Nonlinear Manifold Learning II

Laplacian Eigenmaps
Hessian Eigenmaps
Charting

Erik Sudderth

6.454 Presentation
October 15, 2002

Manifold Learning: Last Week

Goal: Determine the low-dimensional structure underlying a set of high-dimensional data

Linear Structure:

- Principal Components Analysis (PCA)
- Multidimensional Scaling (MDS)

Nonlinear Structure:

- Locally Linear Embedding (LLE)
- Isometric Feature Mapping (IsoMap)

Manifold Learning: This Week

Laplacian Eigenmaps: (Belkin & Niyogi, NIPS 2001)

- Closely related to LLE
- New theoretical framework for understanding LLE

Hessian Eigenmaps: (Donoho & Grimes, 2003)

- Attempts to minimize local curvature of embedding
- Asymptotically accurate (more generally than IsoMap)

Charting: (Brand, NIPS 2002)

- Manifold learning as density estimation
- Greater robustness to noisy or sparsely sampled data

Problem Formulation

$\mathcal{Y} \subset \mathbb{R}^q$ $\longrightarrow$ coordinate space

$\rho : \mathcal{Y} \to \mathbb{R}^p$ $\longrightarrow$ smooth mapping (p > q)

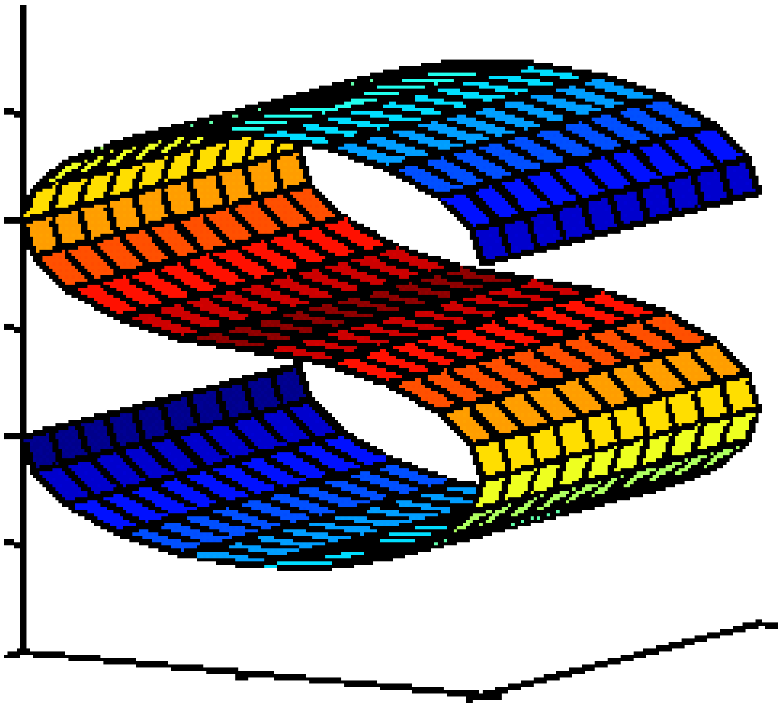
$\mathcal{X} = \rho(\mathcal{Y})$ $\longrightarrow$ manifold of dimension q

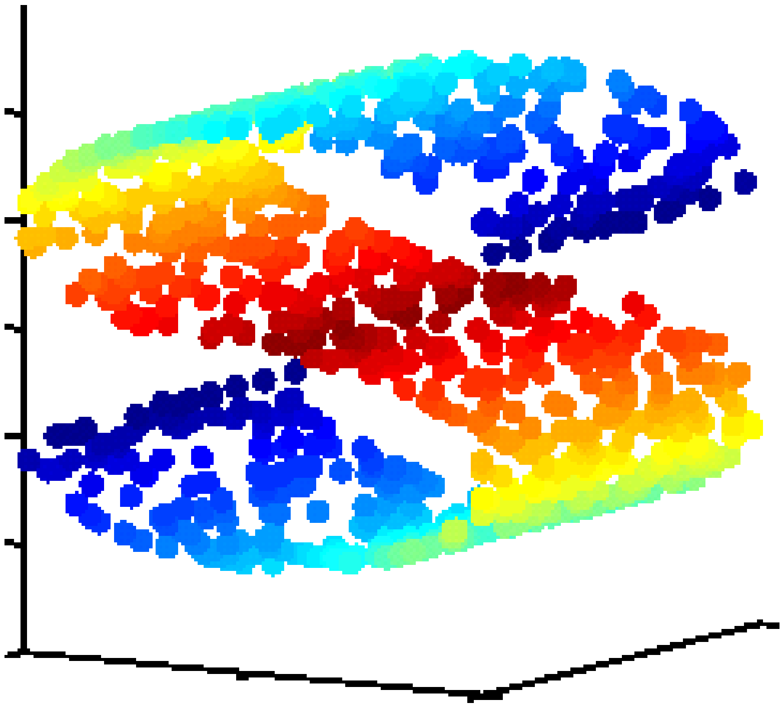
Embedding: Given data $X = [x_1, x_2, \ldots, x_n]$ from the manifold, estimate the corresponding coordinate points $Y = [y_1, y_2, \ldots, y_n]$

Mapping: Given data $X = [x_1, x_2, \ldots, x_n]$ from the manifold, determine an approximation to the embedding function ρ and its inverse

$\longrightarrow$ Both problems only solvable up to a rigid transform (translation, rotation, scale, reflection)

Canonical Problem: Swiss Roll





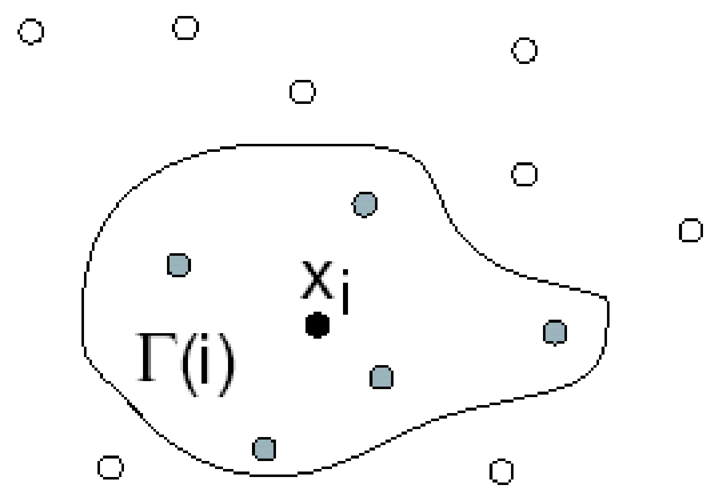
Locally Linear Embedding: Step 1

Find Nearest Neighbors

$\Gamma(i) =$ set of neighbors of variable x_i

- K nearest neighbors
- Points within a ball of radius ε

Manifold should be approximately linear within this local neighborhood



Locally Linear Embedding: Step 2

Determine Reconstruction Weights

• Determine the weights W which best reconstruct each point in terms of its local neighbors by minimizing

