Spectral Methods for Dimensionality Reduction

Prof. Lawrence Saul
Dept of Computer & Information Science
University of Pennsylvania

UCLA IPAM Tutorial, July 11-14, 2005
Nonlinear dimensionality reduction

Given high dimensional data sampled from a low dimensional submanifold, how to compute a faithful embedding?
Notation

• Inputs (high dimensional)
  \[ \vec{x}_i \in \mathbb{R}^D \text{ with } i = 1, 2, \ldots, n \]

• Outputs (low dimensional)
  \[ \vec{y}_i \in \mathbb{R}^d \text{ where } d \ll D \]

• Goals
  Nearby points remain nearby.
  Distant points remain distant.
  (Estimate \( d \).)
Linear vs nonlinear

What computational price must we pay for nonlinear dimensionality reduction?
Yesterday

• **Linear methods**
  - principal components analysis (PCA)
  - metric multidimensional scaling (MDS)

• **Isomap**
  1. \( k \)-nearest neighbors
  2. Shortest paths through graph
  3. MDS on geodesic distances

*A nonlinear method with most of the advantages of linear ones...*
Yesterday vs today

• MDS and Isomap
  - preserve global pairwise distances
  - construct large, dense matrices
  - compute top eigenvectors

• “Local” methods
  - preserve local geometric relationships
  - construct large, sparse matrices
  - compute bottom eigenvectors
2000
Isomap
(Tenenbaum, de Silva, & Langford)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)

2005
Conformal eigenmaps
(Sha & Saul)

Locally Linear Embedding
(Roweis & Saul)
Questions for today

• **How to exploit local linearity?**
  Manifolds are globally nonlinear, but locally linear.

• **Isn’t this an old idea?**
  - k-means, k-subspaces
  - mixture models
  - self-organizing maps
How (not) to use local linearity

iterative clustering,
subspace quantization
k-means clustering

• Goal
Map each continuous input $\vec{x}_i$ to a discrete label $y_i \in \{1, 2, ..., k\}$. 

• Algorithm
1. Randomly choose $k$ “centroids” $\vec{\mu}_\alpha$.
2. Set $y_i = \arg\min_{\alpha} ||\vec{x}_i - \vec{\mu}_\alpha||$.
3. Set $\vec{\mu}_\alpha$ to mean of inputs with $y_i = \alpha$.
4. Iterate steps 2-3 until convergence.
k-means clustering

• Generalizations
  - ellipsoidal vs spherical clusters
  - unbalanced vs balanced clusters
  - soft (probabilistic) assignment
  - k-lines, k-planes, k-subspaces
Problem solved?

Can simple iterative clustering algorithms, properly generalized, solve the problem of manifold learning?
Problem solved? No.

Iterative clustering algorithms are sensitive to initial conditions, with many spurious local minima.
Also, remember the goal…

- **Inputs** (high dimensional)
  \[ \vec{x}_i \in \mathbb{R}^D \text{ with } i = 1, 2, \ldots, n \]

- **Outputs** (low dimensional)
  \[ \vec{y}_i \in \mathbb{R}^d \text{ where } d \ll D \]

- **Goals**
  Nearby points remain nearby.  
  Distant points remain distant.  
  (Estimate \( d \).)
Local vs global

Clustering algorithms do not map their inputs into a single continuous global coordinate system of lower dimensionality.
Locally Linear Embedding

“Think globally, fit locally.”
Algorithm

• Steps
  1. Nearest neighbor search.
  2. Least squares fits.

• Properties
  – Obtains highly nonlinear embeddings.
  – Not prone to local minima.
  – Sparse graphs yield sparse problems.
Step 1. Identify neighbors.

• **Examples of neighborhoods**
  – $k$ nearest neighbors
  – Neighbors within radius $r$
  – Metric based on prior knowledge

• **Assumptions**
  – Data is sampled from a manifold.
  – Manifold is well sampled.
Nearest neighbor graph

Assumptions:
• Graph is connected.
• Neighborhoods on the graph correspond to neighborhoods on the manifold.
Step 2. Compute weights.

- Characterize local geometry of each neighborhood by weights $W_{ij}$.

- Compute weights by reconstructing each input (linearly) from neighbors.
Linear reconstructions

• Local linearity

Neighbors lie on locally linear patches of a low dimensional manifold.

