Workshop on Graph Partitioning in Vision and Machine Learning

Nonlinear Dimensionality Reduction

John Langford

Workshop on Graphs in Vision and Machine Learning

Nonlinear Dimensionality Reduction

John Langford

The Problem

Given: m points in R^n ($n \simeq 10^6$)

Find: "good" Projection into R^d ($d \simeq 2$)

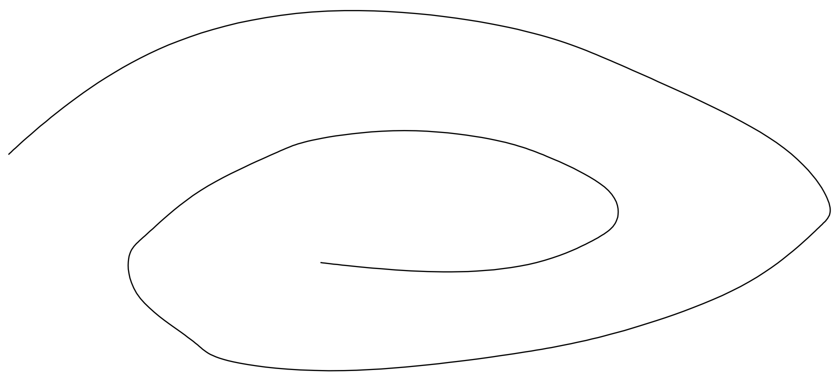
The Problem

Given: m points in R^n ($n \simeq 10^6$)

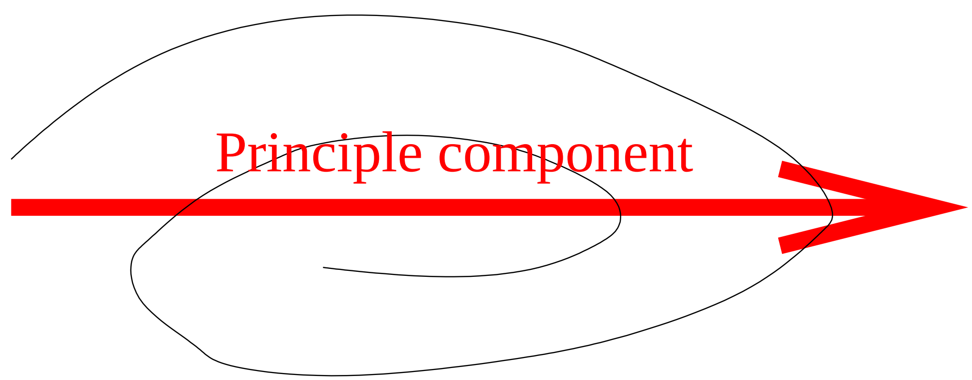
Find: "good" Projection into R^d ($d \simeq 2$)