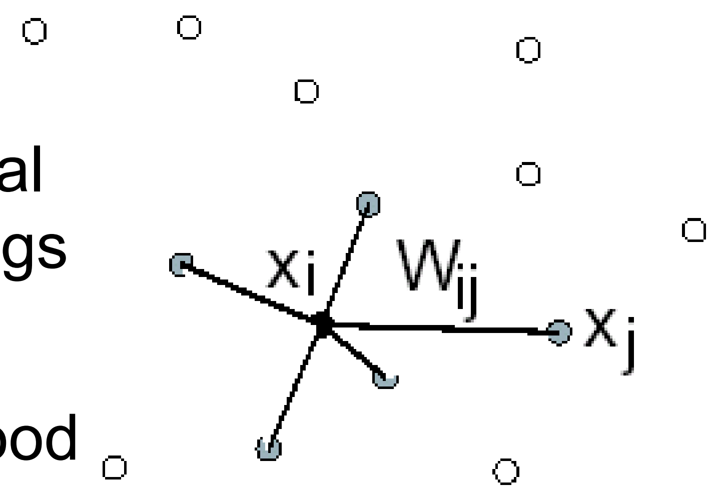
$$\Psi_{\mathrm{lle}}(W) = \sum_{i=1}^{n} ||x_i - \sum_{j \in \Gamma(i)} W_{ij} x_j||^2$$

subject to

$$\sum_{j \in \Gamma(i)} W_{ij} = 1 \qquad \text{for all } i$$

• Optimal weights invariant to local translations, rotations, and scalings

• Optimize via decoupled least squares in each local neighborhood



Locally Linear Embedding: Step 3

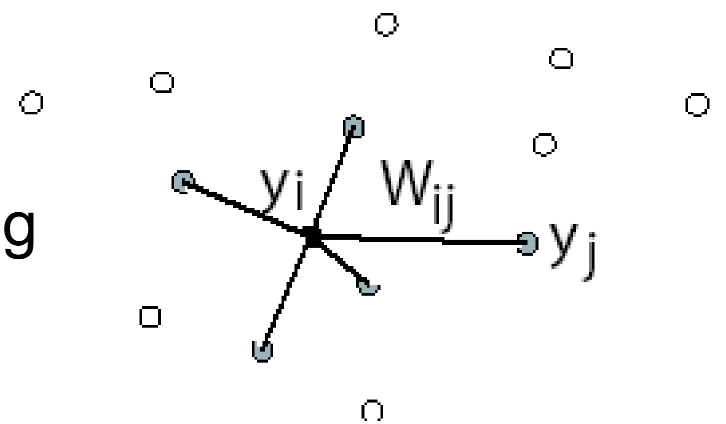
Find Low-Dimensional Embedding

- Determine the embedding points Y which best respect the local relationships captured by W. Minimize

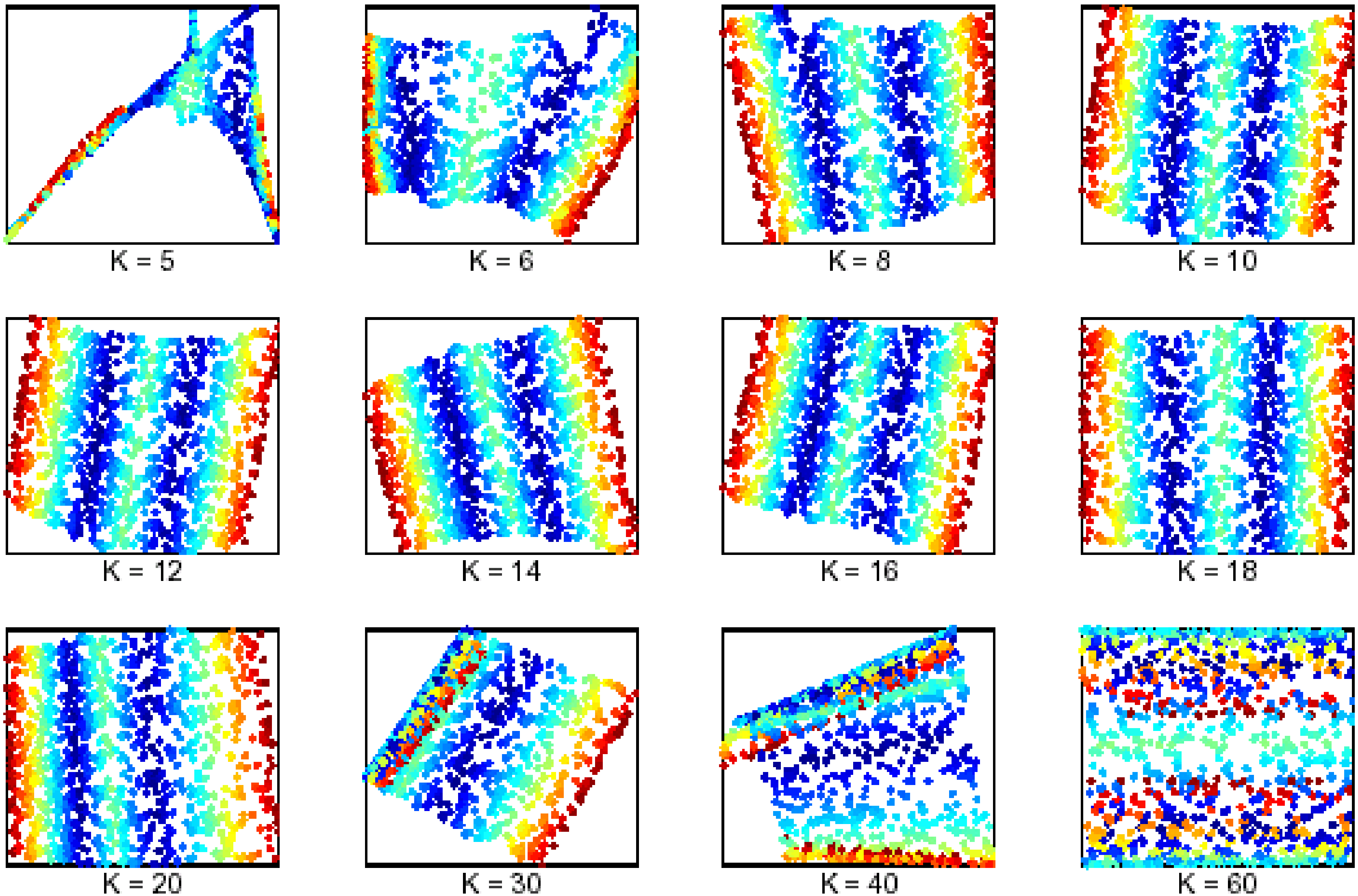
$$\Phi_{\text{lle}}(Y) = \sum_{i=1}^{n} ||y_i - \sum_{j \in \Gamma(i)} W_{ij} y_j||^2 = \text{trace}(Y(I-W)^T(I-W)Y)$$

subject to $\sum_{i=1}^{n} y_i = 0$ $\quad \frac{1}{n}\sum_{i=1}^{n} y_i y_i^T = I$

