Geometric Methods and Manifold Learning

Mikhail Belkin, Ohio State University,

prepared jointly with Partha Niyogi

High Dimensional Data

When can we avoid the curse of dimensionality?

- Smoothness

  rate $\approx (1/n)^{\frac{s}{d}}$

  splines,kernel methods, L_2 regularization...

- Sparsity

  wavelets, L_1 regularization, LASSO, compressed sensing..

- **Geometry**

  graphs, simplicial complexes, laplacians, diffusions

Geometry and Data: The Central Dogma

- Distribution of natural data is non-uniform and concentrates around low-dimensional structures.
- The shape (geometry) of the distribution can be exploited for efficient learning.

Manifold Learning

Learning when data $\sim \mathcal{M} \subset \mathbb{R}^N$

- Clustering: $\mathcal{M} \to \{1, \ldots, k\}$

  connected components, min cut

- Classification: $\mathcal{M} \to \{-1, +1\}$

  P on $\mathcal{M} \times \{-1, +1\}$

- Dimensionality Reduction: $f : \mathcal{M} \to \mathbb{R}^n \quad n << N$

- $\mathcal{M}$ unknown: what can you learn about $\mathcal{M}$ from data?

  e.g. dimensionality, connected components
  holes, handles, homology
  curvature, geodesics

Formal Justification

- Speech

  speech $\in l_2$ generated by vocal tract

  Jansen and Niyogi (2005)

- Vision

  group actions on object leading to different images

  Donoho and Grimes (2004)

- Robotics

  configuration spaces in joint movements

- Graphics

Manifold + Noise may be generic model in high dimensions.

Take Home Message

- Geometrically motivated approach to learning

  nonlinear, nonparametric, high dimensions

- Emphasize the role of the Laplacian and Heat Kernel
  - Semi-supervised regression and classification
  - Clustering and Homology
  - Randomized Algorithms and Numerical Analysis

Principal Components Analysis

Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^D$
Find $y_1, \ldots, y_n \in \mathbb{R}$ such that

$$y_i = \mathbf{w} \cdot \mathbf{x}_i$$

and

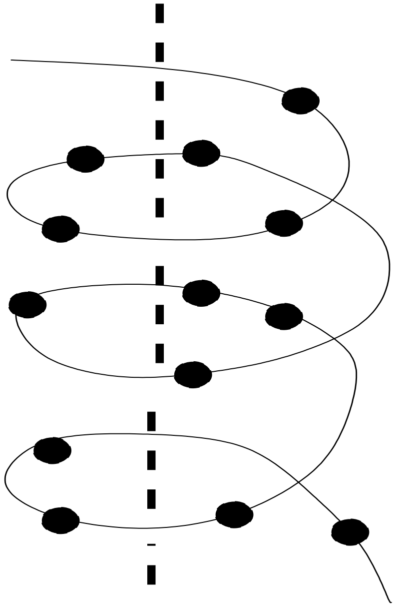
$$\max_{\mathbf{w}} \text{Variance}(\{y_i\}) = \sum_i y_i^2 = \mathbf{w}^T \left(\sum_i \mathbf{x}_i \mathbf{x}_i^T \right) \mathbf{w}$$

$$\mathbf{w}_* = \text{leading eigenvector of } \sum_i x_i x_i^T$$

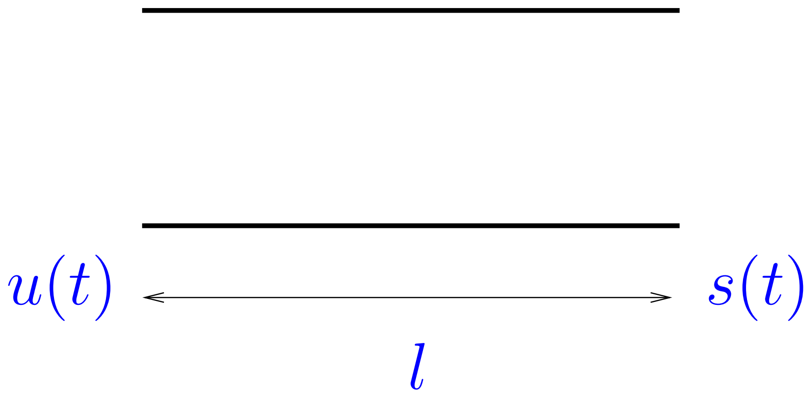
Manifold Model

Suppose data does not lie on a linear subspace.

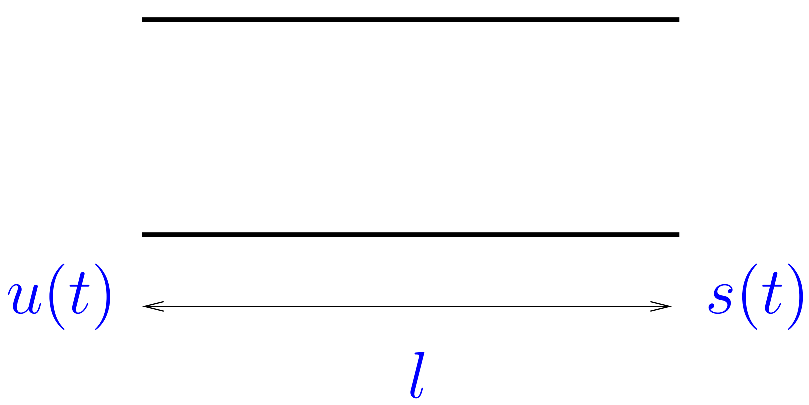
Yet data has inherently one degree of freedom.



An Acoustic Example



An Acoustic Example



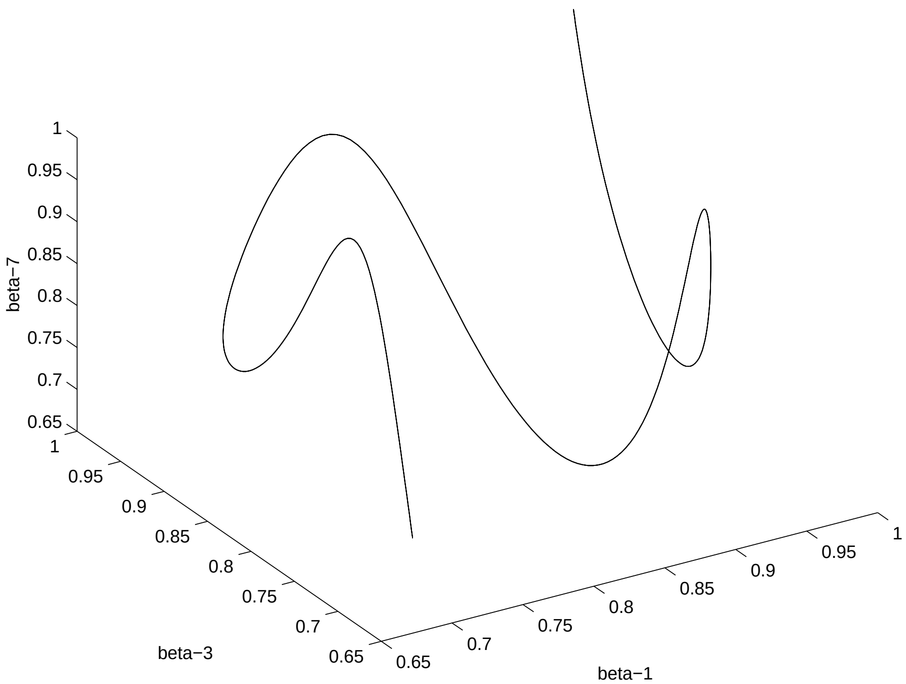
One Dimensional Air Flow

(i) $\frac{\partial V}{\partial x} = -\frac{A}{\rho c^2}\frac{\partial P}{\partial t}$

(ii) $\frac{\partial P}{\partial x} = -\frac{\rho}{A}\frac{\partial V}{\partial t}$

$V(x,t) =$ volume velocity
$P(x,t) =$ pressure

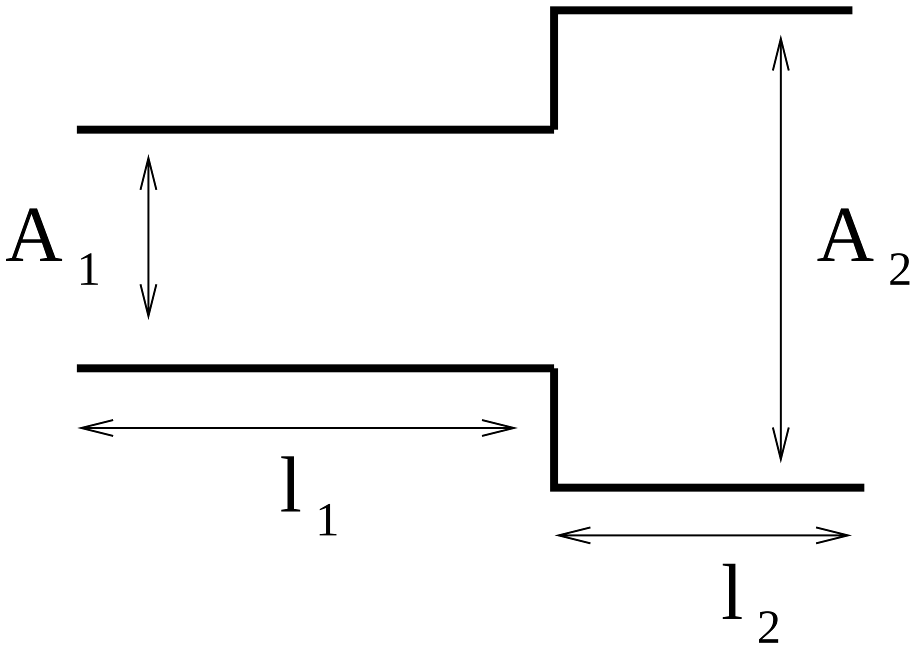
Solutions



$$u(t) = \sum_{n=1}^{\infty} \alpha_n \sin(n\omega_0 t) \in l_2$$

$$s(t) = \sum_{n=1}^{\infty} \beta_n \sin(n\omega_0 t) \in l_2$$

Acoustic Phonetics

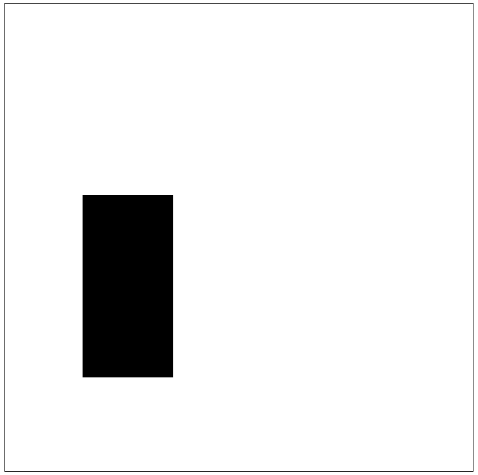


Vocal Tract modeled as a sequence of tubes.
(e.g. Stevens, 1998)

Vision Example

$$f : \mathbb{R}^2 \rightarrow [0, 1]$$

$$\mathcal{F} = \{f | f(x, y) = v(x - t, y - r)\}$$



Robotics

$$g : S^2 \times S^2 \times S^2 \to \mathbb{R}^3$$

$$\langle(\theta_1, \phi_1), (\theta_2, \phi_2), (\theta_3, \phi_3)\rangle \to (x, y, z)$$

Manifold Learning

Learning when data $\sim \mathcal{M} \subset \mathbb{R}^N$

- Clustering: $\mathcal{M} \to \{1, \ldots, k\}$

  connected components, min cut

- Classification/Regression: $\mathcal{M} \to \{-1, +1\}$ or $\mathcal{M} \to \mathbb{R}$

  P on $\mathcal{M} \times \{-1, +1\}$ or P on $\mathcal{M} \times \mathbb{R}$

- Dimensionality Reduction: $f : \mathcal{M} \to \mathbb{R}^n \quad n << N$

- $\mathcal{M}$ unknown: what can you learn about $\mathcal{M}$ from data?

  e.g. dimensionality, connected components
  holes, handles, homology
  curvature, geodesics

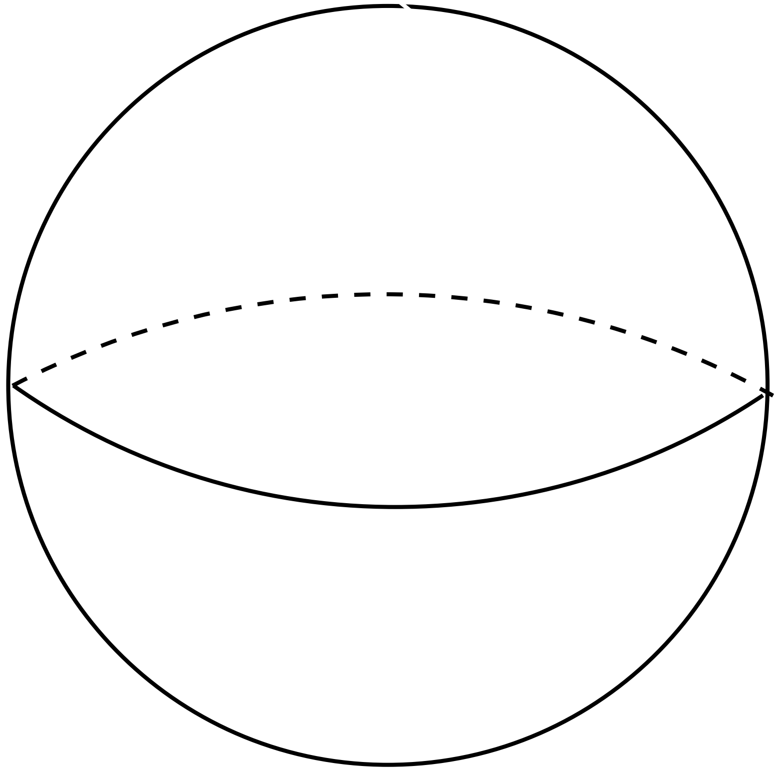
Differential Geometry

All you wanted to know about differential geometry but were afraid to ask, in 10 easy slides!

Embedded manifolds

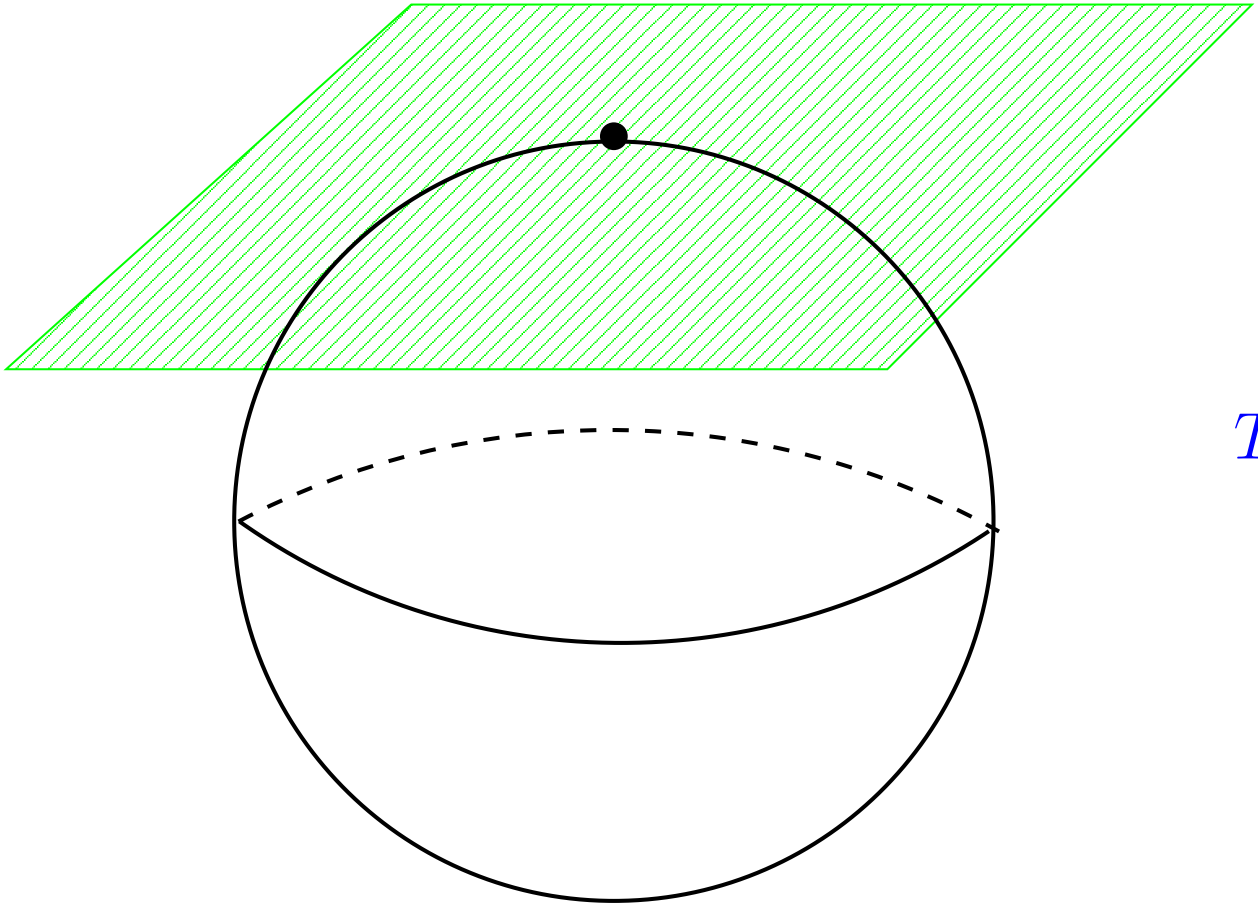
$$\mathcal{M}^k \subset \mathbb{R}^N$$

Locally (not globally) looks like Euclidean space.