• Reconstruction errors

Least squared errors should be small:

\[ \Phi(W) = \sum_i \left( \bar{x}_i - \sum_j W_{ij} \bar{x}_j \right)^2 \]
Least squares fits

• Local reconstructions
  Choose weights to minimize:
  \[ \Phi(W) = \sum_{i} \left| \bar{x}_i - \sum_{j} W_{ij} \bar{x}_j \right|^2 \]

• Constraints
  Nonzero \( W_{ij} \) only for neighbors.
  Weights must sum to one:
  \[ \sum_{j} W_{ij} = 1 \]

• Local invariance
  Optimal weights \( W_{ij} \) are invariant to rotation, translation, and scaling.
Symmetries

• **Local linearity**
  If each neighborhood map looks like a translation, rotation, and rescaling...

• **Local geometry**
  ...then these transformations do not affect the weights $W_{ij}$: they remain valid.
Thought experiment

• Reconstruction from landmarks

Clamp subset of inputs ("landmarks"), then reconstruct others by minimizing:

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

with respect to $\vec{x}_i$!

n=2000 inputs

Number of landmarks: $L = 15$, $L = 10$, $L = 5$
Thought experiment (con’t)

- **Locally linear reconstruction**
  - Very accurate for sufficiently large number of landmarks.
  - Increasingly linearized with decreasing number of landmarks.

Number of landmarks: $L = 15$, $L = 10$, $L = 5$, $L = 0$?
Step 3. “Linearization”

- **Low dimensional representation**
  Map inputs to outputs: \( \tilde{x}_i \in \mathbb{R}^D \) to \( \tilde{y}_i \in \mathbb{R}^d \)

- **Minimize reconstruction errors.**
  Optimize outputs for fixed weights:
  \[
  \Psi(y) = \sum_i \left| \tilde{y}_i - \sum_j W_{ij} \tilde{y}_j \right|^2
  \]

- **Constraints**
  Center outputs on origin: \( \sum_i \tilde{y}_i = \bar{0} \)
  Impose unit covariance matrix:
  \[
  \frac{1}{N} \sum_i \tilde{y}_i \tilde{y}_i^T = I_d.
  \]
Sparse eigenvalue problem

- **Quadratic form**

\[ \Psi(y) = \sum_{ij} \Psi_{ij} (\tilde{y}_i \cdot \tilde{y}_j) \text{ with } \Psi = (I - W)^T (I - W) \]

- **Rayleigh-Ritz quotient**

Optimal embedding given by bottom \(d+1\) eigenvectors.

- **Solution**

Discard bottom eigenvector \([1 \ 1 \ \ldots \ 1]\). Other eigenvectors satisfy constraints.
Summary of LLE

• Three steps
  1. Compute k-nearest neighbors.
  2. Compute weights $W_{ij}$.
  3. Compute outputs $\vec{y}_i$.

• Optimizations

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

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

N=1000 inputs

k=8 nearest neighbors

D=3 d=2 dimensions
Translated faces

N=961 images

k=4 nearest neighbors

D=3009 pixels

d=2 manifold
Pose and expression

N=1965 images
k=12 nearest neighbors
D=560 pixels
d=2 (shown)
**Lips**

- \(N = 15960\) images
- \(K = 24\) neighbors
- \(D = 65664\) pixels
- \(d = 2\) (shown)
Vowels: /aa/ ("hot") vs /ae/ ("hat")

N=3000 log-power spectra
K=10 nearest neighbors
D=400 window size
Word-document counts

n=5000 words
k=20 nearest neighbors
D=31000 documents
d=3,4,5 (shown)
What computational price must we pay for nonlinear dimensionality reduction?
Properties of LLE

• **Strengths**
  – Polynomial-time optimizations
  – No local minima
  – Non-iterative (one pass thru data)
  – Non-parametric
  – Only heuristic is neighborhood size.

• **Weaknesses**
  – Sensitive to “shortcuts”
  – No out-of-sample extension
  – No estimate of dimensionality
LLE versus Isomap

• Many similarities
  – Graph-based, spectral method
  – No local minima

• Essential differences
  – Does not estimate dimensionality
  – No theoretical guarantees

+ Constructs sparse vs dense matrix

? Preserves weights vs distances

Conformal mapping
Algorithms

2000
Isomap
(Tenenbaum, de Silva, & Langford)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)
(Sun, Boyd, Xiao, & Diaconis)

2005
Conformal eigenmaps
(Sha & Saul)

Locally Linear Embedding
(Roweis & Saul)
Laplacian eigenmaps

- **Key idea:**
  Map nearby inputs to nearby outputs, where nearness is encoded by graph.

