Manifold Learning in Computer Vision – Part 1

Instructor: Hui Wu

IBM Research

Outline

- Dimensionality reduction
- Applications of manifold learning in computer vision

Outline

- **Dimensionality reduction**

- Applications of manifold learning in computer vision

Common High-Dimensional Data Sets

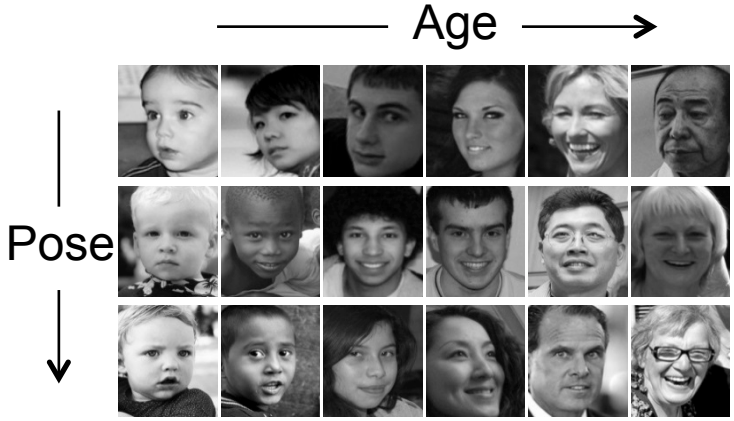
- Number of potential features can be huge
- Image data
  - A 100 x 100 image: 10,000-D
  - Deep features: > 4000-D
- Genomic data
  - Expression levels of the genes: > 1000-D
- Text categorization
  - Frequencies of phrases in a document: > 10,000-D

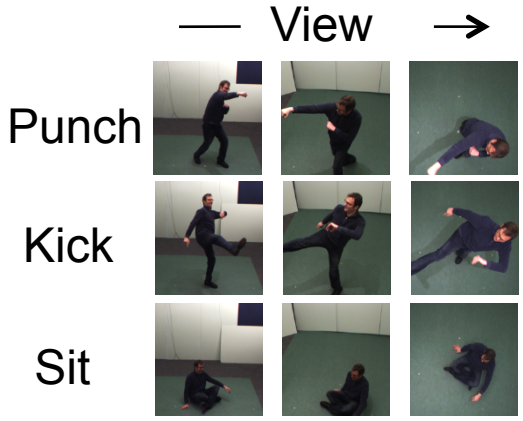
Why Dimensionality Reduction

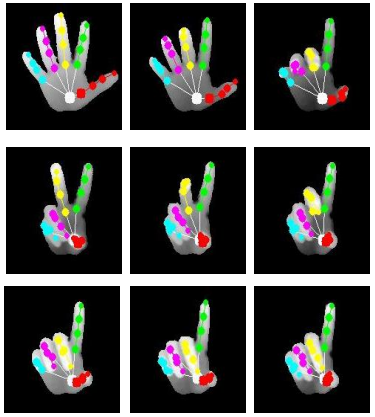
- More features
  - Ideally, more information and higher accuracy
  - Unfortunately, harder to train a good learner
    - Curse of dimensionality
    - Data are highly sparse in the high-D space
    - Needs lots of training data
- One solution: start with as many potentially useful features as possible, then reduce the number of features

Why DR is Possible

- Real data are confined to a region of the high-D space
- Images usually change due to a few variables
  - Lots of redundancy in feature vectors

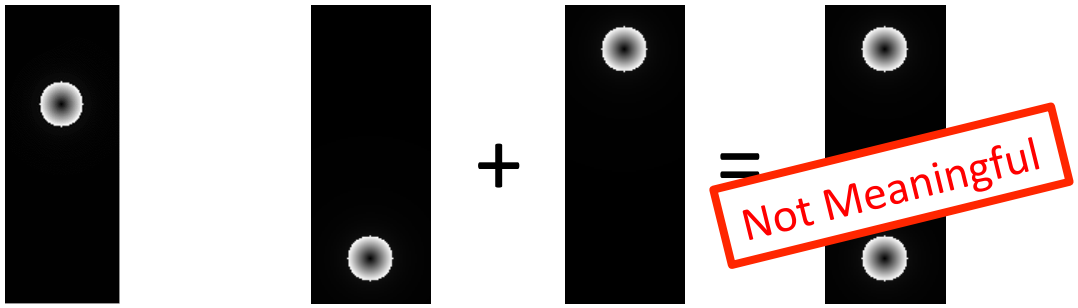


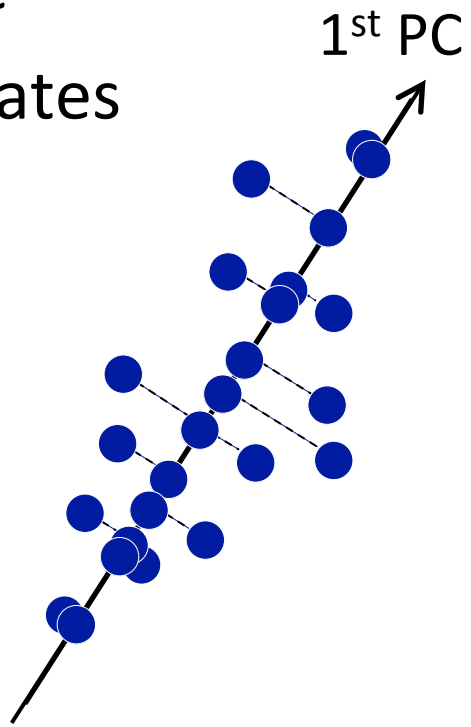




Principal Component Analysis (PCA)

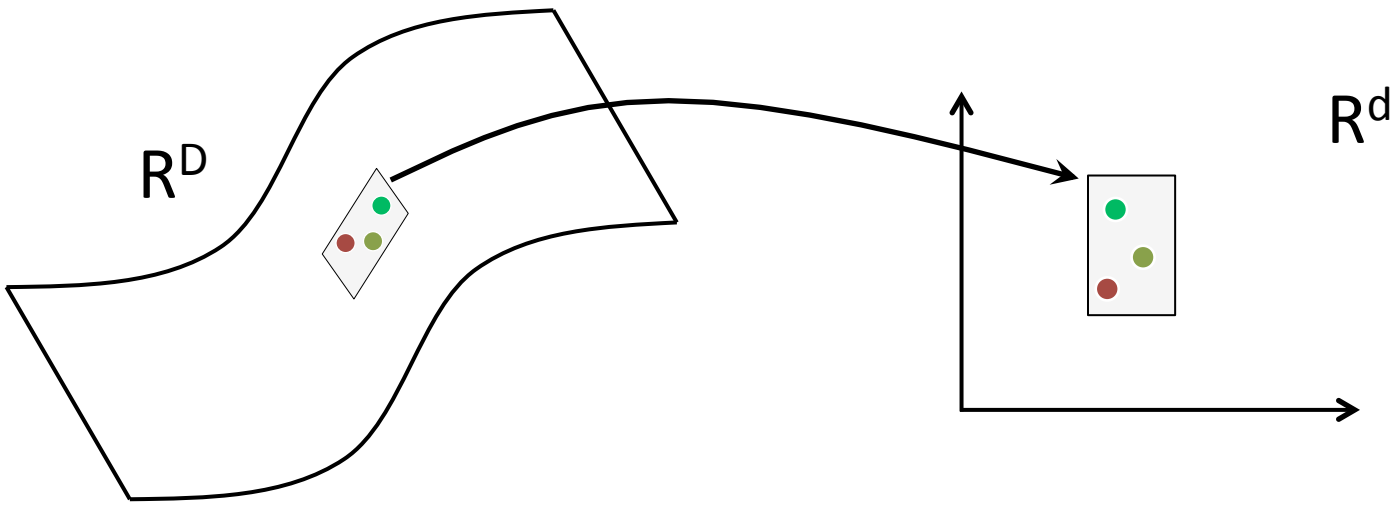
- Steps:
  - Finds the direction of maximum variance
  - Projects points to the linear subspace
  - Obtains the low-dimensional coordinates
- Problem





Manifolds

- A subset of R^D, usually nonlinear
- Locally resembles an Euclidean space, R^d, $d<D$



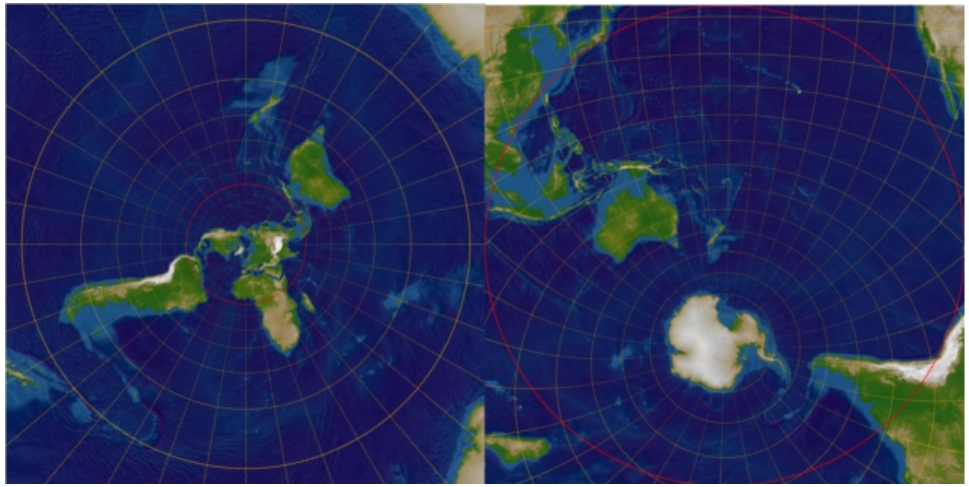
Manifolds

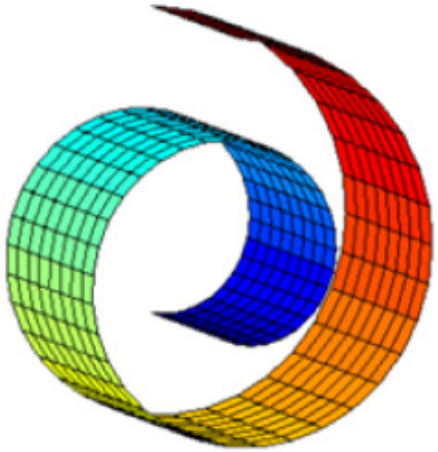
- A subset of R^D, usually nonlinear
- Locally resembles an Euclidean space, R^d, $d<D$

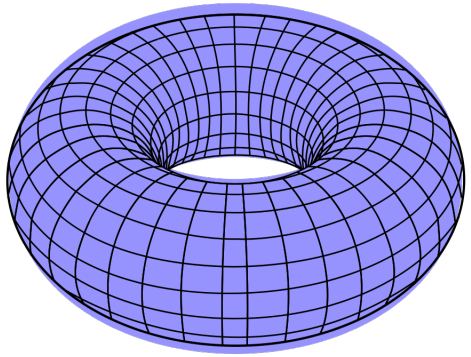
- There may not be such coordinate systems globally
- The coordinate in R^d is called the coordinate of the point on the manifold

Manifold Examples

- Sphere, swiss roll, torus, …







Manifolds for Dimensionality Reduction

- Assumption: data points lie closely to a manifold, M

- If there is a global indexing scheme for M
  - Find **z** on M that is closest to the data **x**
  - Map **z** to its manifold coordinates, **y**
  - Interpret **y** as the low-dimensional representation of **x**

Image Manifold Learning

- There is an underlying manifold, $\mathcal{M}$
- Images are noisy samples from $\mathcal{M}$

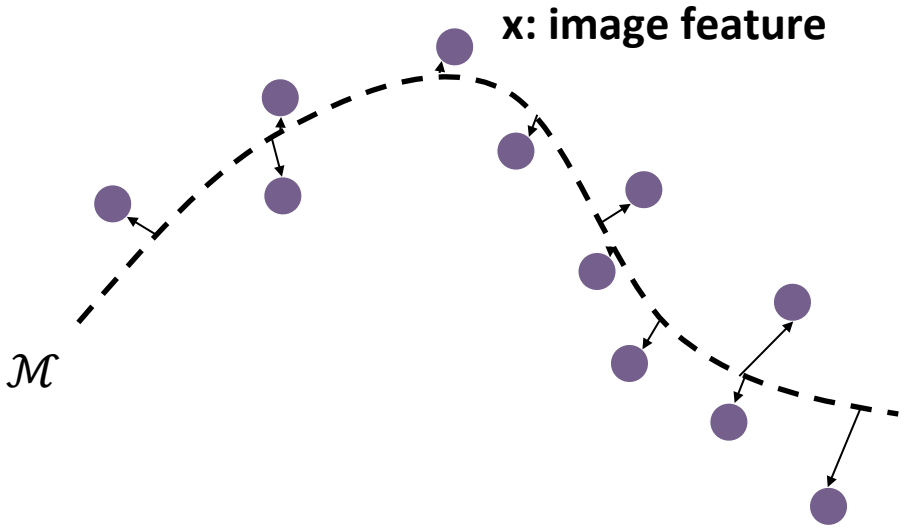


Image Manifold Learning

- There is an underlying manifold, $\mathcal{M}$
- Images are noisy samples from $\mathcal{M}$

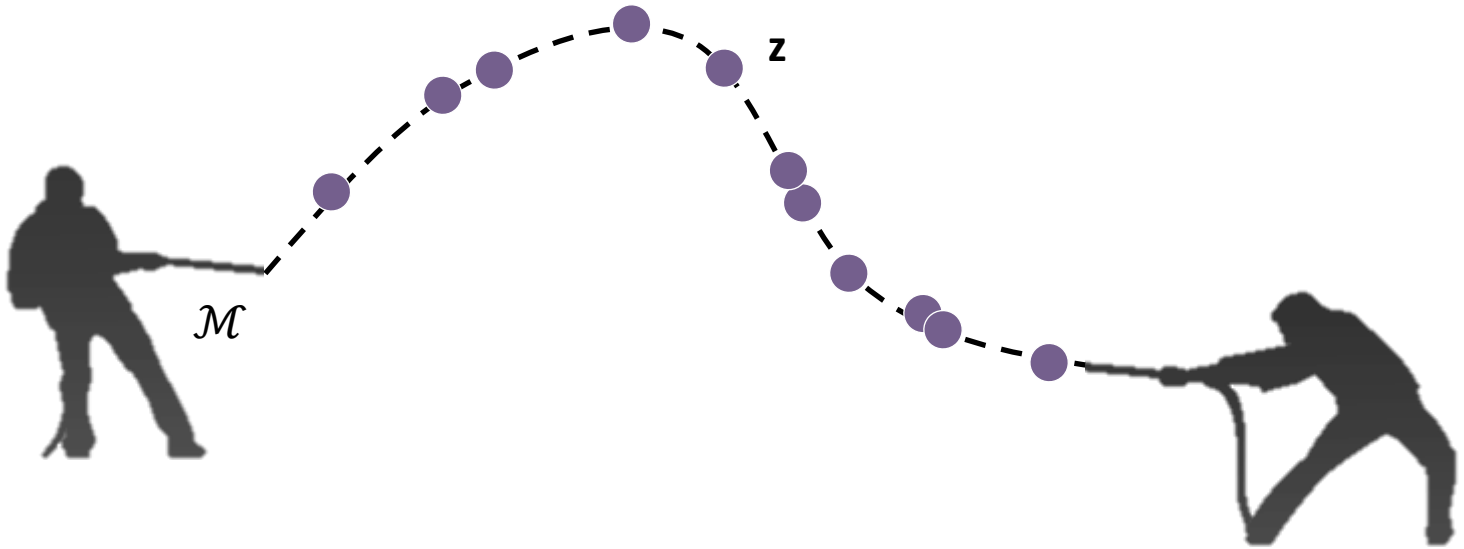
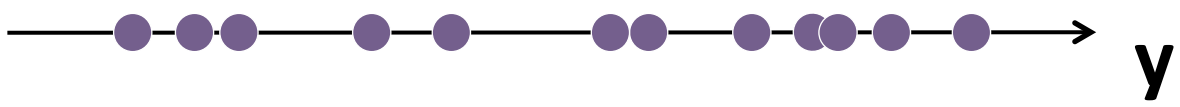


Image Manifold Learning

- There is an underlying manifold, $\mathcal{M}$
- Images are noisy samples from $\mathcal{M}$



Taxonomy of Nonlinear Dimensionality Reduction Algorithms

- Is there any explicit notion of manifold?
- Parametric or non-parametric?
- Can the algorithm take a dissimilarity (or similarity) matrix as input?
- Is local neighborhood in R^D used?