$S^2 \subset \mathbb{R}^3$

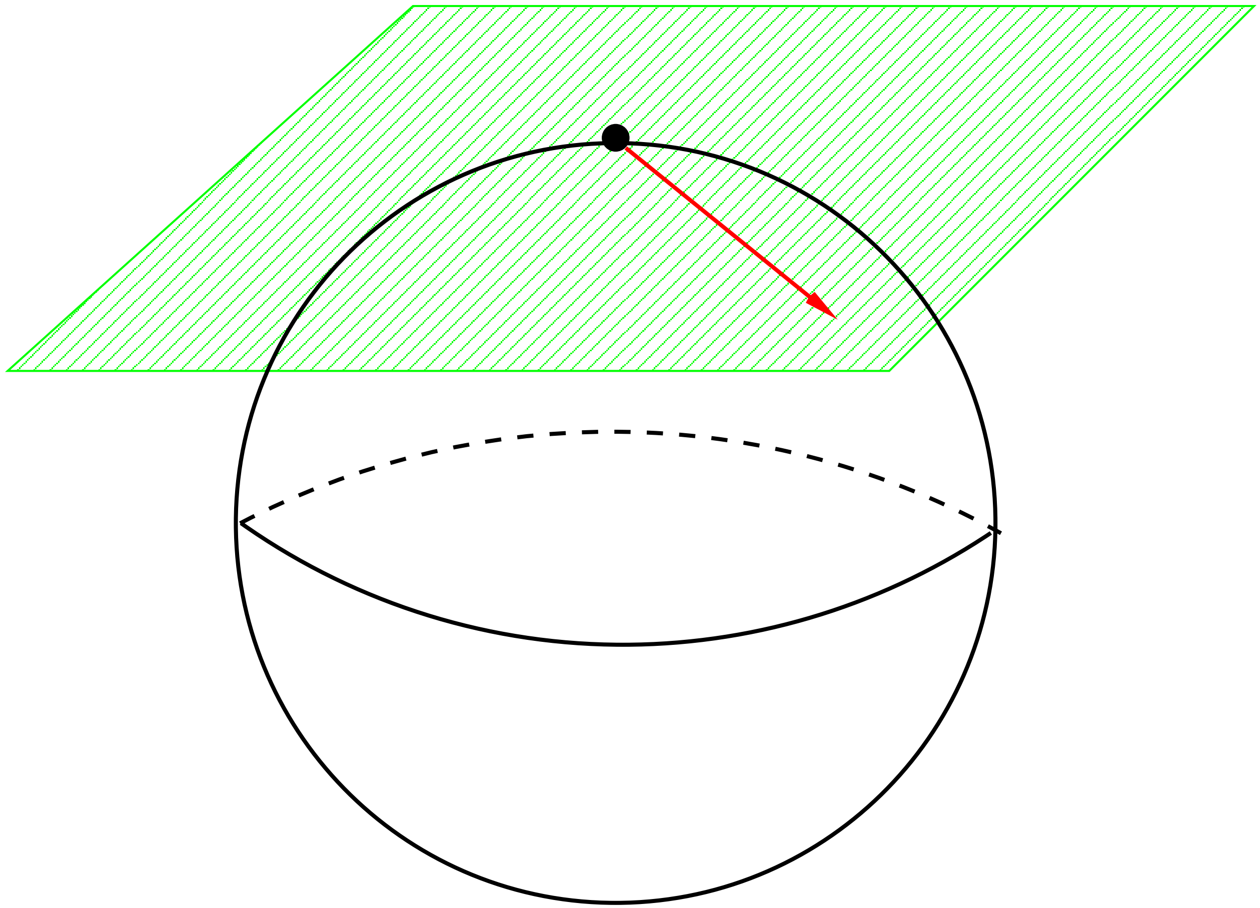
Tangent space



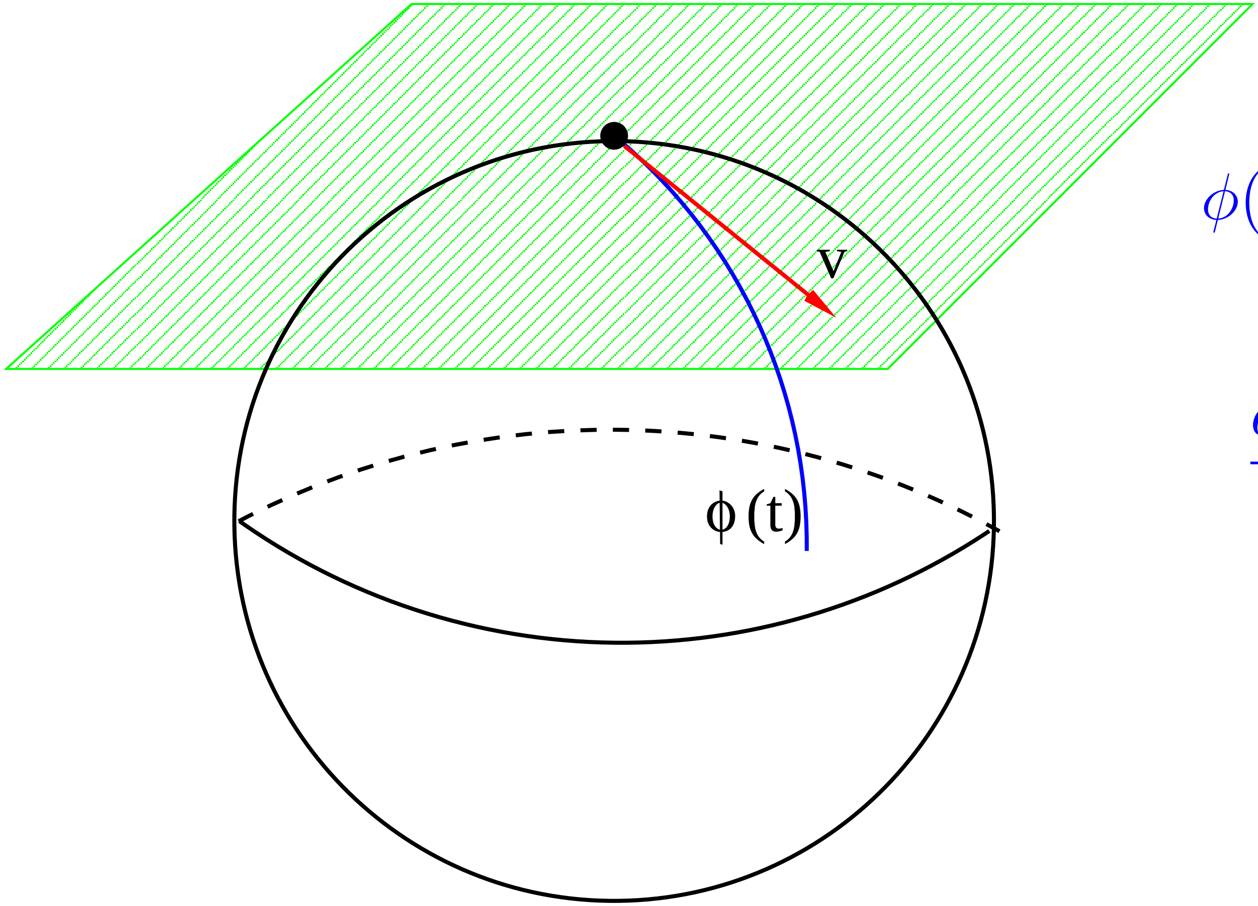
$$T_p\mathcal{M}^k \subset \mathbb{R}^N$$

k-dimensional affine subspace of $\mathbb{R}^N$.

Tangent vectors and curves



Tangent vectors and curves

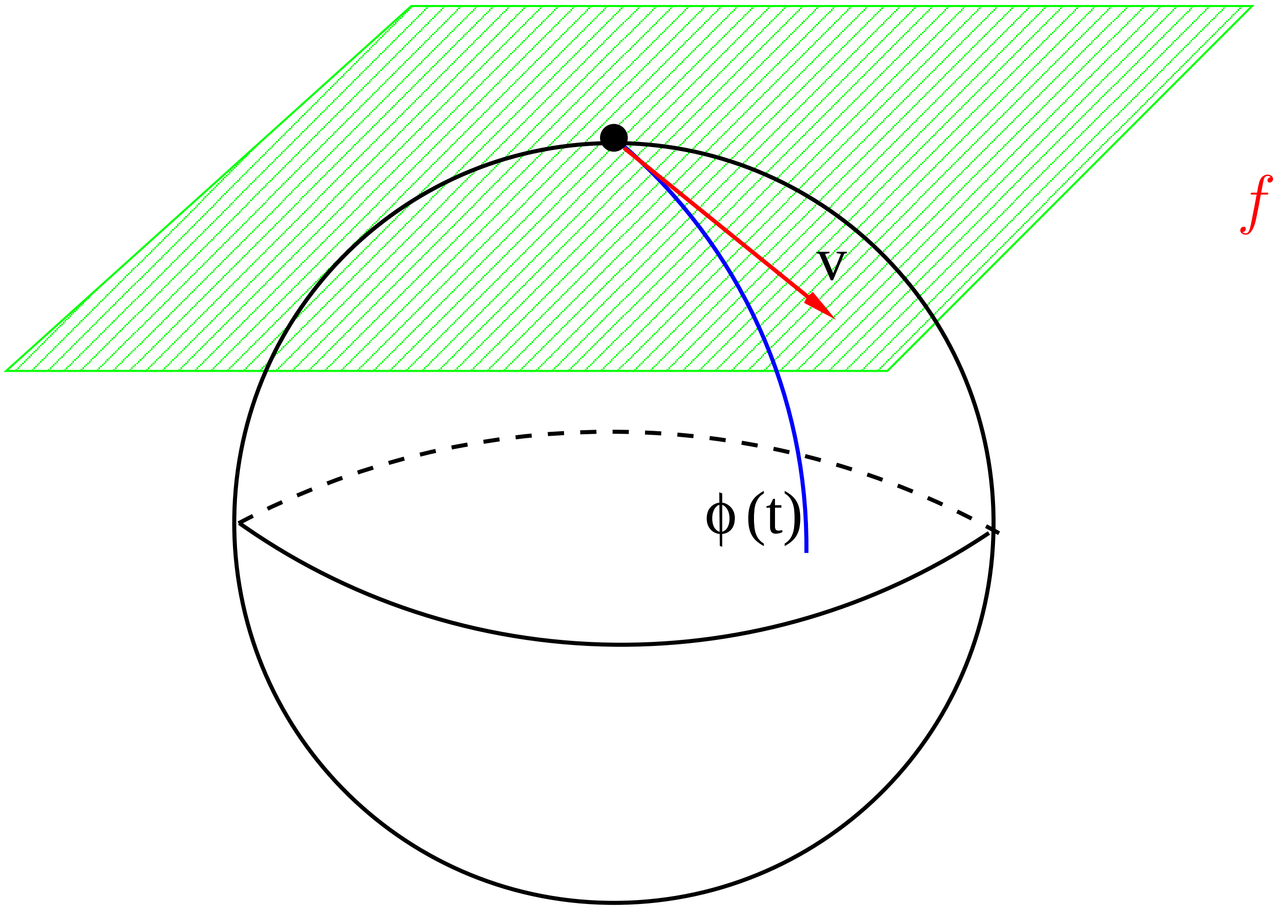


$$\phi(t) : \mathbb{R} \to \mathcal{M}^k$$

$$\left. \frac{d\phi(t)}{dt} \right|_0 = V$$

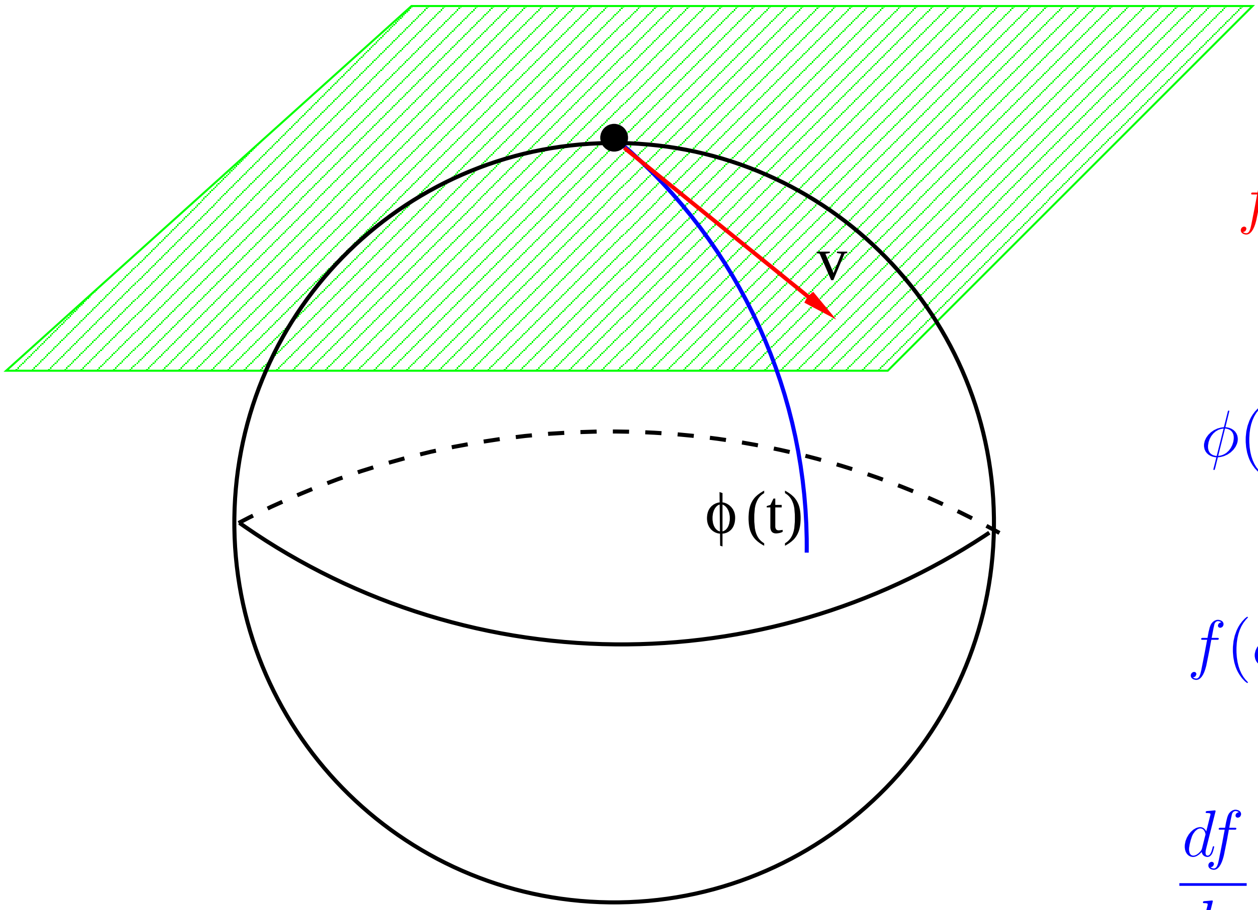
Tangent vectors <———> curves.

