu et al. *Appl Optics* 1983



← Rectangular metal plates

← Shaped metal plates

Learnt embedding:
ex: connectome harmonics



Atasoy et al. *Nat Comm* 2016

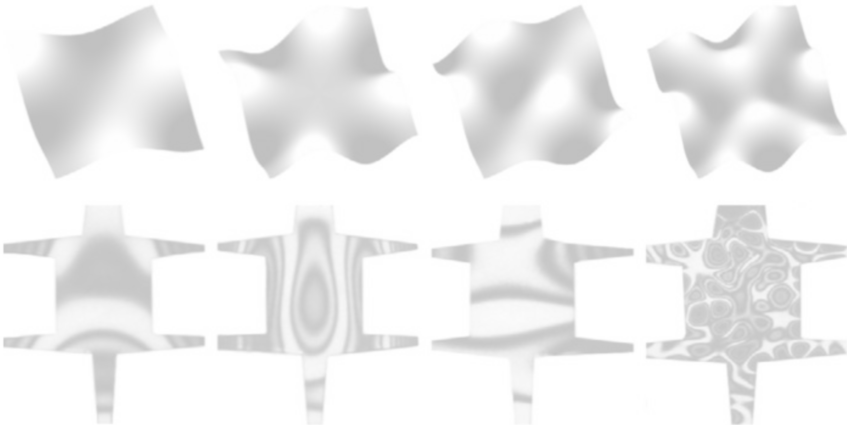
3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Laplacian eigenmaps**

Imaged vibrations (interferometry)