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit Manifold					
Parametric					
Dissimilarity matrix					
Local neighborhood					

Autoencoder

- Given: a data set $\mathbf{x}=\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in R^D

- Train a neural network
  - input = target output = **X**
  - A "middle" layer of *d* units
  - Loss function: sum of squares error

- The low-D representation can be extracted from the node values of the middle hidden units

Autoencoder

Input: $\mathbf{x}_i$

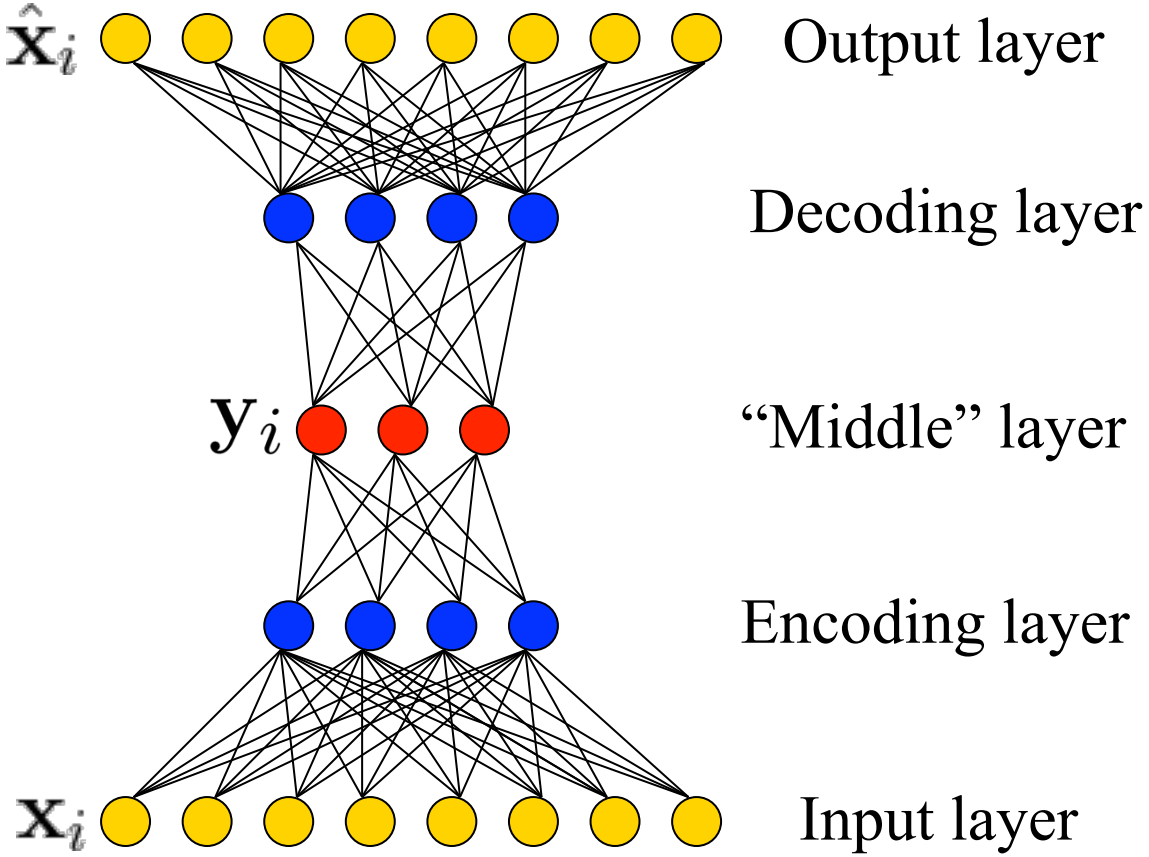
Target output: $\mathbf{x}_i$

Actual output: $\hat{\mathbf{x}}_i$

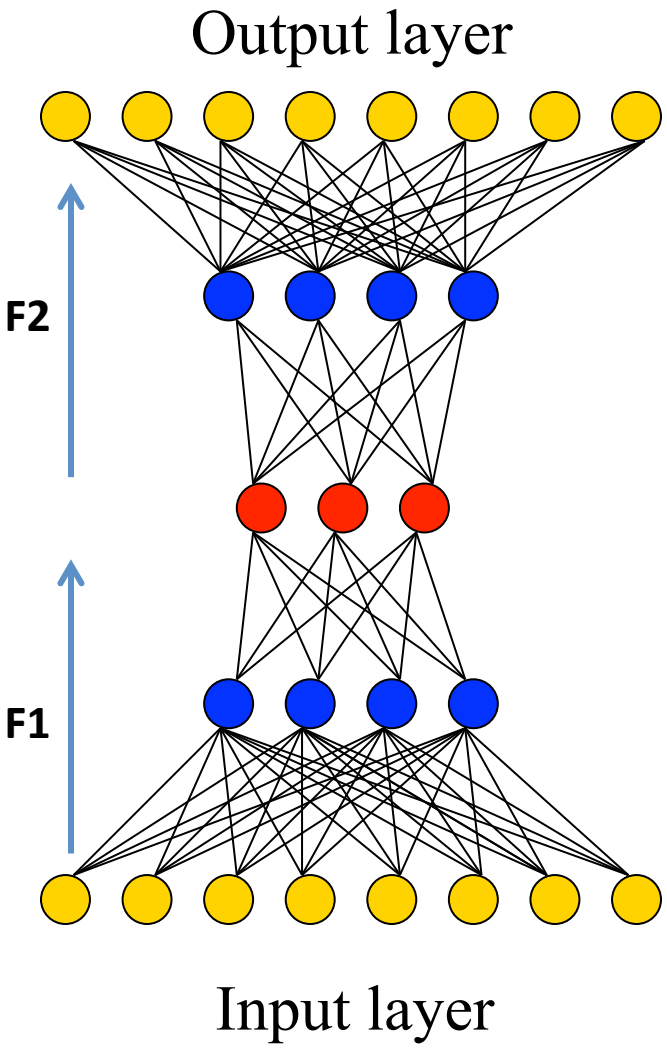
Find weight that minimize

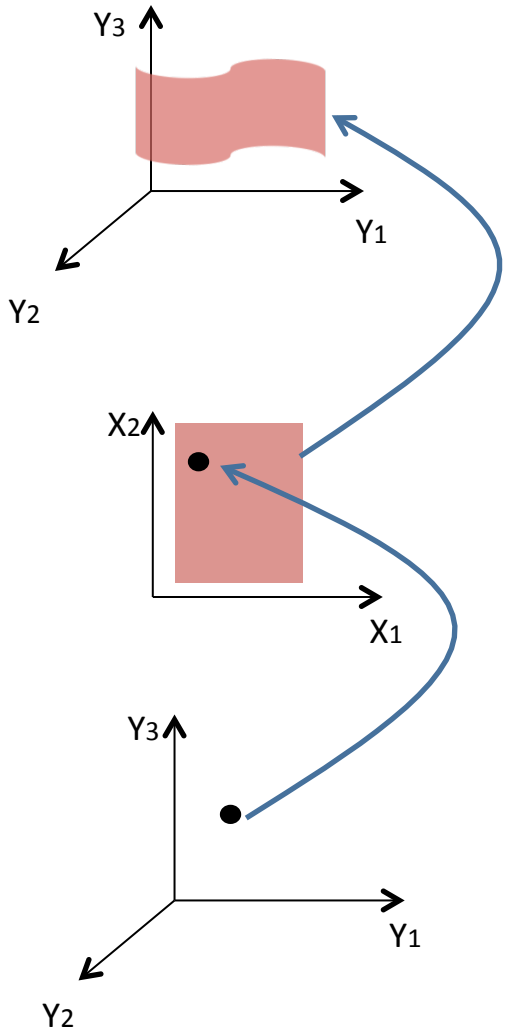
$$\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$$

Transformed data: $\mathbf{y}_i$



Autoencoder



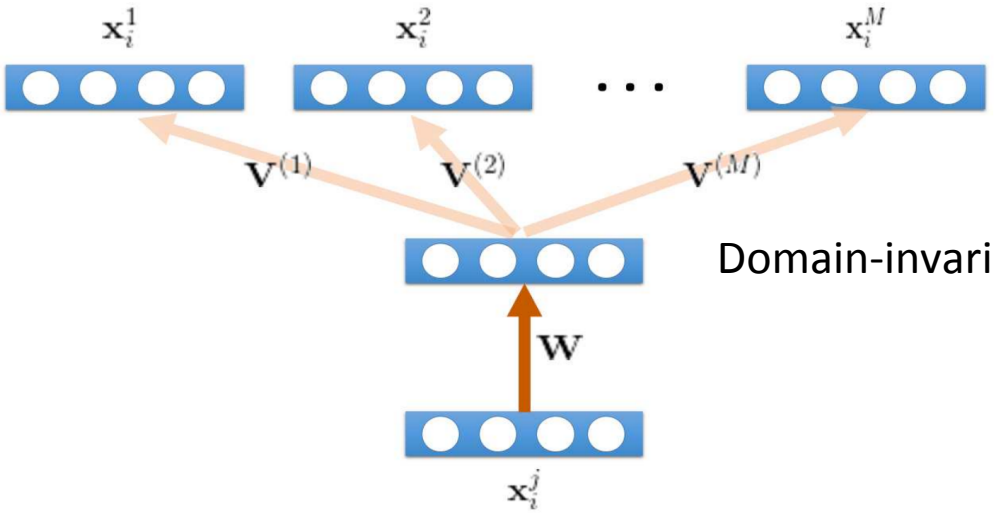


F2
Defines how the low-dimensional space can be embedded in the original space.

F1
Defines a projection of high-dimensional points onto the lower-dimensional subspace.

Autoencoder

- Intuition: $\mathbf{y}_i$ is a "compressed" version of $\mathbf{x}_i$ as it can reconstruct $\mathbf{x}_i$ approximately
- What if the transfer function is linear?
  - One hidden layer and linear activation is used
- Domain generalization: transfer original images to analogs in multiple domains



Domain-invariant representation

Domain Generalization for Object Recognition with Multi-task Autoencoders, ICCV 2015

Summary of the Algorithms

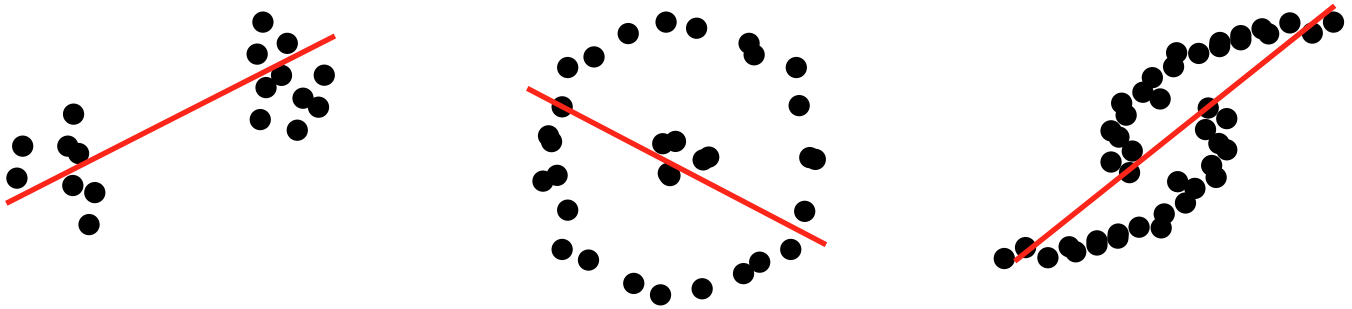
	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold					
Parametric					
Dissimilarity matrix					
Local neighborhood					

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold	No				
Parametric	Yes				
Dissimilarity matrix	No				
Local neighborhood	No				

Kernel PCA (KPCA)

- PCA: finds the principal component of maximum variance
- What happens to these cases …



- Kernel PCA
  - Advantage: can represent nonlinear structures

PCA Steps

$$\underset{D \times n}{X} = \underset{D \times d}{U} \quad \underset{d \times d}{\Sigma} \quad \underset{d \times n}{V^T}$$

- Low-D representation: $Y = U^T X$
- Notice that:
  - U is the eigenvector of covariance matrix: XX^T
  - V is the eigenvector of inner product matrix: $X^T X$
  - $U^T X = \Sigma V^T$

Kernel PCA

Original data space $\xrightarrow{\phi(\mathrm{x})}$ Feature space

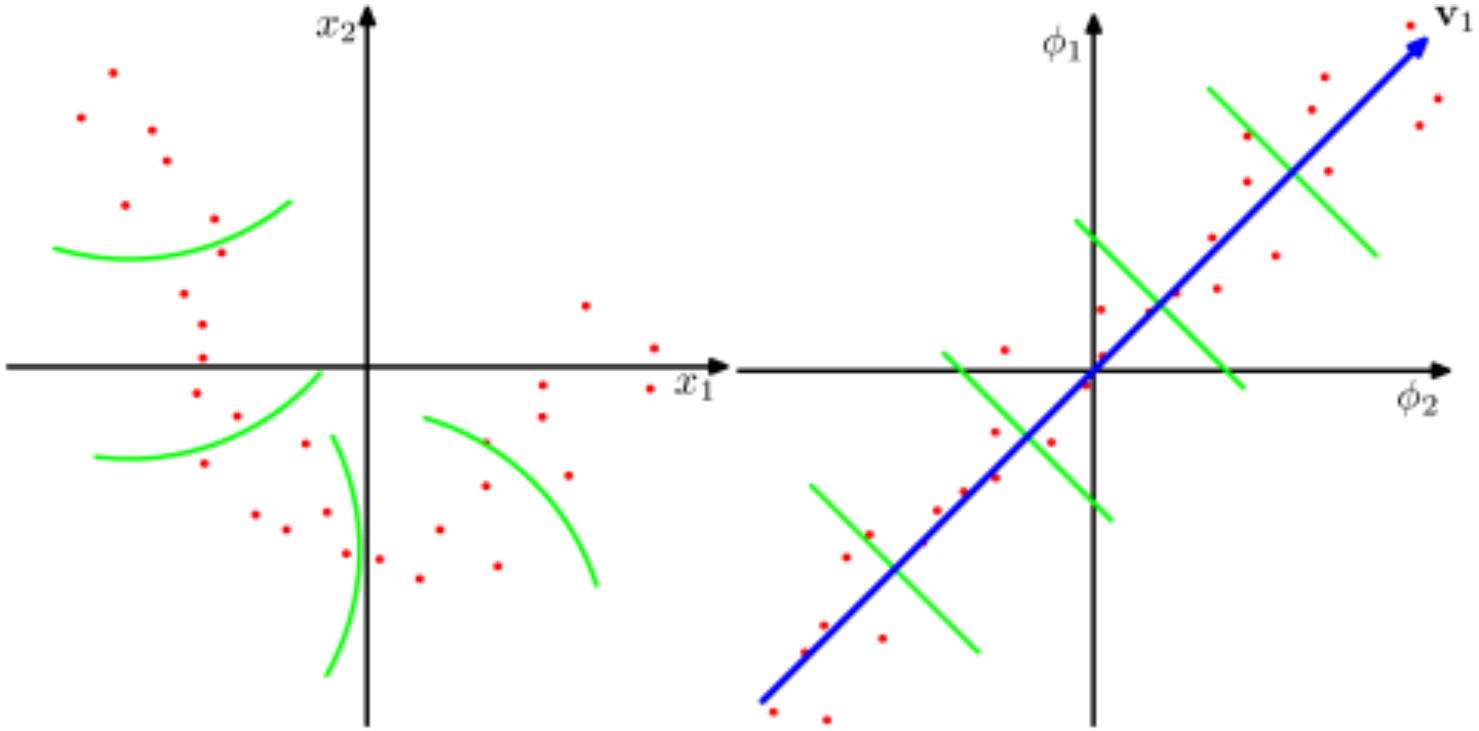


Figure from Pattern Recognition and Machine Learning, Bishop 2006

Kernel PCA

- $\mathbf{x} \rightarrow \phi(\mathbf{x})$
  - We may not know the exact form of $\phi(\cdot)$
  - But we are given the kernel matrix K
    - $K_{i,j} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$: inner product in feature space