Tangent vectors as derivatives



$f : \mathcal{M}^k \to \mathbb{R}$

Tangent vectors as derivatives



$$f : \mathcal{M}^k \rightarrow \mathbb{R}$$

$$\phi(t) : \mathbb{R} \rightarrow \mathcal{M}^k$$

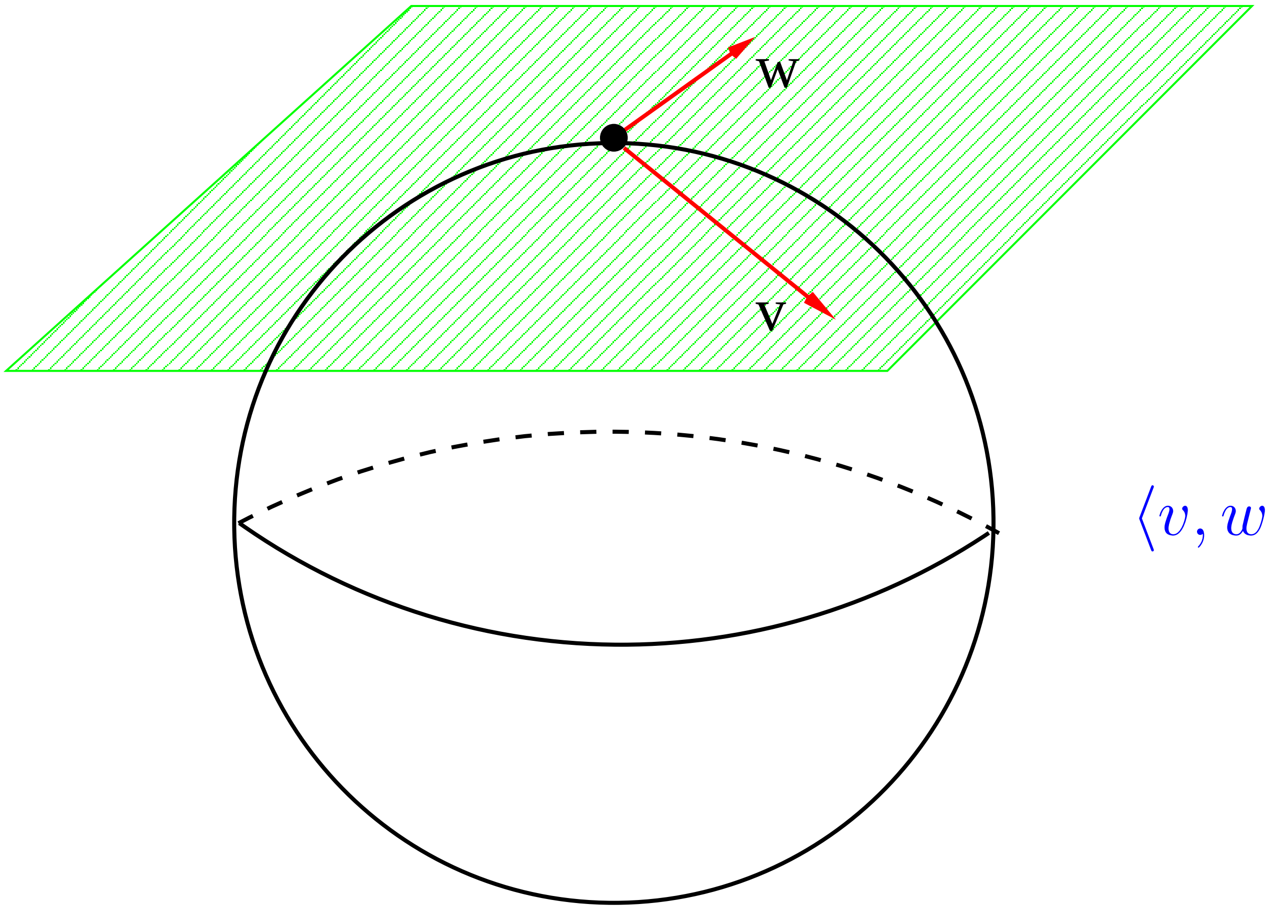
$$f(\phi(t)) : \mathbb{R} \rightarrow \mathbb{R}$$

$$\frac{df}{dv} = \left.\frac{d\, f(\phi(t))}{d\, t}\right|_0$$

Tangent vectors <———> Directional derivatives.

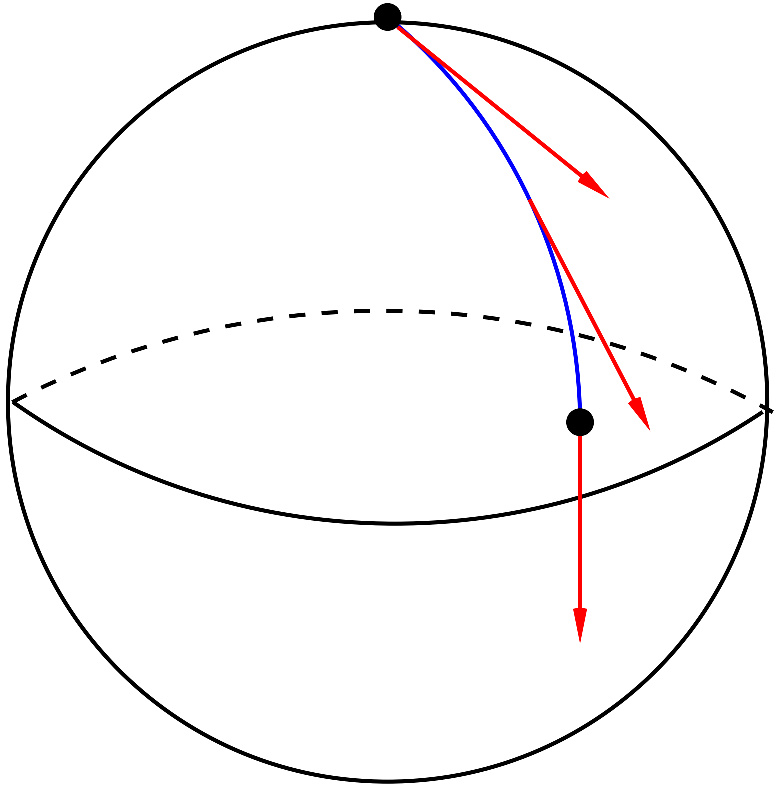
Riemannian geometry

Norms and angles in tangent space.



$\langle v, w \rangle \qquad \|v\|, \ \|w\|$

Length of curves and geodesics



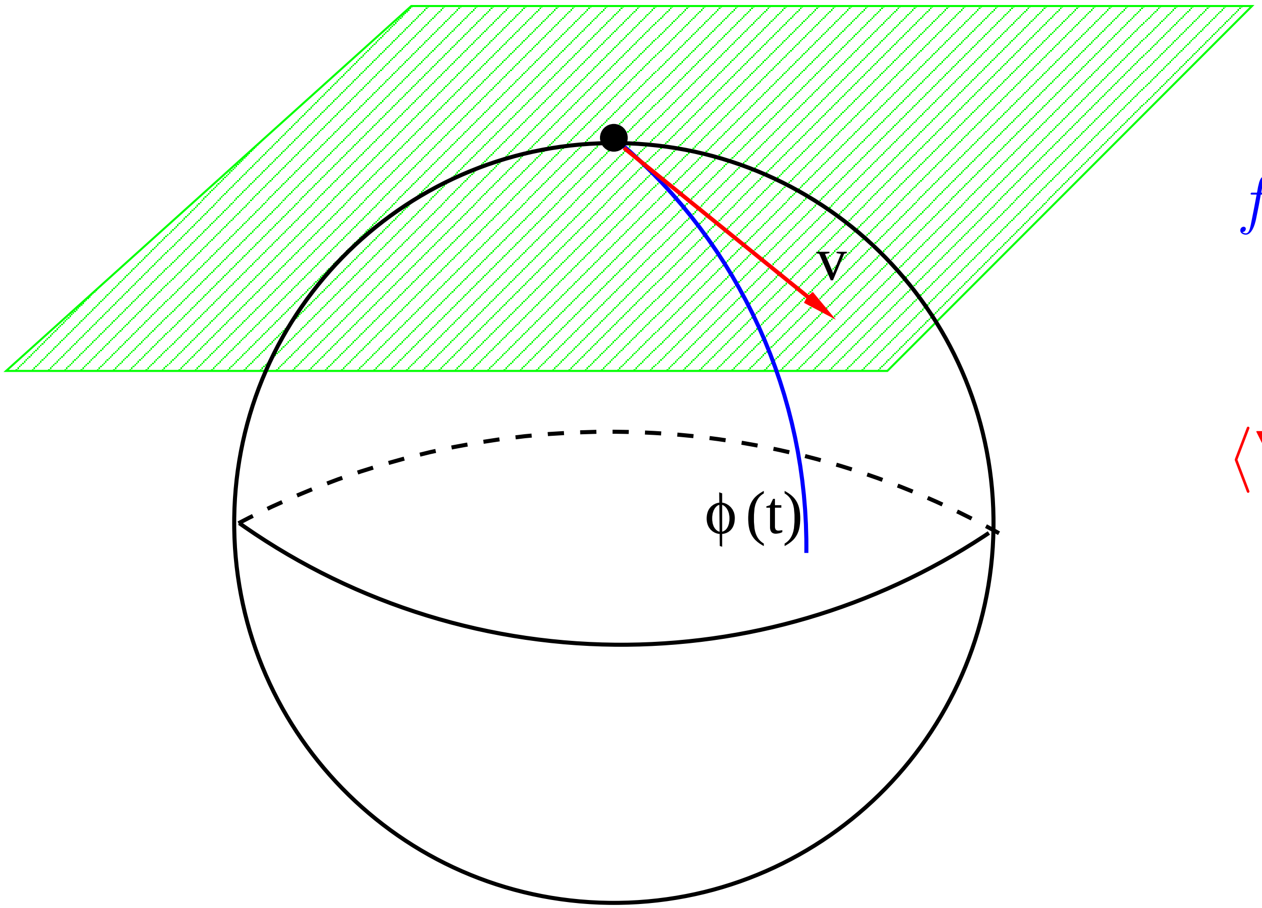
$$\phi(t) : [0,1] \to \mathcal{M}^k$$

$$l(\phi) = \int_0^1 \left\| \frac{d\phi}{dt} \right\| dt$$

Can measure length using norm in tangent space.

Geodesic — shortest curve between two points.

Gradient



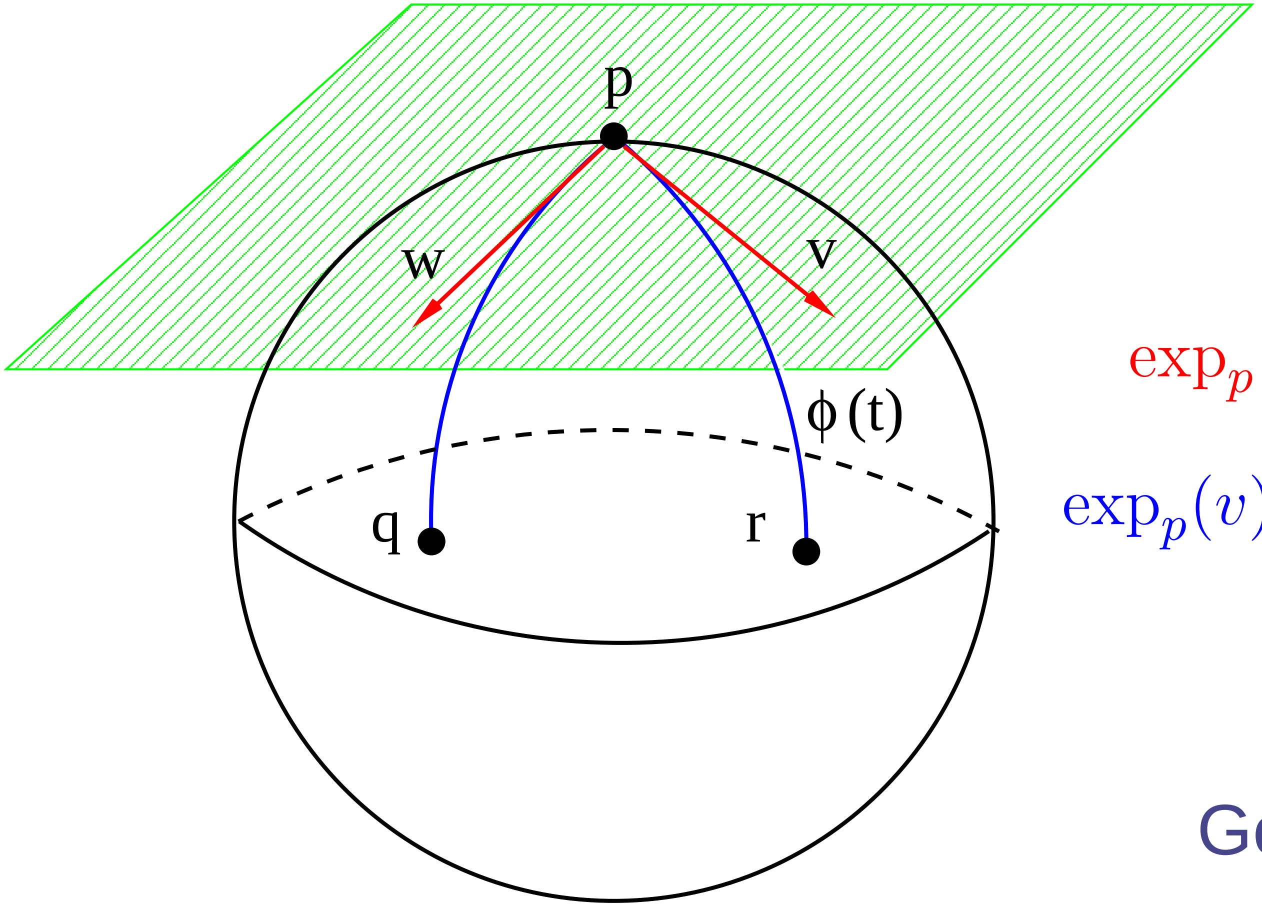
$$f : \mathcal{M}^k \to \mathbb{R}$$

$$\langle \nabla f, v \rangle \equiv \frac{df}{dv}$$

Tangent vectors <———> Directional derivatives.

Gradient points in the direction of maximum change.

Exponential map



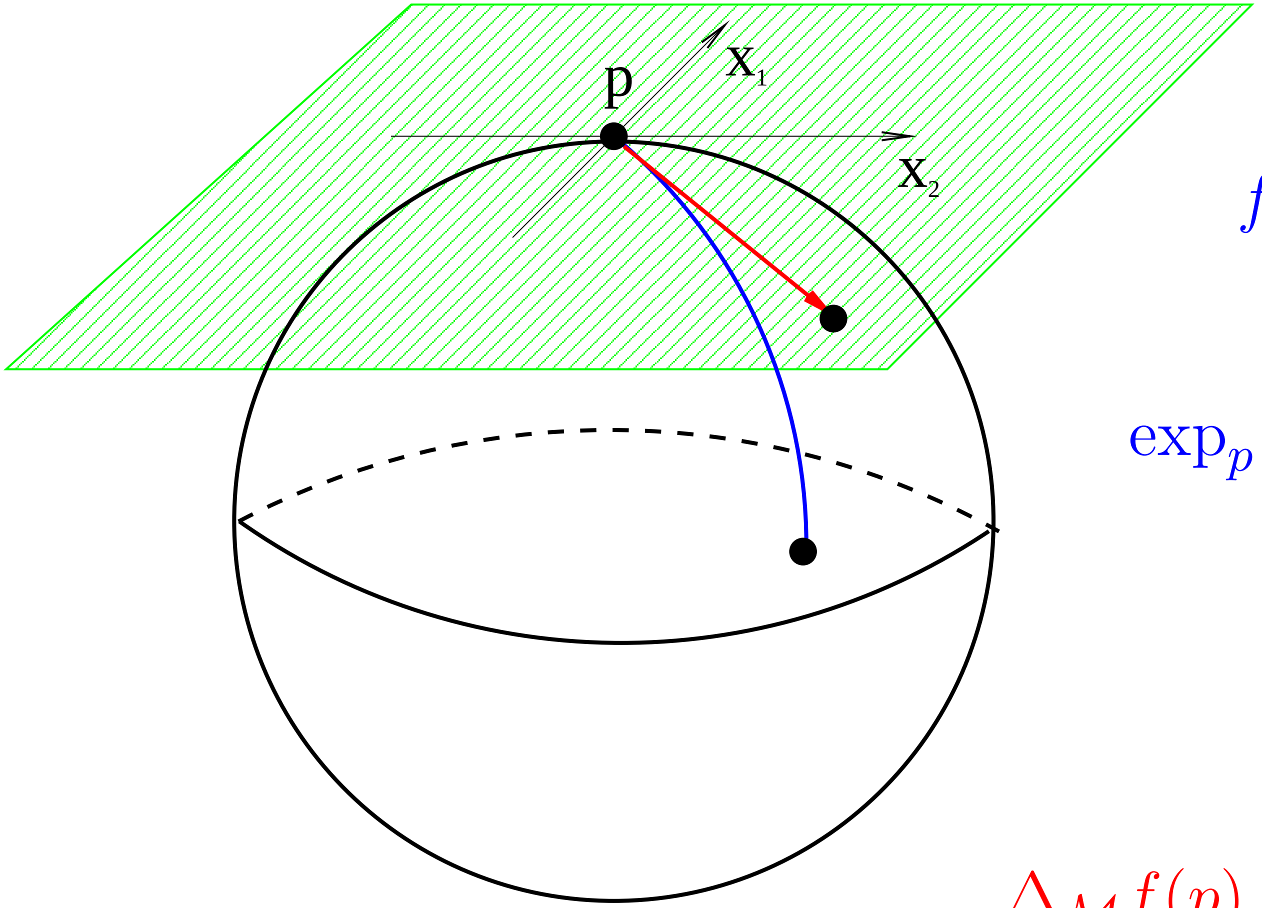
$$\exp_p : T_p\mathcal{M}^k \rightarrow \mathcal{M}^k$$

$$\exp_p(v) = r \quad \exp_p(w) = q$$

Geodesic $\phi(t)$

$$\phi(0) = p, \quad \phi(\|v\|) = q \qquad \left.\frac{d\phi(t)}{dt}\right|_0 = v$$

