Non-linear dimensionality reduction

Michel Verleysen
Université catholique de Louvain (Louvain-la-Neuve, Belgium)
Electricity department

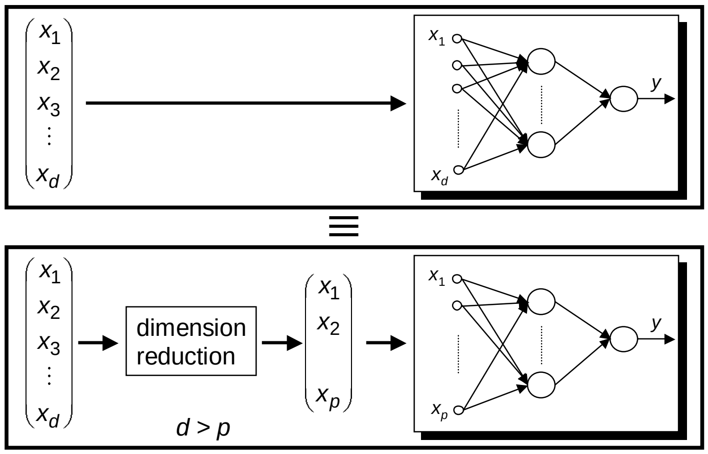
June 2002

Motivation

- High-dimensional data are
  - difficult to represent
  - difficult to understand
  - difficult to analyze
- Example: MLP (Multi-Layer Perceptron) or RBFN (Radial-Basis Function Network) with many inputs: difficult convergence, local minima, etc.
- Need to **reduce the dimension of data while keeping information content**!

Motivation: example

- Supervised learning with MLP

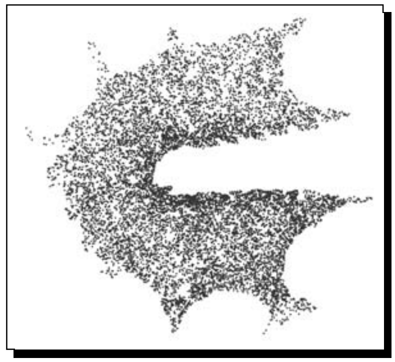


What we have:

- High-dimensional numerical data coming from:
  - sensors
  - pictures,
  - biomedical measures (EEG/ECG),
  - etc.

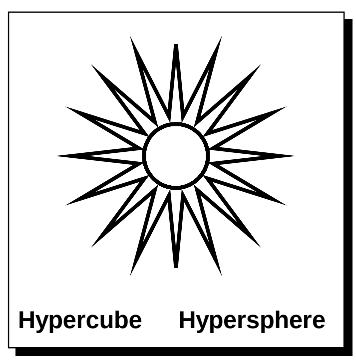
What we would like to have:

- A low-dimensional representation of the data in order to:
  - visualize
  - compress,
  - preprocess,
  - etc.



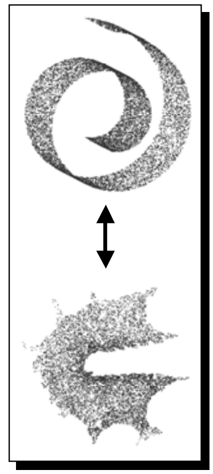
Why ?

- Empty space phenomenon:
  - # points necessary for learning grows exponentially with space dimension
- Curse of dimensionality
  - « Spiky » hypercube
  - Empty hypersphere
  - Narrow spectrum of distances



How ?

- Build a (bijective) relation between
  - the data in the original space
  - the data in the projected space
- If bijection:
  - possibility to switch between representation spaces (« information » rather than « measure »)
- Problems to consider:
  - noise
  - twists and folds
  - impossibility to build a bijection



Content

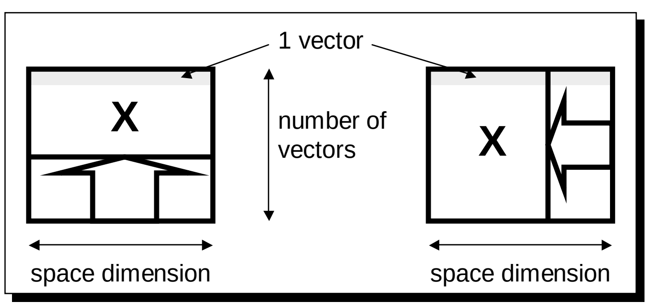
- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