- If we perform eigendecomposition on K
  - We can obtain V and $\sum$
  - So the low-D representation is:

$$Y = U^T \phi(X) = \Sigma V^T$$

Steps for Kernel PCA

1. Calculate the kernel matrix $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$
2. Center data in feature space

$$\mathbf{K} - \mathbf{K}\mathbf{1}_n - \mathbf{1}_n\mathbf{K} + \mathbf{1}_n\mathbf{K}\mathbf{1}_n$$

with $(\mathbf{1}_n)_{i,j} = \frac{1}{n}$

3. Compute the first d eigenvectors of the kernel matrix, denote as V
4. Project the high-D data, $Y = \Sigma V^T$
5. Project new data point, $y_{new} = \Sigma^{-1} V^T K(X, x_{new})$

Kernel PCA: Example

- The principal components in the feature space can be visualized by a contour plot

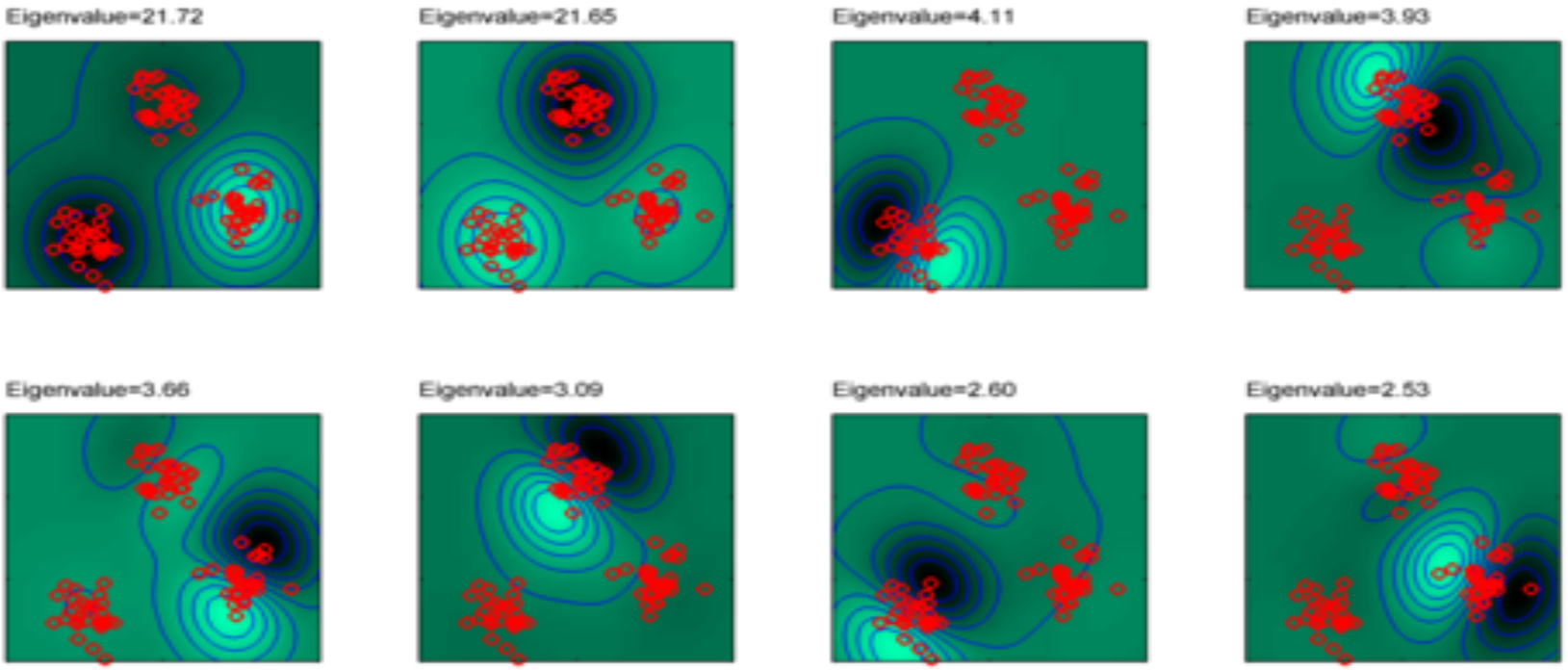
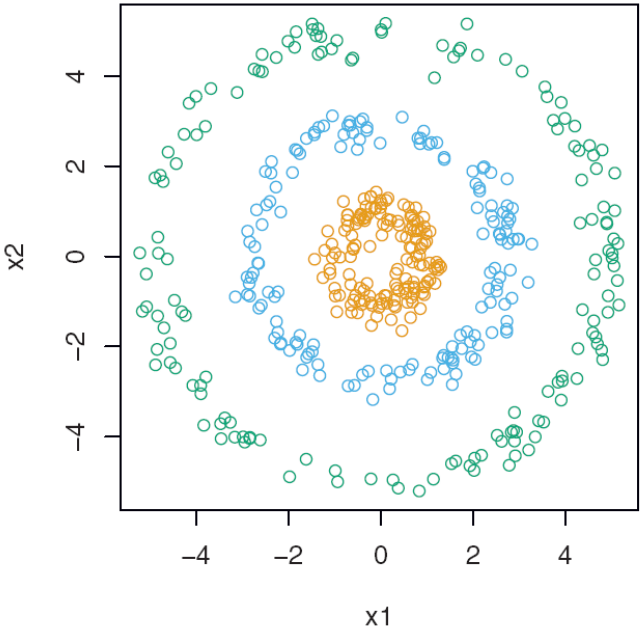


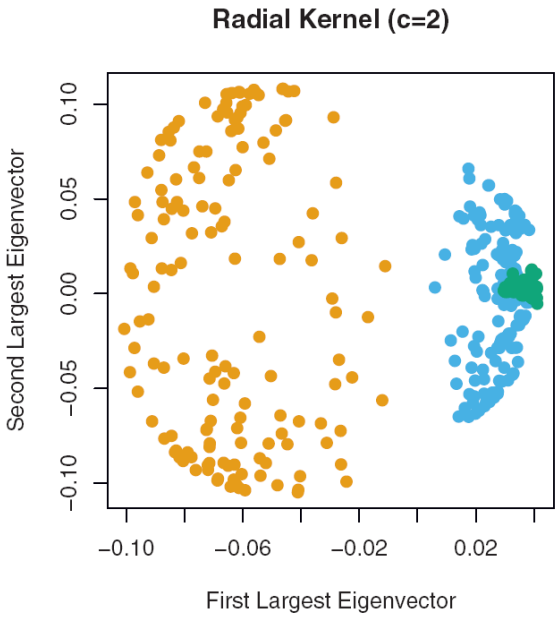
Figure from Pattern Recognition and Machine Learning, Bishop 2006

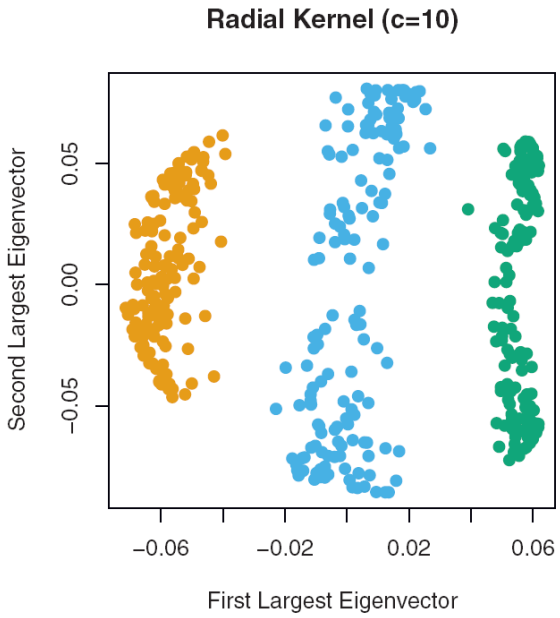
KPCA: Example

Original feature space



Transformed space





$$K(x, x') = \exp(-\|x - x'\|^2/c)$$

Figure from Elements of Statistical Learning, Hastie et al.

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold	No				
Parametric	Yes				
Dissimilarity matrix	No				
Local neighborhood	No				

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold	No	No			
Parametric	Yes	Yes			
Dissimilarity matrix	No	Yes			
Local neighborhood	No	No			

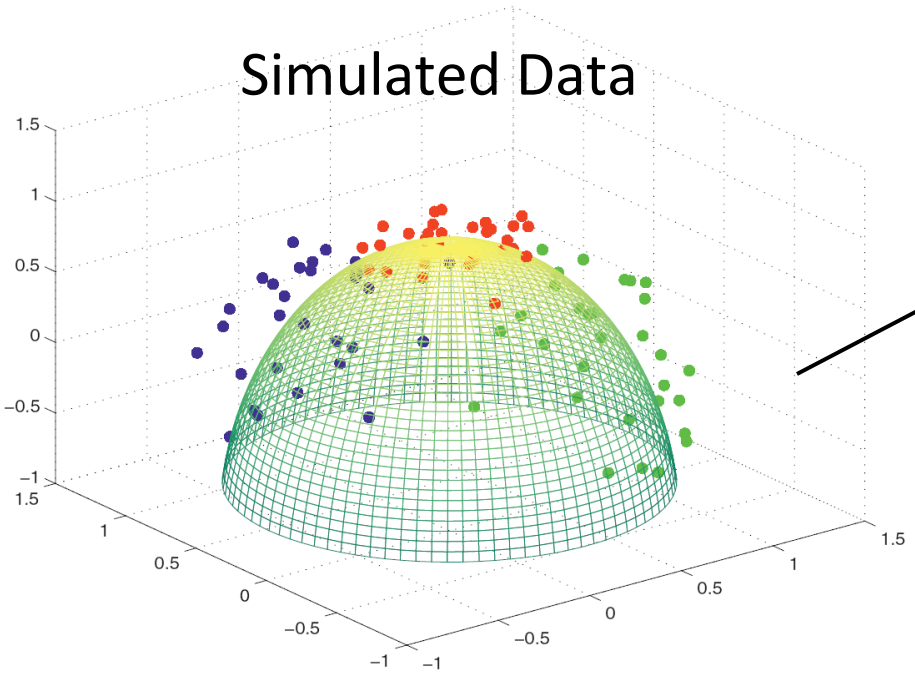
Multidimensional Scaling (MDS)

- Input: Pairwise distances, D
- Output: Point positions whose pairwise distance match D

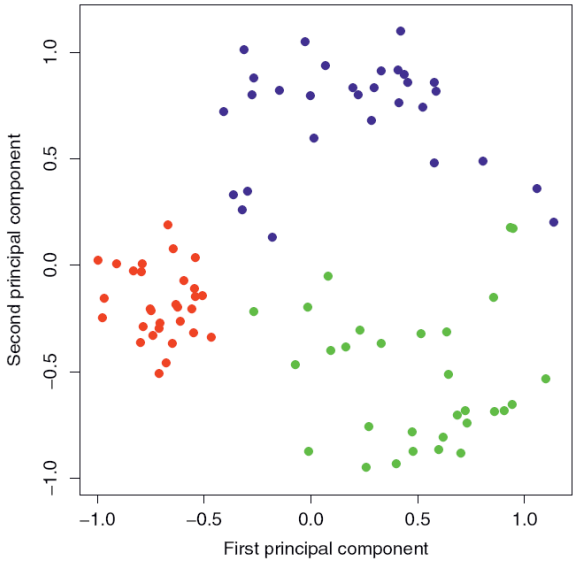
	Chi	NY	LA
Chi	0	719	1749
NY	719	0	2462
LA	1749	2462	0

Major Cities

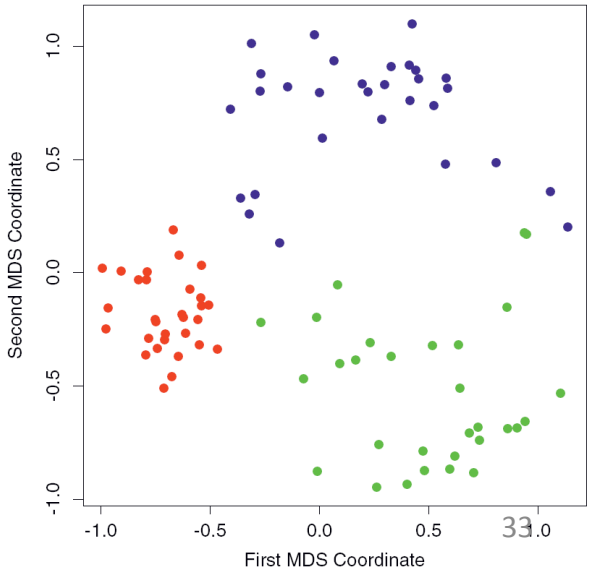
Example: MDS Vs. PCA

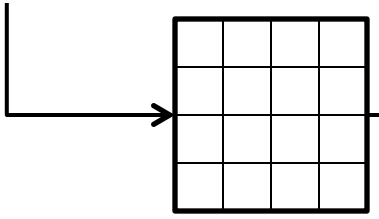


PCA



MDS





Distance Matrix

Figures from Elements of Statistical Learning, Hastie et al.

Notes on MDS

- MDS on points with Euclidean distances (from high dimensional space), gives the SAME embedding as PCA

- No explicit mapping
  - If a new item comes, we do not know where it should be mapped to

- Lots of extensions of MDS since it takes distance matrix as input. (We will see one soon)

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold	No	No			
Parametric	Yes	Yes			
Dissimilarity matrix	No	Yes			
Local neighborhood	No	No			

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit manifold	No	No	No		
Parametric	Yes	Yes	No		
Dissimilarity matrix	No	Yes	Yes		
Local neighborhood	No	No	No		

Isomap & LLE

- Most famous manifold learning algorithms
- Both published in Science 2000

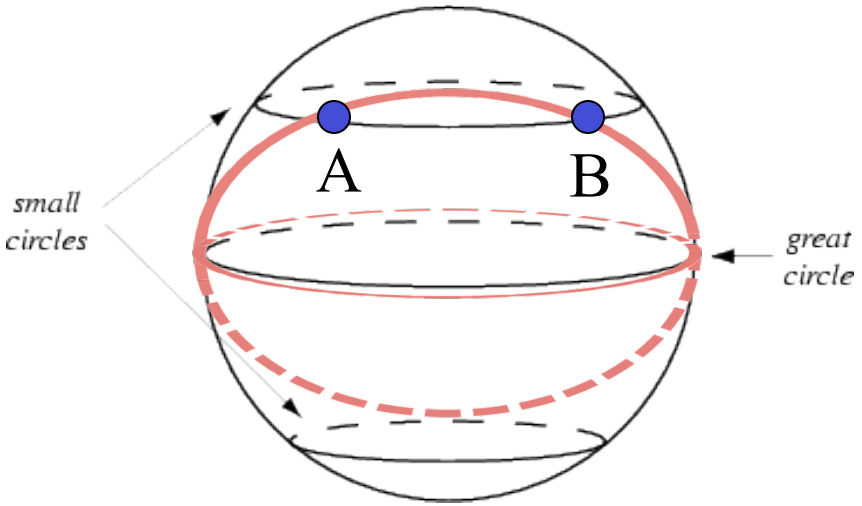
[PDF] A Global Geometric Framework for Nonlinear Dimensionality
wearables.cc.gatech.edu/paper_of_week/**isomap**.pdf ▾
by JB Tenenbaum - 2000 - Cited by 9038 - Related articles

[PDF] Nonlinear Dimensionality Reduction by Locally Linear ...
https://www.cs.cmu.edu/.../roweis-science-00.... ▾ Carnegie Mellon University ▾
by ST Roweis - 1993 - Cited by 9507 - Related articles

- Very similar in lots of aspects
  - Explicitly use the concept of manifold
  - Stimulated a lot research in this area

Geodesic Distance

- Geodesic: the shortest curve that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- Geodesic distance: length of the geodesic