Laplace-Beltrami operator



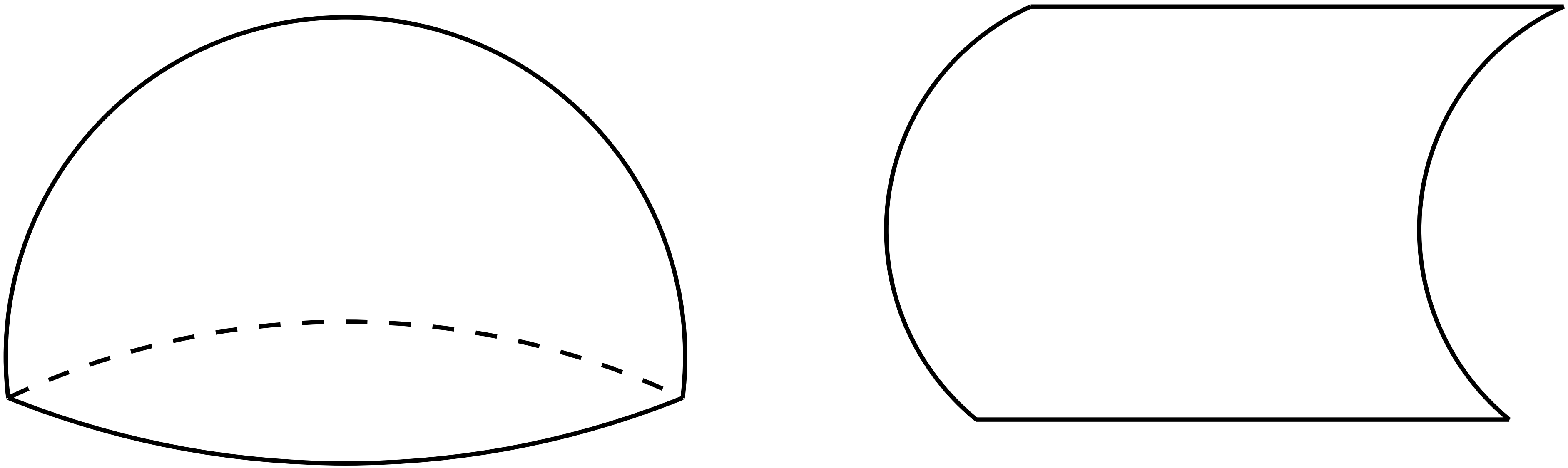
$$f : \mathcal{M}^k \to \mathbb{R}$$

$$\exp_p : T_p\mathcal{M}^k \to \mathcal{M}^k$$

$$\Delta_{\mathcal{M}} f(p) \equiv \sum_i \frac{\partial^2 f(\exp_p(x))}{\partial x_i^2}$$

Orthonormal coordinate system.

Intrinsic Curvature



cannot flatten —— can flatten

nonzero curvature —— zero curvature

No accurate map of Earth exists – Gauss's theorem.

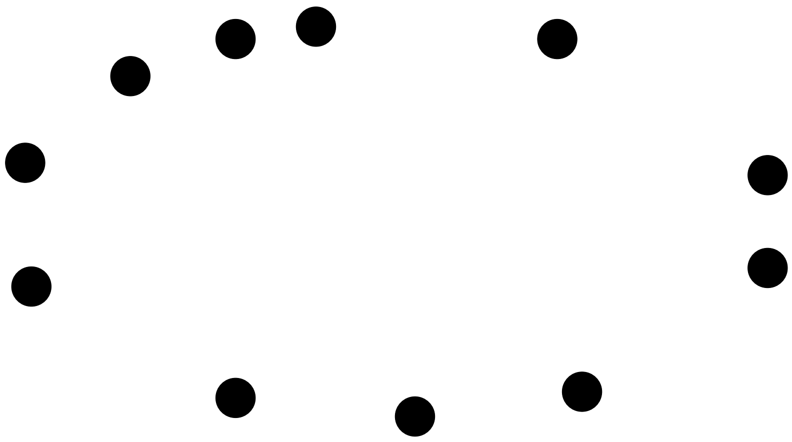
Dimensionality Reduction

Given $x_1, \ldots, x_n \in \mathcal{M} \subset \mathbb{R}^N$,
Find $y_1, \ldots, y_n \in \mathbb{R}^d$ where $d << N$

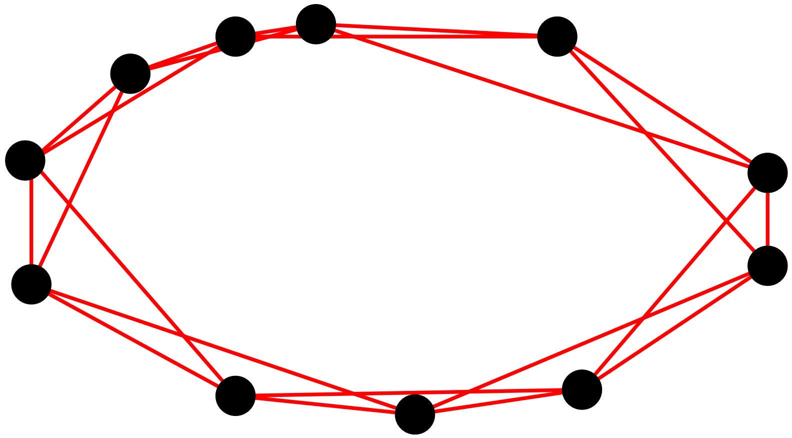
- ISOMAP (Tenenbaum, et al, 00)
- LLE (Roweis, Saul, 00)
- Laplacian Eigenmaps (Belkin, Niyogi, 01)
- Local Tangent Space Alignment (Zhang, Zha, 02)
- Hessian Eigenmaps (Donoho, Grimes, 02)
- Diffusion Maps (Coifman, Lafon, et al, 04)

Related: Kernel PCA (Schoelkopf, et al, 98)

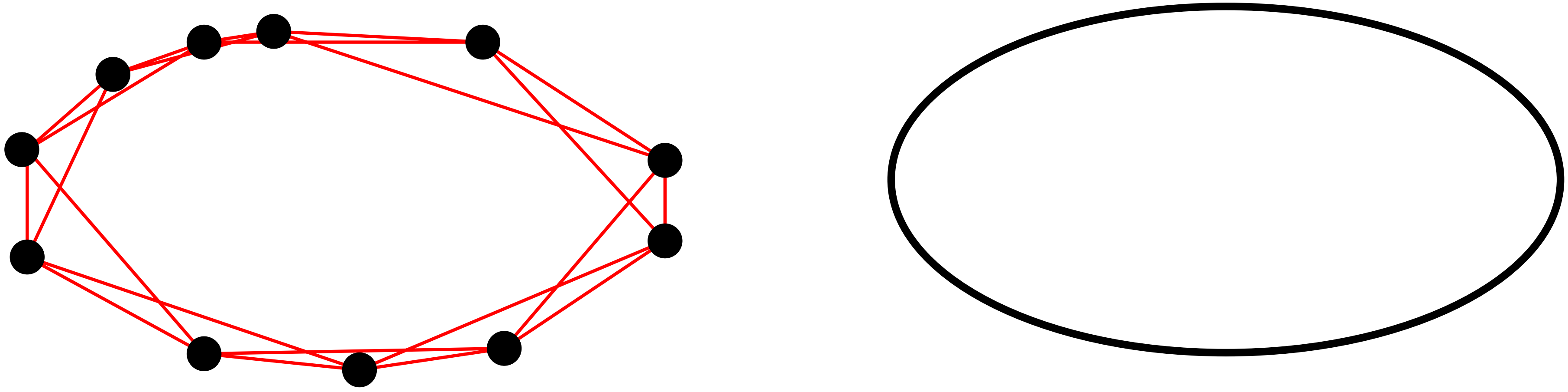
Algorithmic framework



Algorithmic framework



Algorithmic framework



Neighborhood graph common to all methods.

Isomap

1. Construct Neighborhood Graph.
2. Find shortest path (geodesic) distances.

   D_{ij} is $n \times n$

3. Embed using Multidimensional Scaling.

Multidimensional Scaling

Idea: Distances $\rightarrow$ Inner products $\rightarrow$ Embedding

1. Inner product from distances:

$$\langle \mathbf{x}, \mathbf{x} \rangle - 2\langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle = \|\mathbf{x} - \mathbf{y}\|^2$$

$$A_{ii} + A_{jj} - 2A_{ij} = D_{ij}$$

Answer:

$$A = -\frac{1}{2} HDH \text{ where } H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$$

In general only an approximation.

Multidimensional Scaling

2. Embedding from inner products (same as PCA!).

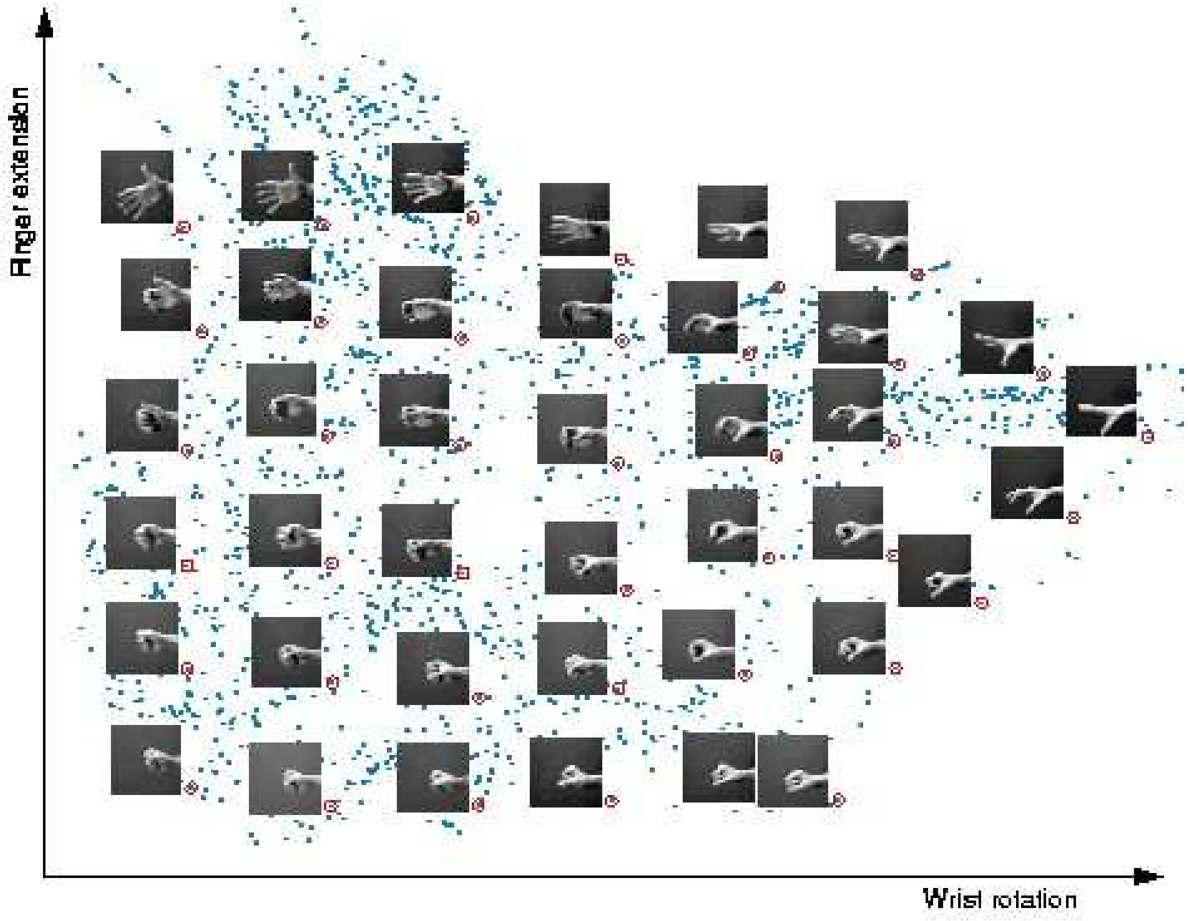
Consider a positive definite matrix A. Then A_{ij} corresponds to inner products.

$$A = \sum_{i=1}^{n} \lambda_i \phi_i \phi_i^T$$

Then for any $x \in \{1, \ldots, n\}$

$$\psi(x) = \left(\sqrt{\lambda_1} \phi_i(x), \ldots, \sqrt{\lambda_k} \phi_k(x) \right) \in \mathbb{R}^k$$

Isomap



From Tenenbaum, et al. 00

Unfolding flat manifolds

Isomap:
“unfolds” a flat manifold isometric to a convex domain in $\mathbb{R}^n$.

Hessian Eigenmaps:
“unfolds” and flat manifold isometric to an arbitrary domain in $\mathbb{R}^n$.

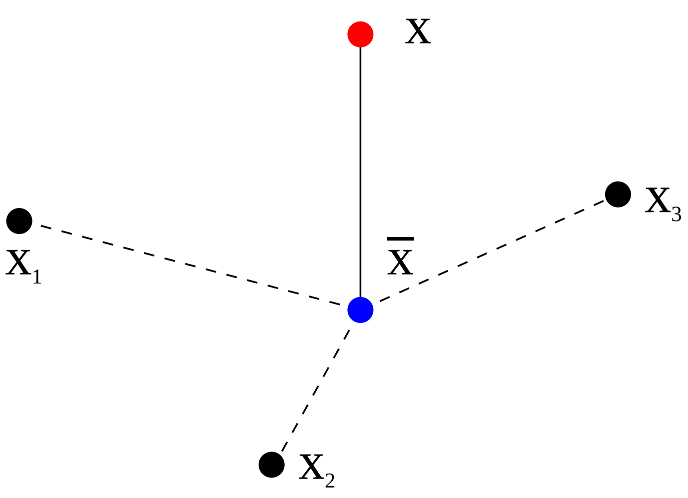
LTSA can also find an unfolding.

Locally Linear Embedding

1. Construct Neighborhood Graph.

2. Let $x_1, \ldots, x_n$ be neighbors of x. Project x to the span of $x_1, \ldots, x_n$.

3. Find barycentric coordinates of $\bar{x}$.



$$\bar{x} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

$$w_1 + w_2 + w_3 = 1$$

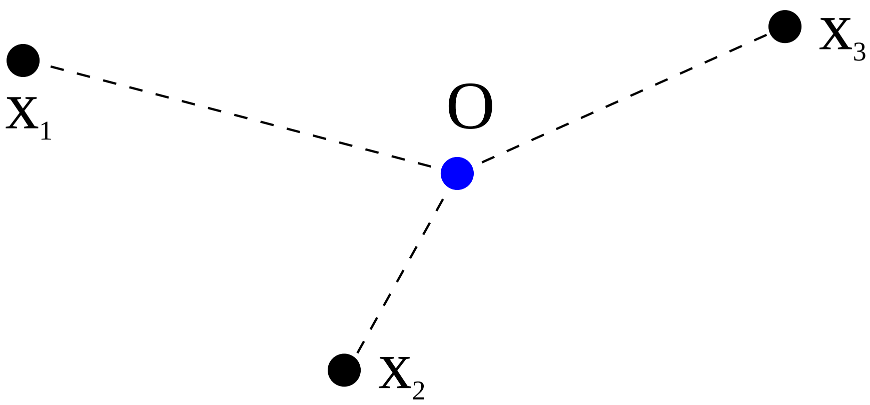
Weights w_1, w_2, w_3 chosen, so that $\bar{x}$ is the center of mass.