NLP <-> VQ

- Non-Linear Projection
- Vector Quantization



- Reduction of the *dimension* of the data (from *d* to *p)*
- Reduction of the *number* of data (from *N* to *M*)

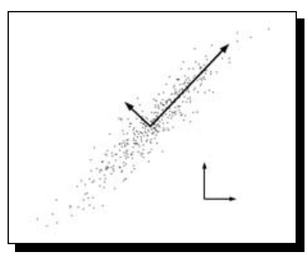
Warning: « lines and columns » convention adopted in linear algebra – contrary to most neural network courses and books…

Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Principal Component Analysis (PCA)

- Goal:
  - To project linearly while keeping the variance of the data
- Computation:
  1. Covariance matrix C of the data
     $C = E\{X_i X_i^T\} = 1/N \; X \; X^T$
  2. Eigenvectors and eigenvalues of C
     V_i = main directions
     λ_i = variance along each direction
  3. Projection & Reconstruction
     $Y = V_{1 \le i \le p}^T \; X$ ← projection
     $X \approx Z = V_{1 \le i \le p} \; Y$ ← « unprojection »



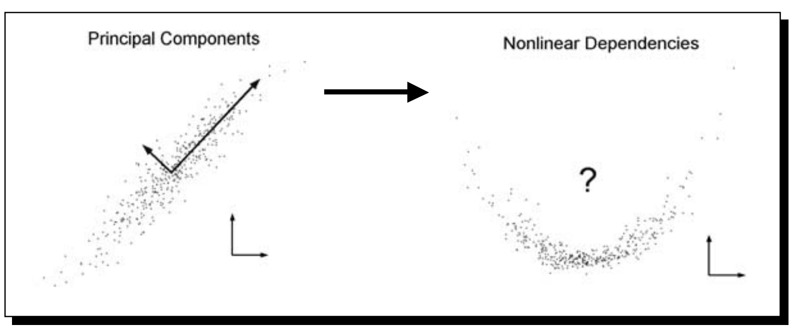
- Also called « Karhunen-Loeve » transform

Metric Multi-Dimensional Scaling (MDS)

- Goal:
  - To project linearly while keeping the (N-1)*N/2 pairwise distances
- Computation:
  1. Matrix D of the squared distances
     $D = [\, d^2_{i,j} \,] = [\, (X_i - X_j)^T (X_i - X_j) \,]$
  2. EigenVectors and eigenvalues of D aften centering (= X X^T)
     V_i = *coordinates* along the main directions
     λ_i = variance along each direction
  3. Projection
     $Y = \mathrm{sqrt}(\, \mathrm{diag}(\, \lambda_{1 \le i \le p})) \, V_{1 \le i \le p}^T$
- Result of PCA = result of metric MDS !!!
- Only distances are needed -> more independent from representation !

Limitations of linear projections

- Detection of linear dependencies only
- What happens with non-linear dependencies?



Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Content

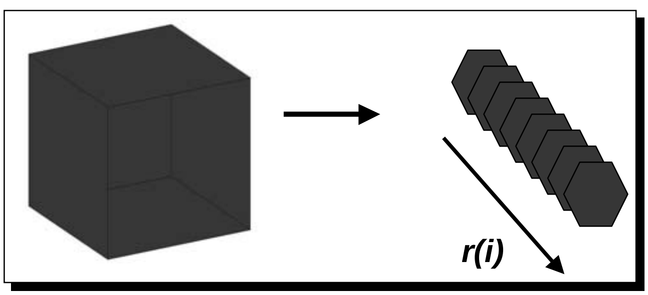
- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation → Local PCA, Kernel PCA
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Local PCA (1/2)

- Criterion:
  - Preserve variance (like PCA) *locally*
- Calculation:
  1. Vector quantization:
     prototypes C_r = representave points of data X_i
  2. Tesselation:
     Voronoï zones = set of X_i with same BMU index $r(i)$
  3. PCA on each zone:
     the model is locally linear and globally non linear
  4. Encoding:
     X_i (dimension d) transformed in $r(i)$ & Y_i (dimension p)

Local PCA (2/2)

- Example



- Shortcomings:
  - No « continuous » representation
  - Mosaic of « disconnected » coordinate systems

Kernel PCA (1/3)