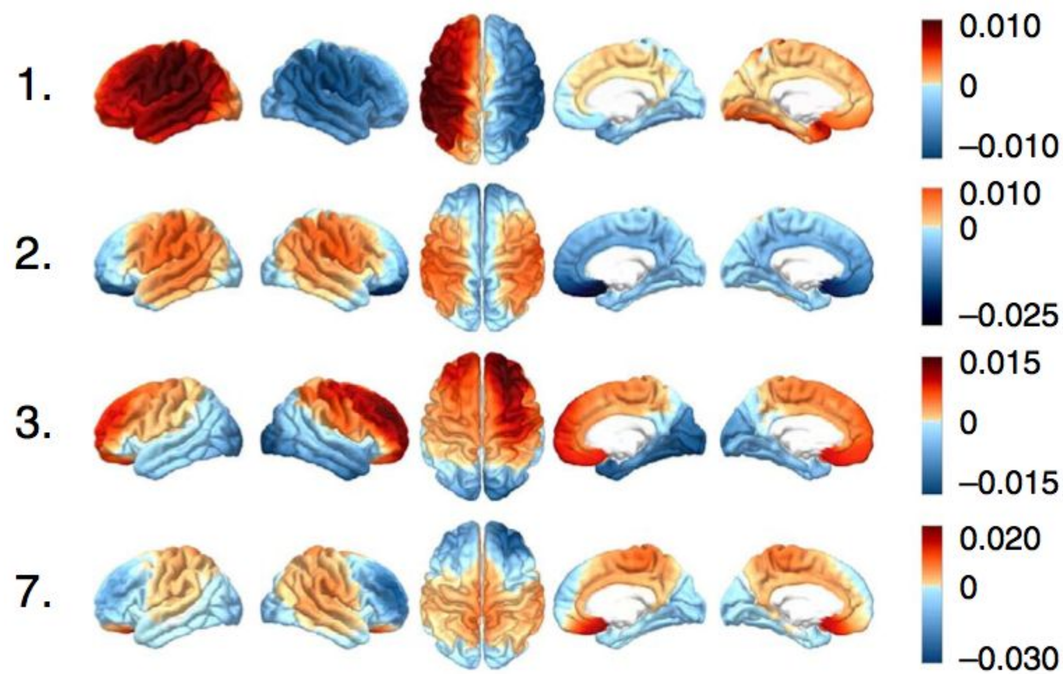
Xu et al. *Appl Optics* 1983



← Rectangular metal plates

← Shaped metal plates

Learnt embedding:
ex: connectome harmonics



Atasoy et al. *Nat Comm* 2016

3. Unsupervised learning

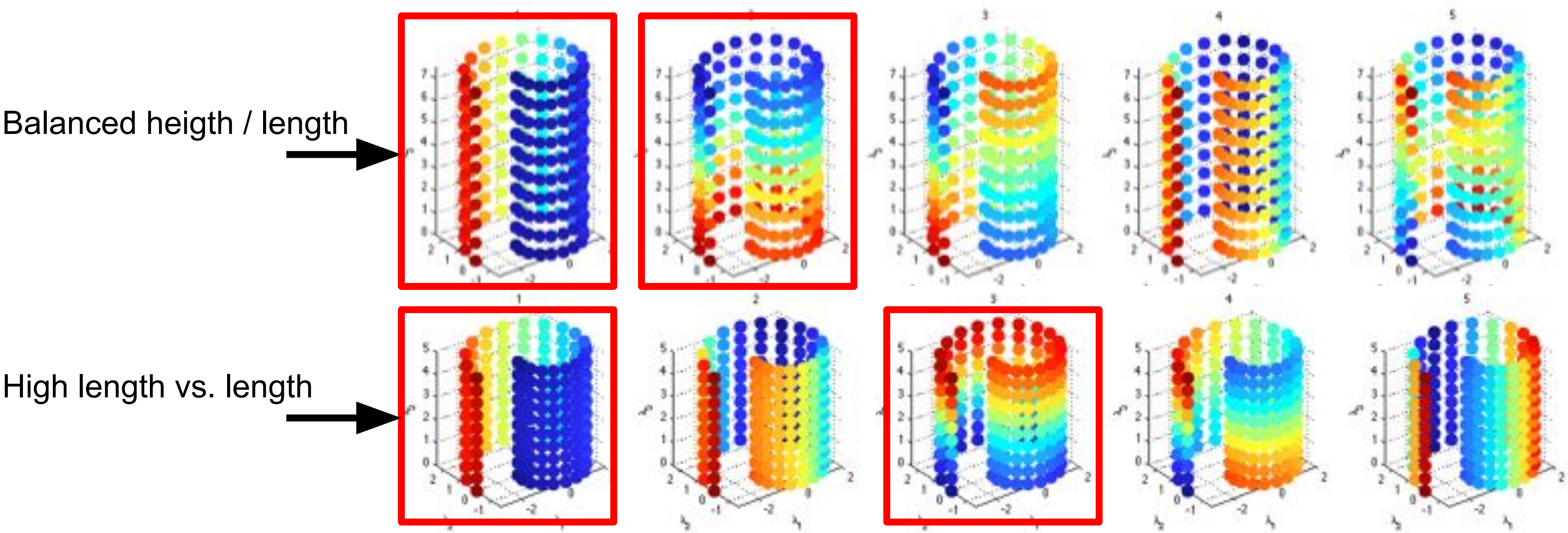
(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Laplacian eigenmaps**

Beware ! meaningful variations **not always ordered** as dimensions (1,2,3,…)

➔ Careful interpretation vs. the **spread of the data space (Nadler et al. 2008)**

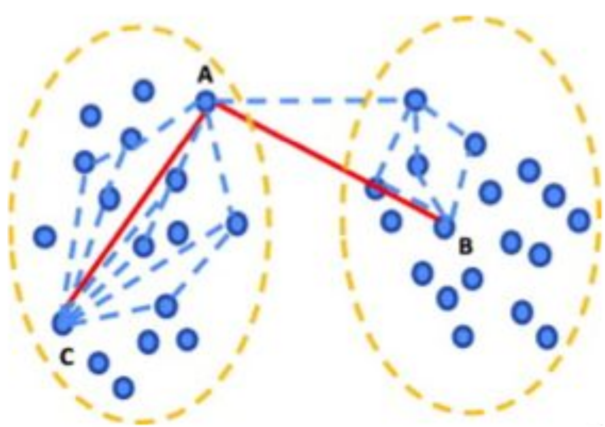
ex: Spiral with varying height vs. length



(low-dimensional) embedding: ~~linear~~ non-linear

To go further...	Works on...	
Laplacian eigenmaps	Graph Laplacian L = D-W	Belkin & Niyogi *Neur Comput* 2003
Diffusion maps	Normalized Laplacian P	Coifmain & Lafon *Appl Comput Harm* 2006

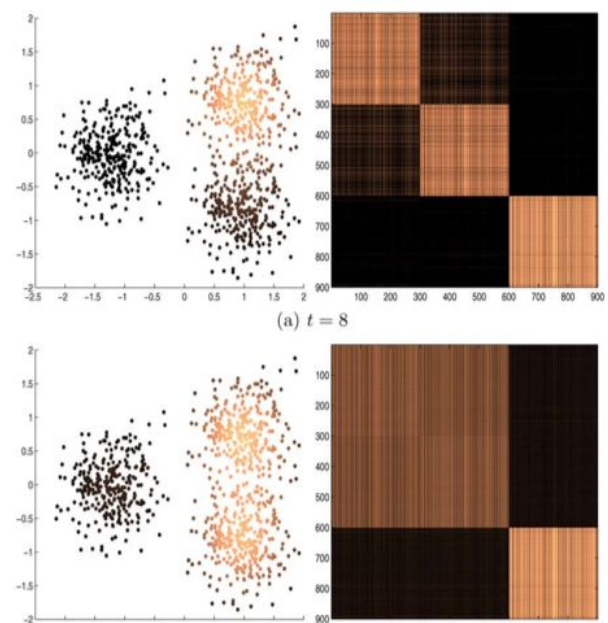
P_{ij} Probability of moving from sample *i* to sample *j* in *t* steps