Note: ill defined

The old way: PCA



The old way: PCA



The Fabled Outline

1. Applications

2. Algorithms & Analysis

3. Open Problems

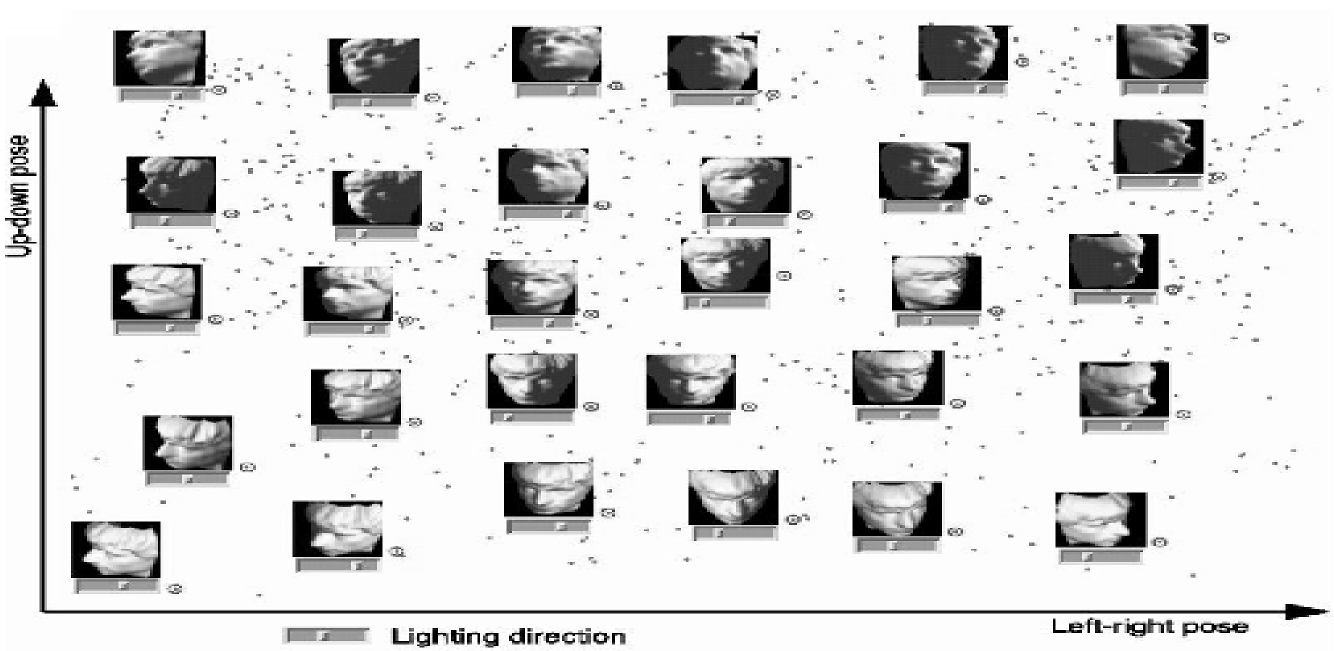
Applications

1. pictures from multiple viewpoints
2. pictures of a scene with moving objects
3. Spectra of stars
4. Sensor data?

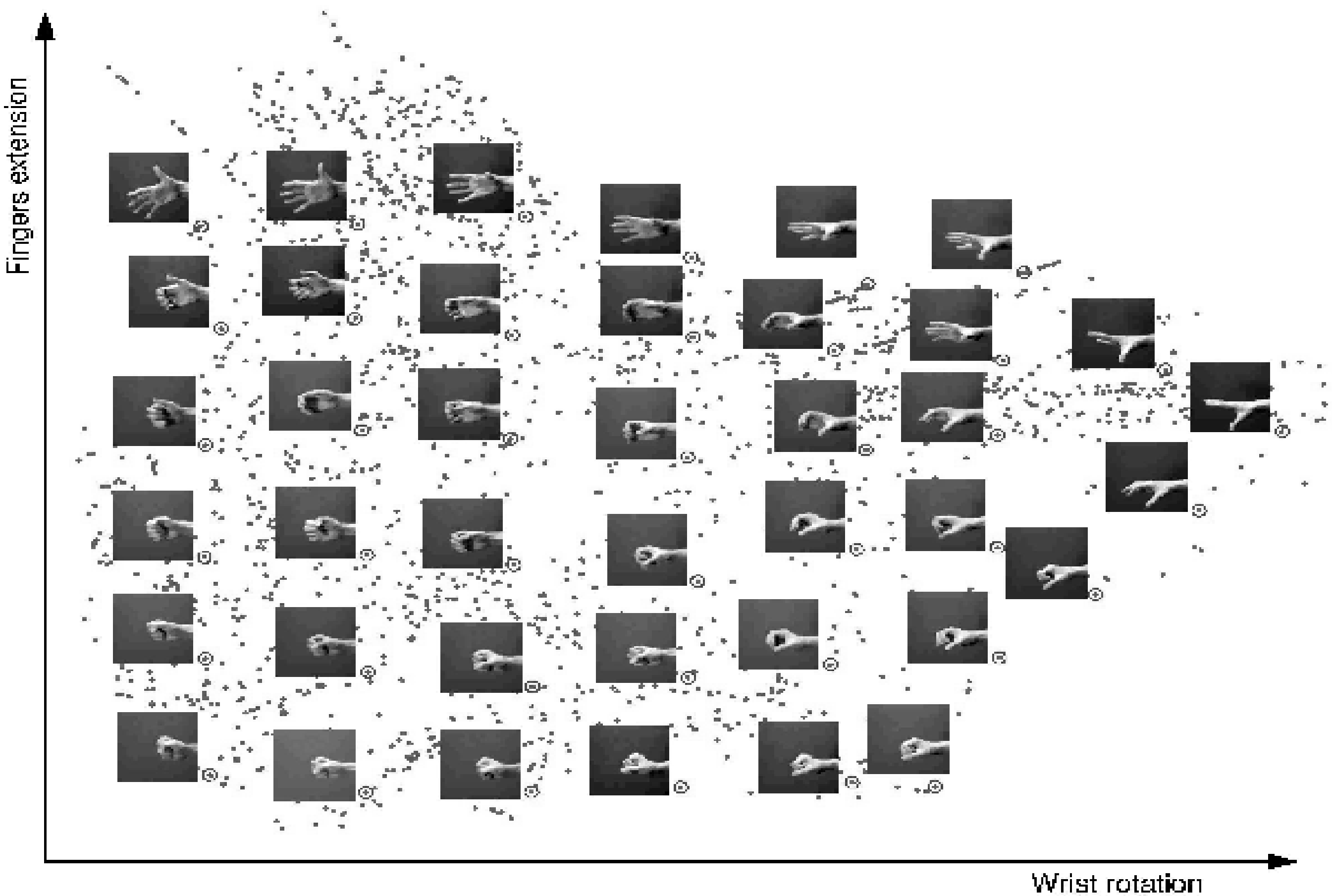
Manifold = the set of viewpoints embedded in image space