- Optimal solution takes rows of Y as the eigenvectors of $(I-W)^T(I-W)$ with minimal eigenvalues, excluding the constant eigenvector



LLE Swiss Roll Embedding



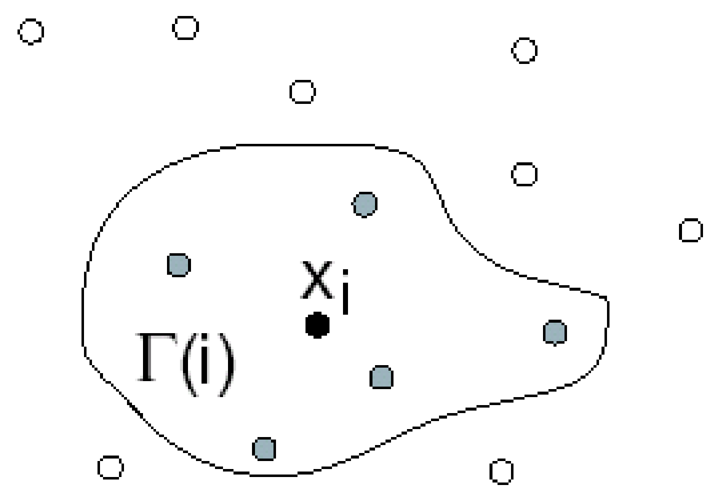
Laplacian Eigenmaps: Step 1

Find Nearest Neighbors

$\Gamma(i) =$ set of neighbors of variable x_i

- K nearest neighbors
- Points within a ball of radius ε

Neighborhoods must be chosen to be symmetric; otherwise like LLE



Laplacian Eigenmaps: Step 2

Construct Weighted Adjacency Matrix

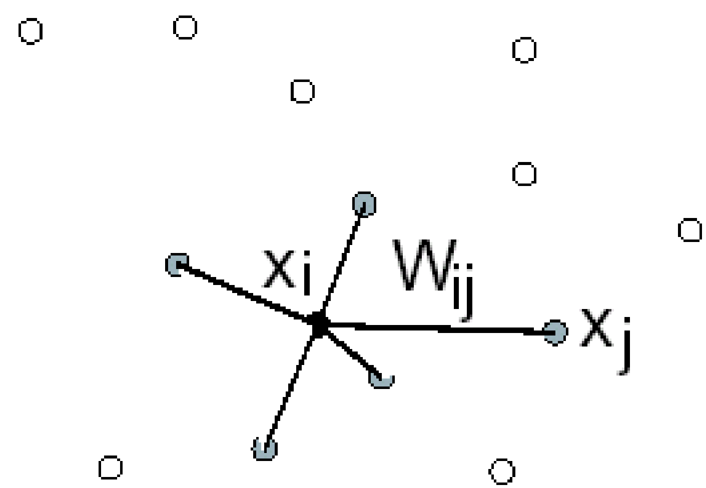
- Construct a symmetric matrix W as follows:

$$W_{ij} = \exp\left\{-\frac{1}{2\sigma^2}||x_i - x_j||^2\right\} \qquad \text{if} \quad j \in \Gamma(i)$$

$$W_{ij} = 0 \qquad \text{otherwise}$$

- Weights largest for nearby points

- Bandwidth σ should be chosen to match radius within which the manifold is approximately linear



Laplacian Eigenmaps: Step 3

Find Embedding From Normalized Laplacian

$D \longrightarrow$ diagonal matrix of row/col sums of W

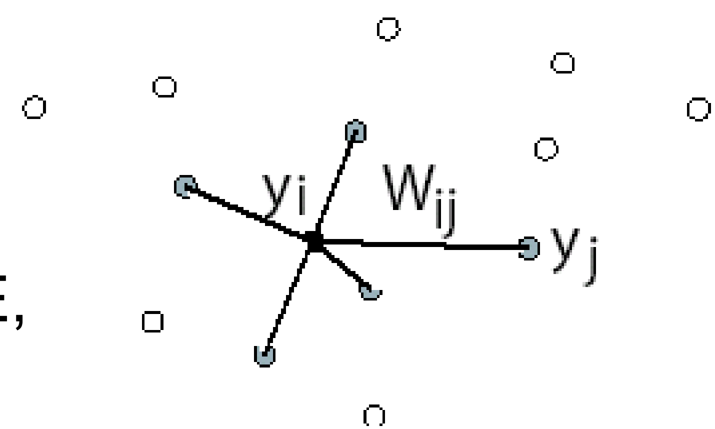
$\tilde{W} = D^{-1/2} W D^{-1/2} \longrightarrow$ normalized weights

$$\Phi_{\text{lap}}(Y) = \sum_{i=1}^{n} \sum_{j \neq i} \tilde{W}_{ij} ||y_i - y_j||^2 = \frac{1}{2} \text{trace}(Y^T L Y) \qquad L = I - \tilde{W}$$

- L is the *normalized Laplacian* of the graph with weighted adjacency W

- Imposing the same constraints as LLE, we again set Y to be the eigenvectors with smallest values

$\longrightarrow$ Third stage equivalent to LLE, but uses different weights W



Locality-Preserving Functions

$T_x(\mathcal{X}) \longrightarrow$ tangent space of manifold at $x \in \mathcal{X}$

- Affine subspace spanned by vectors tangent to the manifold, and passing through the point x

$f : \mathcal{X} \to \mathbb{R} \quad \longrightarrow$ smooth mapping of manifold to real line

- Using the (non-unique) coordinate system that $T_x(\mathcal{X})$ inherits from the manifold, we may compute $\nabla f(x)$
- Although gradient not uniquely defined, norm is, and one can derive the following bound:

$$|f(z) - f(x)| \leq ||\nabla f(x)|| \; ||z - x|| + o(||z - x||)$$

Laplace-Beltrami Operator

- Functions f which map neighboring points to nearby locations should have small

$$\widetilde{\Phi}_{\mathrm{lap}}(f) = \int_{\mathcal{X}} ||\nabla f||^2 = \int_{\mathcal{X}} \Delta(f) f$$