Liu et al. *CVIU* 2012

t=8

t=64



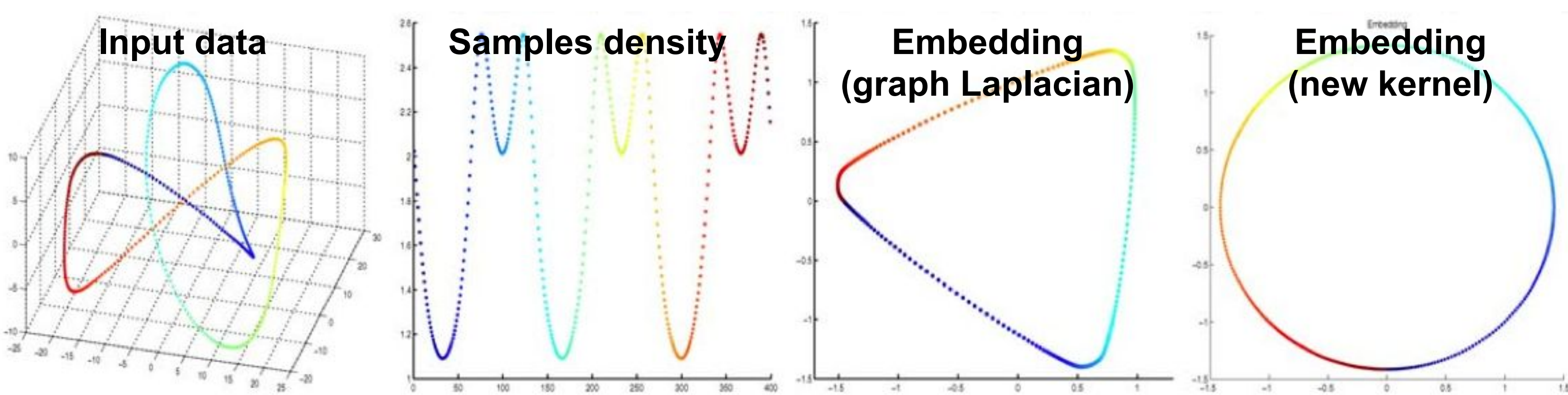
Coifman & Lafon *Appl Comput Harm* 2006

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

To go further...	**Works on...**	
Laplacian eigenmaps	Graph Laplacian L = D-W	Belkin & Niyogi *Neur Comput* 2003
Diffusion maps	Normalized Laplacian P	Coifmain & Lafon *Appl Comput Harm* 2006

Why? Robustness to **non-uniform density** of the samples
(critical in real-life applications !!!)



Coifman & Lafon *Appl Comput Harm* 2006

3. Unsupervised learning

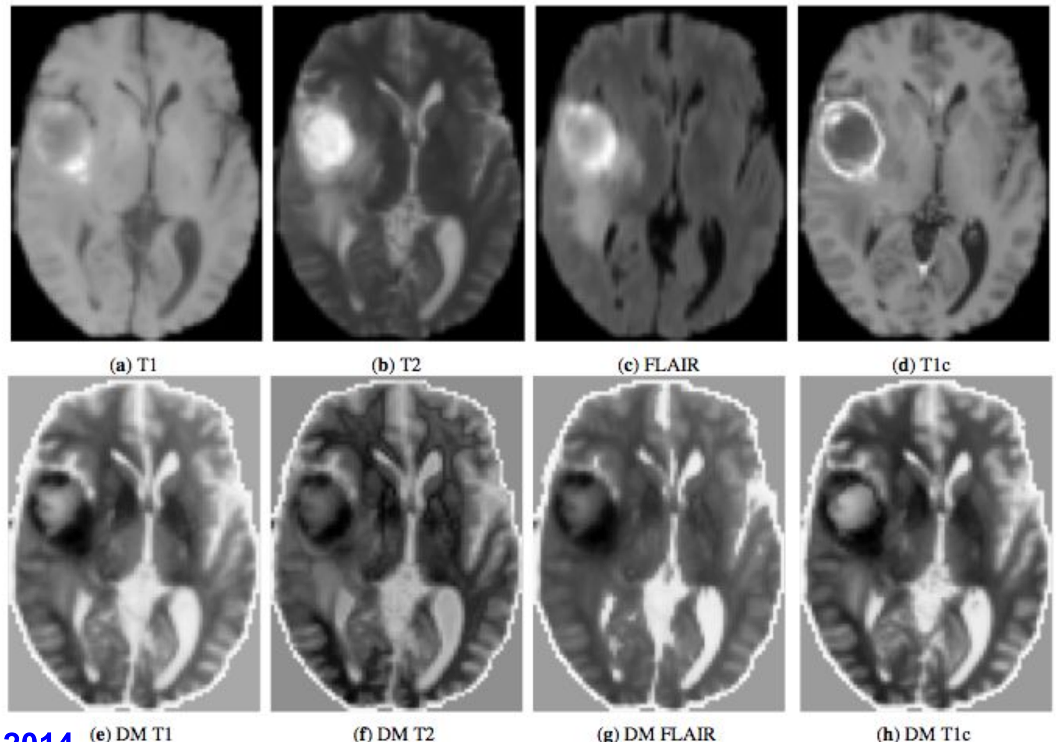
(low-dimensional) embedding: ~~linear~~ non-linear