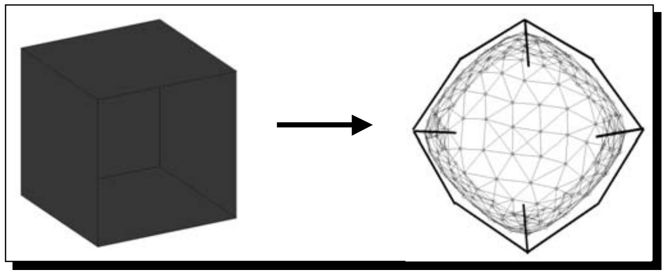
- Criterion:
  - To preserve variance (like PCA) of *transformed* data
- How ?
  - To transform data non-linearly
    (in fact, to transform non-linearly the MDS distance matrix)
  - Transformation: allows to give more weigth to small distances
  - Tranformation used: often Gaussian
  - Interesting theoretical properties:
    - non-linear mapping to high-dimensional spaces
    - Mercer's condition on Gaussian kernels
    - …

Kernel PCA (2/3)

- Calculation:
  1. Dual Problem (cfr PCA <-> MDS):
     ($C = X X^T$) $D = X^T X = [\ X_i^T X_j\ ]$
  2. Nonlinear transformation of data:
     $D' = [\ \Phi(X_i, X_j)\ ]$ with Φ s.t. $\Phi(u,v) = \varphi(u)\ \varphi(v)$ (Mercer condition)
  3. Centering of D'
  4. Eigenvalues and eigenvectors of D':
     V_i = *coordinates* along the main directions
  5. Projection:
     $Y = V_{1 \leq i \leq p}{}^T$

Kernel PCA (3/3)

- Example:



- Shortcomings:
  - Eigenvalues = 0.138, 0.136, *0.099*, 0.029,…
  - Dimensionality reduction is not guaranteed…

Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS) → Sammon's NLM, CCA / CDA, Isomap
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Sammon's Non-Linear Mapping (NLM) 1/2

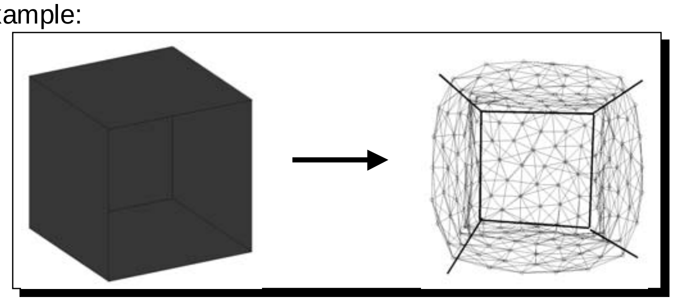
- Criterion to be optimized:
  - Distance preservation (cfr metric MDS)
  - Sammon's stress = $\frac{1}{\sum_{i<j} \delta_{i,j}} \sum_{i<j} \frac{\left(\delta_{i,j} - d_{i,j}\right)^2}{\delta_{i,j}}$

    distances in original space

    distances in projection space
  - Preservation of small distances firstly
- Calculation:
  - Minimization by gradient descent

Sammon's Non-Linear Mapping (NLM) 2/2

- Example:



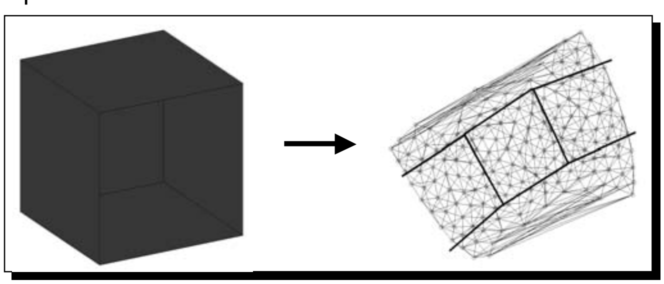
- Shortcomings:
  - Global gradient: lateral faces are « compacted »
  - Computational load (preprocess with VQ)
  - Euclidean distance (use curvilinear distance)

Curvilinear Component Analysis (1/2)

- Criterion to be optimized:
  - Distance preservation
  - Preservation of small distances firstly (but « tears » are allowed)
  -
- Calculation:
  1. Vector Quantization as preprocessing
  2. Minimization by stochastic gradient descent (±)
  3. Interpolation

Curvilinear Component Analysis (2/2)

- Example:

