Dimension Reduction for Fun and Profit

2018-08-06, Oxford

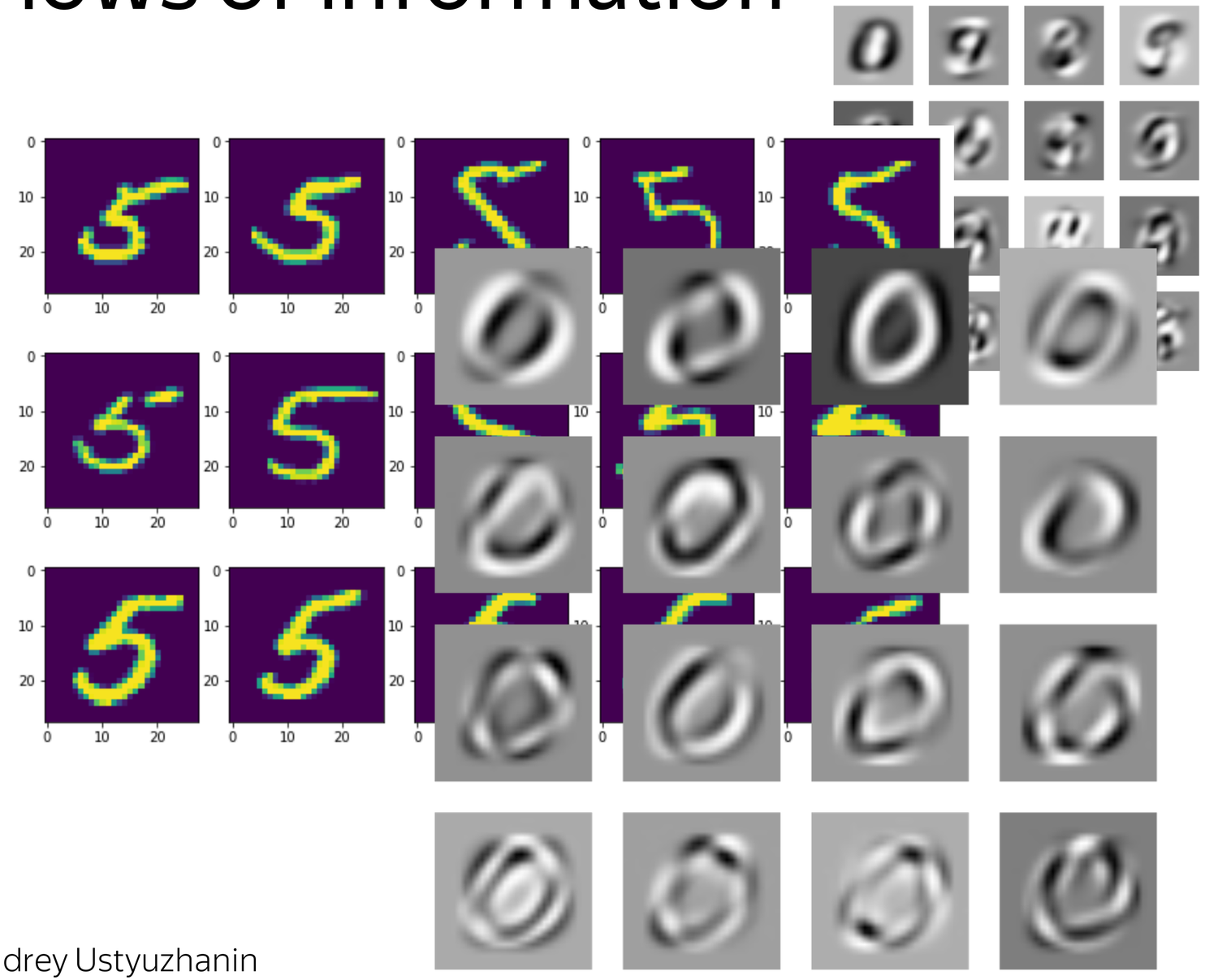
Andrey Ustyuzhanin

NRU HSE

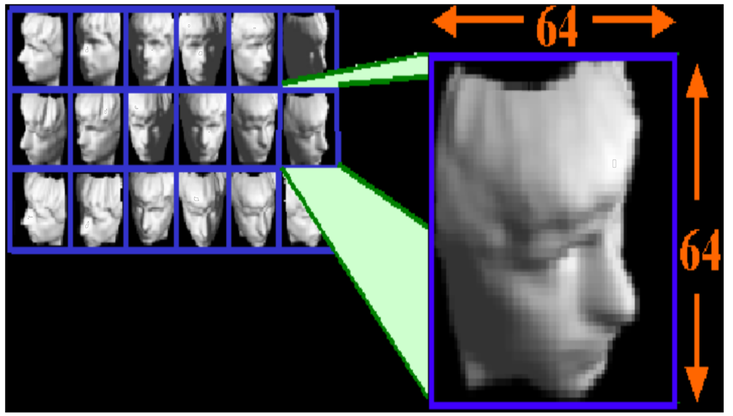
YSDA

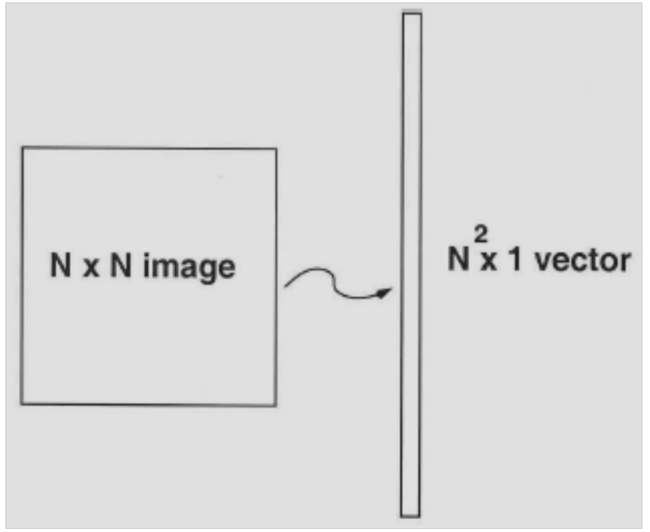
ICL