Locally Linear Embedding

4. Construct sparse matrix W. i th row is barycentric coordinates of $\bar{x}_i$ in the basis of its nearest neighbors.

5. Use lowest eigenvectors of $(I - W)^t(I - W)$ to embed.

Laplacian and LLE



$$\sum w_i x_i = 0$$

$$\sum w_i = 1$$

Hessian H. Taylor expansion :

$$f(x_i) = f(0) + x_i^t \nabla f + \frac{1}{2} x_i^t H x_i + o(\|x_i\|^2)$$

$$(I - W) f(0) = f(0) - \sum w_i f(x_i) \approx f(0) - \sum w_i f(0) - \sum_i w_i x_i^t \nabla f - \frac{1}{2} \sum_i x_i^t H x_i =$$

$$= -\frac{1}{2} \sum_i x_i^t H x_i \approx -trH = \Delta f$$

Laplacian Eigenmaps

Step 1 [*Constructing the Graph*]

$$e_{ij} = 1 \Leftrightarrow \mathbf{x}_i \text{ “close to” } \mathbf{x}_j$$

1. ϵ-neighborhoods. [*parameter* $\epsilon \in \mathbb{R}$] Nodes i and j are connected by an edge if

$$||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon$$

2. n nearest neighbors. [*parameter* $n \in \mathbb{N}$] Nodes i and j are connected by an edge if i is among n nearest neighbors of j or j is among n nearest neighbors of i.

Laplacian Eigenmaps

Step 2. [*Choosing the weights*].

1. Heat kernel. [*parameter* $t \in \mathbb{R}$]. If nodes i and j are connected, put

$$W_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{t}}$$

2. Simple-minded. [*No parameters*]. $W_{ij} = 1$ if and only if vertices i and j are connected by an edge.

Laplacian Eigenmaps

Step 3. [*Eigenmaps*] Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$Lf = \lambda D f$$

D is diagonal matrix where

$$D_{ii} = \sum_j W_{ij}$$

$$L = D - W$$

Let $\mathbf{f}_0, \ldots, \mathbf{f}_{k-1}$ be eigenvectors.

Leave out the eigenvector $\mathbf{f}_0$ and use the next m lowest eigenvectors for embedding in an m-dimensional Euclidean space.

Diffusion Distance

Heat diffusion operator H^t.

δ_x and δ_y initial heat distributions.

Diffusion distance between x and y:

$$\|H^t\delta_x - H^t\delta_y\|_{L^2}$$

Difference between heat distributions after time t.

Diffusion Maps

Embed using weighted eigenfunctions of the Laplacian:

$$x \longrightarrow (e^{-\lambda_1 t}\mathbf{f}_1(x), e^{-\lambda_2 t}\mathbf{f}_2(x), \ldots)$$

Diffusion distance is (approximated by) the distance between the embedded points.

Closely related to random walks on graphs.

Justification

Find $y_1, \ldots, y_n \in R$

$$\min \sum_{i,j} (y_i - y_j)^2 W_{ij}$$

Tries to preserve locality

A Fundamental Identity

But

$$\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y}$$

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = \sum_{i,j}(y_i^2 + y_j^2 - 2y_i y_j) W_{ij}$$

$$= \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{i,j} y_i y_j W_{ij}$$

$$= 2\mathbf{y}^T L \mathbf{y}$$

Embedding

$$\lambda = 0 \rightarrow \mathbf{y} = \mathbf{1}$$

$$\min_{\mathbf{y}^T \mathbf{1} = 0} \mathbf{y}^T L \mathbf{y}$$

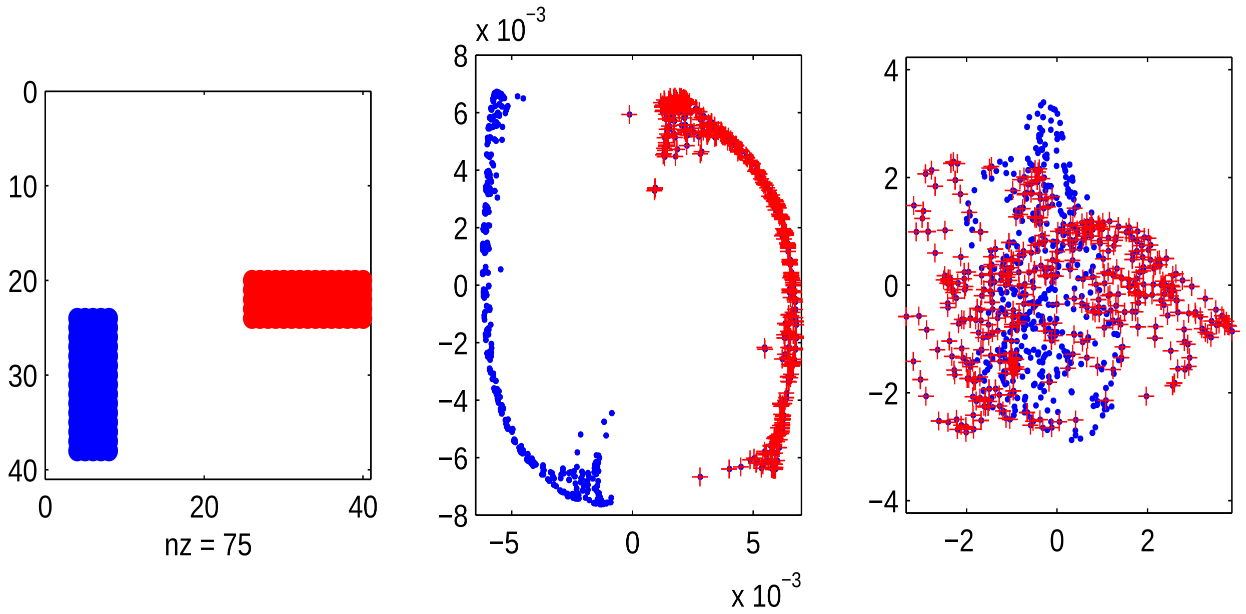
Let $Y = [\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_m]$

$$\sum_{i,j} ||Y_i - Y_j||^2 W_{ij} = \text{trace}(Y^T L Y)$$

subject to $Y^T Y = I$.

Use eigenvectors of L to embed.

PCA versus Laplacian Eigenmaps



On the Manifold

smooth map $f : \mathcal{M} \rightarrow R$

$$\int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 \approx \sum_{i \sim j} W_{ij}(f_i - f_j)^2$$

Recall standard gradient in $\mathbb{R}^k$ of $f(z_1, \ldots, z_k)$

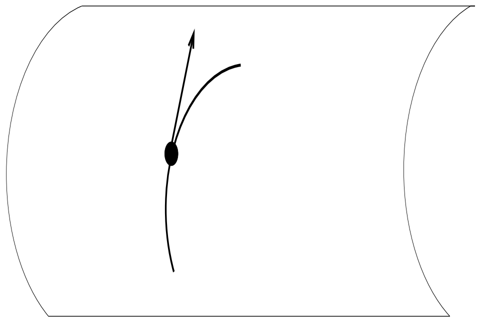
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial z_1} \\ \frac{\partial f}{\partial z_2} \\ \cdot \\ \cdot \\ \frac{\partial f}{\partial z_k} \end{bmatrix}$$

Curves on Manifolds

Consider a curve on $\mathcal{M}$

$$c(t) \in \mathcal{M} \qquad t \in (-1,1) \qquad p = c(0);\ q = c(\tau)$$

$$f(c(t)) : (-1,1) \to \mathbb{R}$$



$$|f(0) - f(\tau)| \lesssim d_G(p,q) \|\nabla_M f(p)\|$$

Stokes Theorem

A Basic Fact

$$\int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 = \int f \cdot \Delta_{\mathcal{M}} f$$

This is like

$$\sum_{i,j} W_{ij}(f_i - f_j)^2 = \mathbf{f}^T \mathbf{L} \mathbf{f}$$

where

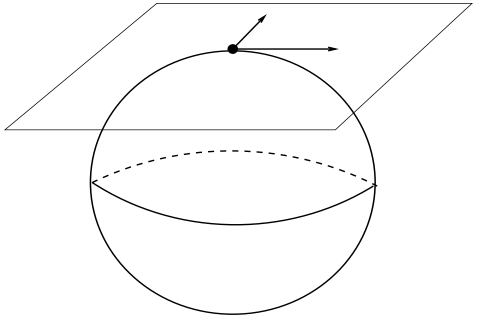
$\Delta_{\mathcal{M}} f$ is the manifold Laplacian

Manifold Laplacian

Recall ordinary Laplacian in $\mathbb{R}^k$
This maps

$$f(x_1, \ldots, x_k) \to \left(-\sum_{i=1}^{k} \frac{\partial^2 f}{\partial x_i^2} \right)$$

Manifold Laplacian is the same on the tangent space.



Properties of Laplacian

Eigensystem

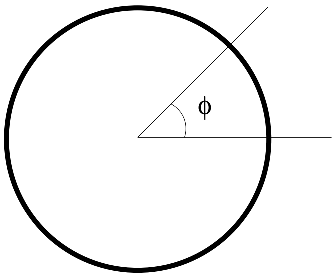
$$\Delta_{\mathcal{M}} f = \lambda_i \phi_i$$

$\lambda_i \geq 0$ and $\lambda_i \to \infty$

$\{\phi_i\}$ form an orthonormal basis for $L^2(\mathcal{M})$

$$\int \|\nabla_{\mathcal{M}} \phi_i\|^2 = \lambda_i$$

The Circle: An Example



$$-\frac{d^2u}{dt^2} = \lambda u \text{ where } u(0) = u(2\pi)$$

Eigenvalues are

$$\lambda_n = n^2$$

Eigenfunctions are

$$\sin(nt), \cos(nt)$$

From graphs to manifolds

$$f : \mathcal{M} \to \mathbb{R} \quad x \in \mathcal{M} \quad x_1, \ldots, x_n \in \mathcal{M}$$

Graph Laplacian:

$$L_n^t(f)(x) = f(x) \sum_j e^{-\frac{\|x - x_j\|^2}{t}} - \sum_j f(x_j) e^{-\frac{\|x - x_j\|^2}{t}}$$

Theorem [pointwise convergence] $t_n = n^{-\frac{1}{k+2+\alpha}}$

$$\lim_{n \to \infty} \frac{(4\pi t_n)^{-\frac{k+2}{2}}}{n} L_n^{t_n} f(x) = \Delta_{\mathcal{M}} f(x)$$

Belkin 03, Lafon Coifman 04, Belkin Niyogi 05, Hein et al 05

From graphs to manifolds

Theorem [convergence of eigenfunctions]

$$\lim_{t \to 0, n \to \infty} Eig[L_n^{t_n}] \to Eig[\Delta_{\mathcal{M}}]$$

Belkin Niyogi 06

Estimating Dimension from Laplacian

$$\lambda_1 \leq \lambda_2 \ldots \leq \lambda_j \leq \ldots$$

Then

$$A + \frac{2}{d}\log(j) \leq \log(\lambda_j) \leq B + \frac{2}{d}\log(j+1)$$

Example: on S^1

$$\lambda_j = j^2 \implies \log(\lambda_j) = \frac{2}{1}\log(j)$$

(Li and Yau; Weyl's asymptotics)

Visualization

Data representation, dimensionality reduction, visualization

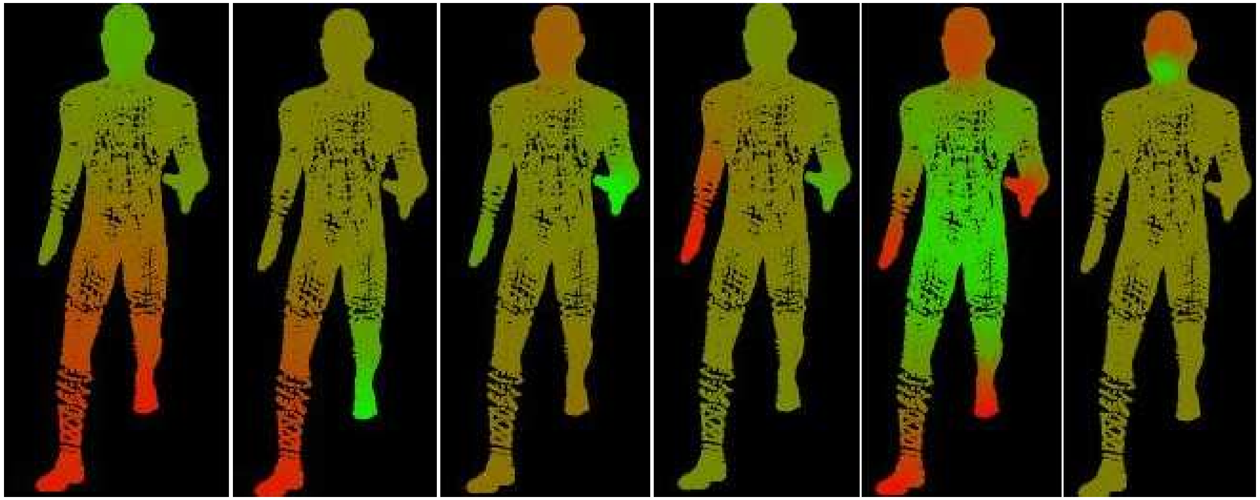
Visualizing spaces of digits.

Partiview, Ndaona, Surendran 04

Motion estimation

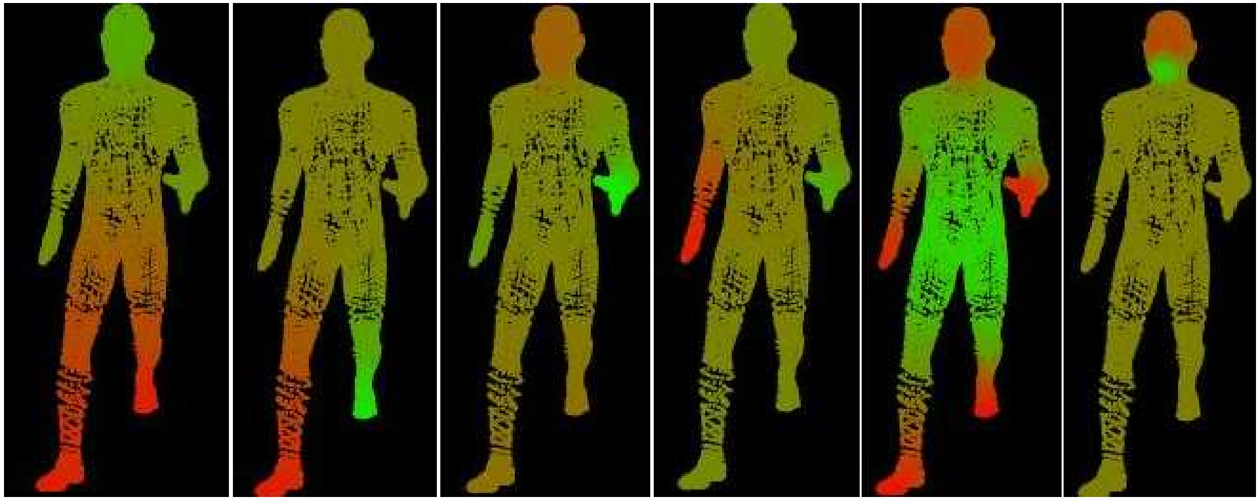
Markerless motion estimation: inferring joint angles.

Corazza, Andriacchi, Stanford Biomotion Lab, 05, Partiview, Surendran



Isometrically invariant representation. [link]
Eigenfunctions of the Laplacian are invariant under isometries.

Graphics, etc



Laplacian from meshes/non-probabilistic point clouds.