Geodesic Distance

- Euclidean distance may not be a good measure between two points on a manifold
  - Length of geodesic is more appropriate
- Example: Swiss roll

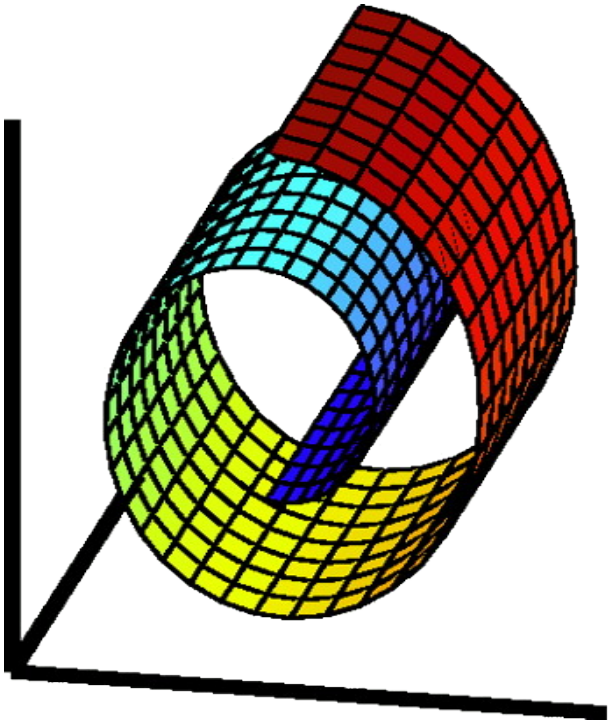


Figure from LLE paper

Isometric Feature Mapping (ISOMAP)

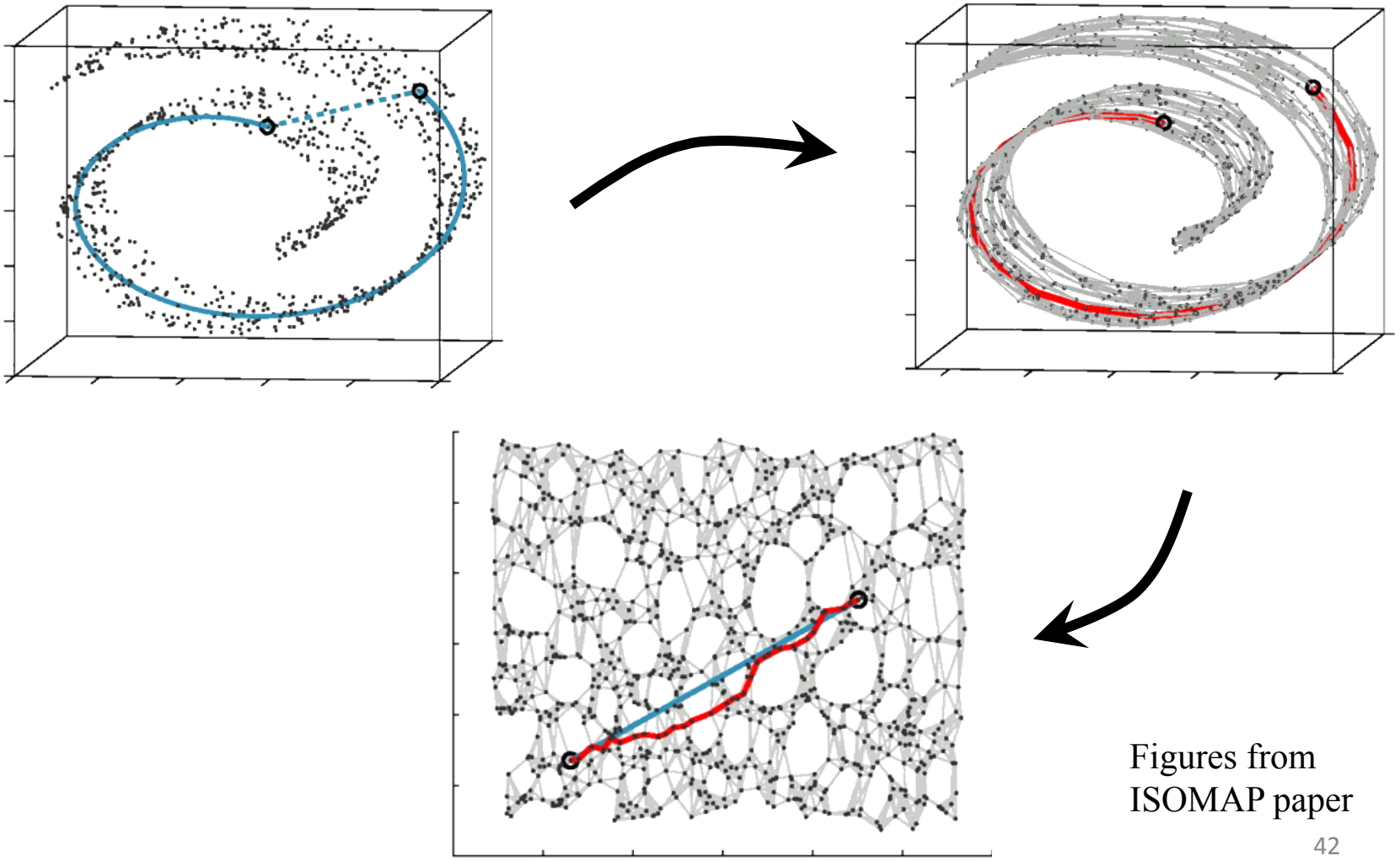
- Take a distance matrix $\{\gamma_{ij}\}$ as input
- Estimate geodesic distance between any two points by "a chain of short paths"
- Perform MDS on the matrix of geodesic distances to obtain final projection

Steps to Estimate Geodesic Distances

1. Find the "neighbors" of all data items $\mathbf{x}_i$
2. Construct a weighted undirected graph
   - Vertex *i* corresponds to $\mathbf{x}_i$
   - An edge between the vertex *i* and *j* if $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors, and its weight is γ_{ij}
3. Find the shortest distance between all pairs of vertices in the graph
   - Floyd or Dijkstra

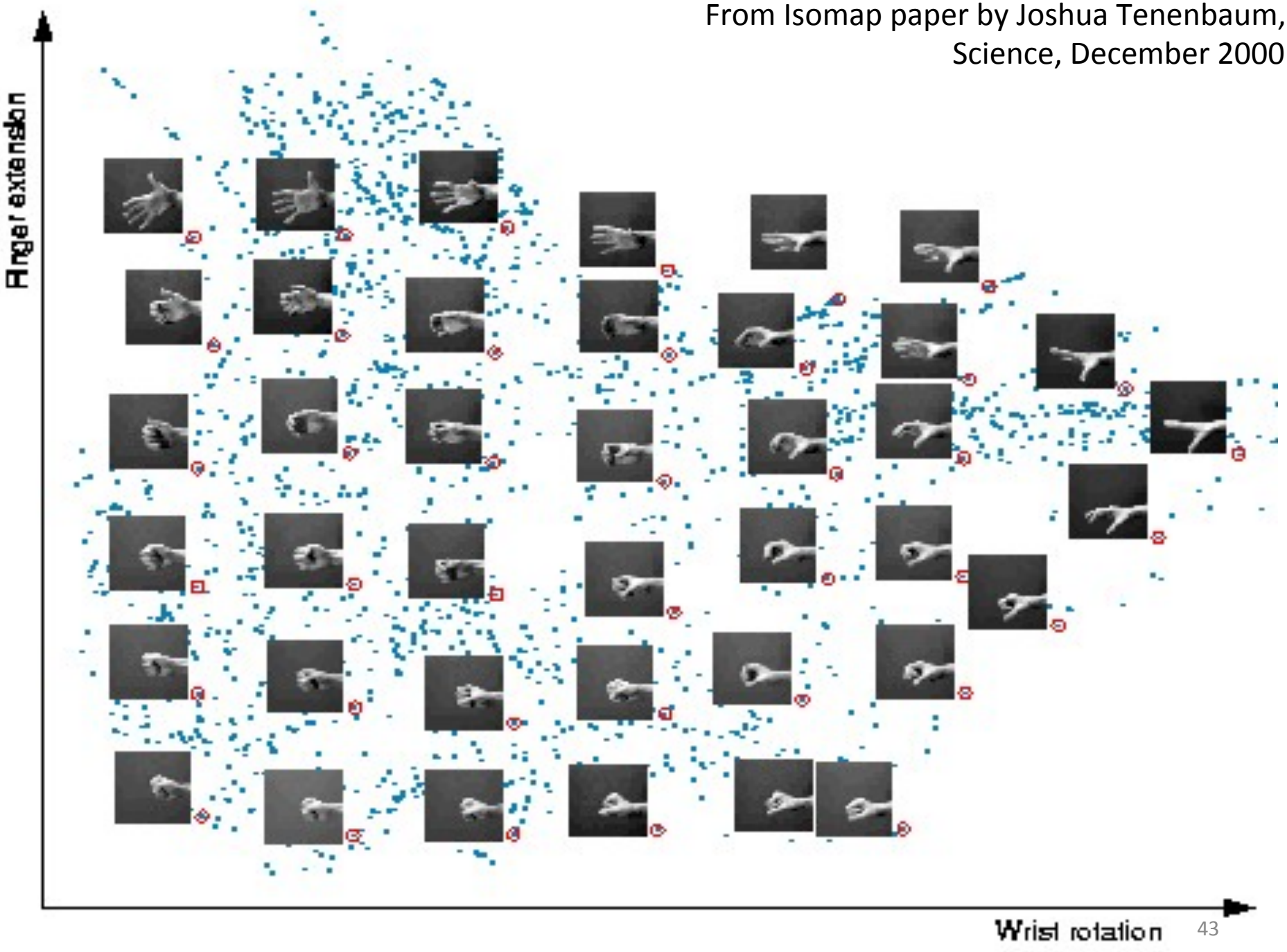
- The shortest distance between vertices *i* and *j* in the graph is the estimated geodesic distance between $\mathbf{x}_i$ and $\mathbf{x}_j$

Rationale for the Geodesic Distance Estimation



Figures from ISOMAP paper

From Isomap paper by Joshua Tenenbaum,
Science, December 2000



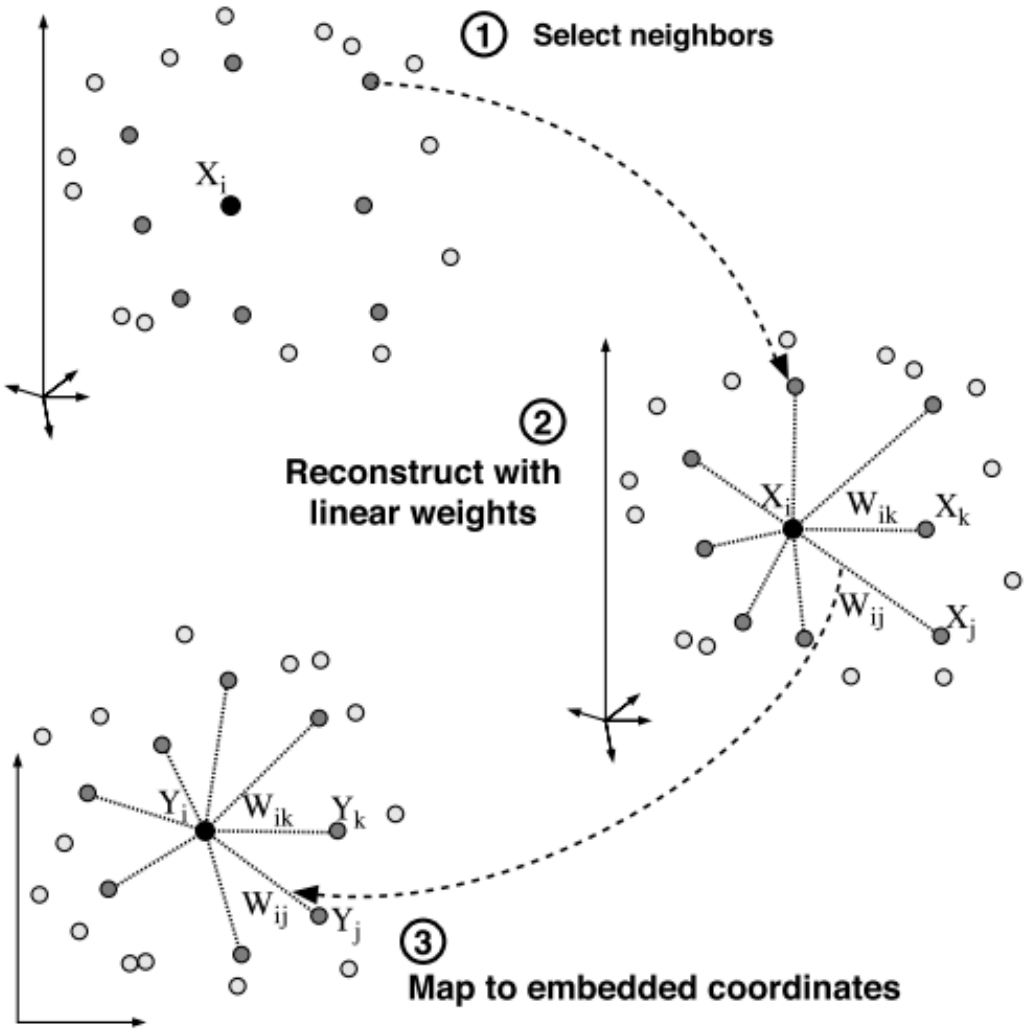
Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit Manifold	No	No	No		
Parametric	Yes	Yes	No		
Dissimilarity matrix	No	Yes	Yes		
Local neighborhood	No	No	No		

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit Manifold	No	No	No	Yes	
Parametric	Yes	Yes	No	No	
Dissimilarity matrix	No	Yes	Yes	Yes	
Local neighborhood	No	No	No	Yes	

Locally Linear Embedding (LLE)



LLE Steps

- For each data point, , $\mathbf{x}_i$, find K nearest neighbors
- Represent each point as a weighted sum of its neighbors $\mathcal{N}(i)$ by minimizing the reconstruction error

$$E(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

Reconstruction weights

- Find the points $\mathbf{y}_i$ in the low-D space based on the reconstruction weights by minimizing

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

LLE

2000
20 x 28
grayscale
images

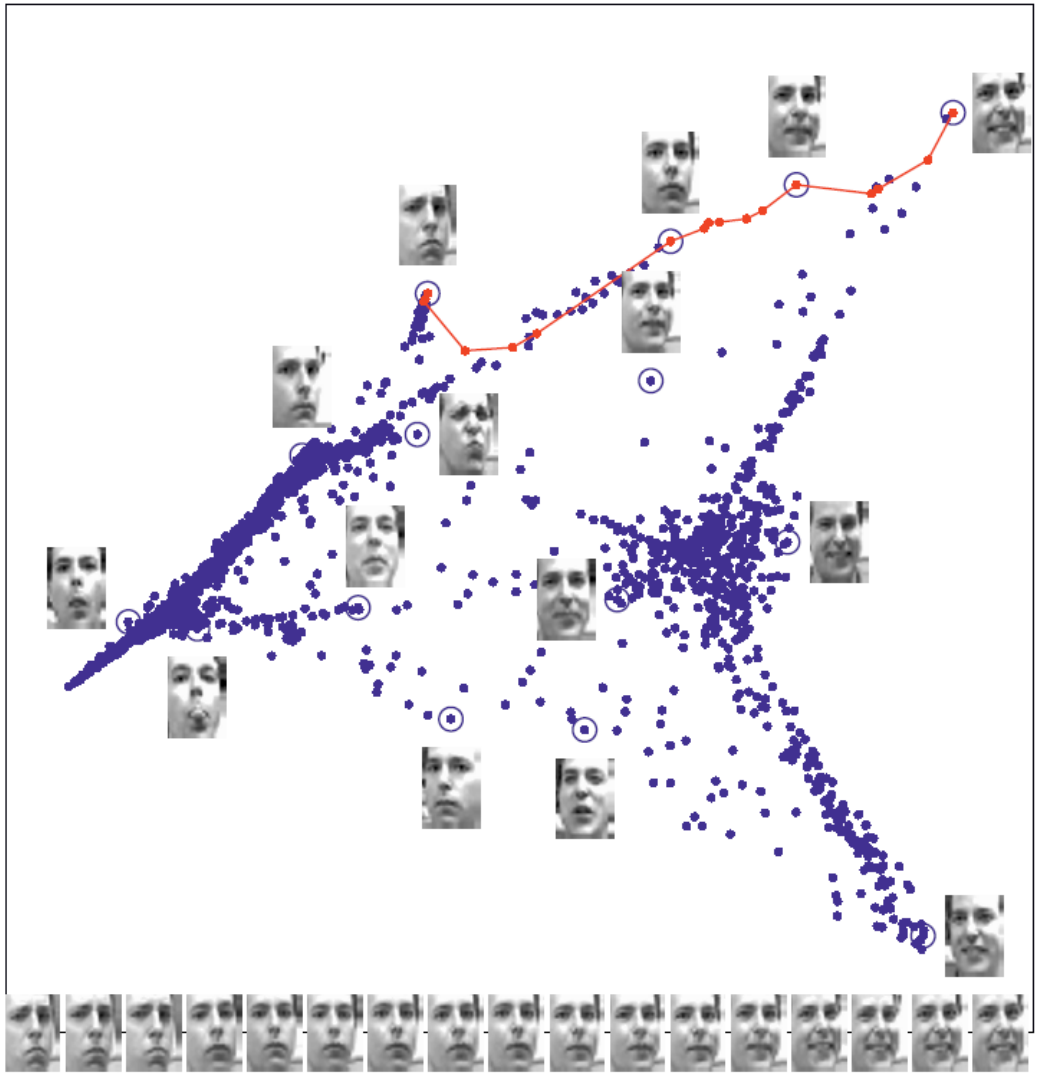


Figure from Elements of Statistical Learning, Hastie et al.