To go further...	**Works on...**	
Laplacian eigenmaps	Graph Laplacian L = D-W	Belkin & Niyogi *Neur Comput* 2003
Diffusion maps	Normalized Laplacian P	Coifmain & Lafon *Appl Comput Harm* 2006

ex: Preprocessing for robust multimodal registration

Multimodal images →

Diffusion maps output →

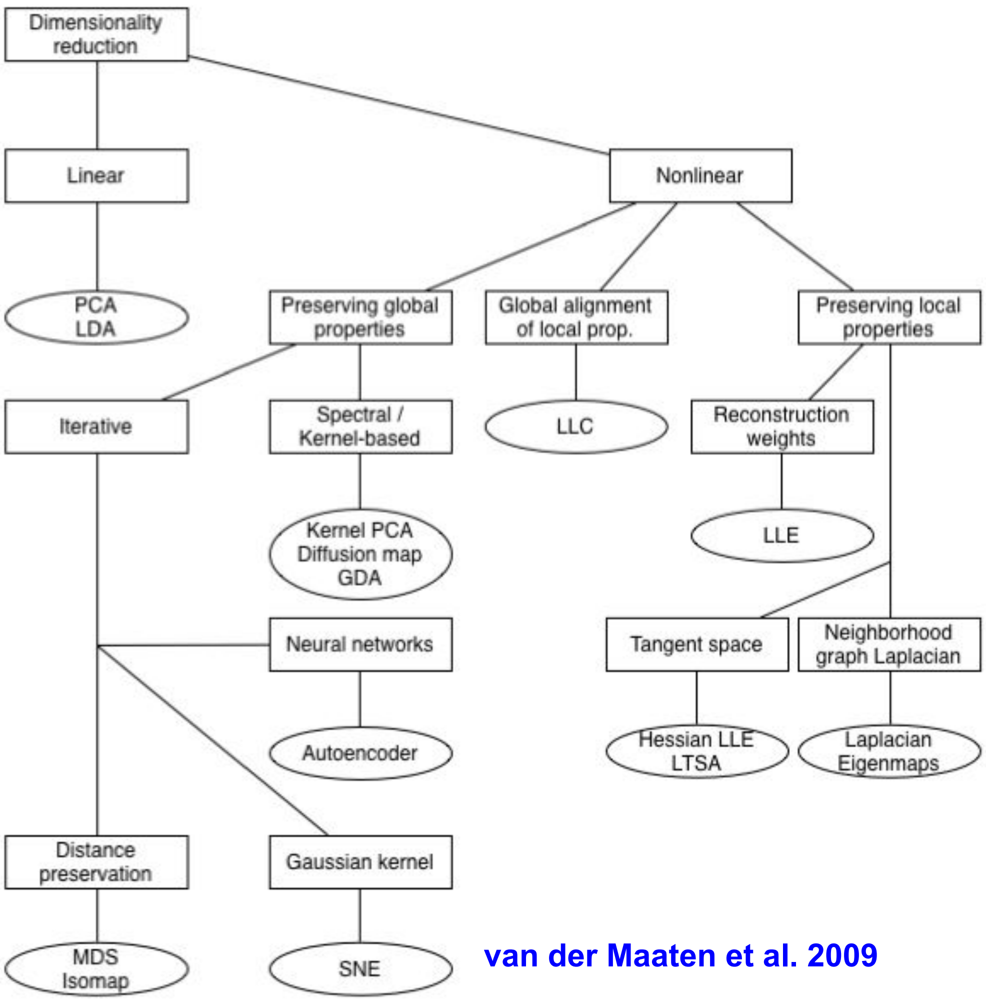


Piella *Sensors* 2014

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ … and many other algorithms !



van der Maaten et al. 2009

Depends on:

- Your knowledge on data
- Your objectives (distance=?)