Input:

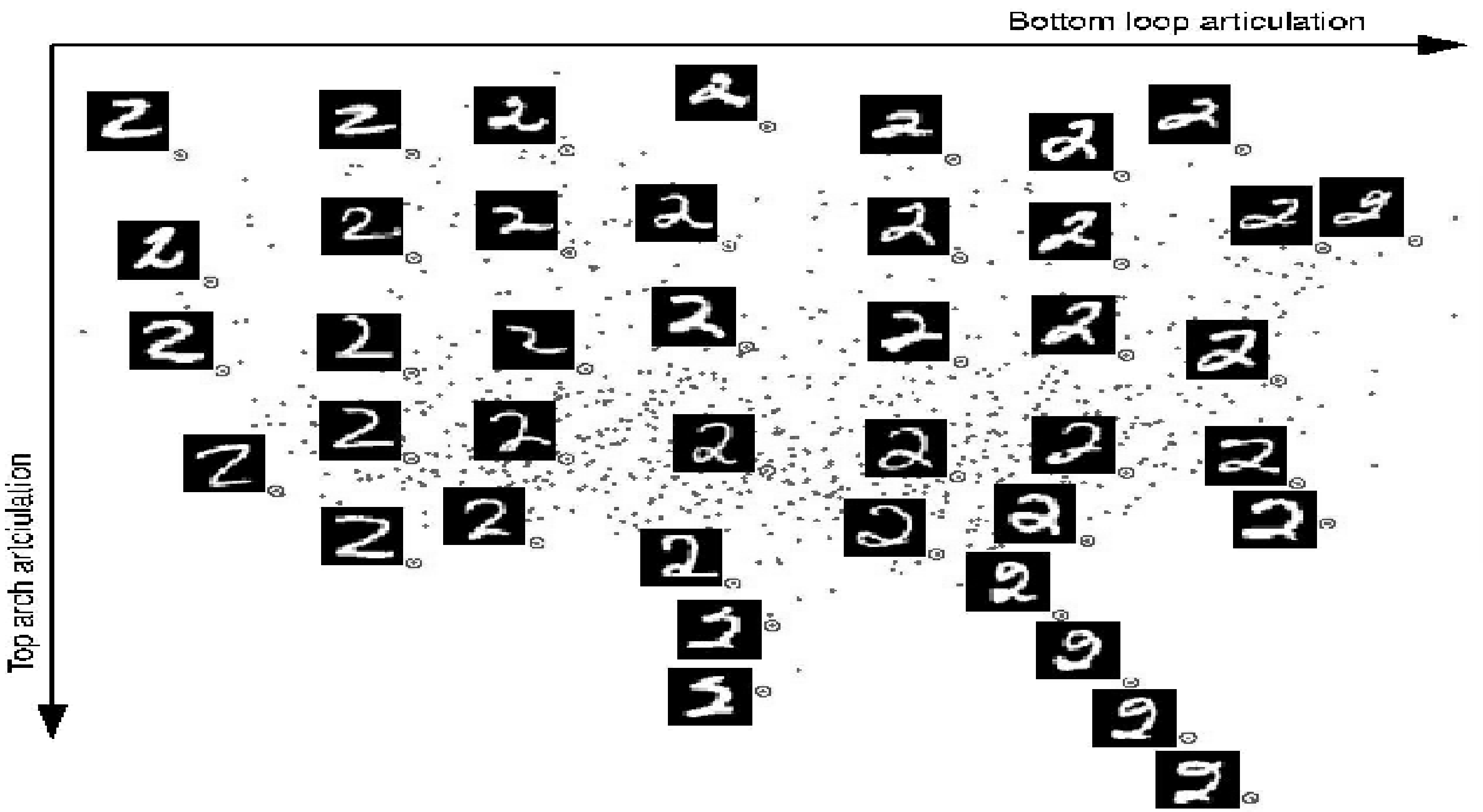
Output:



Manifold = the set of hand configurations embedded in image



Manifold = the set of hand-written “2”



Algorithms:

1. Techniques with local minima

2. Isomap (Josh Tenenbaum, Vin de Silva, Myself)

   (a) Conformal Isomap (T & dS)

   (b) Sparse forms (T & dS)

   (c) Local Isomap (Carrie Grimes and David Donoho)

