Manifold Learning

Generative modeling of high-dimensional data
Manifolds as Representation

images: 100 x 100 pixels

Ack: A. Efros, original images from hormel corp.
Learning to represent
Representations in AI

A representation for an object is a “frame” or collection of parameters associated with the object, the relations between them, and also a set of rules and functions for solving problems on the object.

The set of variables and the predicates defined on them are determined by a knowledge engineer.

Q. Can we learn representations?
Role of Perception?

Newborns (10-24 day old) in dark room work hard to position hand so it is visible in a narrow beam of light. ...

Q. Can perception help in learning a representation?

[A. van der Meer, 1997: Keeping the arm in the limelight]
Learning to represent: robot motions
Representations in AI

A representation for an object is a “frame” or collection of parameters and function associated with the object.

How to represent a “robot”? Must include: degrees of freedom (2) parameters $(\theta_1, \theta_2)$ + rules / functions
Manifolds
Linear dimensionality reduction

project data onto subspace of maximum variance

PCA: principal components analysis

\[ [A] = \text{top eigenvectors of covariance matrix } [XX^T] \]

\[ Y = [A] X \]
Manifolds

A manifold is a topological space which is locally Euclidean.

\[
\text{nbrhood } N \text{ in } \mathbb{R}^n \leftrightarrow \text{ball } B \text{ in } \mathbb{R}^d
\]

(homeomorphic)

**Homeomorphic:** Every \( x \) in \( N \) has a map to a \( y \) in \( B \)

Dimensionality of manifold = \( d \)

Embedding dimension = \( n \)
Manifolds

A manifold is a topological space which is locally Euclidean.

\[ \text{nbrhood in } \mathbb{R}^n \leftrightarrow \text{ball in } \mathbb{R}^d \]  
\[ \text{(homeomorphic)} \]

Dimensionality of manifold = \( d \)
Embedding dimension = \( n \)

Real life data (e.g. images) : \( D = 10^5 \)
motions = smooth variation of just a few parameters

\[ \text{DOFs} = \text{pose of faces} \rightarrow d = 1 \]

Ideally, \( d = \text{number of varying parameters} \)
Manifolds in video
Dimensionality of Actions

Weizmann activity dataset:
videos of 10 actions by 12 actors
[Gorelick / Blank / Irani : 2005 / 07]
Reduced dimensionality

Locality Preserving Projection
[He and Niyogi 2003]
Gestures in low dimensions
Recognizing gestures

- HMM
- HMM 1
- HMM2
- HMM3
Recognizing gestures

Keck gesture dataset
Non-Linear Dimensionality Reduction (NLDR) algorithms: ISOMAP
Euclidean or Geodesic distance?

Geodesic = shortest path along manifold
Isomap Algorithm

• Identify neighbors.
  – points within epsilon-ball (ε-ball)
  – $k$ nearest neighbors ($k$-NN)

• Construct neighborhood graph.
  -- $x$ connected to $y$ if $neighbor(x,y)$.
  -- edge length = distance($x,y$)

• Compute shortest path between nodes
  – Dijkstra / Floyd-Warshall algorithm

• Construct a lower dimensional embedding.
  – Multi-Dimensional Scaling (MDS)

[Tenenbaum, de Silva and Langford 2001]
Residual Variance and Dimensionality

residual variance = 1 – \( r^2(D_g, D_y) \); \( r \) = linear correlation coefficient
\( D_g \) = geodesic distance matrix; \( D_y \) = manifold distance
Short Circuits & Neighbourhood selection

neighbourhood size

too big: short-circuit errors
too small: isolated patches

[saxena, gupta mukerjee 04]
Locally-Linear Embedding

\[ J_1(W) = \sum_{i=1}^{N} \left\| x_i - \sum_{j=1}^{K} W_{ij} x_j \right\|^2 \]

\[ J_2(Y) = \sum_{i=1}^{N} \left\| y_i - \sum_{j=1}^{N} W_{ij} y_j \right\|^2 \]

\[ Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix} = \begin{bmatrix} -u_1 & - \cdots & -u_d \end{bmatrix}_{d \times N} \]
Non-isometric maps