3. Unsupervised learning

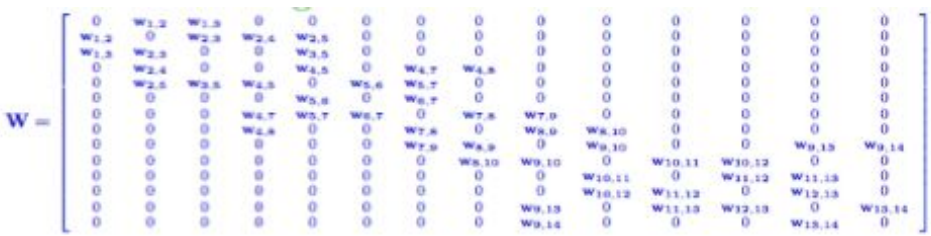
(low-dimensional) embedding: ~~linear~~ non-linear

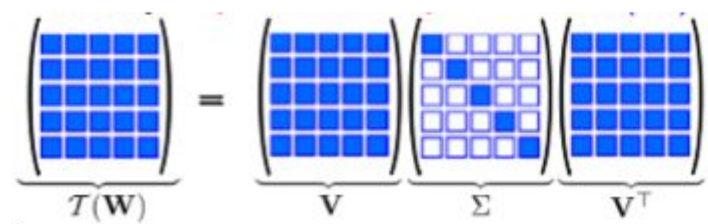
➔ **Unified framework?**

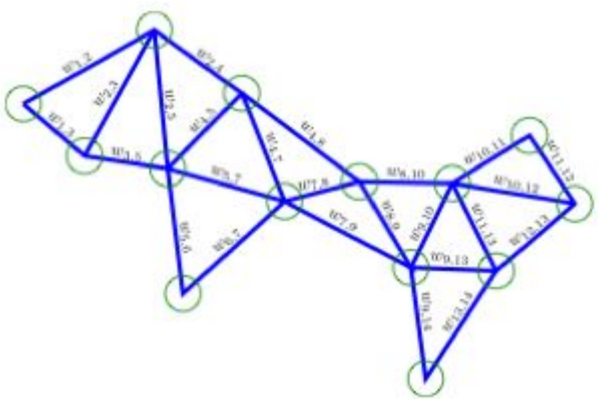
3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ Unified framework?







Method	Operator/Matrix	Preserved	Objective Function
PCA	Covariance matrix	Variance of the dataset / Euclidean distances between data points	$\mathbf{u}^\top \Sigma \mathbf{u}$
Laplacian Eigenmaps	Graph Laplacian	Distances within the local neighbourhood of each data point	$\mathbf{u}^\top L \mathbf{u}$
ISOMAP	Geodesic distance matrix	Geodesic distances between data points	$\mathbf{u}^\top D_G \mathbf{u}$
LLE	Reconstruction weights	Reconstruction weights within the local neighbourhood of each data point	$\mathbf{u}^\top W \mathbf{u}$

Atasoy & Mateus *MICCAI tutorial* 2011

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Unified framework** **Yan et al. *IEEE PAMI* 2007**

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{i,j} w_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 = \underset{\mathbf{Y}}{\operatorname{argmin}} \mathbf{Y}^T \mathbf{L} \mathbf{Y}$$

(cf. Laplacian eigenmaps…)

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Unified framework** Yan et al. *IEEE PAMI* 2007

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{i,j} \boxed{w_{ij}} \|\mathbf{y}_i - \mathbf{y}_j\|^2 = \underset{\mathbf{Y}}{\operatorname{argmin}} \mathbf{Y}^T \mathbf{L} \mathbf{Y}$$

Under the constraint:

Supervised

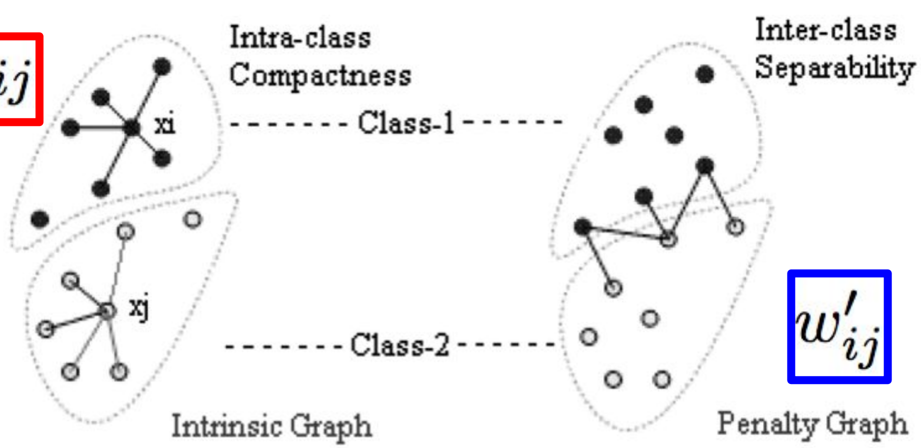
$$\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \boxed{w'_{ij}} = 1$$

Unsupervised

$$\sum_i \|\mathbf{y}_i\|^2 d_{ii} = 1$$

$$d_{ii} = \sum_j w_{ij}$$

$\boxed{w_{ij}}$



$\boxed{w'_{ij}}$

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

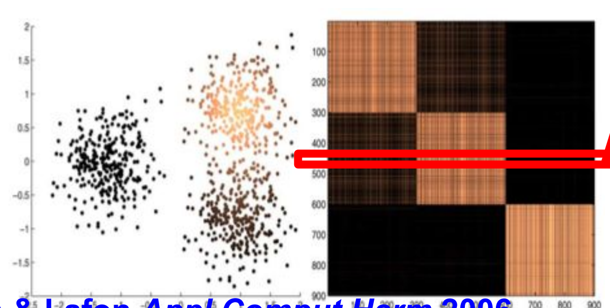
➔ **Unified framework** Yan et al. *IEEE PAMI* 2007