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit Manifold	No	No	No	Yes	
Parametric	Yes	Yes	No	No	
Dissimilarity matrix	No	Yes	Yes	Yes	
Local neighborhood	No	No	No	Yes	

Summary of the Algorithms

	Auto-encoder	KPCA	MDS	ISOMAP	LLE
Explicit Manifold	No	No	No	Yes	Yes
Parametric	Yes	Yes	No	No	No
Dissimilarity matrix	No	Yes	Yes	Yes	Yes
Local neighborhood	No	No	No	Yes	Yes

Recap on Isomap & LLE

- Both methods focus on the notion of manifold, in particular the local property
  - Impact: stimulated a lot of research in this area (we will see some examples soon)

- Some limitations
  - Require dense sampling of the manifold
  - Nonparametric: can not directly apply to new data points

Outline

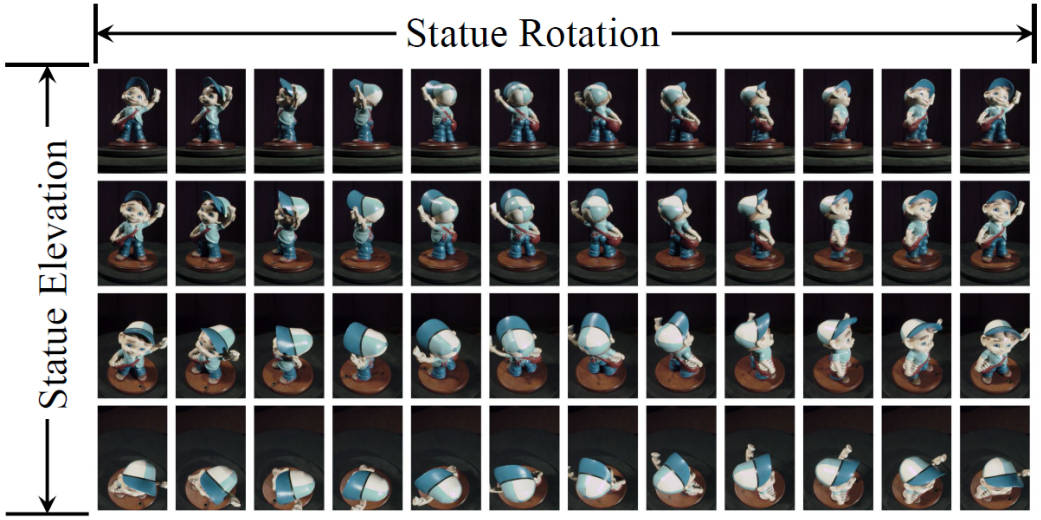
- Dimensionality reduction

- Applications of manifold learning on computer vision
  - A. Semi-supervised Regression on Manifolds

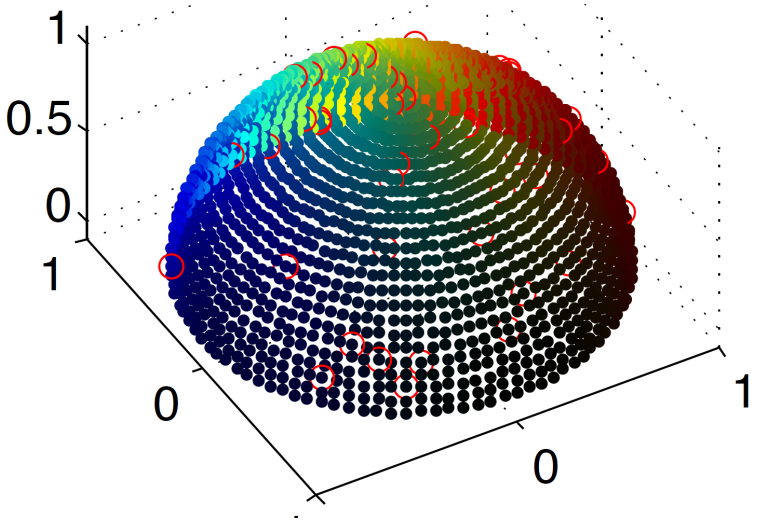
Semi-supervised Regression on Manifolds

- Input: data points (on a manifold), partial labels
- Output: labels on all data points

Images (sampled from an manifold)

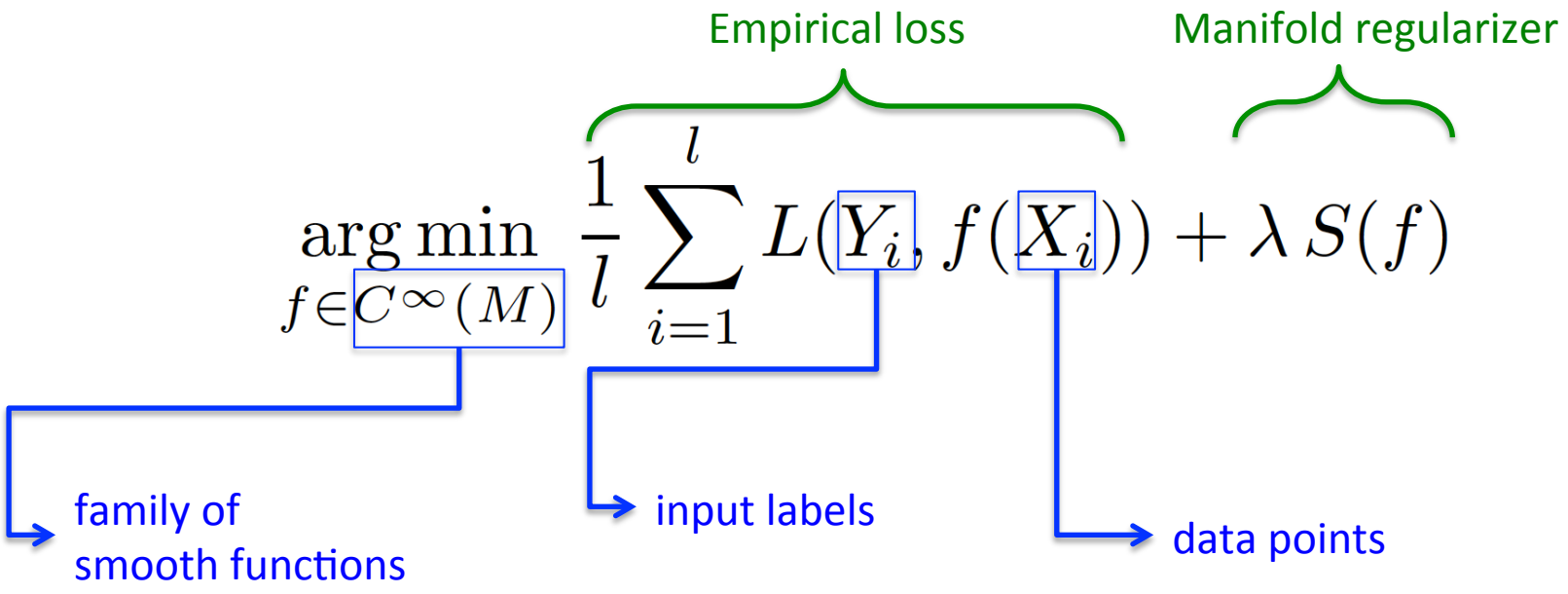


Partial view labels



Semi-supervised Regression on Manifolds

- Input: data points (on a manifold), partial labels
- Output: labels on all data points
- A common framework of semi-supervised regression on manifolds:



Semi-supervised Regression on Manifolds

- Different choices are available for the manifold regularizer
- Hessian regularizer is used here
  - Involves local PCA computation
  - The Hessian matrix: B, N*N PSD matrix
  - Minimizing $f^T B f$ results in locally linear labels on the manifold
- This leaves us:

$$\underset{f \in \mathbb{R}^n}{\arg\min} \underbrace{\frac{1}{l} \sum_{i=1}^{l} (Y_i - f(X_i))^2}_{\text{L2 loss}} + \lambda \underbrace{\langle f, Bf \rangle}_{\text{Hessian energy}}$$

Semi-supervised Regression using Hessian Energy, NIPS 2009

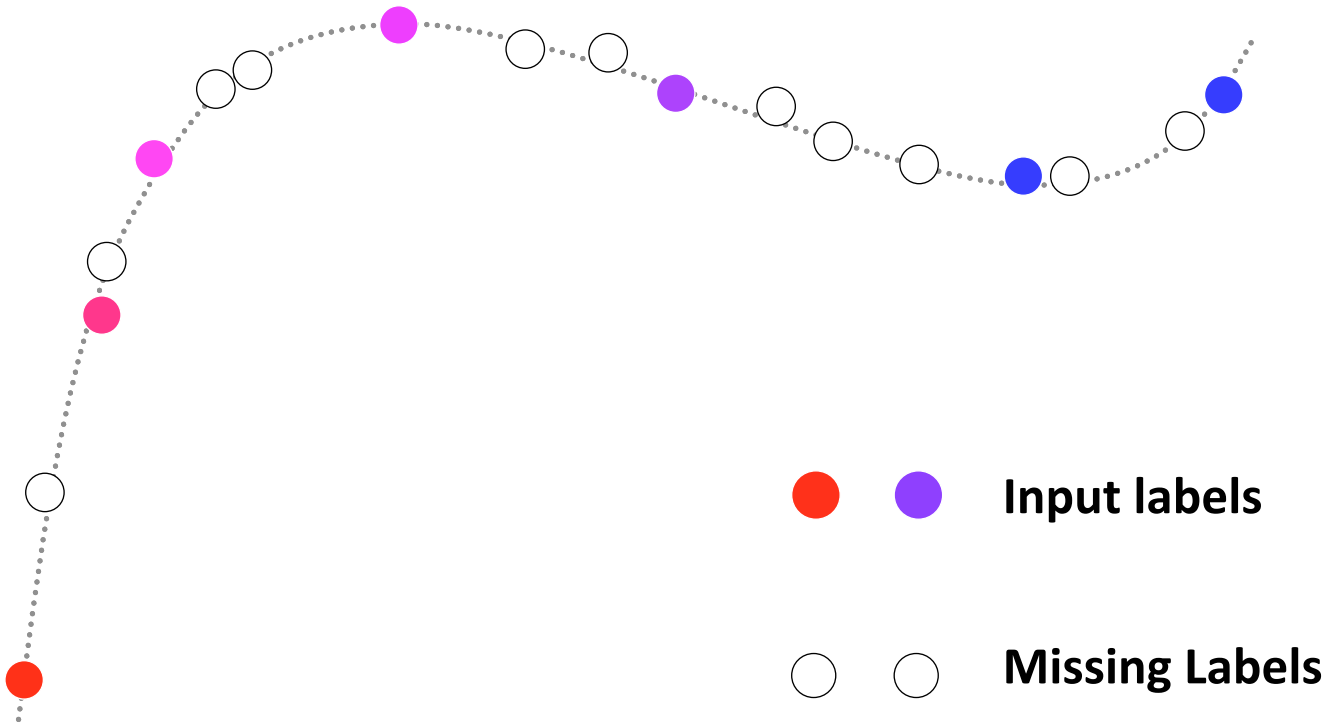
Semi-supervised Regression on Manifolds

- Different choices are available for the manifold regularizer
- Hessian regularizer is used here
  - Involves local PCA computation
  - The Hessian matrix: B, N*N PSD matrix
  - Minimizing $f^T B f$ results in locally linear labels on the manifold
- This leaves us:

$$\arg\min_{f \in \mathbb{R}^n} \frac{1}{l} \sum_{i=1}^{l} (Y_i - f(X_i))^2 + \lambda \langle f, Bf \rangle$$

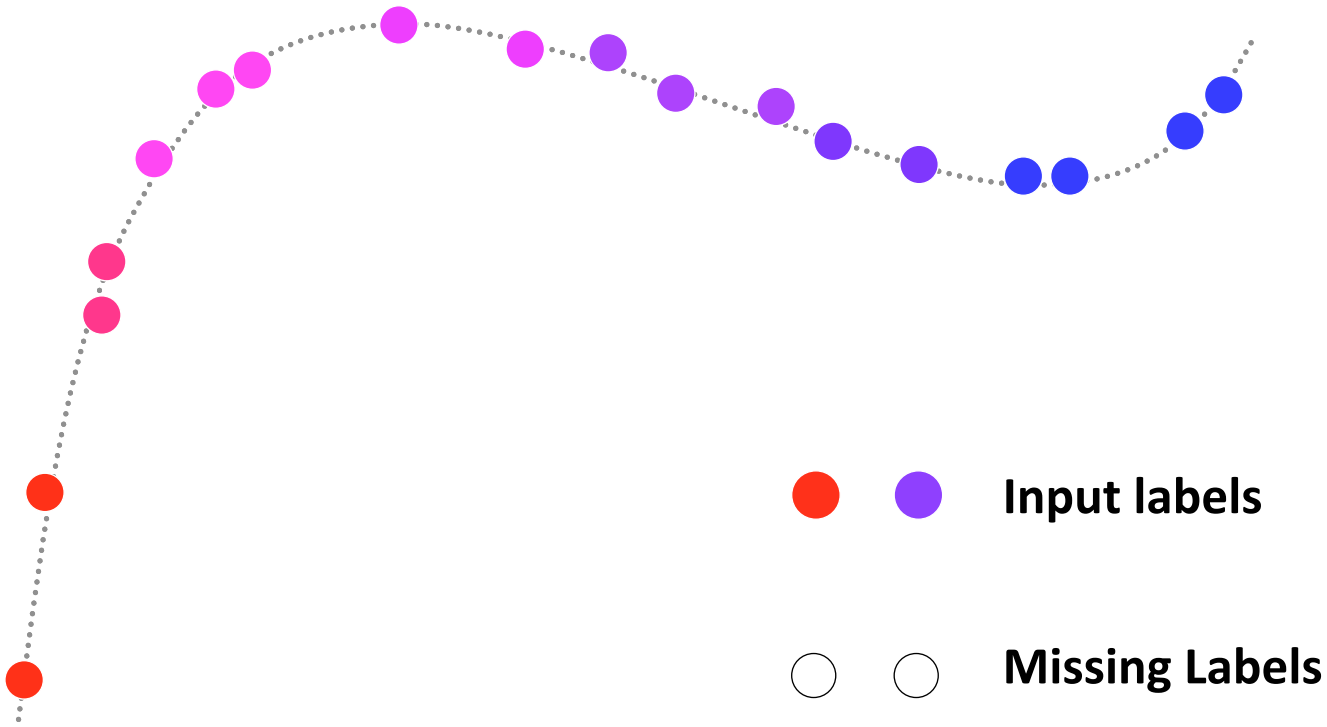
$$\Rightarrow (\mathbb{I}' + l\,\lambda B) f = Y$$ (Sparse linear system)