$\Delta(f) \longrightarrow$ Laplacian on manifold (the Laplace-Beltrami operator), defined as

$$\Delta(f) = \sum_i \frac{\partial^2 f}{\partial z_i^2}$$ in tangent space coordinates z_i

- Optimal solutions f are eigenfunctions of $\Delta(f)$
- Laplacian Eigenmaps is discrete implementation of this, where the Gaussian weight function may be motivated by an analysis of the operator in infinitesimal neighborhoods

Laplacian Eigenmaps & LLE

- Laplacian Eigenmaps & LLE differ only in choice of W
- From Taylor approximation, LLE weights satisfy

$$(I-W)f(x)\Big|_{x=x_i} \approx -\frac{1}{2}\sum_{j\in\Gamma(i)} W_{ij}(x_i-x_j)^T H_i(x_i-x_j)$$

$H_i \longrightarrow$ Hessian of f at x_i

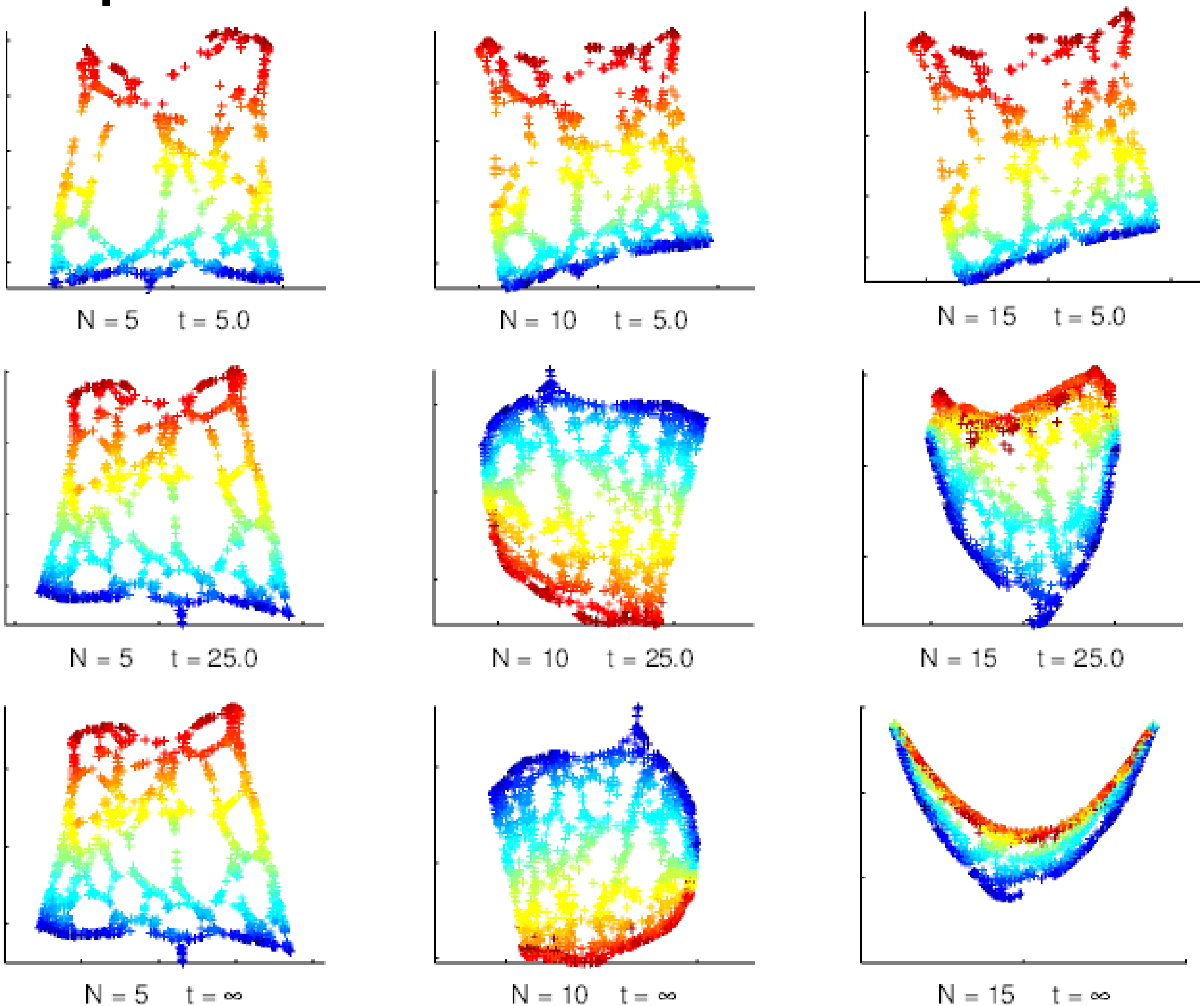
- If vectors to neighboring points form an orthonormal basis for the tangent space defined by H_i, we have

$$\sum_{j\in\Gamma(i)} W_{ij}(x_i-x_j)^T H_i(x_i-x_j) = \mathrm{trace}(H_i) = \Delta(f)$$

- Also, if neighbors uniformly distributed in orientation, expectation of LLE's operator is proportional to $\Delta(f)$

$\longrightarrow$ Equivalent in limit, but for finite data??

Laplacian Swiss Roll Embedding



Hessian Eigenmaps

- Asymptotic Convergence Guarantees
- Theory underlying these guarantees
- Discrete implementation (inspired by LLE)

IsoMap Convergence

$d_{\mathcal{X}}(x, x') \longrightarrow$ geodesic distance between x and x' (shortest path within manifold $\mathcal{X}$)

Given an infinite number of samples from a strictly positive distribution over the manifold, IsoMap will recover the true coordinates (up to a rigid transform) assuming

Isometry: For *all* points x, x' on the manifold,

$$d_{\mathcal{X}}(x, x') = ||y - y'|| \qquad x = \rho(y) \quad x' = \rho(y')$$

Convexity: Coordinate space $\mathcal{Y} \subset \mathbb{R}^q$ is convex

$\longrightarrow$ Poor assumption for image manifolds?