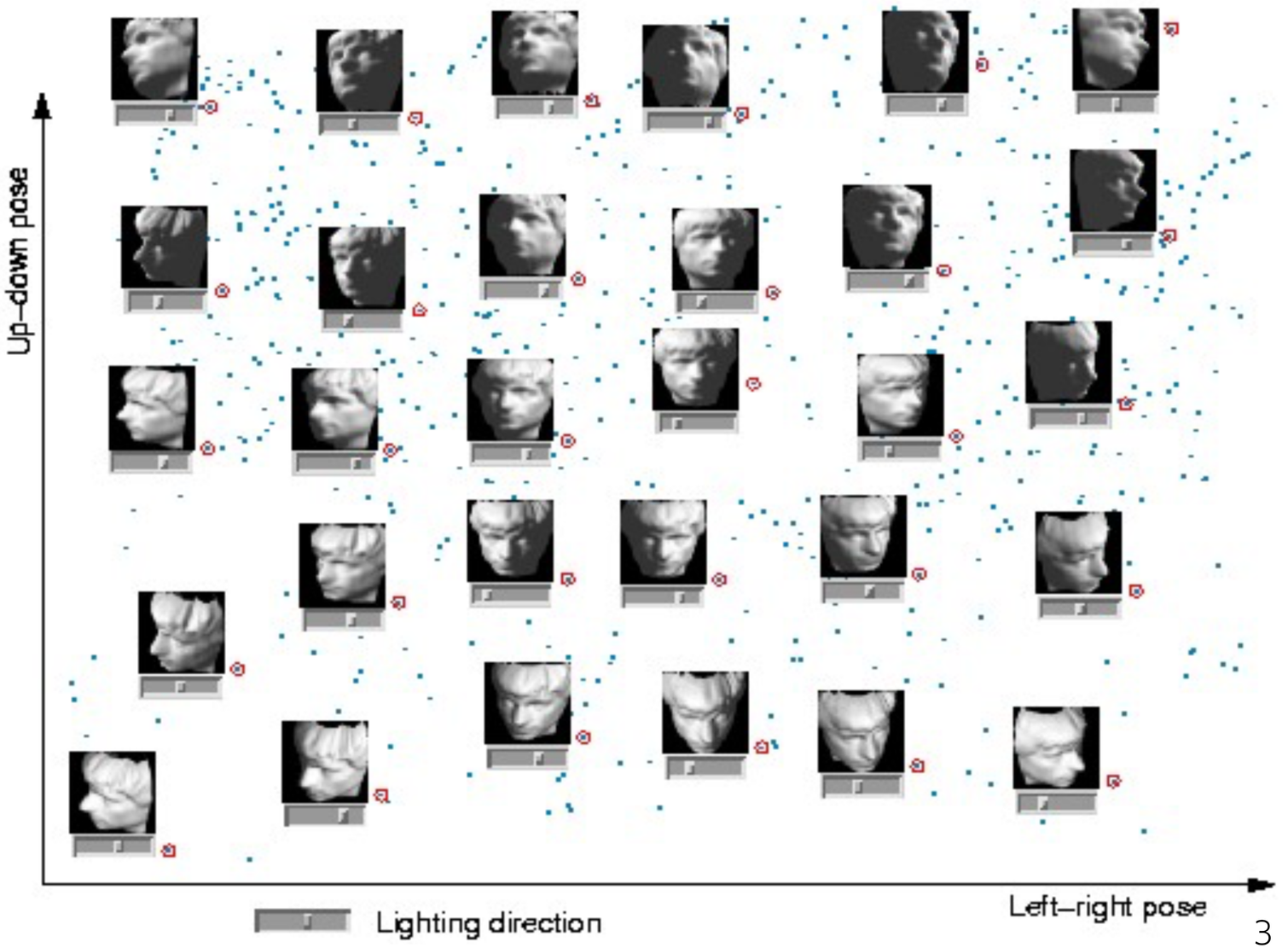
Flows of information



Faces

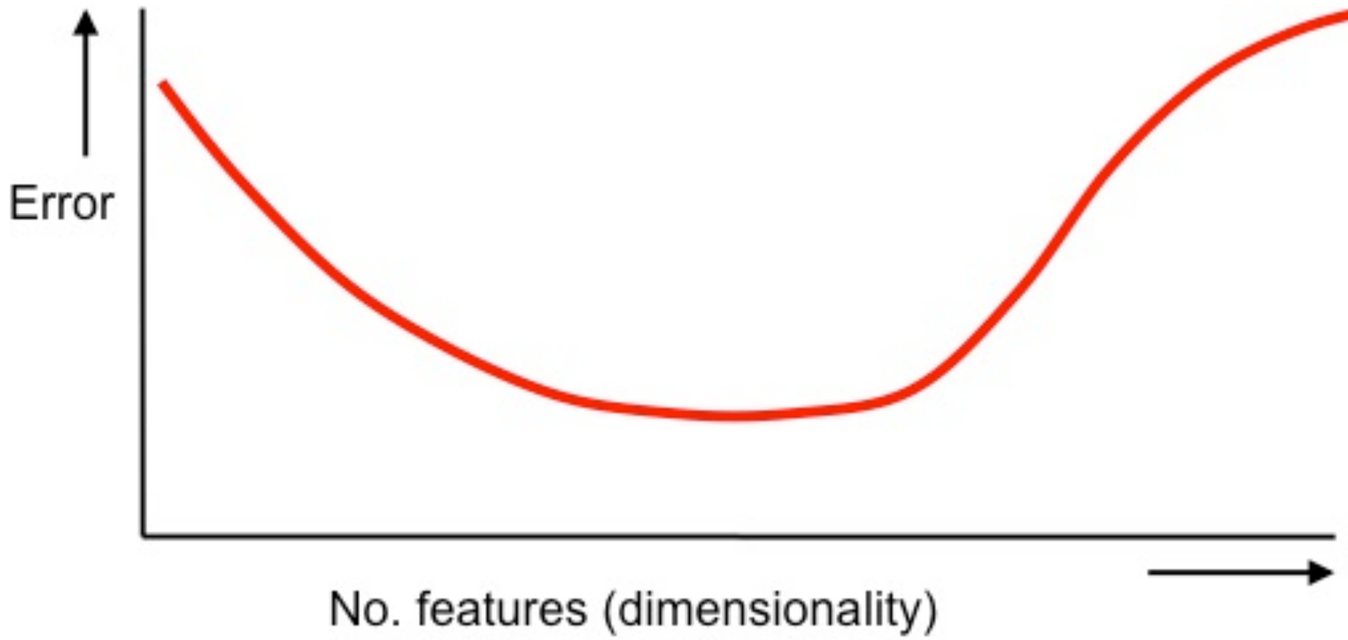

64
64


N x N image
N 2 x 1 vector


Up–down pose
Left–right pose
Lighting direction

Motivation: curse of dimensionality, errors