- **Physical intuition:**
  Find lowest frequency vibrational modes of a mass-spring system.
Summary of algorithm

• Three steps

1. Identify k-nearest neighbors
2. Assign weights to neighbors:
   \[ W_{ij} = 1 \text{ or } W_{ij} = \exp\left(-\beta \| \bar{x}_i - \bar{x}_j \|^2 \right) \]
3. Compute outputs by minimizing:
   \[ \Psi(y) = \sum_{ij} \frac{W_{ij} \| \bar{y}_i - \bar{y}_j \|^2}{\sqrt{D_{ii} D_{jj}}} \text{ where } D_{ii} = \sum_j W_{ij} \]

(sparse eigenvalue problem as in LLE)
Laplacian vs LLE

• More similar than different
  – Graph-based, spectral method
  – Sparse eigenvalue problem
  – Similar results in practice

• Essential differences
  – Preserves locality vs local linearity
  – Uses graph Laplacian

\[ L = D - W \]  \hspace{1cm} \text{(unnormalized)}

\[ \mathcal{L} = I - D^{-1/2}WD^{-1/2} \]  \hspace{1cm} \text{(normalized)}
Analysis on Manifolds

• Laplacian in $\mathbb{R}^d$
  
  Function $f(x_1, x_2, \ldots, x_d)$ has Laplacian:
  
  $$\Box f = -\sum_i \frac{\partial^2 f}{\partial x_i^2}$$

• Manifold Laplacian
  
  Change is measured along tangent space of manifold.

• Stokes theorem
  
  $$\int_M \left\| \nabla f \right\|^2 = \int_M f \Box f$$
Spectral graph theory

• Manifolds and graphs
  Weighted graph is discretized representation of manifold.

• Laplacian operators
  Laplacian measures smoothness of functions over manifold (or graph).

\[
\int_M \| \nabla f \|^2 = \int_M f \Box f \quad \text{(manifold)}
\]

\[
\sum_{ij} W_{ij} (f_i - f_j)^2 = f^T Lf \quad \text{(graph)}
\]
Example: $S^1$ (the circle)

- Continuous
  
  - Eigenfunctions of Laplacian are basis for periodic functions on circle, ordered by smoothness.
  
  - Eigenvalues measure smoothness.

\[ -\frac{\partial^2 f_m}{\partial^2 \theta} = \lambda_m f_m(\theta) \]

\[ f_m(\theta) = \begin{cases} 
  \sin(m\theta) \\
  \cos(m\theta) 
\end{cases} \quad \text{with } \lambda_m = m^2 \]
Example: $S^1$ (the circle)

- Discrete ($n$ equally spaced points)
  - Eigenvectors of graph Laplacian are discrete sines and cosines.
  - Eigenvalues measure smoothness.

Graph embedding from Laplacian eigenmaps:

$$\tilde{y}_k = (\cos(2\pi k/n), \sin(2\pi k/n))$$
Example: Swiss roll
eigenvalues of graph Laplacian
A critical view…

• **LLE and Laplacian eigenmaps**
  – Construct quadratic form over functions on graph.
  – Take $d$ lowest cost (but non-constant) functions as manifold coordinates.

• **Theoretical guarantees?**
  – When do bottom eigenvectors give the “right answer”?
  – Depends on the definition of the “right answer”…
A critical view (con’t)

• Assumption
  – Sample inputs from manifold that is isometrically embedded in $\mathcal{R}^D$.
  – Assume manifold is locally isometric to an open subset of $\mathcal{R}^d$, where $d < D$.

• Hypothesis
  – Isomap’s top $d$ eigenvectors recover parameterization for convex subsets.
  – Can bottom $d$ (nonzero) eigenvectors of sparse matrix method do better?
Algorithms

2000
Isomap
(Tenenbaum, de Silva, & Langford)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)

2005
Conformal components analysis
(Sha & Saul)

Locally Linear Embedding
(Roweis & Saul)
Hessian LLE

• Assumption
  Data manifold $M$ is locally isometric to open, connected subset of $\mathbb{R}^d$.