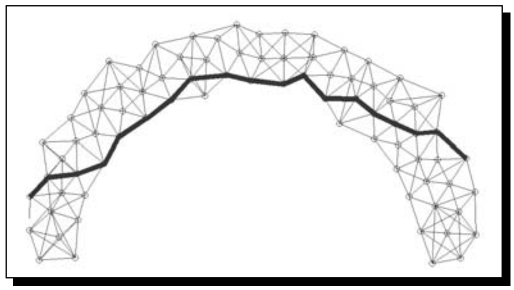


- Shortcomings:
  - Convergence of the gradient descent: « torn » faces
  - Euclidean distance (use curvilinear distance)

NLP: use of curvilinear distance (1/4)

- Principle:

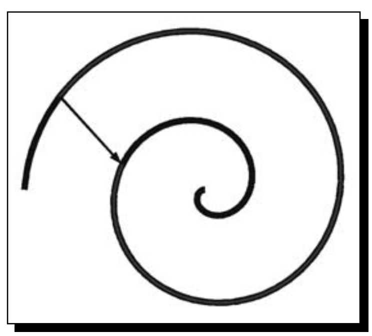
Curvilinear (or geodetic) distance
=
Length of the shortest path from one node to another
in a weighted graph



Michel Verleysen 27

NLP: use of curvilinear distance (2/4)

- Useful for NLP

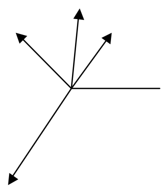


Curvilinear distances are easier to preserve!

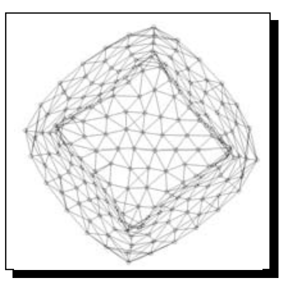
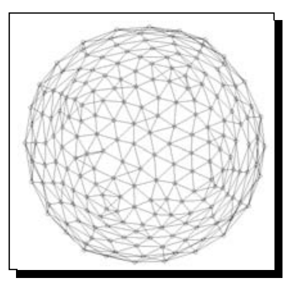
Michel Verleysen 28

NLP: use of curvilinear distance (3/4)

- Integration in projection algorithms:



NLP: use of curvilinear distance (4/4)

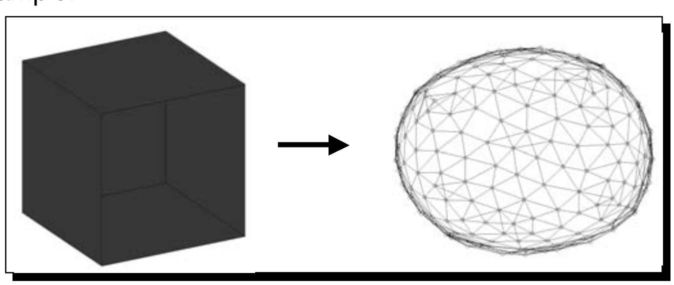
Projected open box: Sammon's NLM with Euclidean distance	Projected open box: Sammon's NLM with curvilinear distance
	
Faces are « compacted »	« Perfect »!

Isomap (1/2)

- Published in *Science* 290 (December 2000):
  ***A global geometric framework for nonlinear dimensionality reduction*.**
- Criterion:
  - Preservation of geodesic distances
- Calculation:
  1. Choice of some representative points (randomly, without VQ!)
  2. Classical MDS, but applied on the matrix of geodesic distances

Isomap (2/2)

- Example:



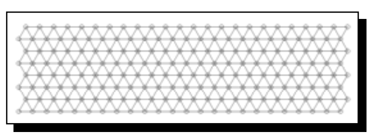
- Shortcomings:
  - No weighting of distances: faces are heavily « compacted »
  - No vector quantization

Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM) → SOM Isotop
  - Minimal reconstruction error
- Comparisons
- Conclusions

Self-Organizing Map (SOM) (1/2)

- Criterion to be optimized:
  - Quantization error & neighborhood preservation
  - No unique mathematical formulation of neighborhood criteria
- Calculation:
  - Preestablished 1D or 2D grid: distance *d(r,s)*



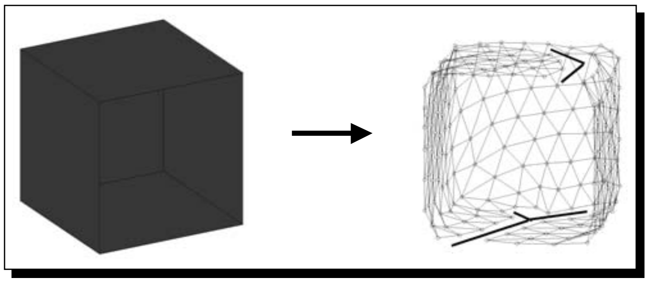
  - Learning rule:

$$r(i) = \arg\min_r \|X_i - C_r\|$$

$$\Delta C_r = \alpha\, e^{\frac{-d^2(r,r(i))}{2\lambda^2}} (X_i - C_r)$$

Self-Organizing Map (SOM) (2/2)