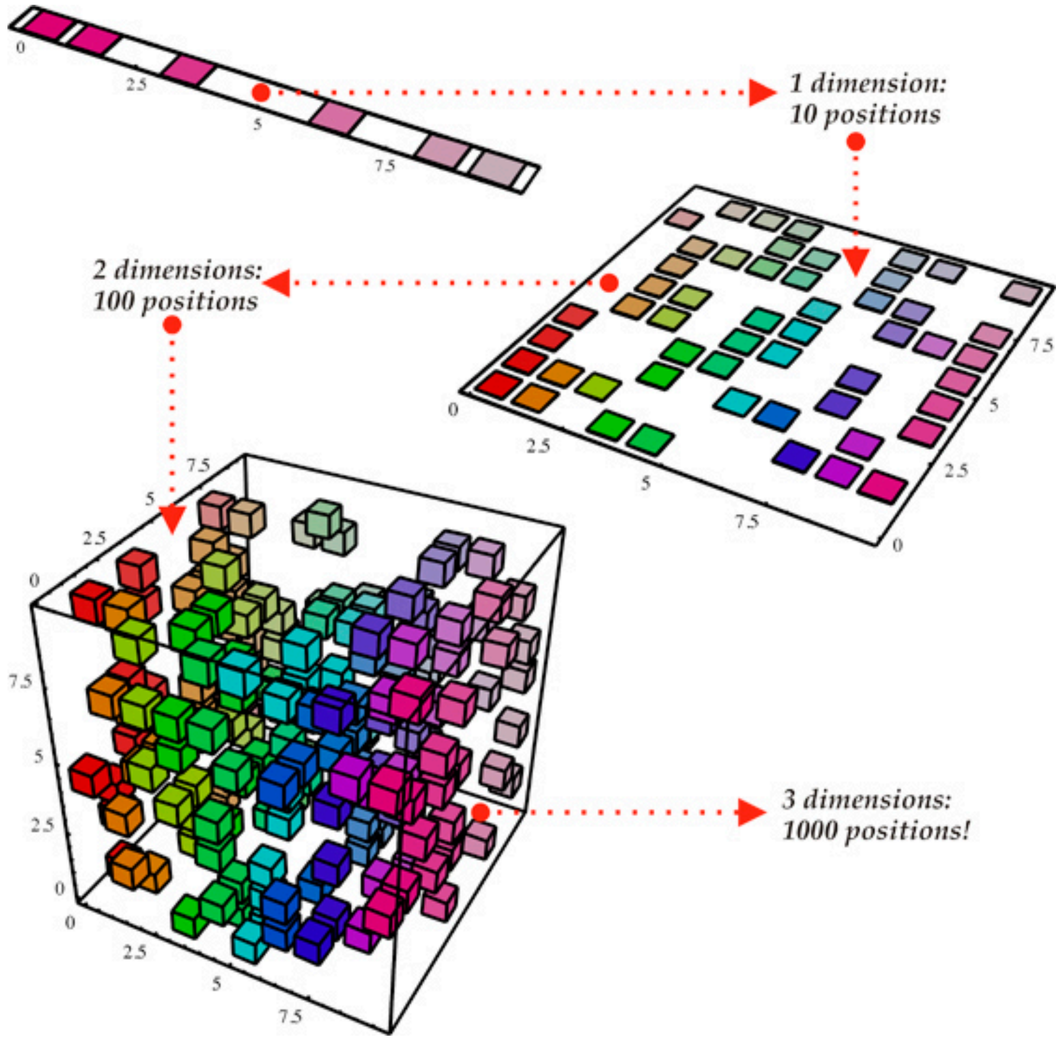
- Too low dimensionality may lead to high variance
- Too high dimensionality may lead to sparsity and high bias
- Classification tasks are harder in higher dimensions