Hessian Eigenmaps Convergence

$d_{\mathcal{X}}(x, x') \longrightarrow$ geodesic distance between x and x' (shortest path within manifold $\mathcal{X}$)

Given an infinite number of samples from a strictly positive distribution over the manifold, Hessian Eigenmaps will recover the true coordinates (up to a rigid transform) assuming

Local Isometry: For sufficiently close x, x' on the manifold,

$$d_{\mathcal{X}}(x, x') = ||y - y'|| \qquad x = \rho(y) \quad x' = \rho(y')$$

Connectedness:

Coordinate space $\mathcal{Y} \subset \mathbb{R}^q$ is open and connected

Hessians in Euclidean Space

$$g : \mathcal{Y} \to \mathbb{R} \qquad \left(H_g^{\text{euc}}\right)_{i,j}(y) = \frac{\partial^2 g(y)}{\partial y_i \partial y_j}$$

Consider the following functional defined for smooth g:

$$\mathcal{H}^{\text{euc}}(g) = \int_{\mathcal{Y}} ||H_g^{\text{euc}}(y)||_F^2 \, dy$$

- Nullspace of this operator consists of all smooth functions with everywhere vanishing Hessian
- This is precisely the space of all affine functions, or the span of the constant function and the q coordinate functions

Basis for $\text{null}\,\mathcal{H}^{\text{euc}}(g)$ $\Longleftrightarrow$ Basis for $\mathcal{Y} \subset \mathbb{R}^q$

Tangent Hessians

$f : \mathcal{X} \to \mathbb{R}$ $\quad$ $H_f^{\mathrm{tan}}(x) =$ Hessian in tangent space at $x \in \mathcal{X}$

• Hessian depends on choice of tangent coordinate system, but its Frobenius norm is unique, so the following functional is well defined:

$$\mathcal{H}(f) = \int_{\mathcal{X}} ||H_f^{\mathrm{tan}}(x)||_F^2 \, dx$$

Key Result

Basis for null $\mathcal{H}^{\mathrm{euc}}(g)$ $\Longleftrightarrow$ Basis for null $\mathcal{H}(f)$

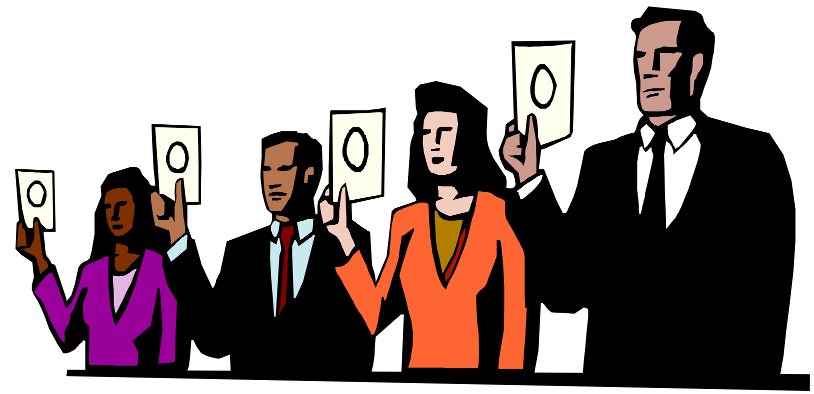
under the correspondence $g = f \circ \rho$

$\longrightarrow$ Find coordinates from empirical estimate of null space of $\mathcal{H}(f)$

Failures of Laplacian Eigenmaps

Laplacian Eigenmaps (and asymptotically LLE) can be seen as minimizing an empirical version of

$$\mathcal{L}(f) = \int_{\mathcal{X}} (\Delta(f))^2$$



• Norm of Laplacian is *not* independent of choice of coordinate frame – is this even well defined?

• There exist functions with zero Laplacian which are not affine (for example $f(x_1,x_2) = x_1 x_2$)

$\longrightarrow$ Searching for a basis for this nullspace may produce axes distorted by these spurious functions

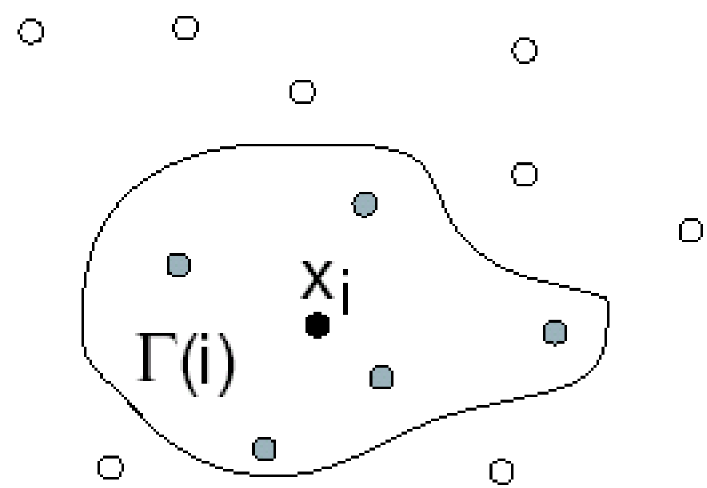
Hessian Eigenmaps: Step 1

Find Nearest Neighbors

$\Gamma(i) =$ set of neighbors of variable x_i

- K nearest neighbors
- Points within a ball of radius ε

Identical to LLE



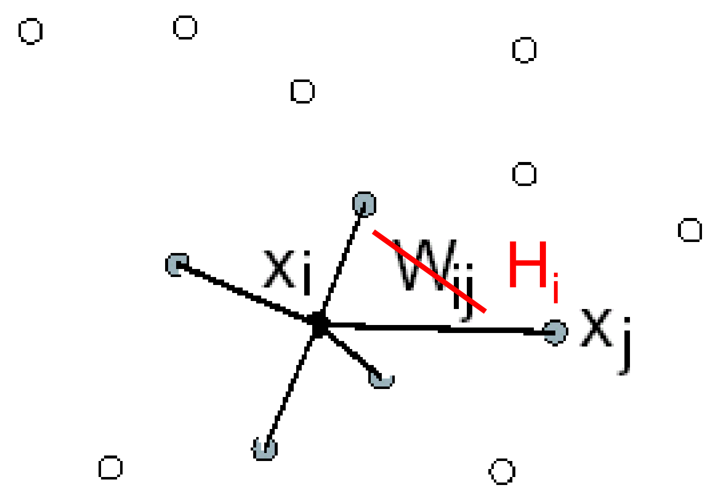
Hessian Eigenmaps: Step 2

Estimate Tangent Hessians

• For each x_i, perform PCA analysis of neighboring points to find best-fitting q-dimensional linear subspace.