Belkin, Sun, Wang 08, 09

Recall

Heat equation in $\mathbb{R}^n$:

$u(x,t)$ – heat distribution at time t.
$u(x,0) = f(x)$ – initial distribution. $x \in \mathbb{R}^n, t \in \mathbb{R}$.

$$\Delta_{\mathbb{R}^n} u(x,t) = \frac{du}{dt}(x,t)$$

Solution – convolution with the heat kernel:

$$u(x,t) = (4\pi t)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy$$

Proof idea (pointwise convergence)

Functional approximation:

Taking limit as $t \to 0$ and writing the derivative:

$$\Delta_{\mathbb{R}^n} f(x) = \frac{d}{dt}\left[(4\pi t)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy\right]_0$$

Proof idea (pointwise convergence)

Functional approximation:

Taking limit as $t \to 0$ and writing the derivative:

$$\Delta_{\mathbb{R}^n} f(x) = \frac{d}{dt}\left[(4\pi t)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy\right]_0$$

$$\Delta_{\mathbb{R}^n} f(x) \approx -\frac{1}{t}(4\pi t)^{-\frac{n}{2}} \left(f(x) - \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy\right)$$

Proof idea (pointwise convergence)

Functional approximation:
Taking limit as $t \to 0$ and writing the derivative:

$$\Delta_{\mathbb{R}^n} f(x) = \frac{d}{dt}\left[(4\pi t)^{-\frac{n}{2}} \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy\right]_0$$

$$\Delta_{\mathbb{R}^n} f(x) \approx -\frac{1}{t}(4\pi t)^{-\frac{n}{2}} \left(f(x) - \int_{\mathbb{R}^n} f(y) e^{-\frac{\|x-y\|^2}{4t}} dy\right)$$

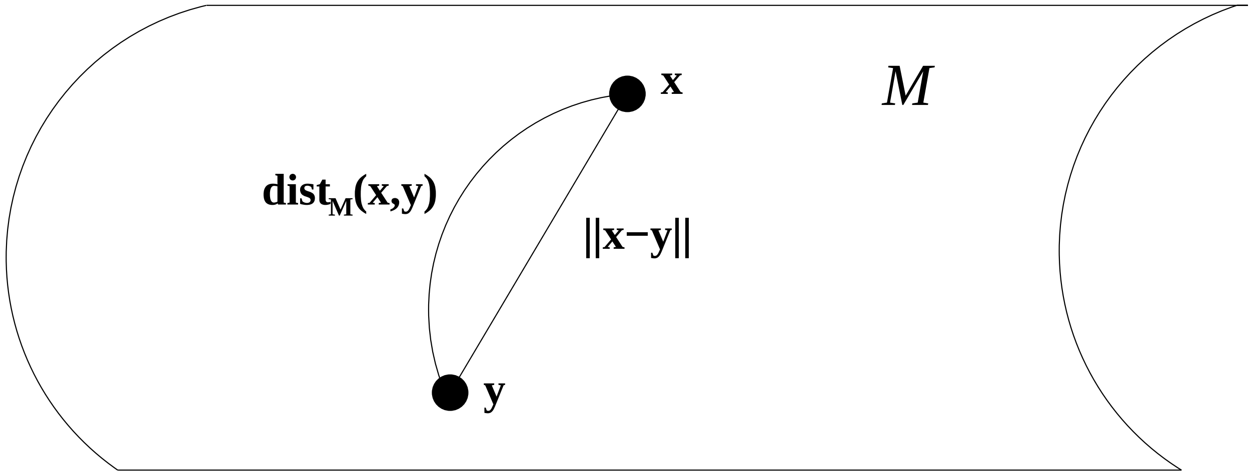
Empirical approximation:
Integral can be estimated from empirical data.

$$\Delta_{\mathbb{R}^n} f(x) \approx -\frac{1}{t}(4\pi t)^{-\frac{n}{2}} \left(f(x) - \sum_{x_i} f(x_i) e^{-\frac{\|x-x_i\|^2}{4t}}\right)$$

Some difficulties

Some difficulties arise for manifolds:

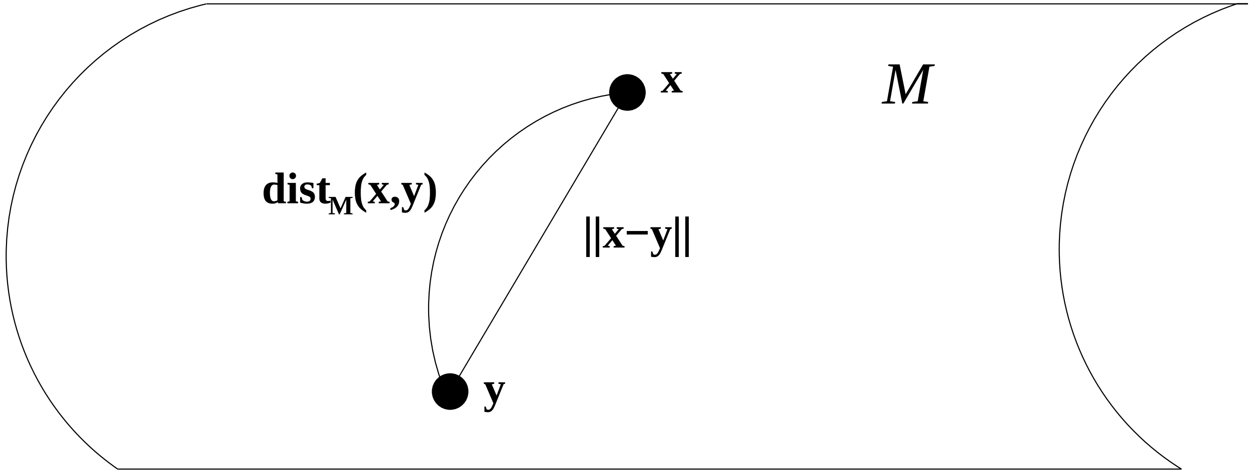
- Do not know distances.
- Do not know the heat kernel.



Some difficulties

Some difficulties arise for manifolds:

- Do not know distances.
- Do not know the heat kernel.



Careful analysis needed.

The Heat Kernel

- $H_t(x,y) = \sum_i e^{-\lambda_i t}\phi_i(x)\phi_i(y)$
- in $\mathbb{R}^d$, closed form expression

$$H_t(x,y) = \frac{1}{(4\pi t)^{d/2}} e^{-\frac{\|x-y\|^2}{4t}}$$

- Goodness of approximation depends on the gap

$$\left| H_t(x,y) - \frac{1}{(4\pi t)^{d/2}} e^{-\frac{\|x-y\|^2}{4t}} \right|$$

- H_t is a Mercer kernel intrinsically defined on manifold. Leads to SVMs on manifolds.

Three Remarks on Noise

1. Arbitrary probability distribution on the manifold: convergence to weighted Laplacian.
2. Noise off the manifold:
   $\mu = \mu_{\mathcal{M}^d} + \mu_{\mathbb{R}^N}$
   Then
   $$\lim_{t \to 0} L^t f(x) = \Delta f(x)$$
3. Noise off the manifold:
   $$z = x + \eta \; (\sim N(0, \sigma^2 I))$$
   We have
   $$\lim_{t \to 0} \lim_{\sigma \to 0} L^{t,\sigma} f(x) = \Delta f(x)$$

NLDR: some references

- A global geometric framework for nonlinear dimensionality reduction.

J.B. Tenenbaum, V. de Silva and J. C. Langford, 00.

- Nonlinear Dimensionality Reduction by Locally Linear Embedding.

L. K. Saul and S. T. Roweis. 00

- Laplacian Eigenmaps for Dimensionality Reduction and Data Representation.

M.Belkin, P.Niyogi, 01.

- Hessian Eigenmaps: new locally linear embedding techniques for high-dimensional data. D. L. Donoho and C. Grimes, 02.
- Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment.

Zhenyue Zhang and Hongyuan Zha. 02.

- Charting a manifold. Matthew Brand, 03
- Diffusion Maps. R. Coifman and S. Lafon. 04.
- Many more: http://www.cse.msu.edu/∼lawhiu/manifold/

Unlabeled data

Reasons to use unlabeled data in inference:

- Pragmatic:

Unlabeled data is everywhere. Need a way to use it.

- Philosophical:

The brain uses unlabeled data.

Geometry of classification

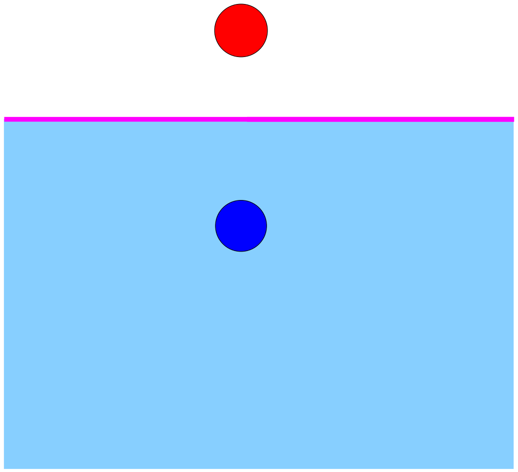
How does shape of the data affect classification?

- Manifold assumption.
- Cluster assumption.

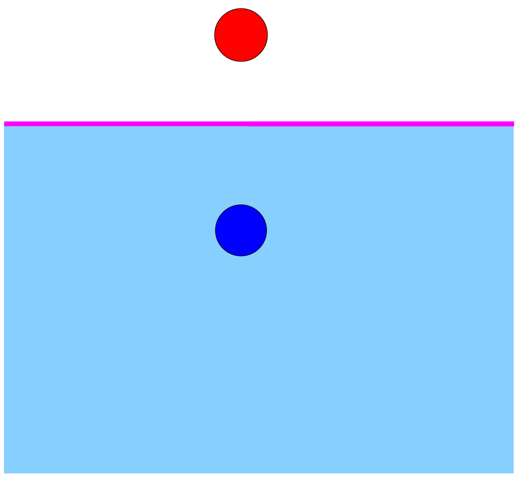
Reflect our understanding of structure of natural data.

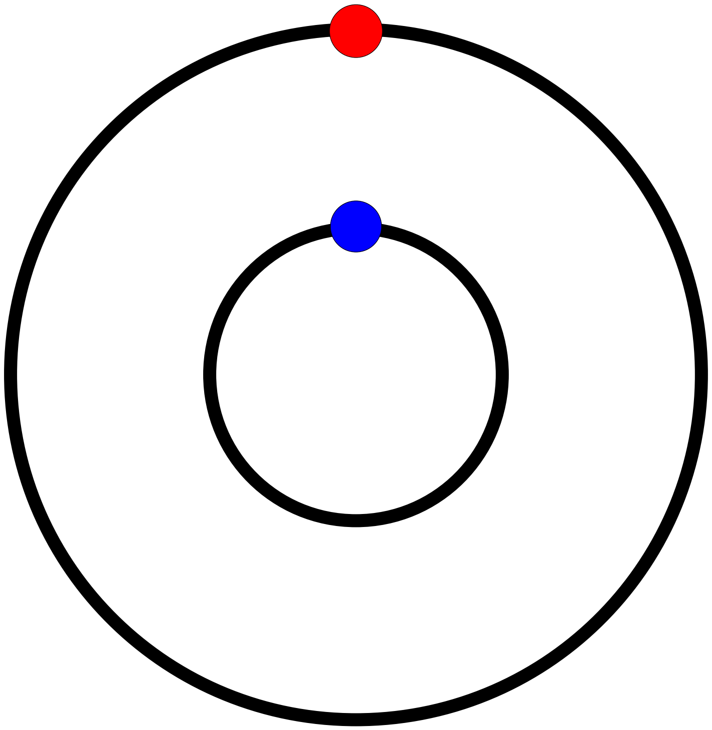
Intuition

Intuition

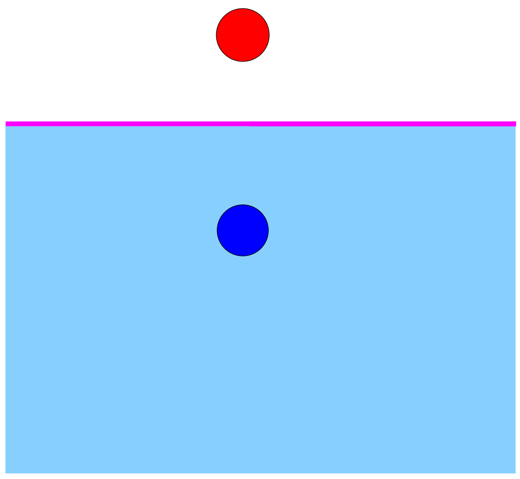


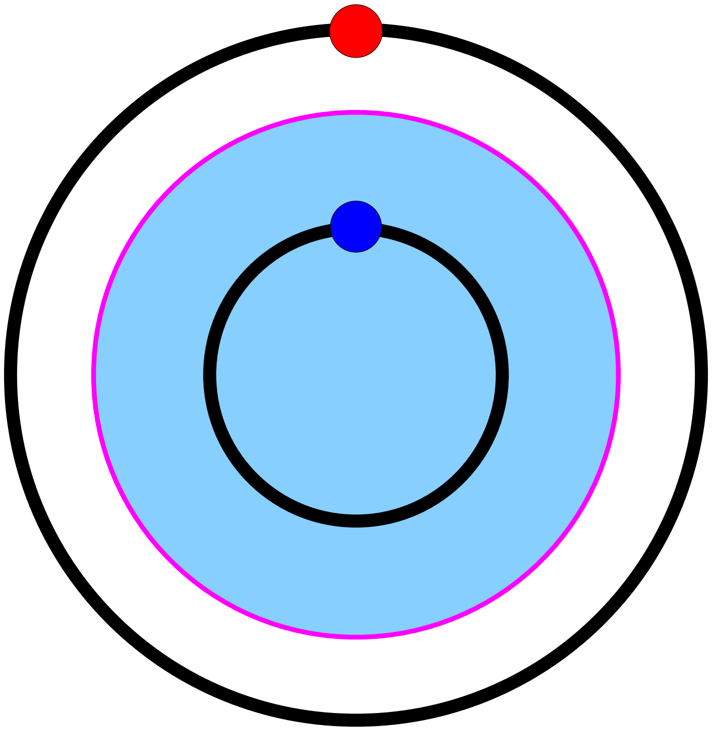
Intuition





Intuition

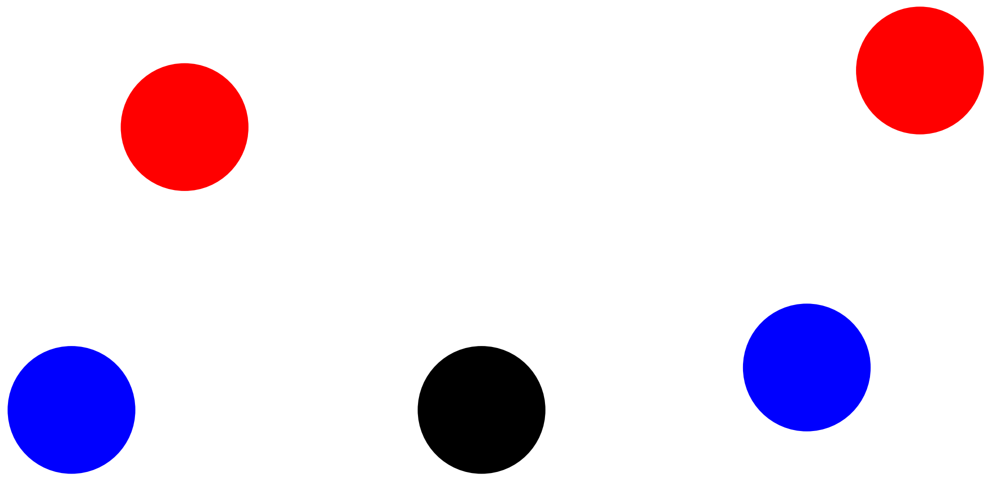




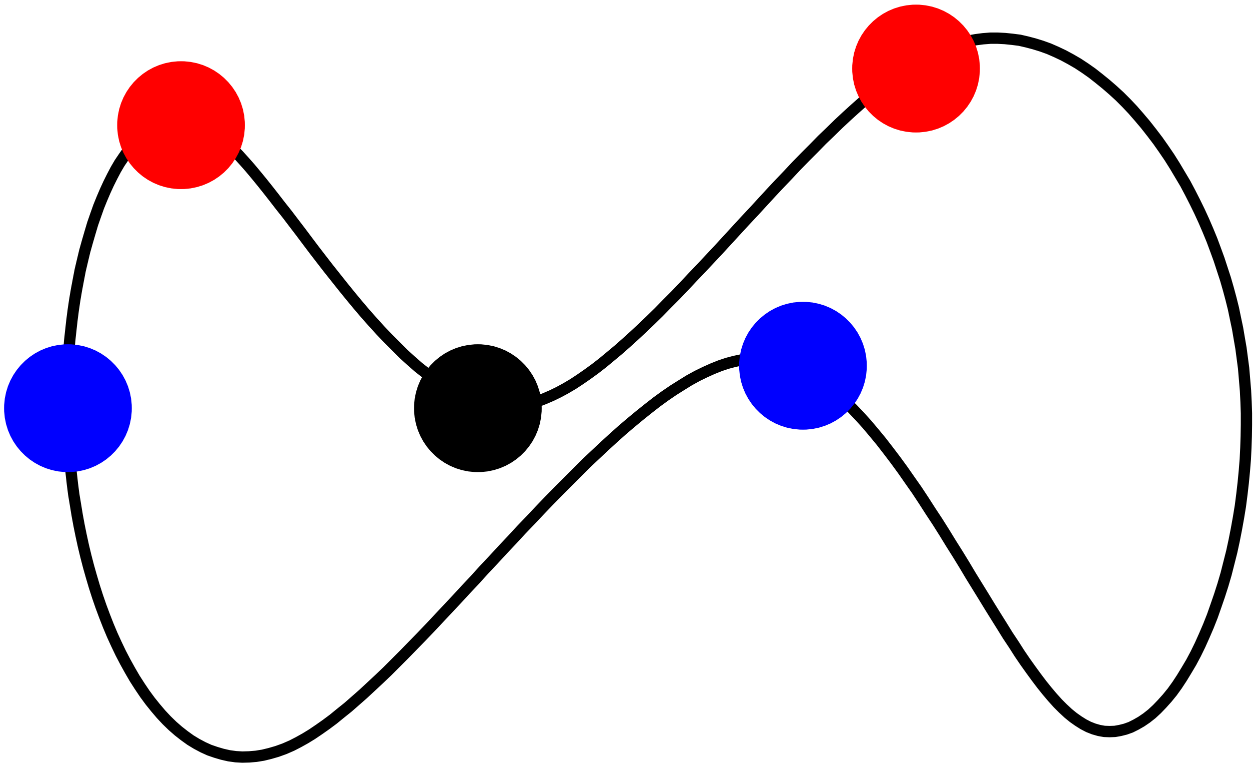
Geometry of data changes our notion of similarity.