- Example:



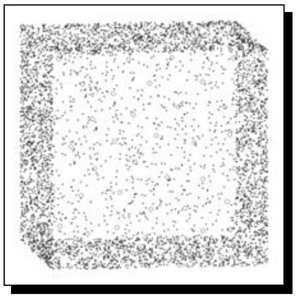
- Shortcomings:
  - Inadequate grid shape: faces are « cracked »
  - 1D or 2D grid only…

Isotop (1/3)

- Inspired from SOM and CCA/CDA
- Criterion:
  - Neighborhood preservation
  - No known math. formula…
- Calculation within 4 steps:
  1. Vector quantification
  2. Linking prototypes C_r
  3. Mapping (between d-dim. and p-dim. spaces)
  4. Linear interpolation

Isotop (2/3)

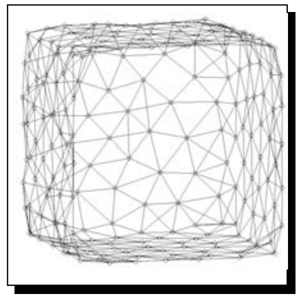
1. Vector quantification



3D

No preestablished shape

2. Linking of all prototypes

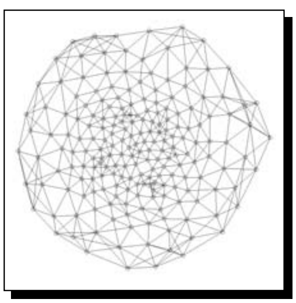


3D

« Data-driven neighborhoods»

Isotop (3/3)

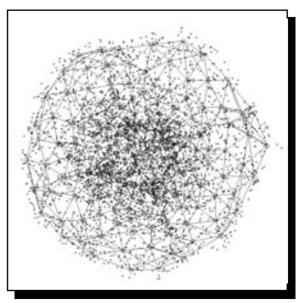
3. Mapping



2D

VQ (~SOM) of a Gaussian pdf

4. Linking of all prototypes



2D

Local linear interpolations

Content

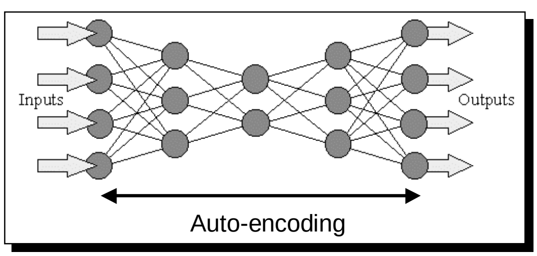
- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error → Autoassociative MLP
- Comparisons
- Conclusions

Autoassociative MLP (1/2)

- Criterion to be minimized:

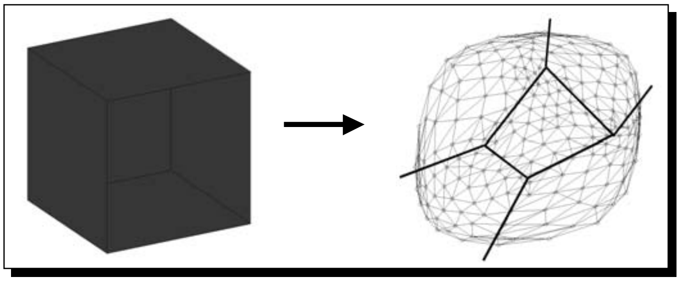
  Reconstruction error (MSE)
  after coding and decoding of the data
  with an autoassociative neural network (MLP)

- Autoassociative MLP: unsupervised (in=out)



Autoassociative MLP (2/2)

- Example:



- Shortcomings:
  - « Non-geometric » method
  - Slow and hasardous convergence (5 layers!)