⟶ Estimate of tangent space at x_i

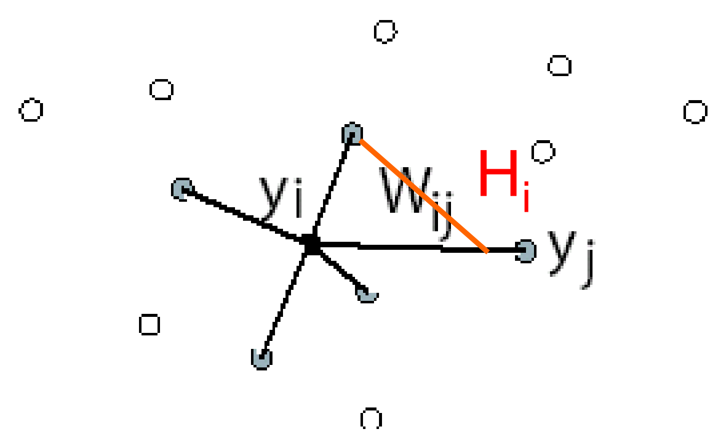
• Compute least-squares Hessian estimate based on projection of all points to this tangent space



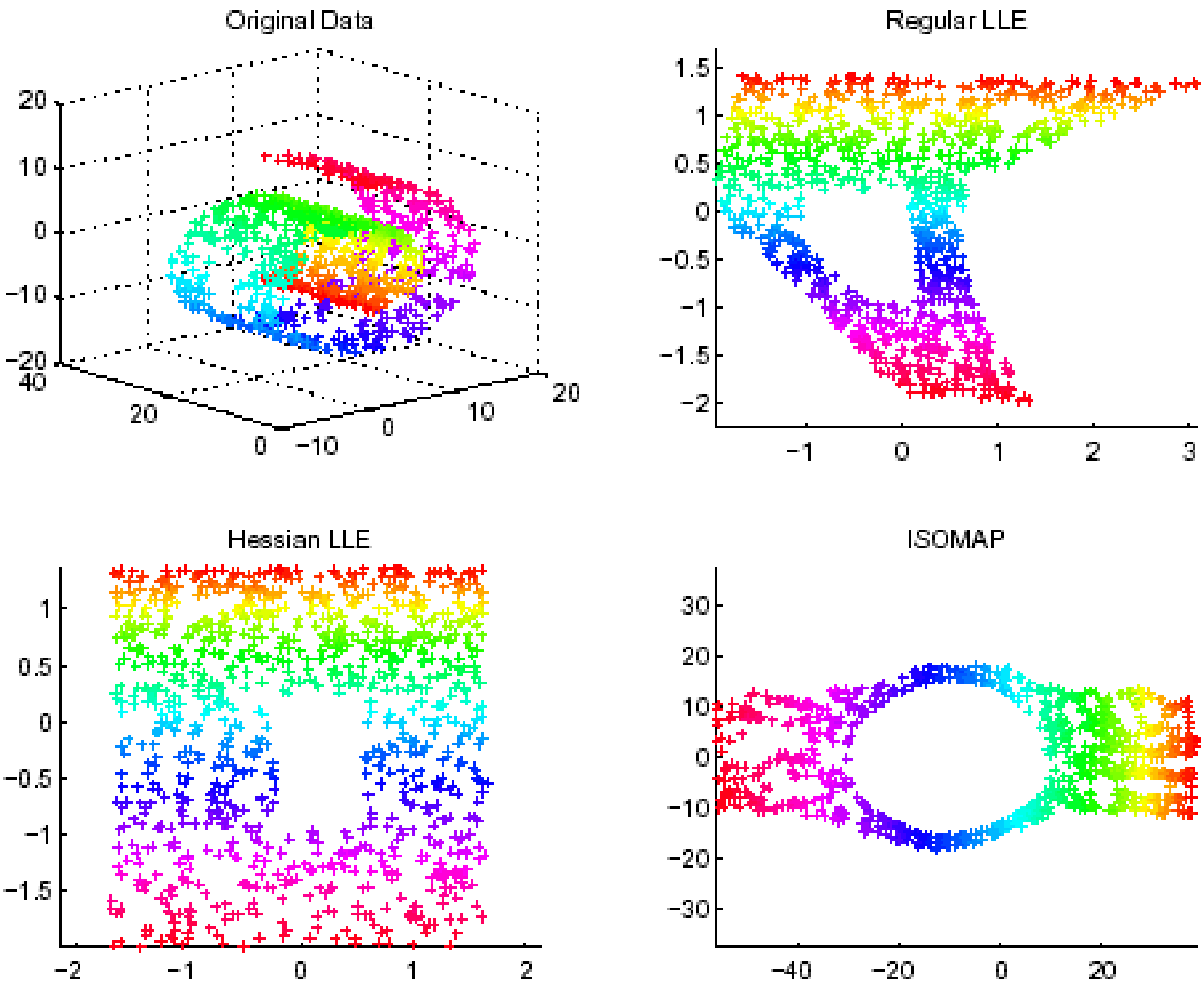
Hessian Eigenmaps: Step 3

Find Low-Dimensional Embedding

- Use discretization implied by data points, and local Hessian estimates H_i, find discrete approximation to $\mathcal{H}(f)$
- Optimal solution takes rows of Y as the eigenvectors of $\mathcal{H}(f)$ with minimal eigenvalues, excluding the constant eigenvector



Hessian Swiss Roll Embedding



Issues with Hessian Eigenmaps

Has strong asymptotic guarantees, but…

- Assumes data points lie precisely on some manifold

⟶ For real data sets, we can at most hope that the data is "close" (in a probabilistic sense) to a manifold

- Requires local, empirical estimates of differential operators

⟶ What happens to these estimates when the manifold is sparsely and/or irregularly sampled?

Charting a Manifold

Manifold Learning as Density Estimation

• Fit Gaussian mixture model to input data. The fitted Gaussians implicitly define local coordinate systems

• Estimate a consistent mapping of these local coordinate systems to a single low-dimensional space

→ The posterior distributions of this density model then define a mapping of all high-dimensional points to the manifold, rather than just an embedding of the given data

Charting: Step 1

Soft Nearest Neighbor Assignment

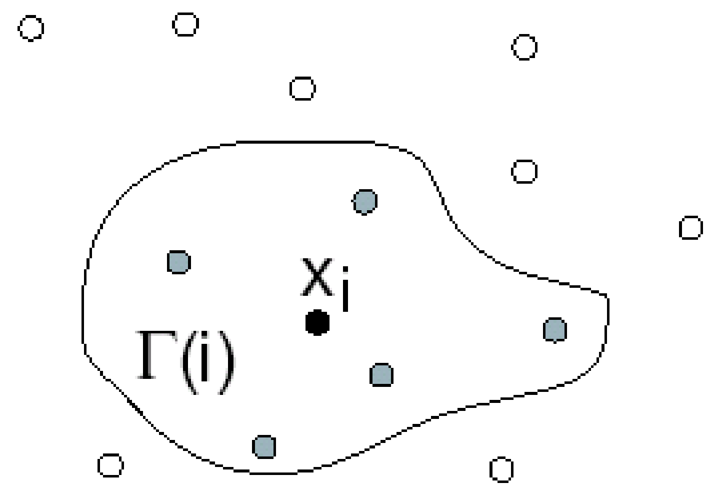
For each x_i, assign a weight to all other points:

$$W_{ij} = \exp\left\{-\frac{1}{2\sigma^2}||x_i - x_j||^2\right\}$$

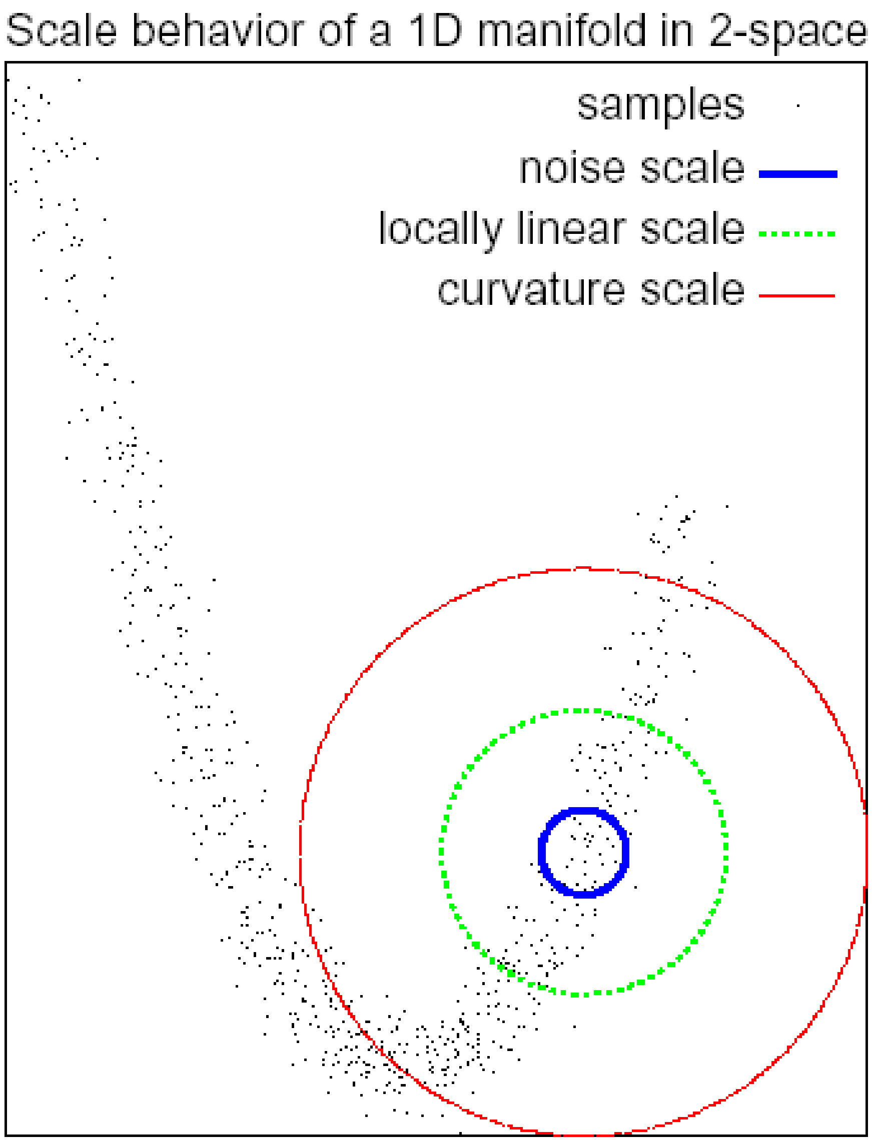
By setting very small weights to zero, we implicitly choose

$\Gamma(i) =$ set of neighbors of variable x_i

- Similar to use of ε ball neighborhoods in other methods
- Relationship to weights in Laplacian eigenmaps?



Automatic Local Scale Selection



$n(r) \rightarrow$ number of neighbors in ball of radius r

$$n(r) \propto r^q$$

only at locally linear scale

- Noise at small scales
- Curvature at large scales

Charting: Step 2

Fit Gaussian Mixture Model

$$p(x|\Lambda) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(x; x_i, \Lambda_i)$$

- ML covariance estimate is degenerate (zero-variance)
- Regularize using prior which constrains nearby mixture components to model similar subspaces:

$$p(\Lambda) = \alpha \,\mathrm{exp}\left\{-\sum_{i\neq j} W_{ij} D(\mathcal{N}(x; x_i, \Lambda_i)\,||\mathcal{N}\big(x; x_j, \Lambda_j\big))\right\}$$

- MAP covariance matrix has closed form solution specified by a system of linear equations

$\longrightarrow$ Robustness by coupling local coordinate estimates

Charting: Step 3

Connect Local Charts

U_k → projection of all data points onto the local coordinate frame (mixture component) centered on x_k

G_k → affine transform mapping local coordinates around x_k to a common, global set of q-dim. coordinates