Manifold assumption

Manifold assumption

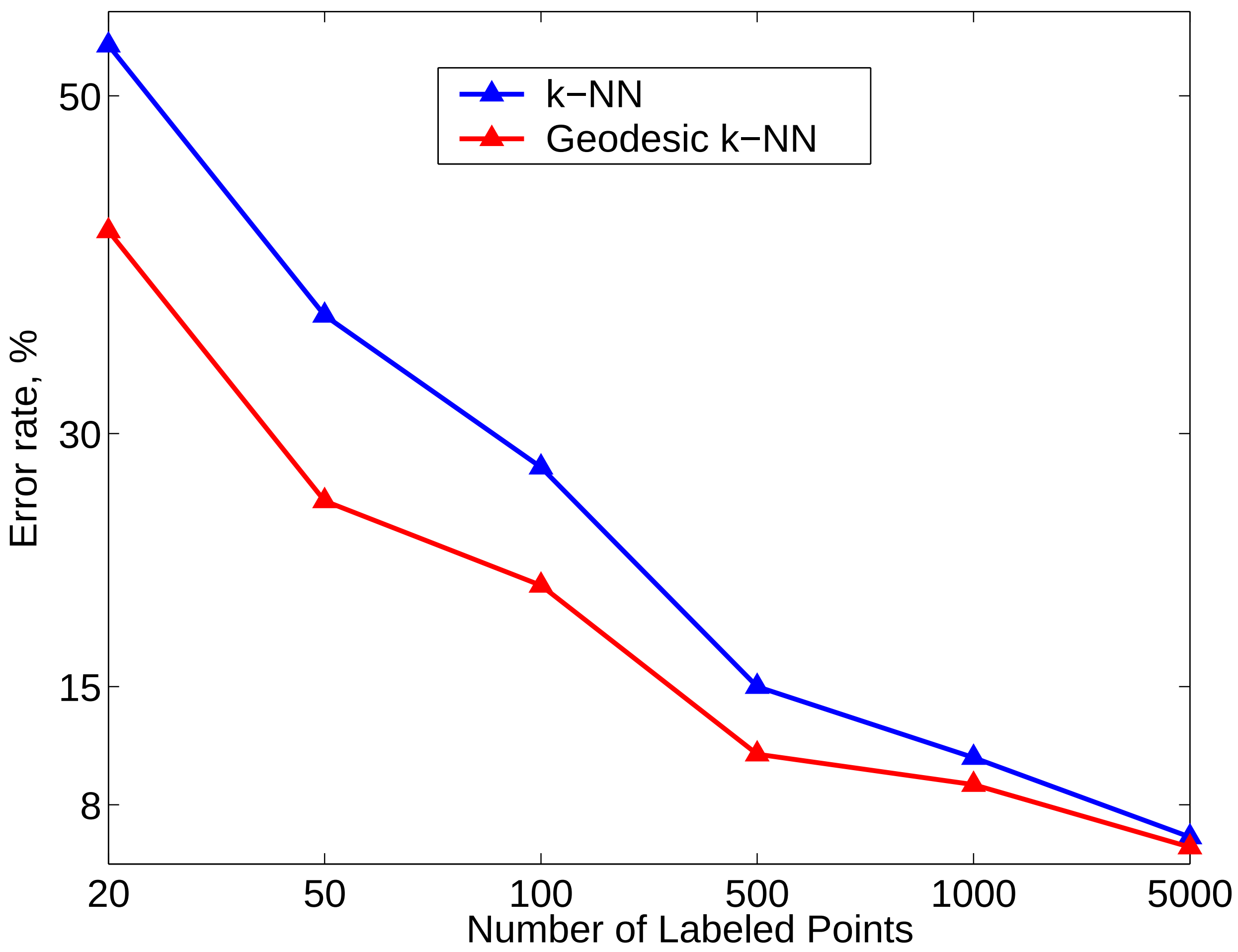


Manifold assumption

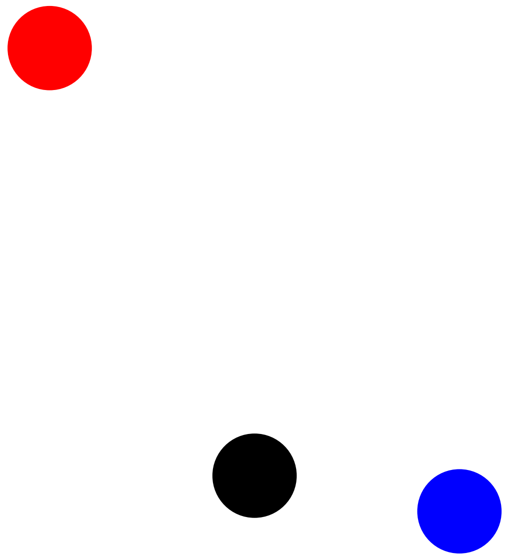


Geometry is important.

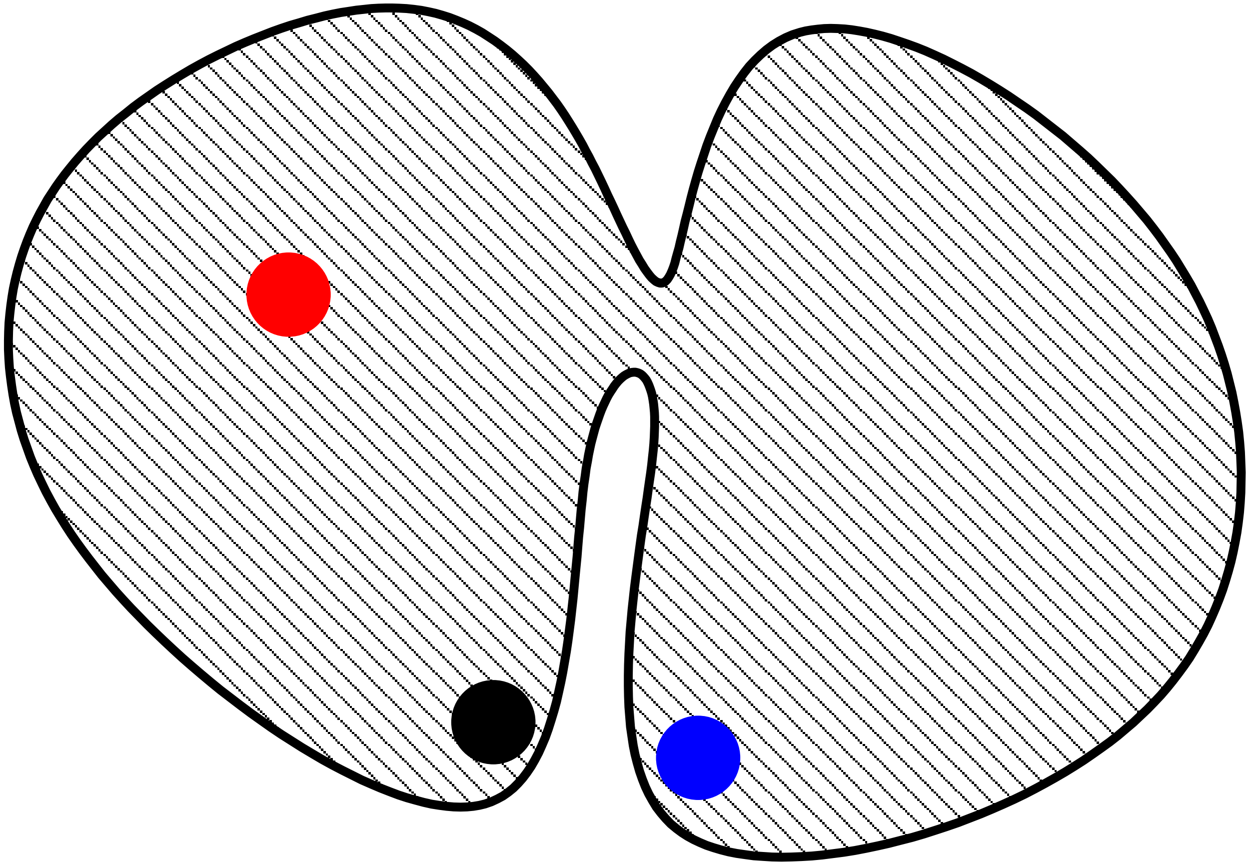
Geodesic Nearest Neighbors



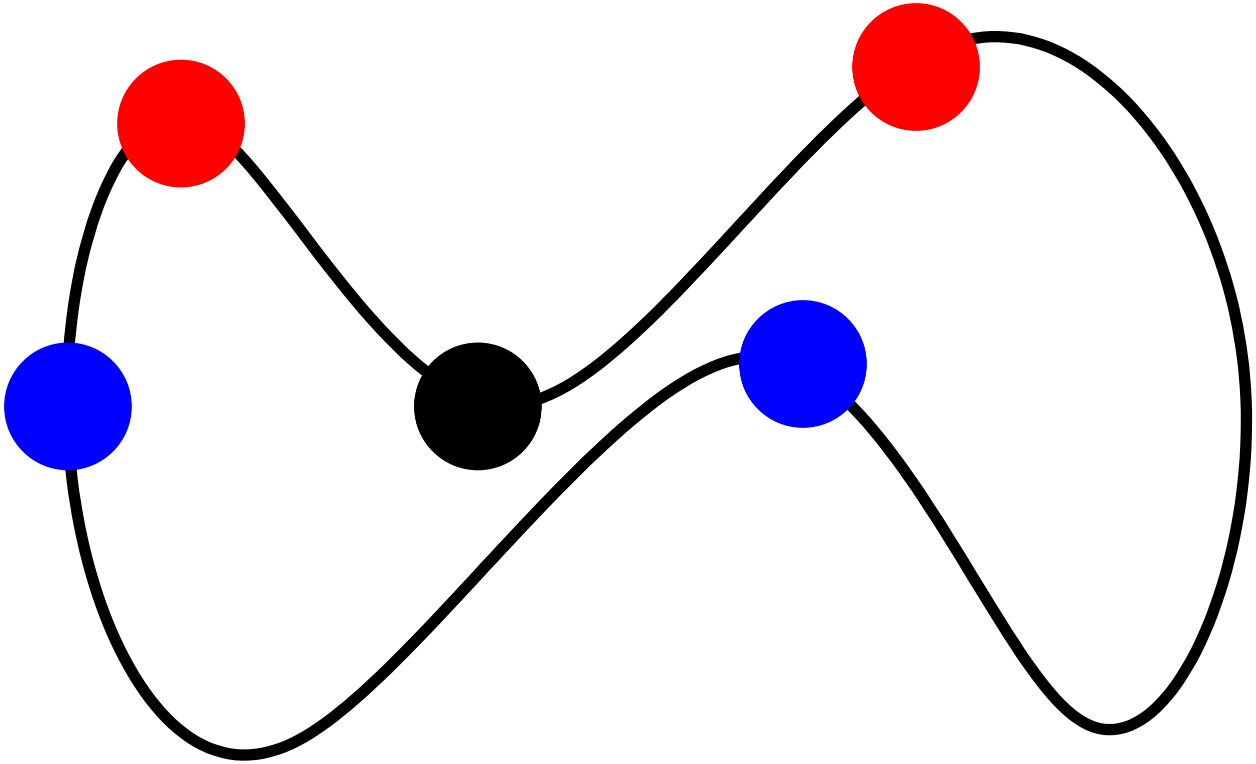
Cluster assumption



Cluster assumption

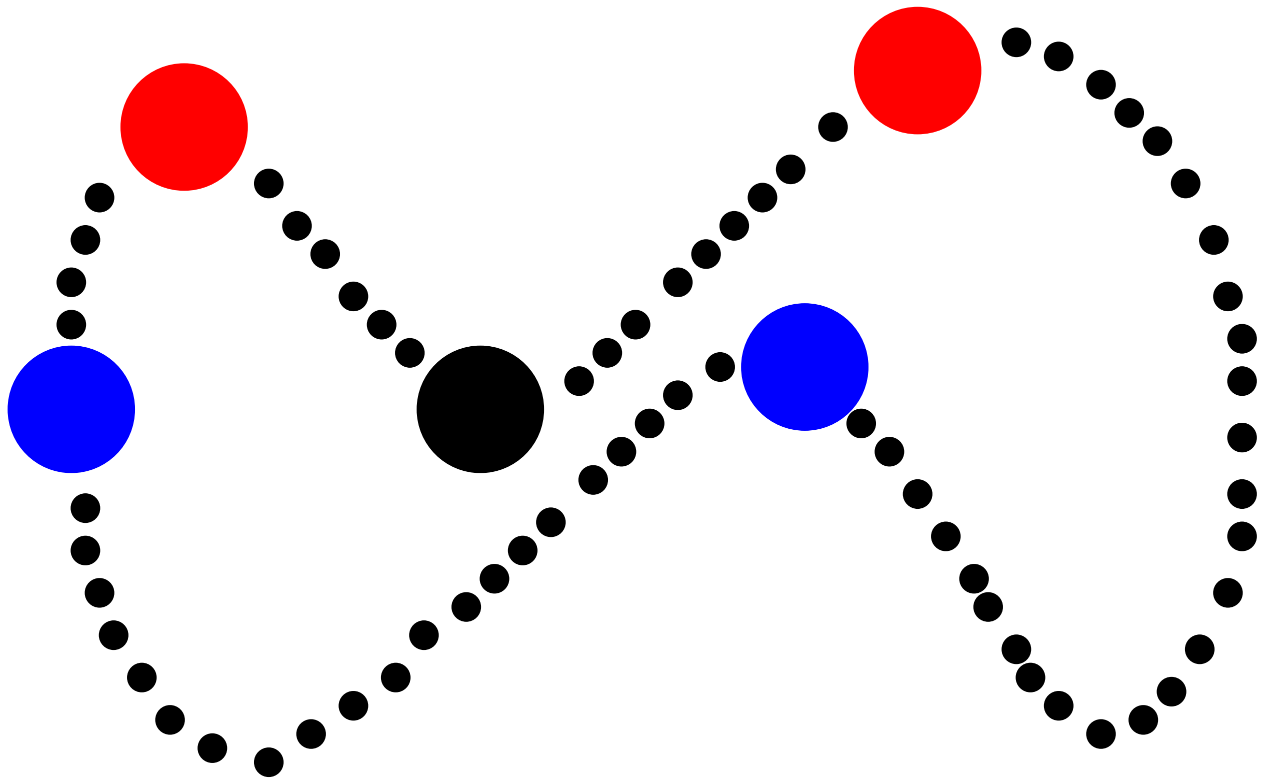


Unlabeled data



Geometry is important.

Unlabeled data



Geometry is important.
Unlabeled data to estimate geometry.

Manifold assumption

Manifold/geometric assumption:

functions of interest are smooth with respect to the underlying geometry.

Manifold assumption

Manifold/geometric assumption:
functions of interest are smooth with respect to the underlying geometry.

Probabilistic setting:
Map $X \to Y$. Probability distribution P on $X \times Y$.

Regression/(two class)classification: $X \to \mathbb{R}$.

Manifold assumption

Manifold/geometric assumption:

functions of interest are smooth with respect to the underlying geometry.

Probabilistic setting:

Map $X \to Y$. Probability distribution P on $X \times Y$.

Regression/(two class)classification: $X \to \mathbb{R}$.