Motivation: curse of dimensionality, sparsity

To cover 10% of N-dimensional (100-cube) volume you have to provide:

- 10 samples (1D)
- 100 samples (2D)
- 1000 samples (3D)
- ...

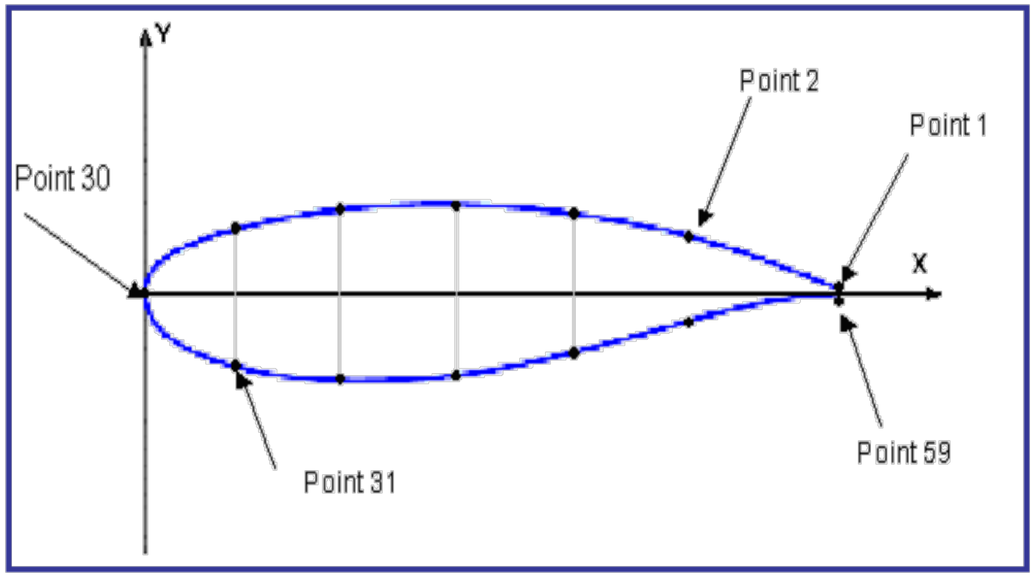


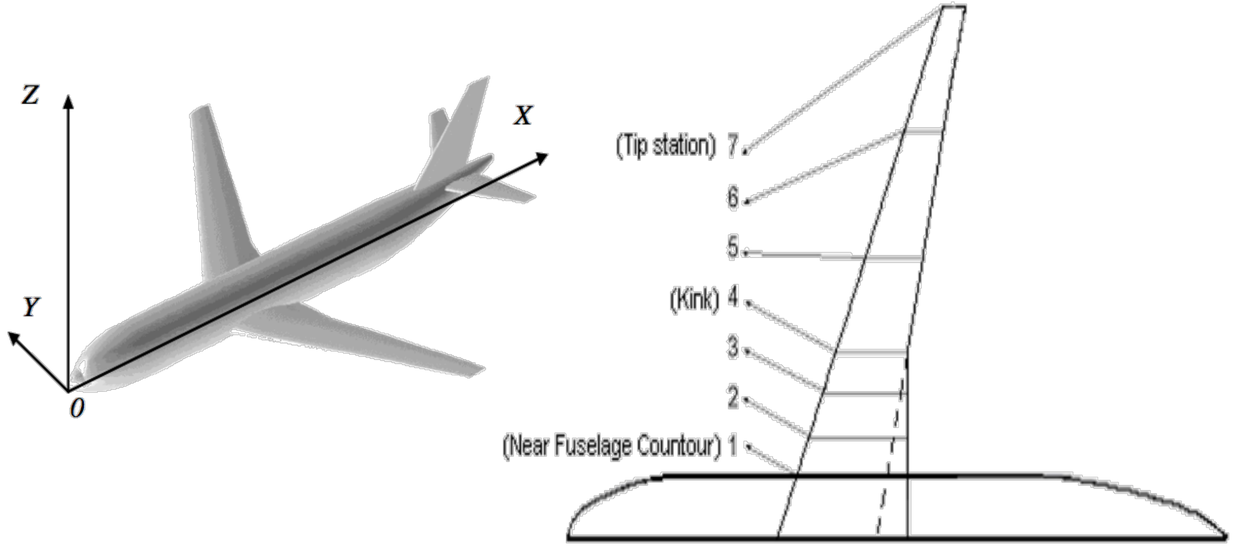