Semi-supervised Regression on Manifolds



Semi-supervised Regression using Hessian Energy, NIPS 2009

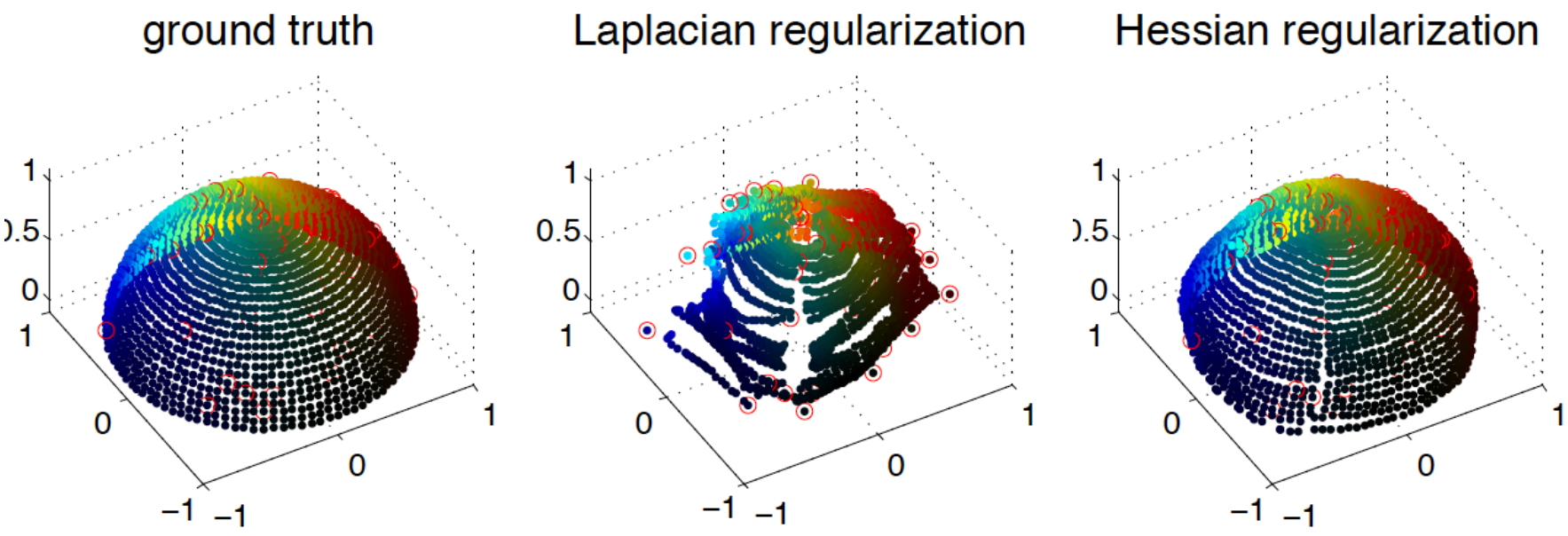
Semi-supervised Regression on Manifolds



Semi-supervised Regression using Hessian Energy, NIPS 2009

Semi-supervised Regression on Manifolds

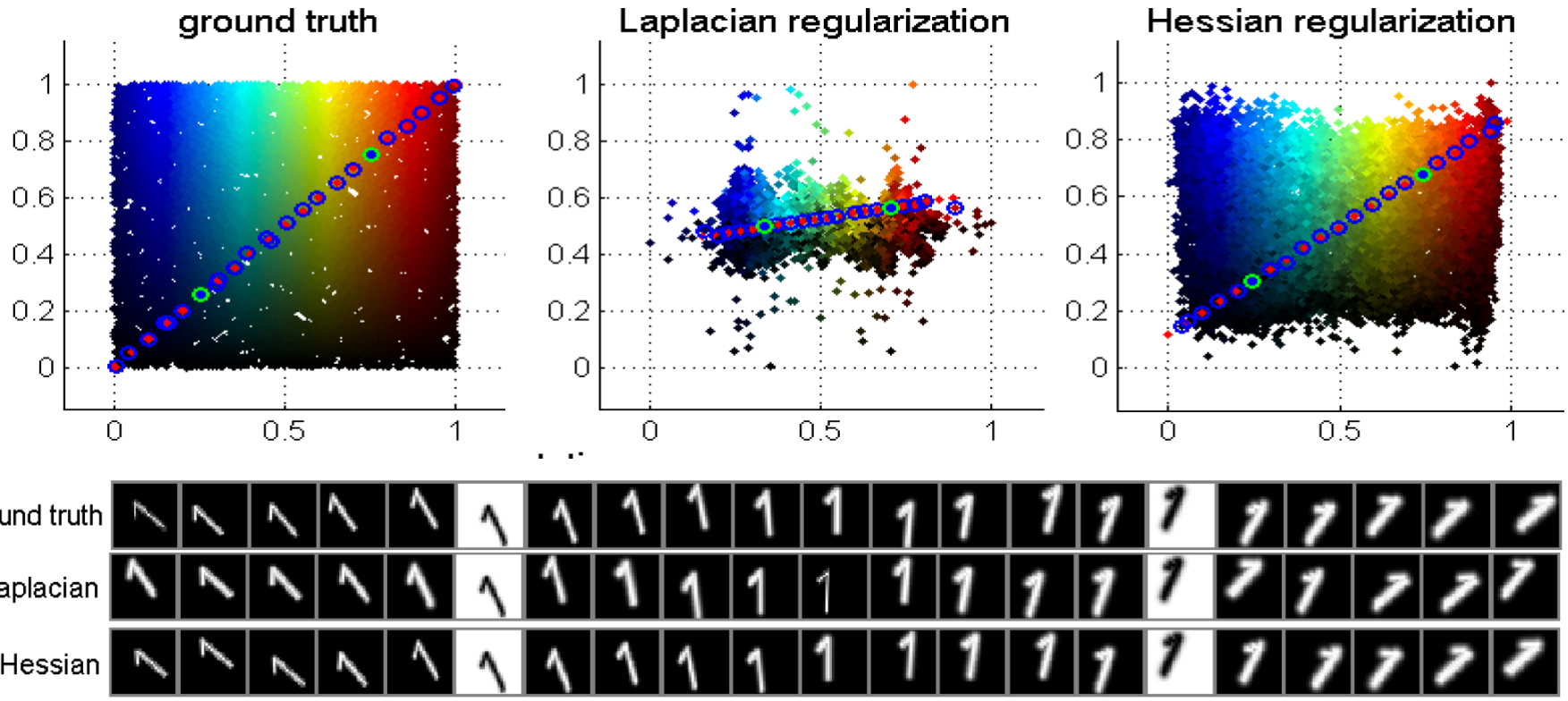
- Image viewpoint estimation



Semi-supervised Regression using Hessian Energy, NIPS 2009

Semi-supervised Regression on Manifolds

- 10,000 input images, 100 labels



Semi-supervised Regression using Hessian Energy, NIPS 2009

Outline

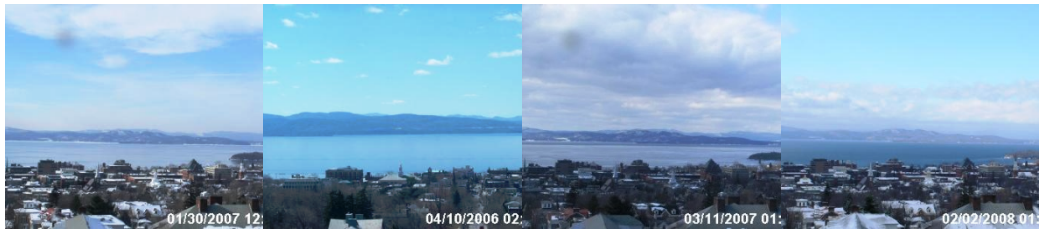
- Dimensionality reduction

- Applications of manifold learning on computer vision
  - A. Semi-supervised Regression on Manifolds
  - B. Robust Manifold Regression for Image Label Denoising

Robust Manifold Regression for Image Label Denoising

- Automated image annotation
  - Efficient but less accurate

AMOS



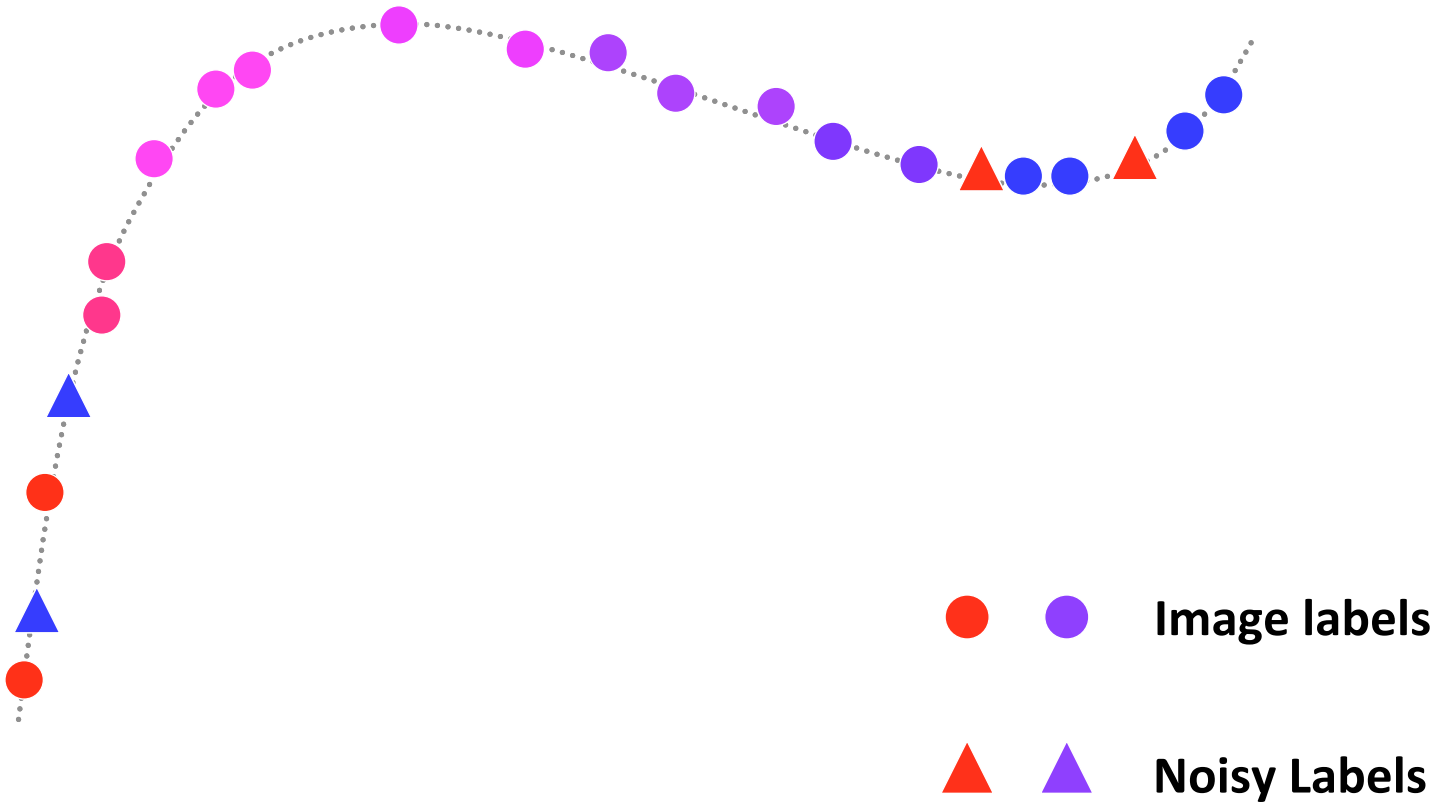
Clear	Clear	Cloudy	Cloudy
Partly	**Clear**	**Cloudy**	**Partly**

GeoFace

-90	0	45	45
-90	**0**	**45**	**90**

Robust regression on image manifolds for ordered label denoising, CVPR 2015

Robust Manifold Regression for Image Label Denoising



Robust regression on image manifolds for ordered label denoising, CVPR 2015

Robust Manifold Regression for Image Label Denoising

- A regularized empirical risk minimization framework

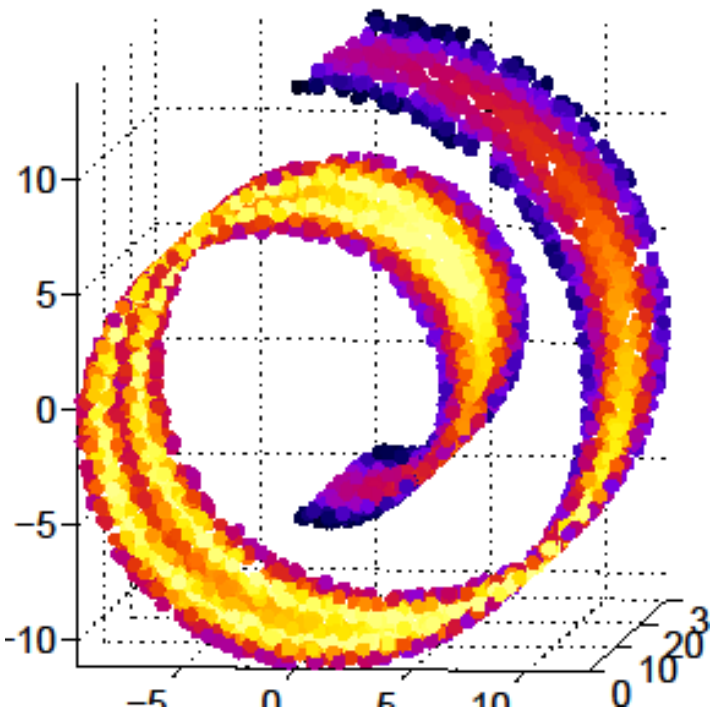
$$\underset{\hat{\mathbf{y}}}{\operatorname{argmin}} \quad \underbrace{\hat{\mathbf{y}}^{\intercal}\mathbf{B}\hat{\mathbf{y}}}_{\text{Hessian regularizer}} + \lambda \underbrace{\|\hat{\mathbf{y}} - \mathbf{y}\|_1}_{\text{L1 loss term}}$$

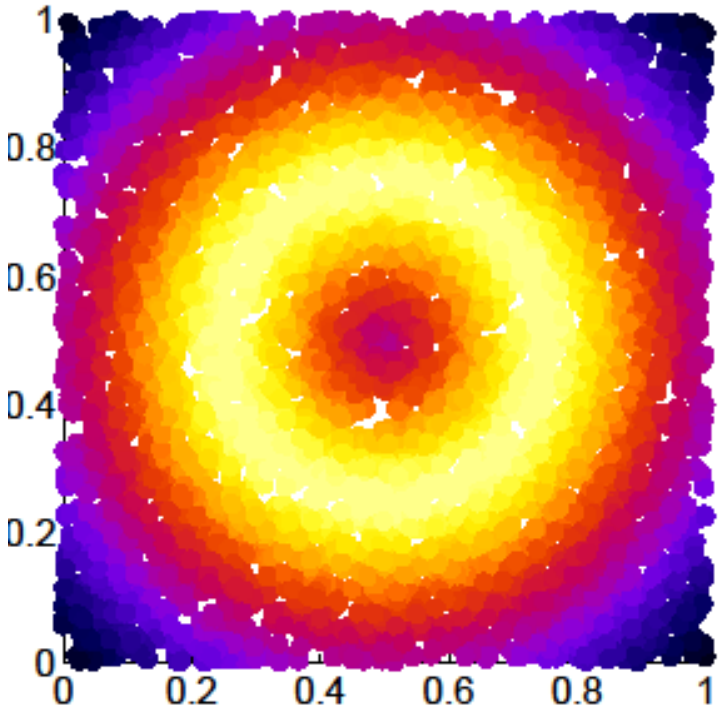
  - L1 norm is robust to high variance in noise
  - Implicitly promotes sparsity in noise: selects the 'good' labels
- An L1-regularized least squares problem
  - Efficient solvers are available for large-scale problems