Content

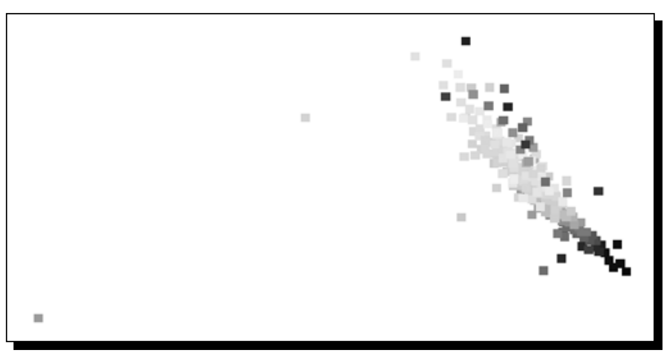
- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Comparisons: dataset

- *Abalone* (UCI Machine learning repository):
  - 4177 shells
  - 8 features (+ sex)
    - Length
    - Diameter
    - Height
    - Whole weight
    - Shucked de la chair
    - Viscera des viscères
    - Shell weight
    - *Age (# rings)*
  - VQ with 200 prototypes
  - Reduction from dimension **7** to **2** and visualization of the age (colors)

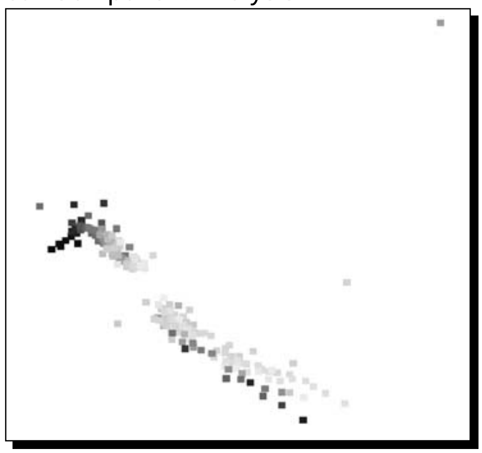
Comparisons: results (1/4)

- Sammon's nonlinear mapping:



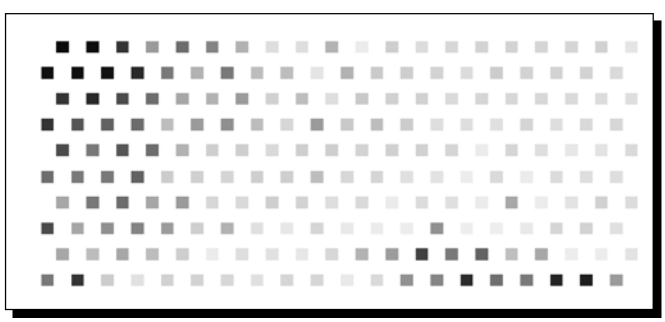
Comparisons: results (2/4)

- Curvilinear Component Analysis:



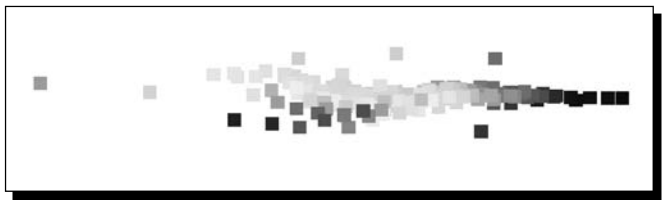
Comparisons: results (3/4)

- Self-organizing map:



Comparisons: results (4/4)

- Isotop:



Comparisons: summary

	Distance preservation	Neighborhood preservation
«Rigid» method	Sammon's mapping (fixed weighting)	Self-Organizing Map (fixed neighborhood)
«Flexible» method	Curv. Comp. Analysis (adaptative weighting)	Isotop (adaptative neighborhoods)

- Warning: model complexity !

Content

- Vector Quantization and Non-Linear Projections
- Limitations of linear methods
  - Principal Component Analysis (PCA)
  - Metric Multi-Dimensional Scaling (MDS)
  - Limitations
- Nonlinear Algorithms
  - Variance preservation
  - Distance preservation (like MDS)
  - Neighborhood preservation (like SOM)
  - Minimal reconstruction error
- Comparisons
- Conclusions

Research directions

- NLP methods
  - Neighborhood decrease in CCA/CDA
- Curvilinear distance (geodesic)
  - Study and implementation
  - Integration in SOM, CCA, Sammon's NLM and Isotop
- Non-Euclidean distances
  - Alternative metrics are considered (L_{inf}, L_1, $L_{0.5}$, etc.)
  - Integration in curvilinear distance, VQ and NLP
- Piecewise linear interpolation
  - Study and implementation
  - Integration in Sammon's NLM, CCA and Isotop
- New algorithm: Isotop

Acknowledgements

- Most of the content of these slides is based on the work of my colleague John Lee