Airfoil parametrization

Aircraft wing can be described by 7 cuts:

- Each cut takes 59 points to describe 2D shape

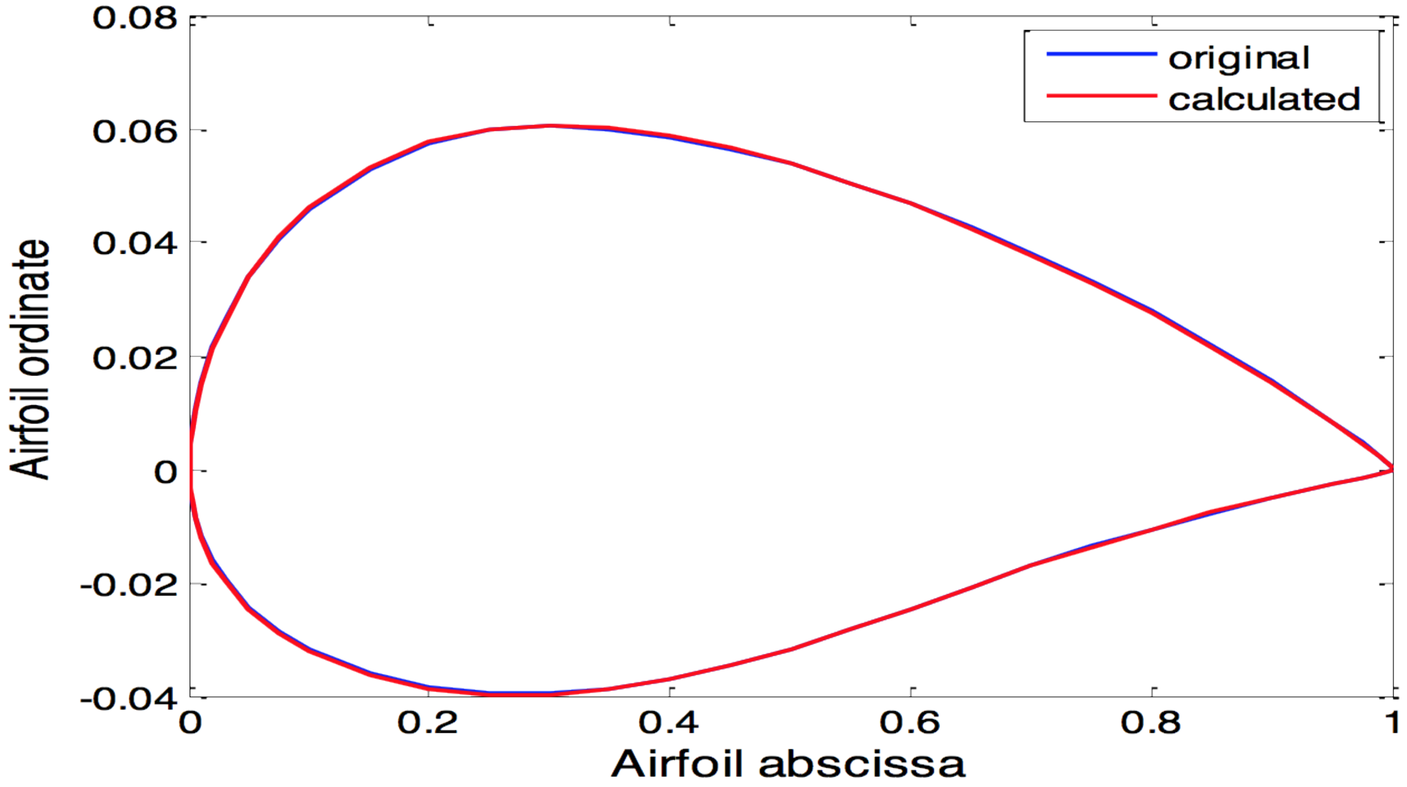
If we could find a better parametrization without losing precision?