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmin}} \sum_{i,j} w_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2 = \underset{\mathbf{Y}}{\operatorname{argmin}} \mathbf{Y}^T \mathbf{L} \mathbf{Y}$$

Under the constraint:

Supervised	Unsupervised
$\sum_{i,j} \|\|\mathbf{y}_i - \mathbf{y}_j\|\|^2 w'_{ij} = 1$	$\sum_i \|\|\mathbf{y}_i\|\|^2 d_{ii} = 1$

$$d_{ii} = \sum_j w_{ij}$$



Coifman & Lafon *Appl Comput Harm* 2006

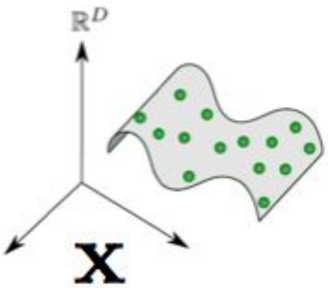
3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Back to our pipeline...**

(low-dimensional) embedding: ~~linear~~ non-linear

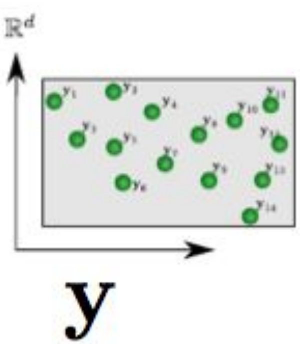
➔ **Back to our pipeline...**



Inputs =
- Single **high-dimensional** descriptors
- **Multiple** scalars

...or **Multiple high-dimensional** descriptors

1) Embedding



Output =
- **Low-dimensional** representation

2) Manifold / latent space

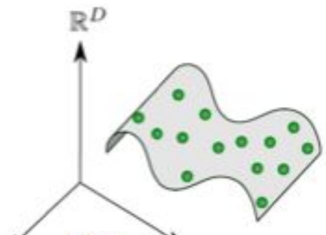
$$D \gg d$$

$$f : \mathbf{x} \in \mathbb{R}^D \mapsto \mathbf{y} \in \mathbb{R}^d$$

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Back to our pipeline...**



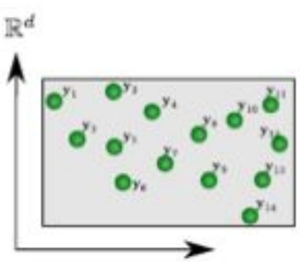
$\mathbf{x}$

<u>Inputs</u> =
- Single **high-dimensional** descriptors
- **Multiple** scalars

  ...or **Multiple high-dimensional** descriptors

1) Embedding

3) Reconstruction???



$\mathbf{y}$

<u>Output</u> =
- **Low-dimensional** representation

2) Manifold / latent space

$$D \gg d$$

$$f : \mathbf{x} \in \mathbb{R}^D \mapsto \mathbf{y} \in \mathbb{R}^d$$

$$g : \mathbf{x} \in \mathbb{R}^D \leftarrow\!\shortmid \mathbf{y} \in \mathbb{R}^d$$

3. Unsupervised learning

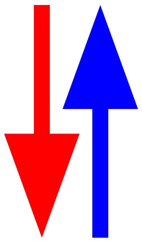
(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Reconstruction**

Analytical formula exists?

ex: PCA = linear change of basis $\Sigma = \mathbf{P}\mathbf{D}\mathbf{P}^T$

$$\mathbf{x} = \mathbf{P}\mathbf{y}$$



$$\mathbf{y} = \mathbf{P}^T\mathbf{x}$$

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

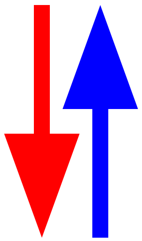
➔ **Reconstruction**

Analytical formula exists?

… in many cases, no !