Robust regression on image manifolds for ordered label denoising, CVPR 2015

Robust Manifold Regression for Image Label Denoising

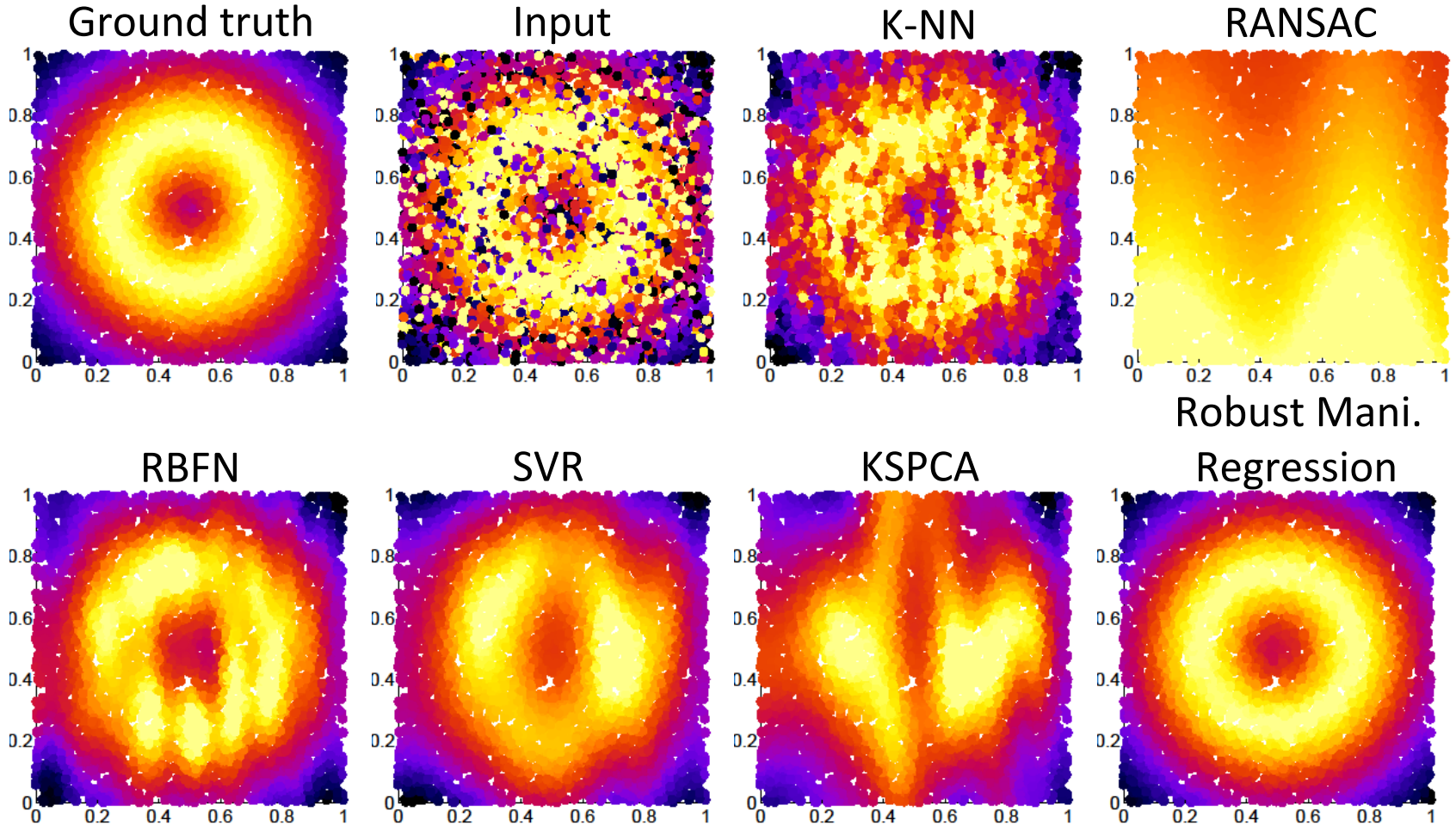
- Extend Swiss Roll data to include data labels (color of the points)





Robust regression on image manifolds for ordered label denoising, CVPR 2015

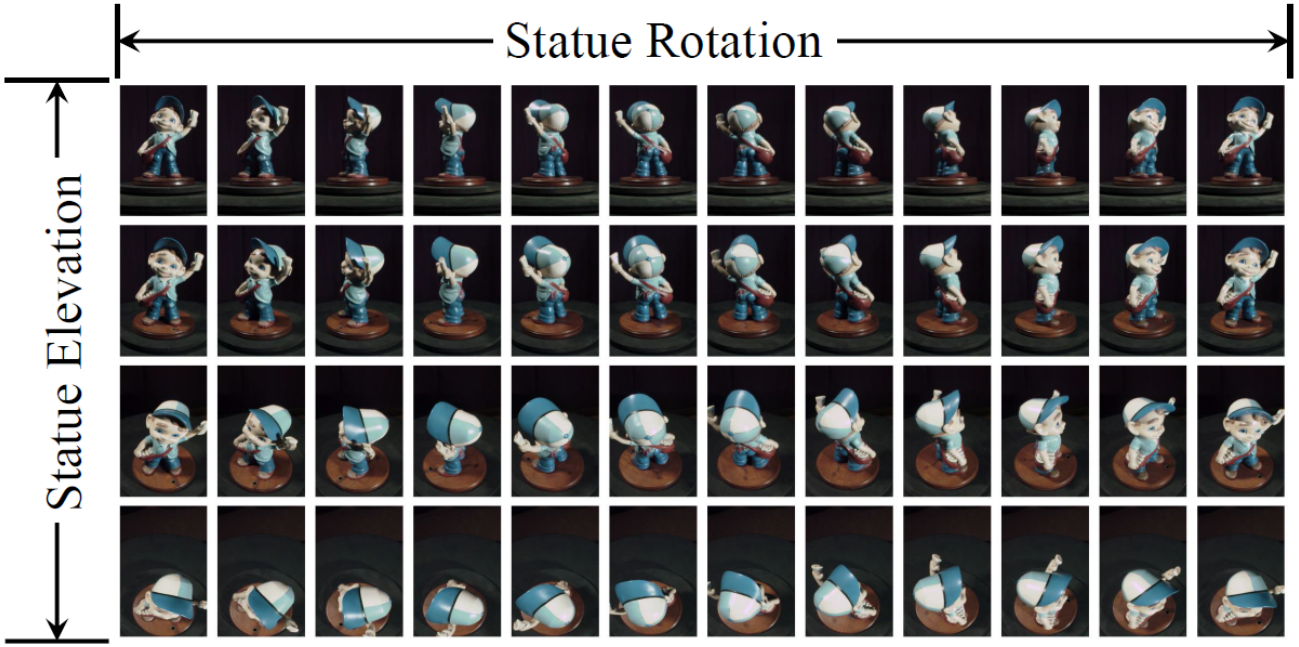
Robust Manifold Regression for Image Label Denoising



Robust regression on image manifolds for ordered label denoising, CVPR 2015

Robust Manifold Regression for Image Label Denoising

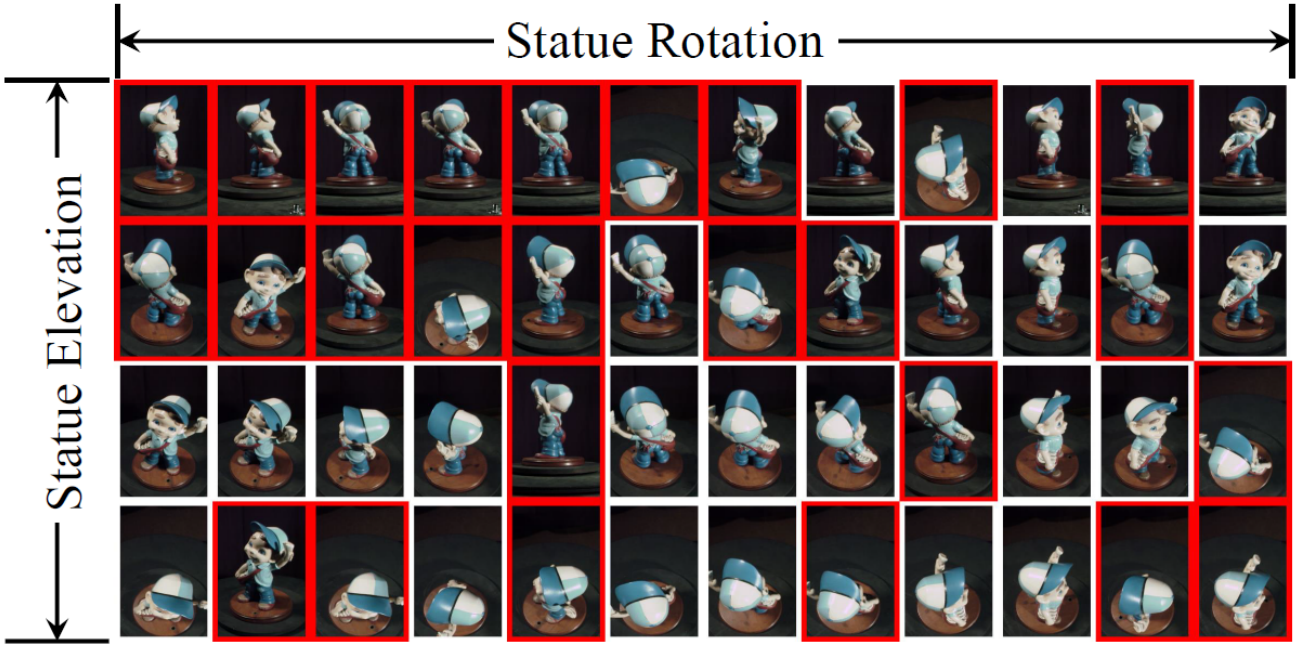
- 840 images collected on a turntable platform
- Images are sampled every 6° in rotation and elevation



Ground truth

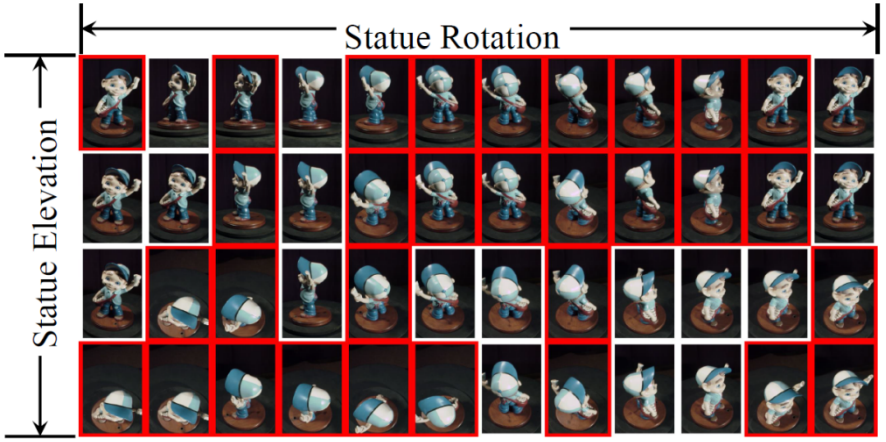
Robust Manifold Regression for Image Label Denoising

- 840 images collected on a turntable platform
- Images are sampled every 6° in rotation and elevation

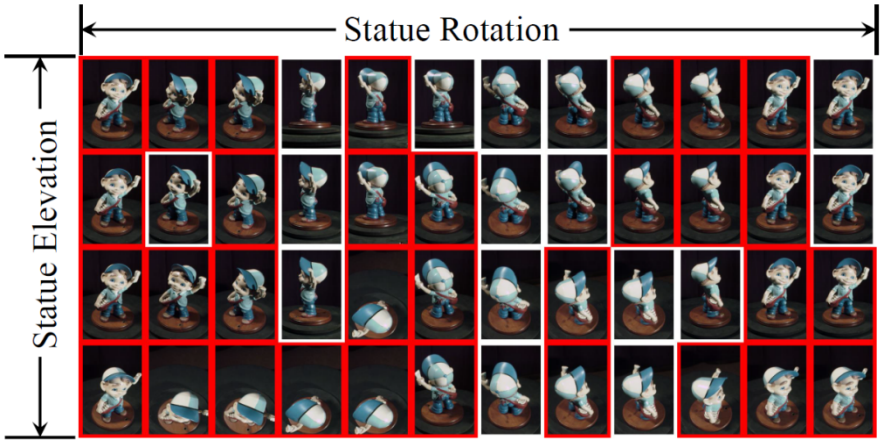


Input (Corrupted)

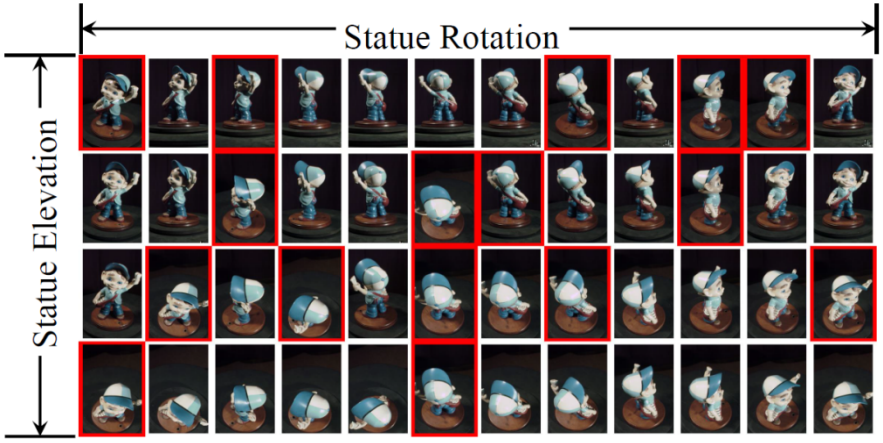
Robust Manifold Regression for Image Label Denoising