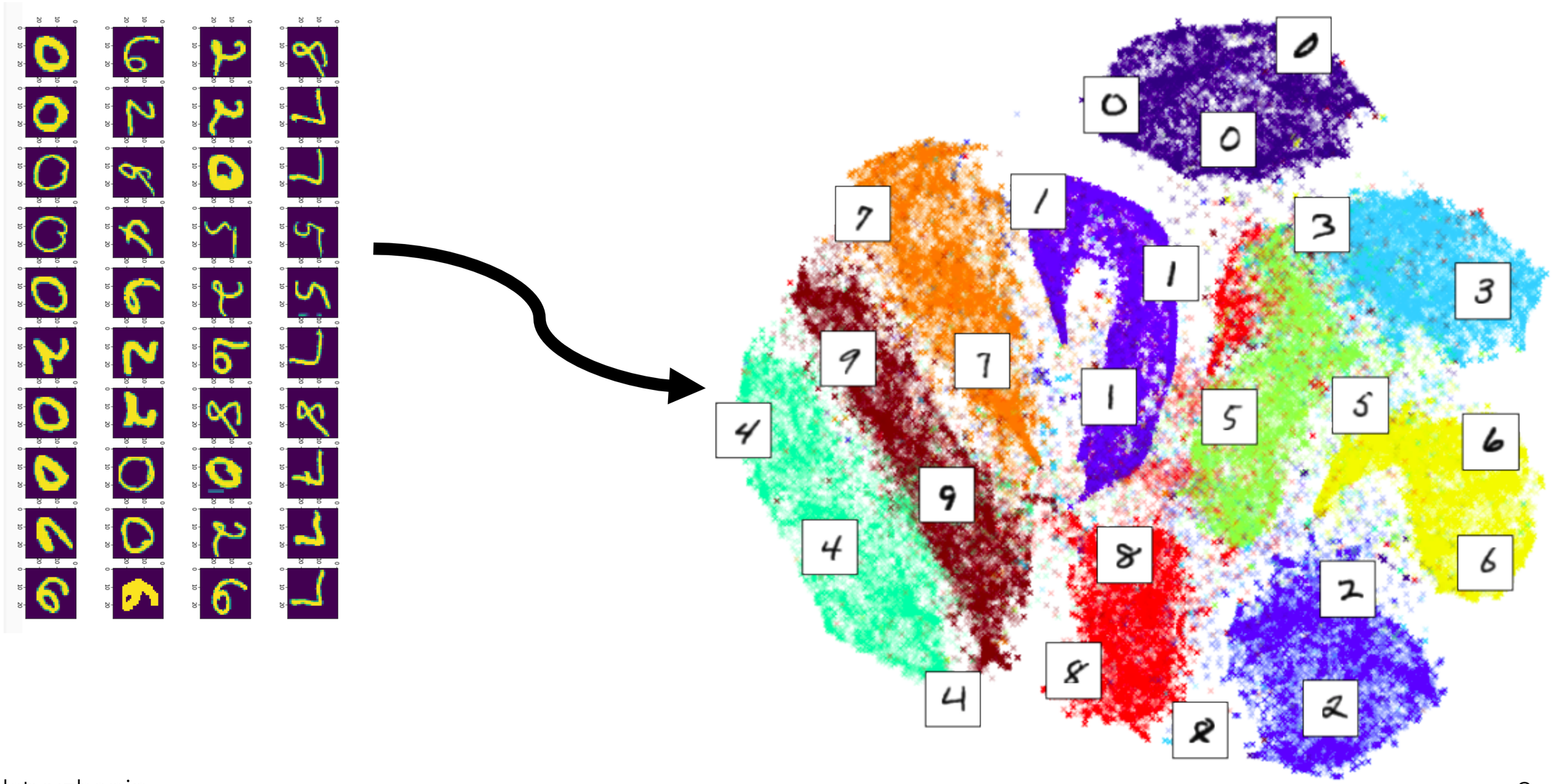


Airfoil parametrization

Blue: original shape (59 points)
Red: 6-parametrization



MNIST dimensionality reduction to 2



Particle Physics examples

Multidimensional objects:

- Hits: Jets, Showers (represented by images)
- Tracks

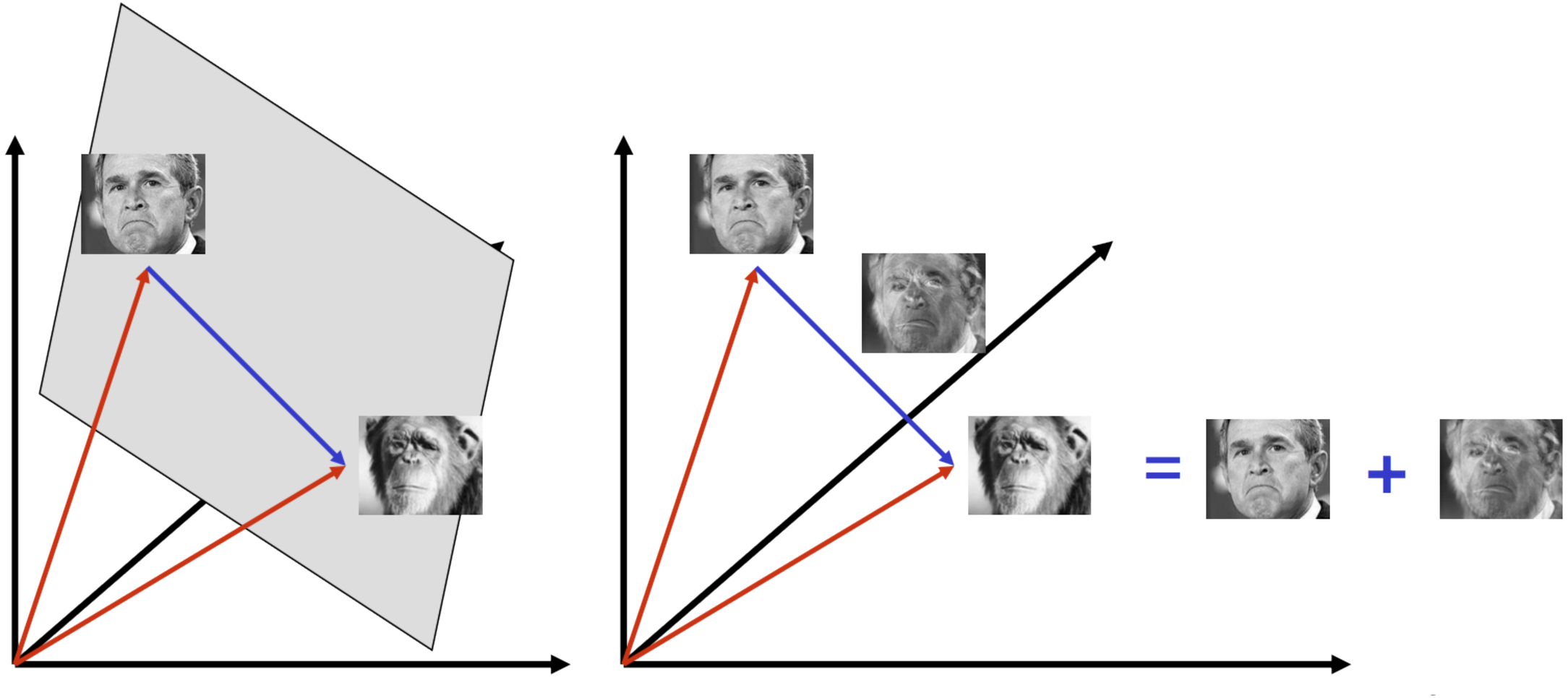
Complex objects

- Events (track collection)
- Trigger/stripping lines
- Collection of subdetector highlevel responses for PID, trigger, etc
- ...

Applications

- Compression (improve operational performance)
  - CPU, Memory, IO
- Visualization
- Remove collinearity (helpful to some models)
- Feature extraction (meaningful representation)
  - Vector operations

Applications. Vector operations



Approaches

Unsupervised learning

- Manifold learning

Basic Problem Statement:

- given set of objects O_i, described by features $X(O_i)$ from R^p
- Find compressed representation $y(O_i)$ from R^q, $q < p$
- Provided no significant loss of information about O_i

Approaches

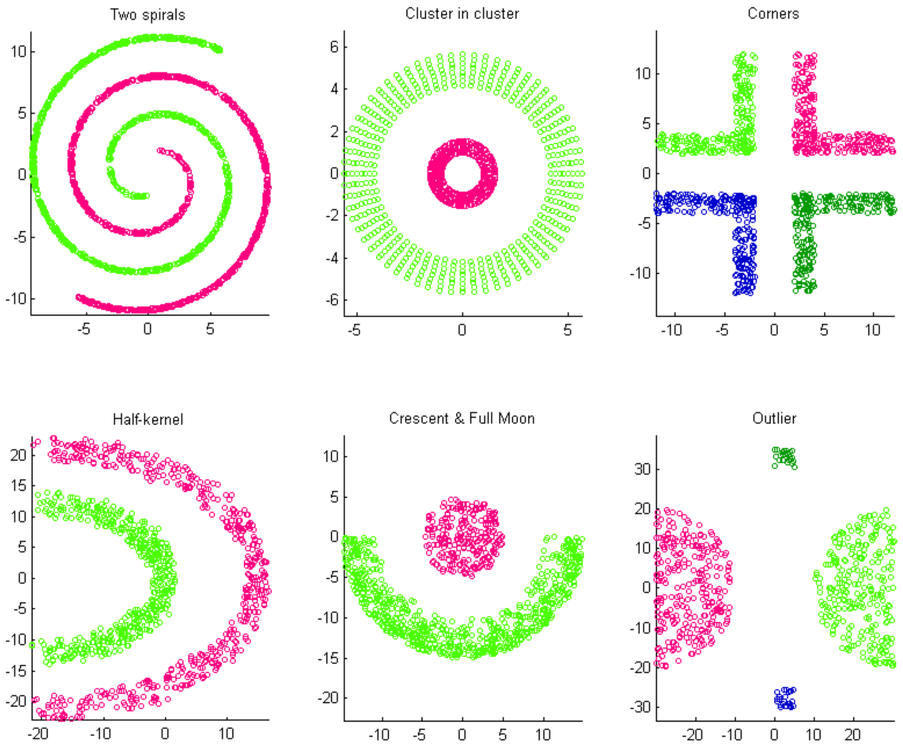
- **Unsupervised learning**
  - Manifold learning
- **Basic Problem Statement:**
  - given set of N objects $\mathbf{O}_i$, each is described by p-vector of features
  - find such mapping from p-space to q-space, p > q,
  - provided no significant information is lost about $\mathbf{O}_i$