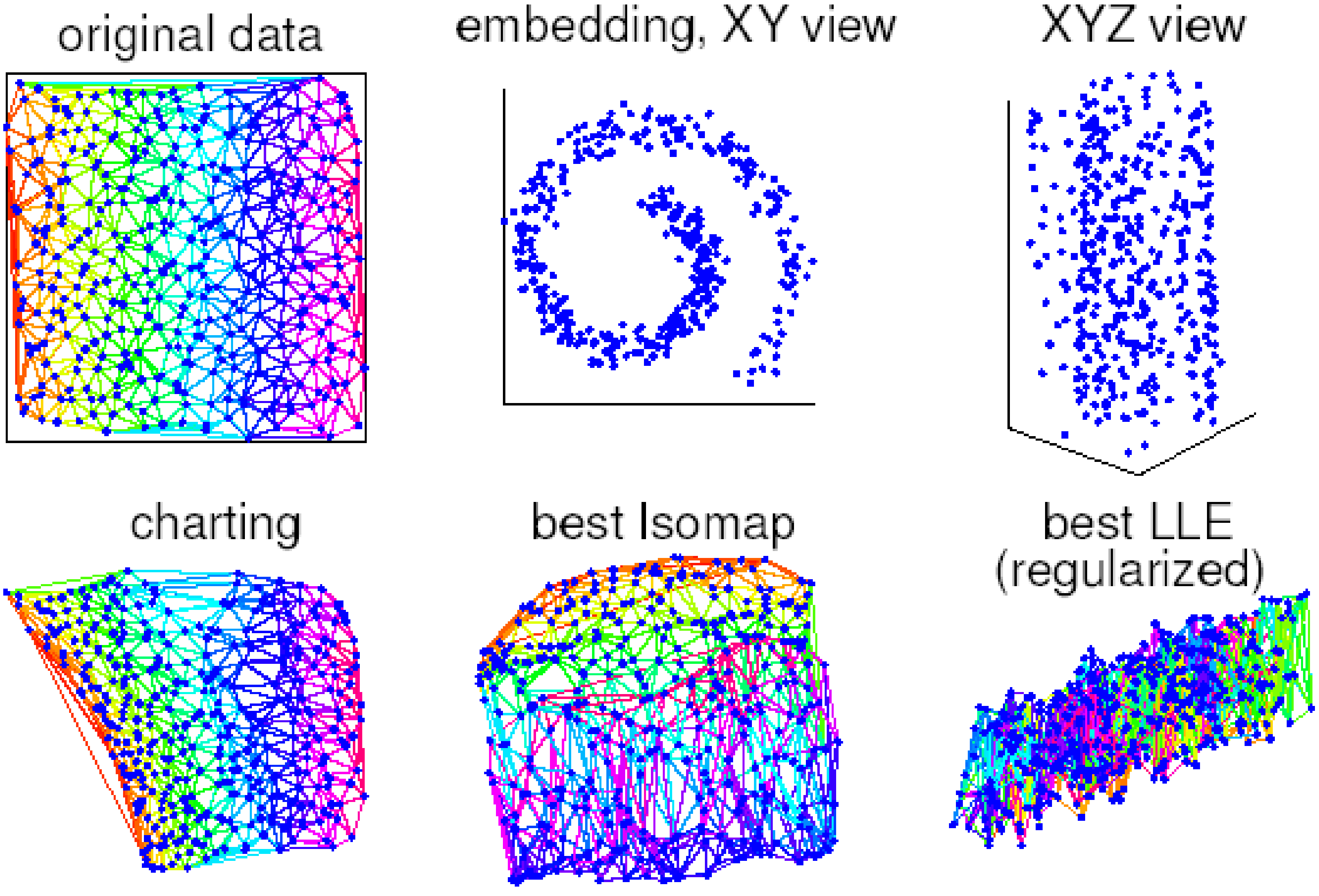
$$\Phi_{\text{chart}}(G) = \sum_{k \neq j} \sum_{i=1}^{n} p_k(x_i) p_j(x_i) \left\| G_k \begin{bmatrix} u_{ki} \\ 1 \end{bmatrix} - G_j \begin{bmatrix} u_{ji} \\ 1 \end{bmatrix} \right\|_F^2$$

$p_k(x_i)$ → posterior probability that x_i sampled from Gaussian centered on x_k

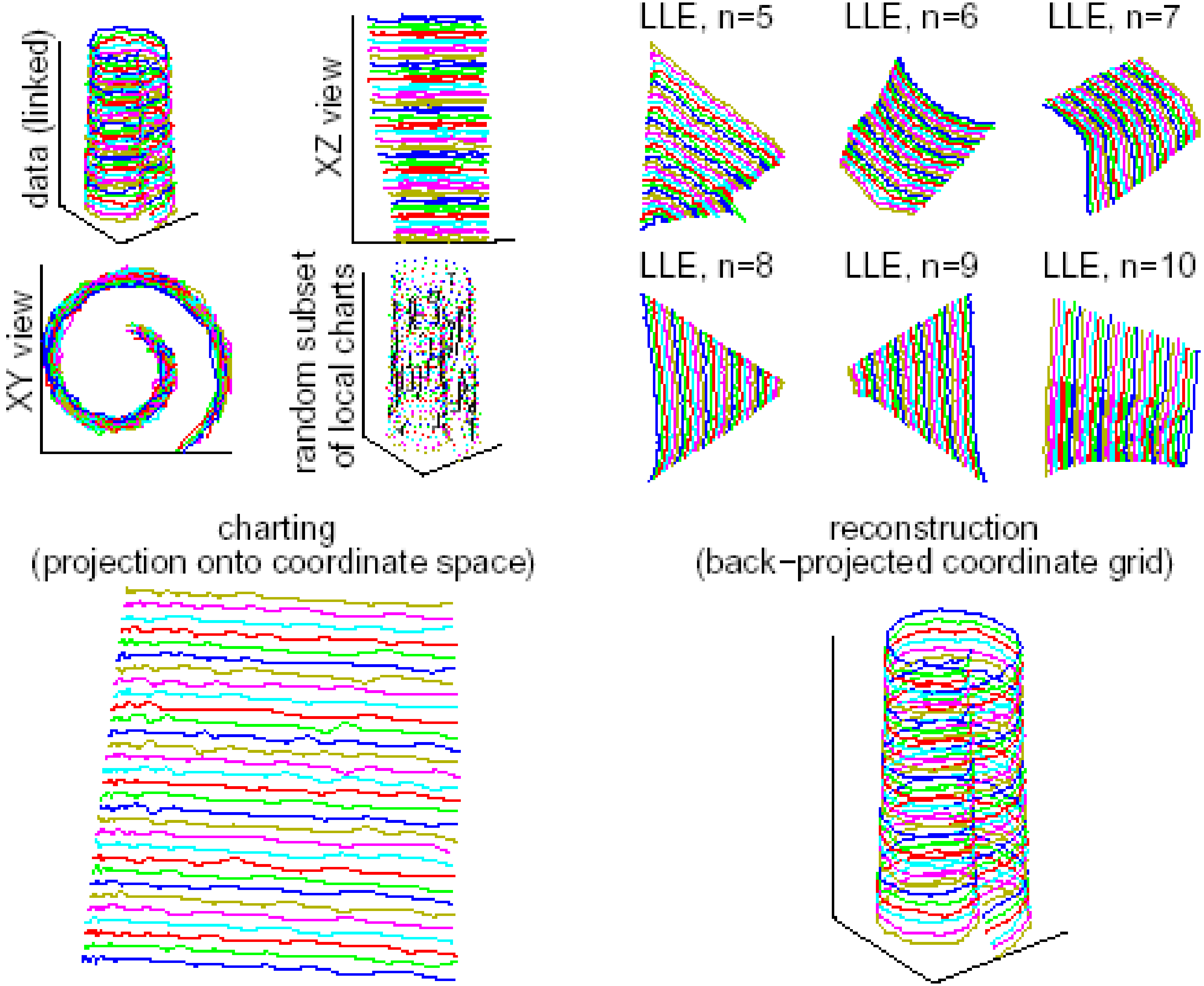
$$\Phi_{\text{chart}}(G) = \text{trace}(GQQ^T G^T)$$

Again find mapping via smallest eigenvectors

Charting a Sparse Swiss Roll



Charting Gives a Mapping



Open Research Directions

- Do probabilistic methods like charting every asymptotically recover the true manifold geometry?
- How does charting's use of Gaussian weight functions relate to the Laplacian eigenmap weights?
- Can probabilistic methods be used to improve the second derivative estimates used by Hessian LLE?
- Can we develop methods which avoid the need to identify linear patches and compute discrete derivative estimates?
- Can prior knowledge be used to regularize the learning of very sparsely sampled manifolds?