• Key ideas
  – Define Hessian via orthogonal coordinates on tangent planes of $M$.
  – Quadratic form $\Psi(f)$ averages Frobenius norm of Hessian over $M$.

\[
\Psi(f) = \int_M \left\| H_f(m) \right\|^2 \, dm
\]

\[
\int_M \left\| \nabla f \right\|^2 = \int_M f \circ f
\]
Hessian LLE

\[ \Psi(f) = \int_M \| H_f(m) \|^2 \, dm \]

- Every function with vanishing Hessian is linear. (Not so for Laplacian.)
- Bottom eigenfunctions in null space of \( H(f) \) yield isometric coordinates.
- Graph-based discretization yields algorithm.

\[ \int_M \| \nabla f \|^2 = \int_M f \Box f \]

\[ \int_M \| \nabla f \|^2 - \int_M f \Box f \]
Hessian LLE

- **Three steps**
  1. Construct graph from kNN.
  2. Estimate Hessian operator at each data point.
  3. Compute bottom eigenvectors of sparse quadratic form.

- **What’s new?**
  (1) and (3) are same as before.
  (2) estimates Hessian. (Details omitted.)

\[
\Psi(f) = \int_M \|H_f(m)\|^2 dm
\]
Relation to previous work

• **Algorithm variant of LLE**
  Replaces least squares fits in LLE by estimation of Hessian.

• **Conceptual variant of Laplacian**
  Substitutes Frobenius norm of Hessian for norm of gradient vector.

• **Sparse matrix variant of Isomap**
  Also looks for isometric coordinates on data manifold.
Theoretical guarantees

• **Asymptotic convergence**
  For data sampled from a submanifold that is isometric to an open, connected subset of Euclidean space, hLLE will recover the subset up to rigid motion.

• **No convexity assumption**
  Convergence is obtained for a larger class of manifolds than Isomap.
Connected but not convex

Hessian LLE yields an isometric embedding, but not Isomap or LLE.
Connected but not convex

- **Occlusion**
  Images of two disks, one occluding the other.

- **Locomotion**
  Images of periodic gait.
Algorithms

2000
Isomap
(Tenenbaum, de Silva, & Langford)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

Locally Linear Embedding
(Roweis & Saul)

What is left to do?
Problem solved?

• For manifolds without “holes”:
  – Isomap with asymptotic guarantees
  – landmark Isomap for large data sets

• More generally:
  – hLLE with asymptotic guarantees?
  – sparse matrix method should scale well to large data sets?

(If it seems too good to be true, it usually is...)
Flies in the ointment

• How to estimate dimensionality?
  Revealed by eigenvalue gap of Isomap, but specified in advance for (h)LLE.

• How to compute eigenvectors?
  Bottom eigenvalues are very closely spaced for large data sets.

• Must we preserve distances?
  Preserving distances may hamper dimensionality reduction.
Computing eigenvectors

• Numerical difficulty
  Inversely proportional to spacing between adjacent eigenvalues.

• Scaling to large data sets
  Bottom eigenvalue spacing shrinks with increased sampling of manifold.

• Conundrum
  Finer discretization of manifold trades off with ability to resolve eigenvectors.
Example

• Lattice model
  Inputs are $n$ sites of hypercubic lattice. Edges connect $2d$ nearest neighbors.

• Fourier diagonalization
  Graph Laplacian has translational symmetry. Eigenvectors: $\exp(i\mathbf{q} \cdot \mathbf{x})$.

• Eigenvalues
  For $n=\infty$, eigenvalues are indexed continuously by $\mathbf{q}$ in $[-\pi, \pi]^d$; no gaps!
Can we combine strengths of:

- **Isomap**

- **hLLE**
  Solves sparse eigenvalue problem. Handles manifolds with “holes”.

- **LLE and Laplacian eigenmaps**
  Aggressive dimensionality reduction. Locality vs distance-preserving maps
See you tomorrow...

2000
Isomap
(Tenenbaum, de Silva, & Langford)

Locally Linear Embedding
(Roweis & Saul)

2002
Laplacian eigenmaps
(Belkin & Niyogi)

2003
Hessian LLE
(Donoho & Grimes)

2004
Maximum variance unfolding
(Weinberger & Saul)
(Sun, Boyd, Xiao, & Diaconis)

2005
Conformal eigenmaps
(Sha & Saul)