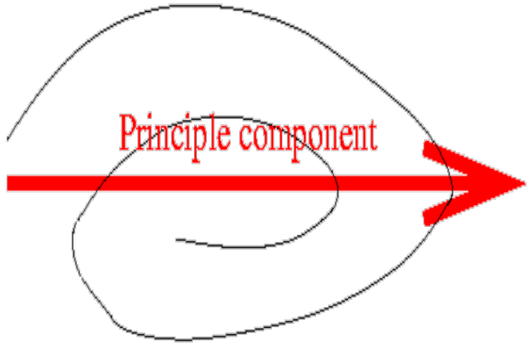
Linear Methods

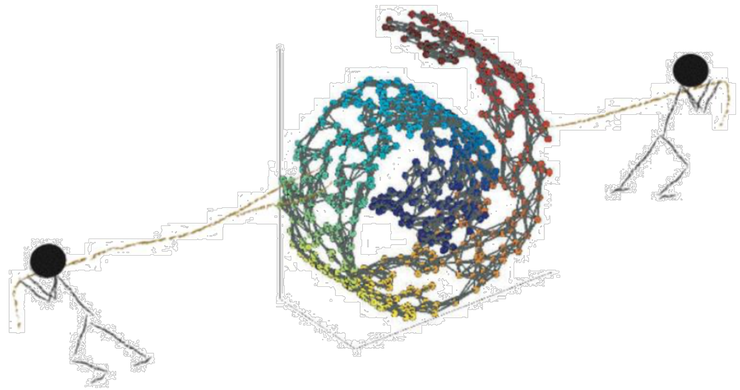
Linear Methods Recap

- **PCA**
  - Should be run on scaled data
  - Relatively easy interpretable
  - Slow to calculate without SVD hacks (randomized, truncated)
- **SVD**
  - Allows to estimate principal components
  - Gives mapping into hidden space
- **Other methods examples**
  - Non-negative matrix factorization, Pursuit Projection, Multi Dimensional Scaling

Finding low-dimensional manifolds

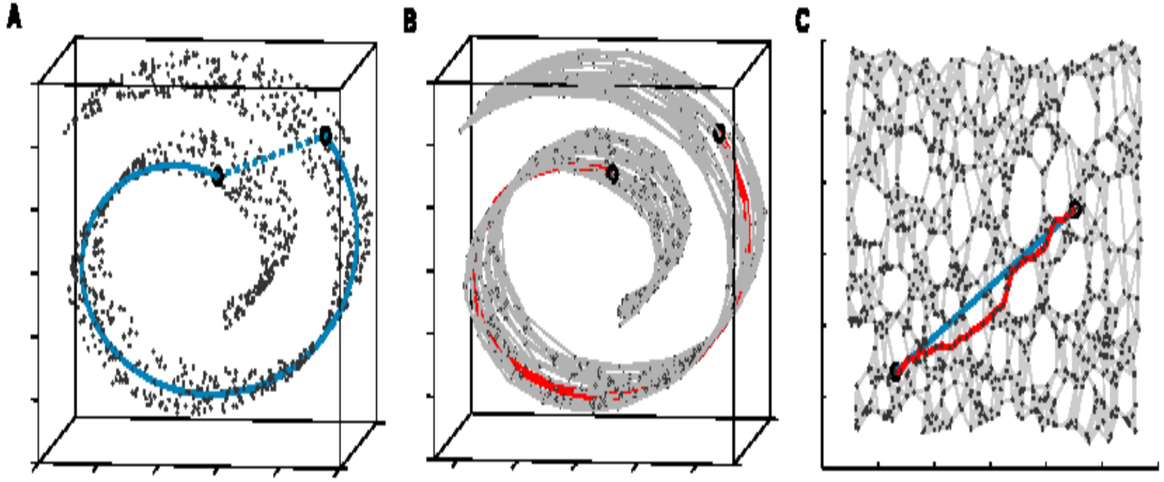






Non-linear methods

- t-distributed stochastic neighbor embedding (t-SNE)
- Locally Linear Embedding (LLE)
- Laplacian Eigenmaps (LE)
- Hessian Eigenmaps (HE)
- ISOmetric MAPping (ISOMAP)
- Kernel PCA
- Riemannian Manifold Learning (RML)
- Local Tangent Space Alignment (LTSA)



T-SNE

Further Tasks

Anomaly detection / identification
Generative model design
Bijective compression
Constrained dimensionality reduction

Conclusion

Helpful methods for solving practical tasks

- Finding more relevant and efficient representation
- Discover intrinsic structure

Further questions:

- Can you restore original representation from embedding?
- Can you specify external conditions to be met while after data transform?

Unsupervised learning example:

- Look, ma, no labels!

References

PCA

http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html
http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/
http://www.math.union.edu/~jaureguj/PCA.pdf

T-SNE

https://oreillymedia.github.io/thebe/examples/t-sne-build.html
https://distill.pub/2016/misread-tsne/
https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/
http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html