= out-of-sample extension problem

$\mathbf{x}$

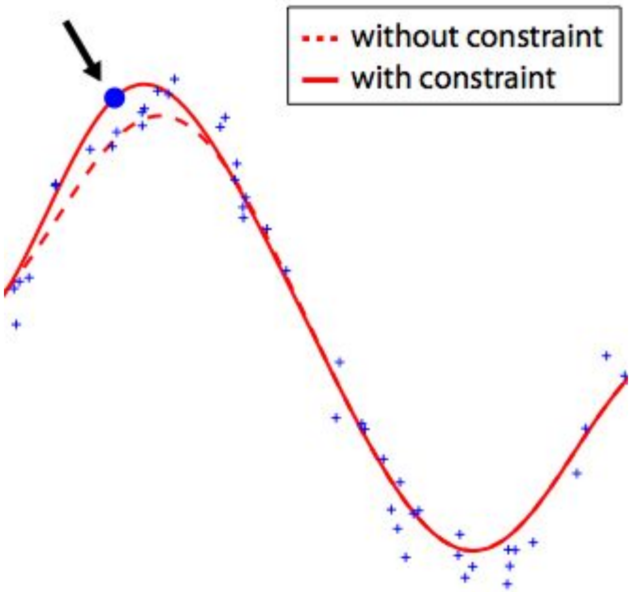


$\mathbf{y}$

> Possibility = **kernel** interpolation:

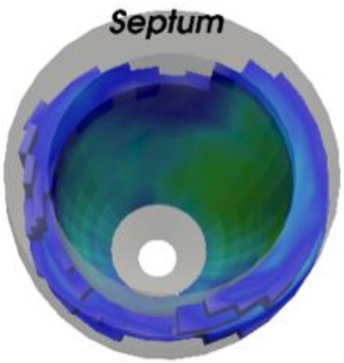
$$\mathbf{x} = \sum_{i=1}^{N} K'(\mathbf{y}, \mathbf{y}_i)\mathbf{c}_i$$