3. LLE (Larry Saul, Sam Roweis)

   (a) Hessian LLE (G & D)

Isomap Algorithm

1. Construct neighborhood graph G

   (a) ϵ-Nearest Neighbor

   (b) K-Nearest Neighbor

2. Compute all-points shortest path in G

3. Use multidimensional scaling (eigenvalue method) to embed graph in R^d

Analysis:

Assume Isometric (distance-preserving) embedding:

1. Isomap rate of convergence given dense samples, not too much curvature, branch seperation (Josh, Vin, myself, Mira Bernstein)

2. Isomap converges asymptotically given convexity & isometry. (G & D)

Locally Linear Embedding

1. Find neighbors of each point

2. For every point, p_i, find equation in terms of linear superposition of neighborpoints

   $p_i = w_{i1}p_1 + w_{i2}p_2 + w_{i3}p_3 + w_{i4}p_4 + ...$

   Minimize errors: $\epsilon(w_{ij}) = \sum_i \left(p_i - \sum_{ij} w_{ij}p_j\right)^2$

3. Find a set of points in R^d (approximately) satisfying these equations:

   Find x_i to minimize $\widehat{\epsilon}(x_i) = \sum_i \left(x_i - \sum_{ij} w_{ij}x_j\right)^2$

LLE vs Isomap

Isomap derives global structure from local structure

LLE uses local structure only

- LLE is more flexible than Isomap (“stretching” is allowed)
- LLE does not recover isometric embeddings (G & D)
- LLE faster (sparse problem)
- Hessian LLE does converge (G & D)

Conformal (=distance preserving up to scale) Isomap

1. Construct neighborhood graph G

   K-Nearest Neighbor

2. Normalize all neighborhoods to have the same size.

3. Compute all-points shortest path in G

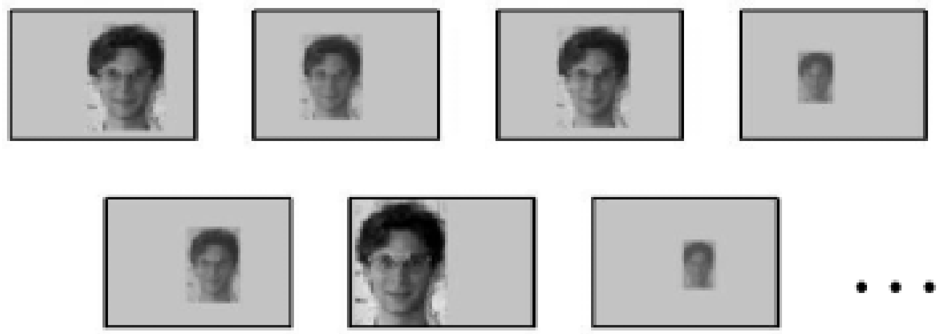
4. Use MDS to embed graph in R^d

Conformal Isomap Analysis

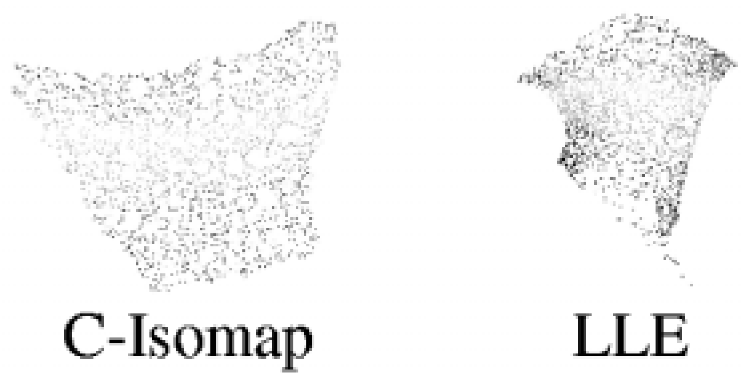
- Allows some stretching, like LLE
- Reverses conformal mappings, assuming uniform sampling

Conformal Isomap Example

Input =



Output =



Landmark Isomap (= attempt to make Isomap faster)

1. Choose $l > d$ “landmark” points

2. Construct neighborhood graph G

3. Compute shortest path in G to landmarks

4. Use MDS to embed landmarks in R^d

5. Embed other points based upon distance to landmarks

Landmark Analysis

- Dominating computation is nearest-neighbor calculations

- Number of landmarks needs to be only slightly larger than l.

Open Problems:

1. Which set of points should be in the "neighborhood"?

   (a) epsilon nearest neighbor?

   (b) K-nearest neighbor?

   (c) The problem of holes.

2. Can the Finite Sample analysis be improved to use a local feature size?

3. What can NOT be done?

4. Incrementalization: Many data sources are continuous