Fishbowl dataset: no isomorphic map to plane

- Conformal mappings: preserve angles, not distances

- Assume data is uniformly distributed in low dim
Kernel PCA
Kernel PCA

PCA: top eigenvectors of covariance matrix $[XX^T]$

Kernel PCA: replace $X$ by $\phi(x)$

$$C = \frac{1}{N} \sum_{n=1}^{N} \phi(x_n) \phi(x_n)^T$$

Eigenvalue expression $Cv_i = \lambda_i v_i$

$$\frac{1}{N} \sum_{n=1}^{N} \phi(x_n) \{\phi(x_n)^T v_i\} = \lambda_i v_i$$

To express in terms of kernel fn $k(x_n, x_m) = \phi(x_n)^T \phi(x_m)$, substitute

$$v_i = \sum_{n=1}^{N} a_{in} \phi(x_n)$$

Bishop section 12.5
Kernel PCA

\[ \frac{1}{N} \sum_{n=1}^{N} \phi(x_n)\phi(x_n)^T \sum_{m=1}^{N} a_{im} \phi(x_m) = \lambda_i \sum_{n=1}^{N} a_{in} \phi(x_n). \]

Multiply both sides by \( \varphi(x_n)^T \)

\[ \frac{1}{N} \sum_{n=1}^{N} k(x_l, x_n) \sum_{m=1}^{m} a_{im} k(x_n, x_m) = \lambda_i \sum_{n=1}^{N} a_{in} k(x_l, x_n). \]

which reduces to

\[ Ka_i = \lambda_i N a_i \]

(a K is removed from both sides – affects only zero \( \lambda_i \)).

Projections \( y_i = \sum_{n=1}^{N} a_{in} k(x, x_n) \)

What happens when we use a linear kernel \( k(x, x') = x^T x' \)?
Kernel PCA: Demonstration

Kernel: \( k(x, x') = \exp \left( -\frac{|x - x'|^2}{0.1} \right) \)

[Scholkopf 98]
Learning representations: Handwritten Digits
handwritten numerals (MNIST)

Modified NIST digits database: 60K + 10K 28x28 images
Importance of choosing a metric

9 9 4
Manifold mapping with Euclidean Distance
"tangent distance"

Pixel space

\[ s(P, \alpha) \]

\[ \mathbf{T} = \frac{\partial F}{\partial \alpha} \]

\[ s(2, \alpha) = \begin{array}{cccccc}
\alpha=-2 & \alpha=-1 & \alpha=0 & \alpha=1 & \alpha=2 \\
\end{array} \]

\[ 2 + \alpha \cdot \mathbf{T} = \begin{array}{cccccc}
\alpha=-2 & \alpha=-1 & \alpha=0 & \alpha=1 & \alpha=2 \\
\end{array} \]
Manifold mapping with
Dimensionality: handwritten digits

Residual variance

Manifold dimension
NLDR algorithms:
Representing a robot
Input = images
Manifold dimension
2-DOF motion
Dimensionality reduction

• Identify neighbors.
  – “neighbours” may have link1 in same pose, but link 2 varying
  – Alternately, link2 similar, but link1 varying
    → variation is along 2 dimensions in image space,

• Construct neighborhood graph.

• Compute geodesics = shortest path between nodes

• Find low-dimensional embedding that preserves geodesics
  – Target dimension for low-D space not known. Just try 1,2,3, … n
Residual variance vs dimension

dofs : 2
Robot Structure Learning

• Consider many images of robot configurations
  – Construct manifold on images
  – Dimensionality that explains variance

• Resultant graph
  -- $\text{neighbor}(x,y) \rightarrow$ neighbouring configurations.
  – Topology (for unbounded theta) = torus
• Latent variables:
  – Distributed on $S^1 \times S^1$ topology
  – Along circumferential path: $\theta_1$; along radial: $\theta_2$
  – Naïve, non-metric representation of $\theta_1, \theta_2$

• Manifold transformation = mapping between input images (workspace) ↔ naive $\theta_1, \theta_2$ (C-space)
  – manifold $\rightarrow$ image $\approx$ naïve forward kinematics
  – image $\rightarrow$ manifold $\approx$ naïve inverse kinematics
Differences with AI representation

• Grounding
  – AI models are defined only in terms of other logical structures → circularity of definitions
  – Manifold-based naïve representations: grounded on sensory data:
• Physics like formulation \((\theta_1, \theta_2)\) may not be needed
  – Topologically consistent representation of \((\theta_1, \theta_2)\)
  – Non-uniform sampling → higher resolution for functionally relevant regions