$$\mathbf{C} = (\mathbf{K}' + \mathbf{I}/\gamma)^{-1}\mathbf{x}$$



(low-dimensional) embedding: ~~linear~~ non-linear

➔ Reconstruction

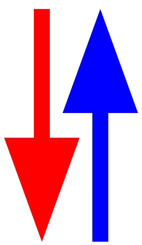


Linear average
Linear average

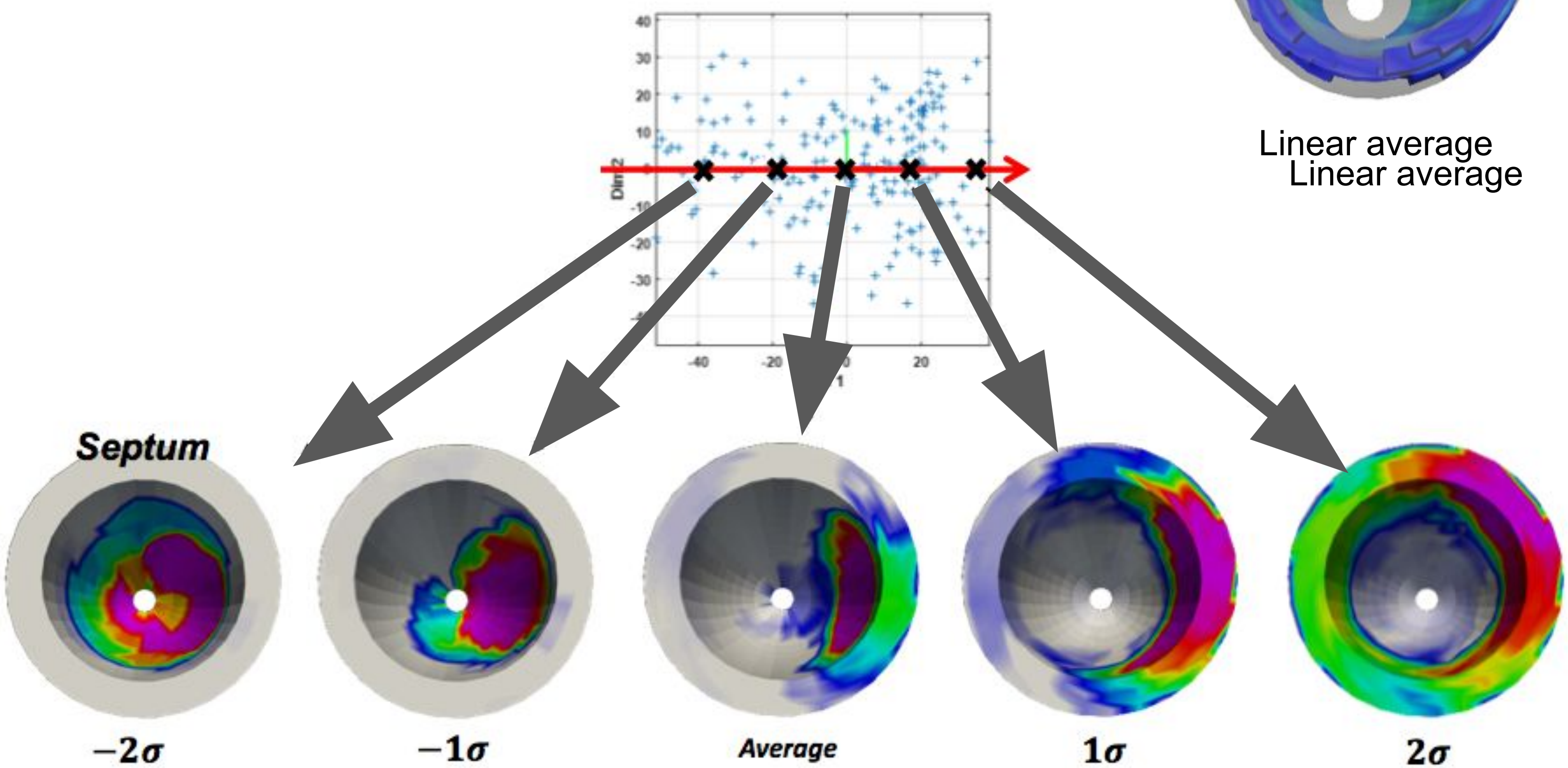
ex: variability in myocardial infarcts

Di Folco et al. *CNIV* 2019

x



y



3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Autoencoders** (unsupervised learning as supervised learning)

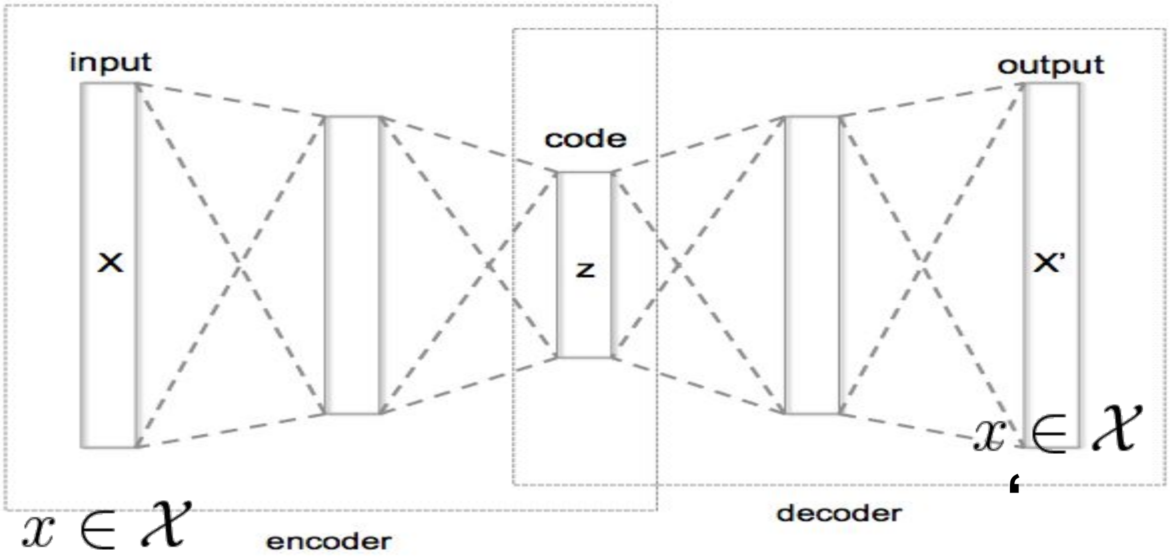
LeCun et al. *PhD* 1987
Hinton & Zemel *NIPS* 1994

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Autoencoders** (unsupervised learning as supervised learning)

LeCun et al. *PhD* 1987
Hinton & Zemel *NIPS* 1994



Simultaneously learn, in a supervised manner

- an **encoder** that maps the input into a **shorter code**
- a **decoder** that maps back a code into an input

Self supervision = minimize the reconstruction error

3. Unsupervised learning

(low-dimensional) embedding: ~~linear~~ non-linear

➔ **Autoencoders** (unsupervised learning as supervised learning)

LeCun et al. *PhD* 1987
Hinton & Zemel *NIPS* 1994

Extensions and other approaches

- Denoising **Autoencoders**, Variational Autoencoders (VAE), Conditional VAE, …
- **Generative Adversarial Networks** (GAN),
    - CGAN, WGAN, …, “gan zoo”
    - VAE-GAN, CVAE-GAN, …
- **Self supervised** learning

Summary

- ➔ **Unsupervised learning, depends on:**
  - ◆ Your data problem: labels / no labels, ...
  - ◆ Your initial question: clustering, outliers, …

- ➔ **Learning a representation is key**, depends on:
  - ◆ Linear / non-linear
  - ◆ Objectives:

	Embedding space	**Reconstruction**
Manifold learning	**Meaningful** distances between *input* samples ~ Euclidean distance between *output* coordinates	Interpolation?
Auto-encoders	Limited statistical meaningi	Optimized encoding/decoding

3. Unsupervised learning

Summary

➔ **Validation not straightforward:**
  - No labels !
  - vs. method's way of working? (ex: short-circuit)
  - vs. application's objectives? (ex: risk analysis, knowledge discovery)

➔ **Still <u>under-used</u> vs. supervised learning**
  - Promising for medical problems !
  - Rising **semi-supervised** learning...