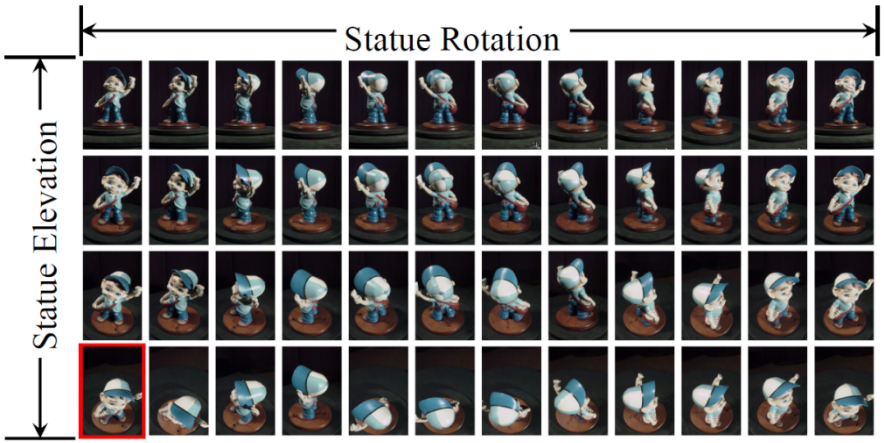
K-NN



KSPCA

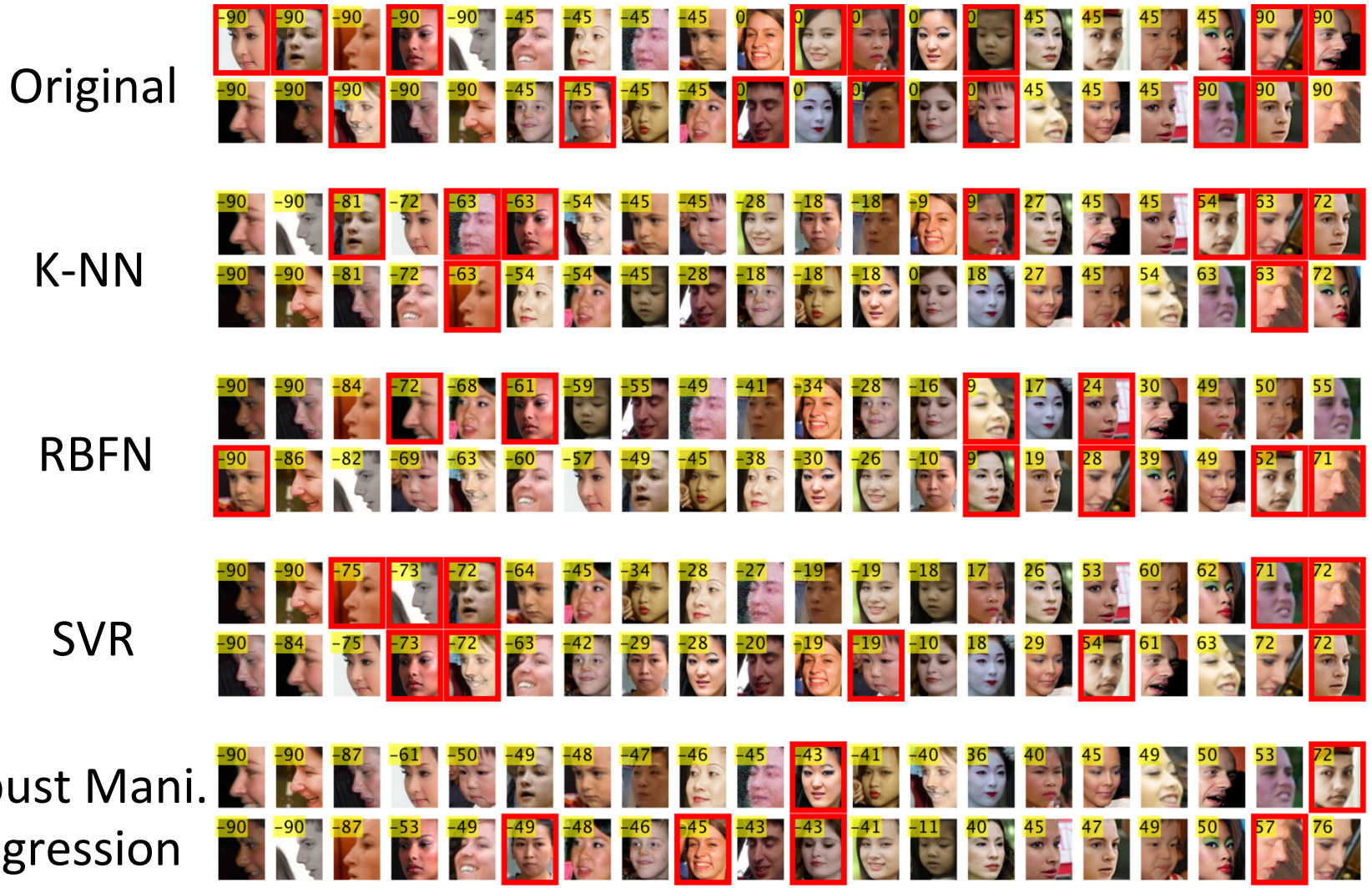


SVR



Robust Manifold Regression

Robust Manifold Regression for Image Label Denoising



Robust regression on image manifolds for ordered label denoising, CVPR 2015

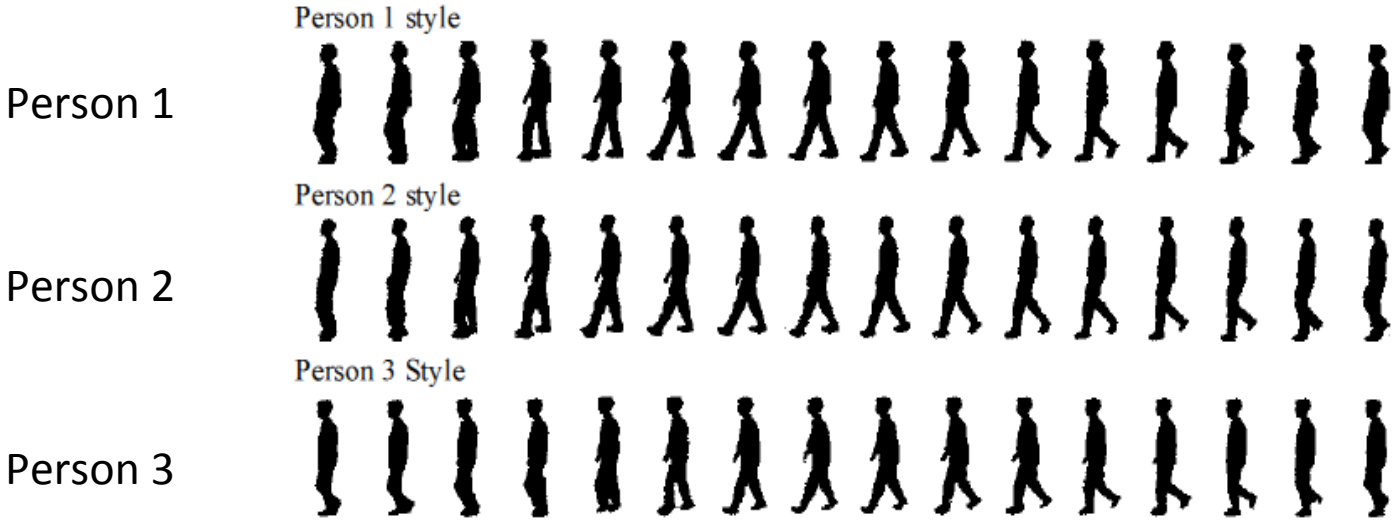
Outline

- Dimensionality reduction

- Applications of manifold learning on computer vision
  - A. Semi-supervised Regression on Manifolds
  - B. Robust Manifold Regression for Image Label Denoising
  - C. Manifold Learning for Human Motion Analysis

Manifold Learning for Human Motion Analysis

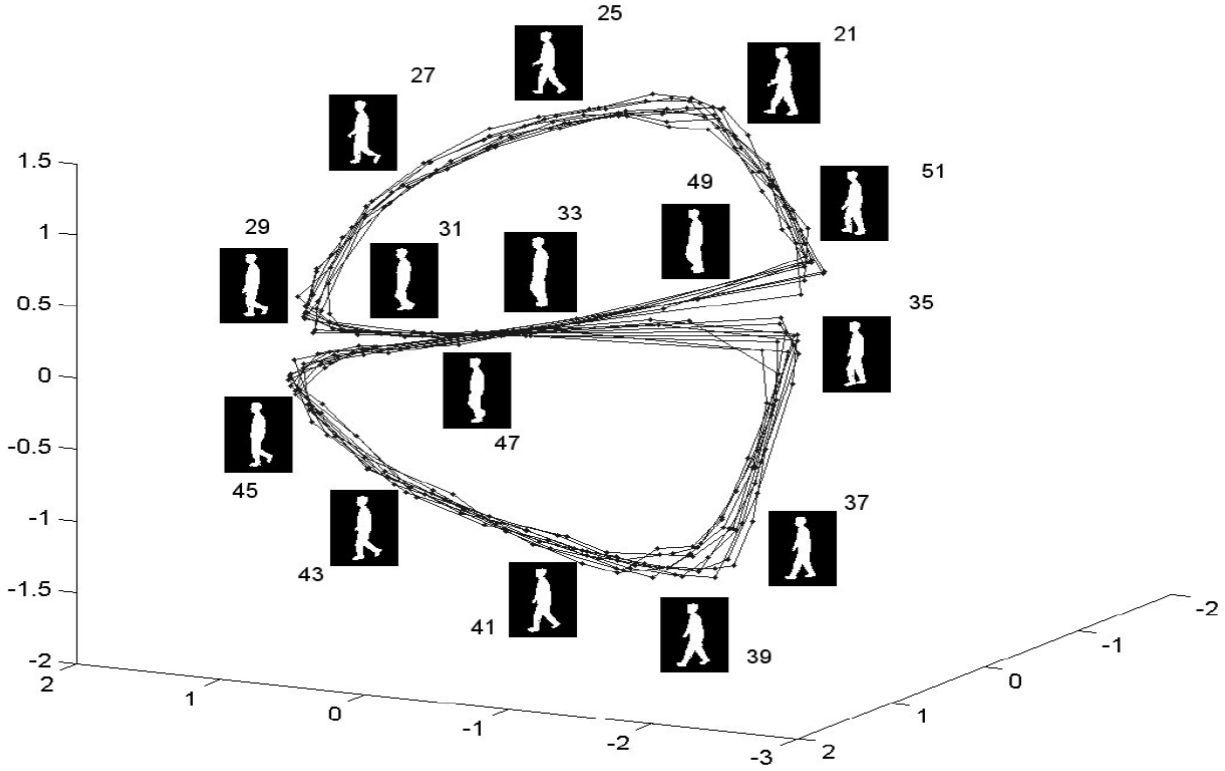
- Goal: separating style and content from images
  - For example
    - Content: the phase in a walking cycle
    - Style: person-specific gait style



Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- For each person, learns the low-D representation using LLE



Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

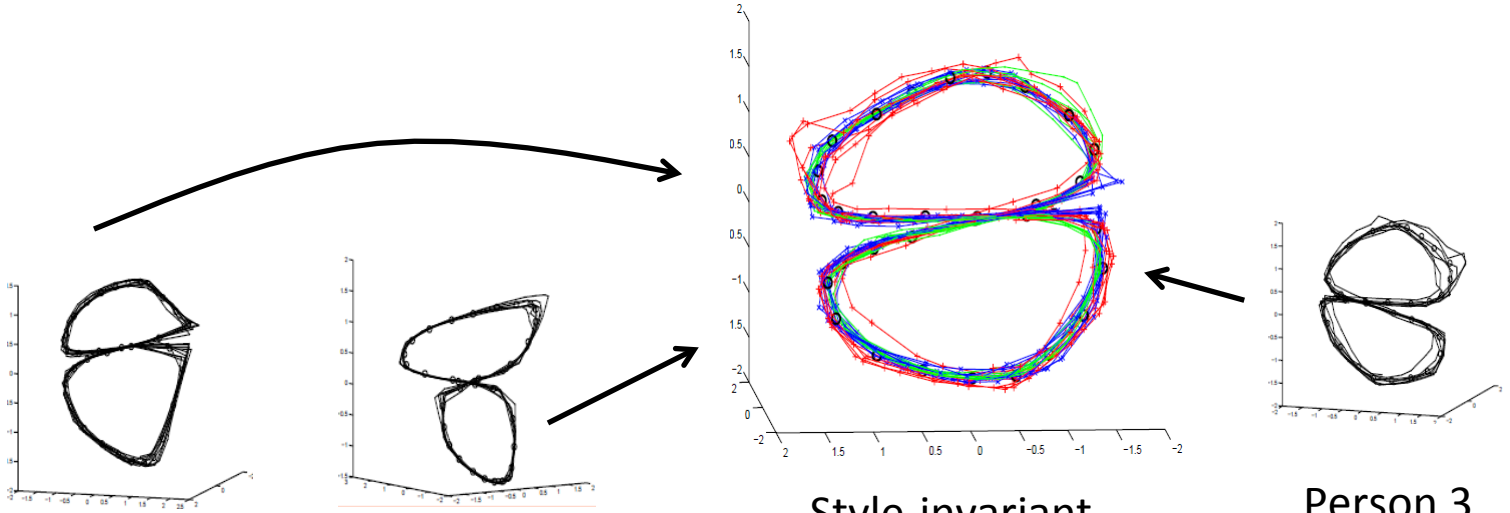
- Align multiple manifolds to compute a style-invariant manifold

$$E(f) = \sum_k \sum_i \|Z(t_i) - f(m^k(t_i); \alpha_k)\|^2 + \lambda \|Lf\|^2$$

mean manifold

mapping from each manifold

smooth term



Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- Learns style-dependent mappings
  - Learns parametric functions from the unified manifold to all input images
  - Let $\psi(x) = [\phi(|x - z_1|) \cdots \phi(|x - z_N|) \;\; 1 \;\; x^\top]^\top$

    A center on the mean manifold — Radial basis function — A point in low-D space

  - Each low-D coordinate maps to an input image y_i^k

$$y_i^k = C^k \cdot \psi(x)$$

  - Given N input images corresponding to the k-th style, C^k can be computed by solving a linear system

Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- Separating style
  - Let C be a tensor containing all style-dependent projections, $C^1, C^2, \cdots, C^K$
  - Perform SVD to obtain content bases and style coefficient

$$\mathcal{C} = \mathcal{A}^c \times_3 B^s$$

The learned style bases

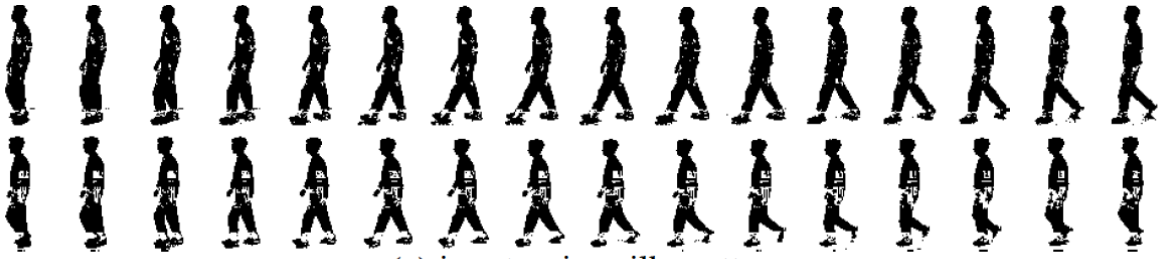
- For a new image, style and content can be solved by minimizing:

$$E(x^c, b^s) = \|y - \mathcal{A} \times b^s \times \psi(x^c)\|^2$$

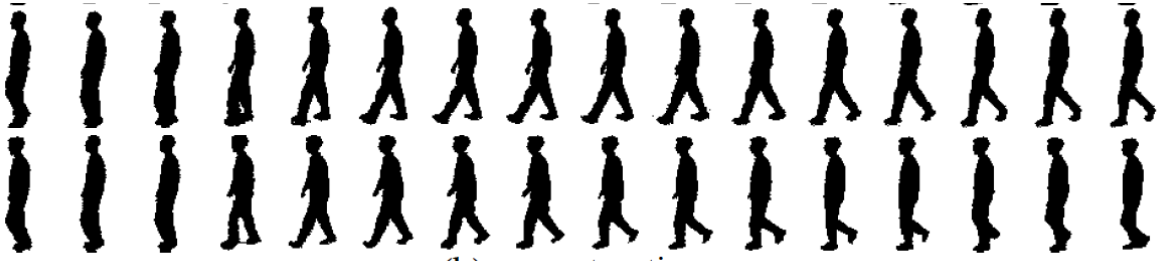
Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- Application #1: denoising
  1. Given each input (noisy) image, solve for style and content
  2. Given style and content factors, generate new image using the learned generative model



(a) input noisy silhouettes

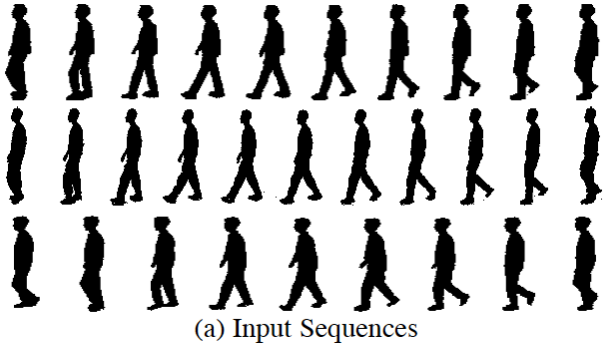


(b) reconstructions

Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- Application #2: content interpolation



(a) Input Sequences

Interpolated walks:

Person 1 style

Person 2 style

Person 3 Style

Separating style and content on a nonlinear manifold, CVPR 2004

Manifold Learning for Human Motion Analysis

- Application #3: style interpolation

Interpolated smiles for four different people

Interpolated smiles at intermediate (new) people styles.

Conclusions

- Manifold models can be incorporated in many applications by using a regularization term
- Classical manifold learning methods lack a parametric mapping from the high-D space to the manifold coordinates, and vice versa
- How to jointly learn a unified manifold given multiple data sets ?
  - For example, data sets from different domains or of different styles

- Next lecture: application of manifold learning on 3D computer vision problems