Probabilistic version:

conditional distributions $P(y|x)$ are smooth with respect to the marginal $P(x)$.

What is smooth?

Function $f : X \to \mathbb{R}$. Penalty at $x \in X$:

$$\frac{1}{\delta^{k+2}} \int_{\text{small } \delta} \left(f(x) - f(x+\delta)\right)^2 p(x) d\,\delta \approx \|\nabla f\|^2 p(x)$$

Total penalty – Laplace operator:

$$\int_X \|\nabla f\|^2 p(x) = \langle f, \Delta_p f \rangle_X$$

What is smooth?

Function $f : X \to \mathbb{R}$. Penalty at $x \in X$:

$$\frac{1}{\delta^{k+2}} \int_{\text{small } \delta} (f(x) - f(x+\delta))^2 p(x) d\,\delta \approx \|\nabla f\|^2 p(x)$$

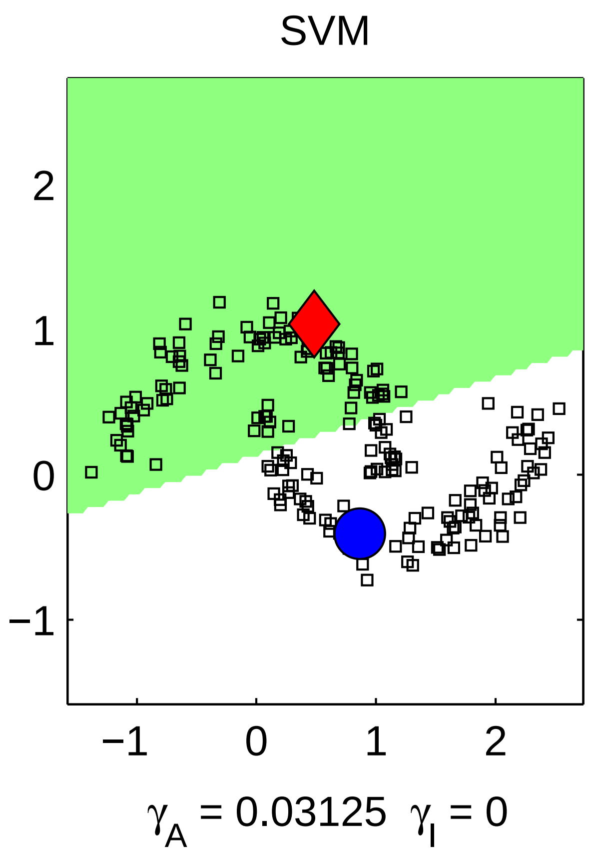
Total penalty – Laplace operator:

$$\int_X \|\nabla f\|^2 p(x) = \langle f, \Delta_p f \rangle_X$$

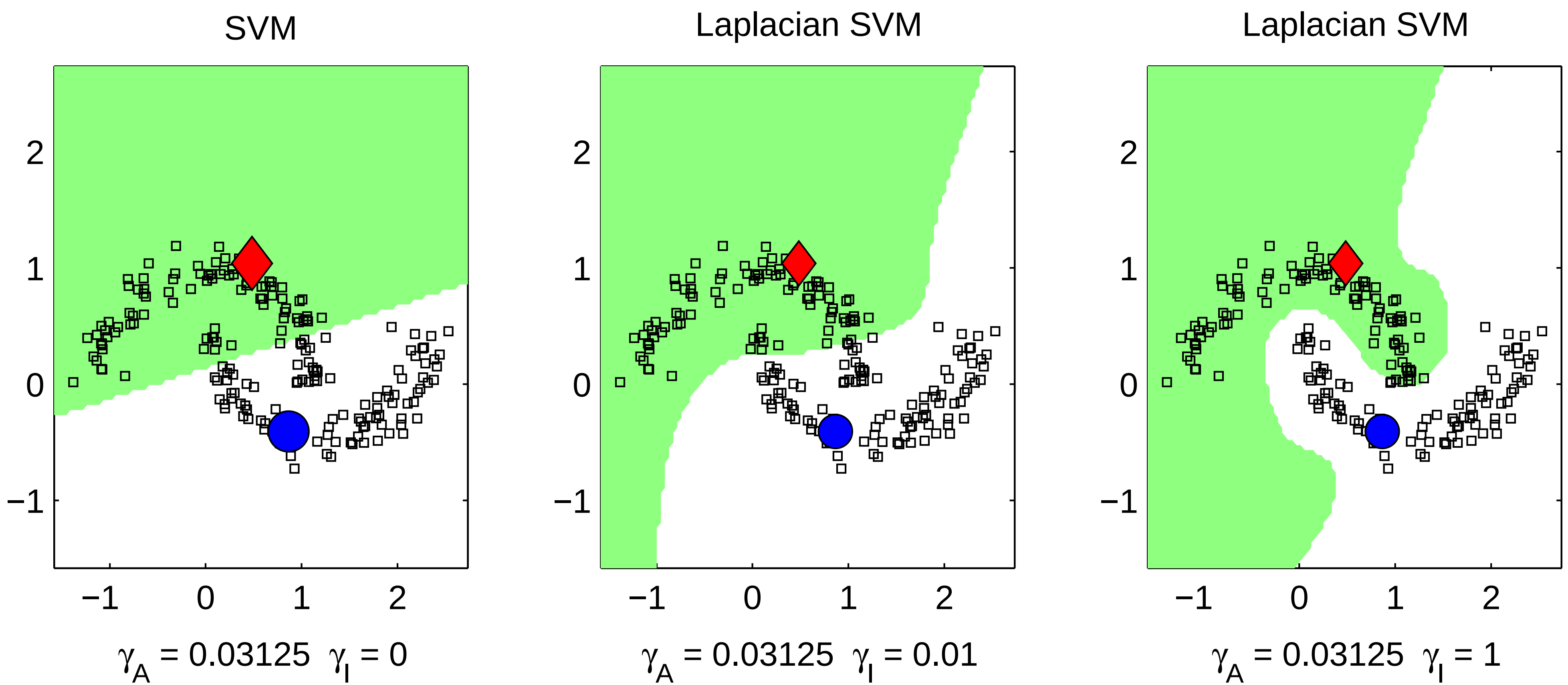
Two-class classification – conditional $P(1|x)$.

Manifold assumption: $\langle P(1|x), \Delta_p P(1|x) \rangle_X$ is small.

Example



Example


SVM
Laplacian SVM
Laplacian SVM
2
1
0
−1
−1
0
1
2
γA = 0.03125 γI = 0
γA = 0.03125 γI = 0.01
γA = 0.03125 γI = 1

Regularization

Estimate $f : \mathbb{R}^N \rightarrow \mathbb{R}$
Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$

Regularized least squares (hinge loss for SVM):

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{l} \sum (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_K^2$$

fit to data + smoothness penalty

$\|f\|_K$ incorporates our smoothness assumptions.
Choice of $\| \ \|_K$ is important.

Algorithm: RLS/SVM

$$\textbf{Solve :} \qquad f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{l} \sum (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_K^2$$

$\|f\|_K$ is a Reproducing Kernel Hilbert Space norm with kernel $K(\mathbf{x}, \mathbf{y})$.

Can solve explicitly (via Representer theorem):

$$f^*(\cdot) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \cdot)$$

$$[\alpha_1, \ldots, \alpha_l]^t = (\mathbf{K} + \lambda I)^{-1} [y_1, \ldots, y_l]^t$$

$$(\mathbf{K})_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$$

Manifold regularization

Estimate $f : \mathbb{R}^N \to \mathbb{R}$

Labeled data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$
Unlabeled data: $\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}$

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{l} \sum (f(\mathbf{x}_i) - y_i)^2 + \lambda_A \|f\|_K^2 + \lambda_I \|f\|_I^2$$

fit to data + extrinsic smoothness + intrinsic smoothness

Empirical estimate:

$$\|f\|_I^2 = \frac{1}{(l+u)^2} [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{l+u})] \, L \, [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{l+u})]^t$$

Laplacian RLS/SVM

Representer theorem (discrete case):

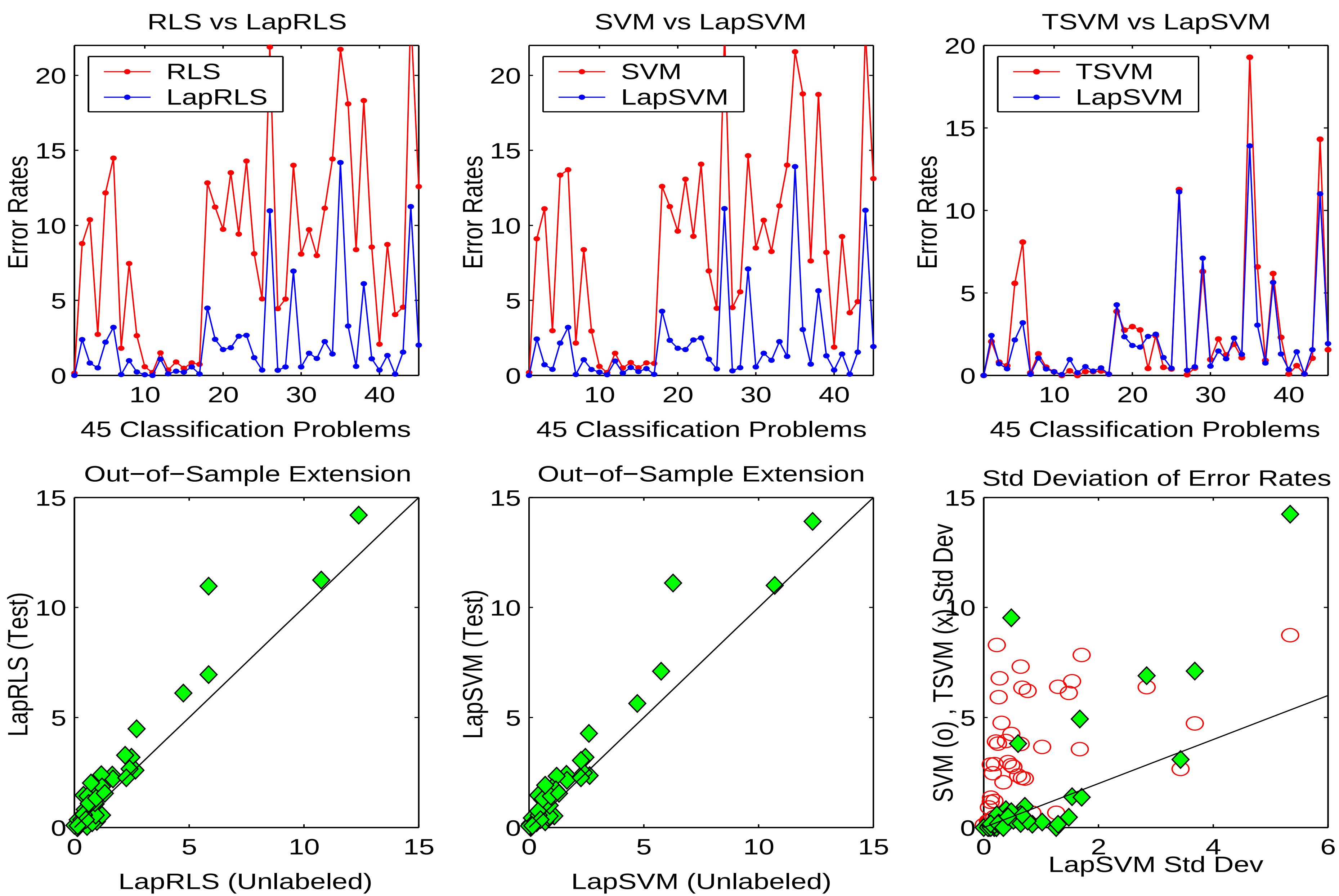
$$f^*(\cdot) = \sum_{i=1}^{l+u} \alpha_i K(\mathbf{x}_i, \cdot)$$

Explicit solution for quadratic loss:

$$\bar{\alpha} = (J\mathbf{K} + \lambda_A l I + \frac{\lambda_I l}{(u+l)^2}\mathbf{L}\mathbf{K})^{-1}[y_1, \ldots, y_l, 0, \ldots, 0]^t$$

$$(\mathbf{K})_{ij} = K(\mathbf{x}_i, \mathbf{x}_j), \qquad J = diag\,(\underbrace{1, \ldots, 1}_{l}, \underbrace{0, \ldots, 0}_{u})$$

Experimental results: USPS



Experimental comparisons

Dataset → Algorithm ↓	g50c	Coil20	Uspst	mac-win	WebKB (link)	WebKB (page)	WebKB (page+link)
SVM (full labels)	3.82	0.0	3.35	2.32	6.3	6.5	1.0
SVM (l labels)	8.32	24.64	23.18	18.87	25.6	22.2	15.6
Graph-Reg	17.30	6.20	21.30	11.71	22.0	10.7	6.6
TSVM	6.87	26.26	26.46	7.44	**14.5**	**8.6**	7.8
Graph-density	8.32	6.43	16.92	10.48	-	-	-
∇TSVM	5.80	17.56	17.61	5.71	-	-	-
LDS	5.62	4.86	15.79	**5.13**	-	-	-
LapSVM	**5.44**	**3.66**	**12.67**	10.41	18.1	10.5	**6.4**

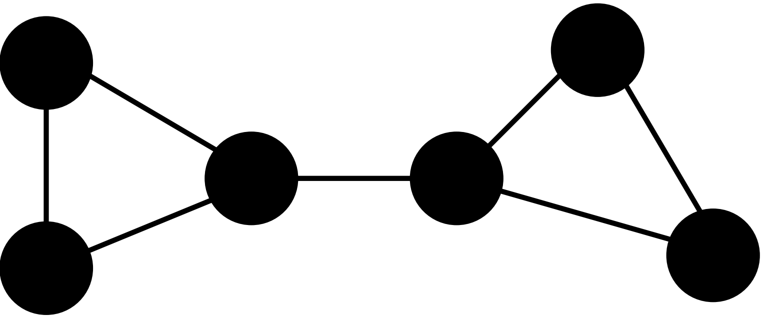
Geometry of clustering

Probability distribution P.

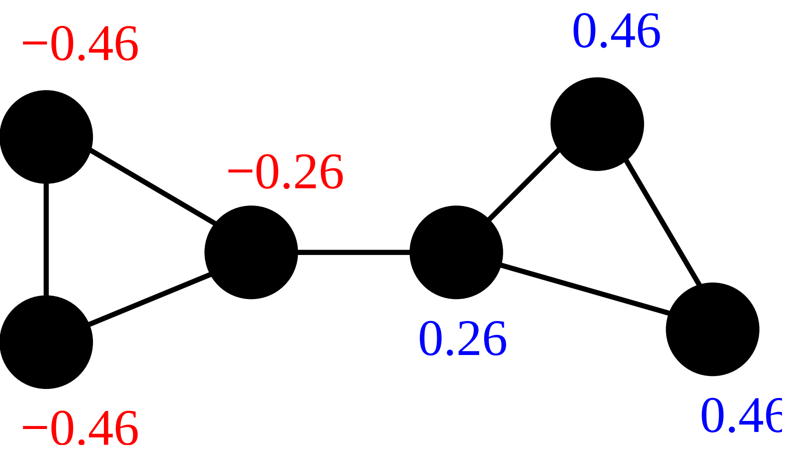
What are clusters? Geometric question.

How does one estimate clusters given finite data?

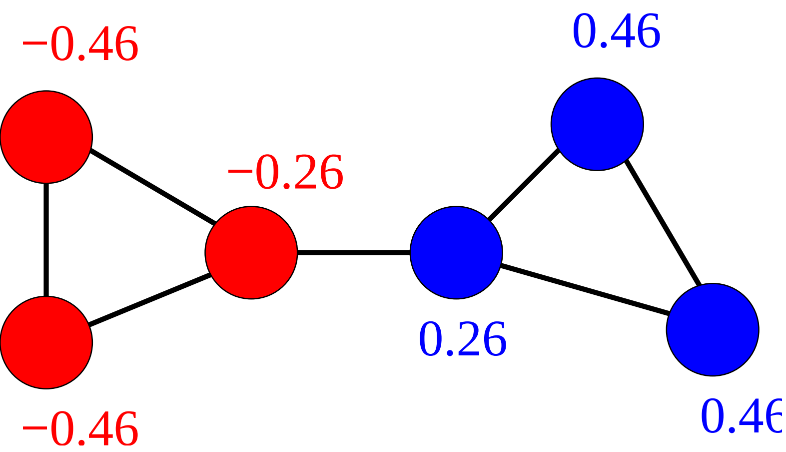
Spectral graph clustering



Spectral graph clustering



Spectral graph clustering

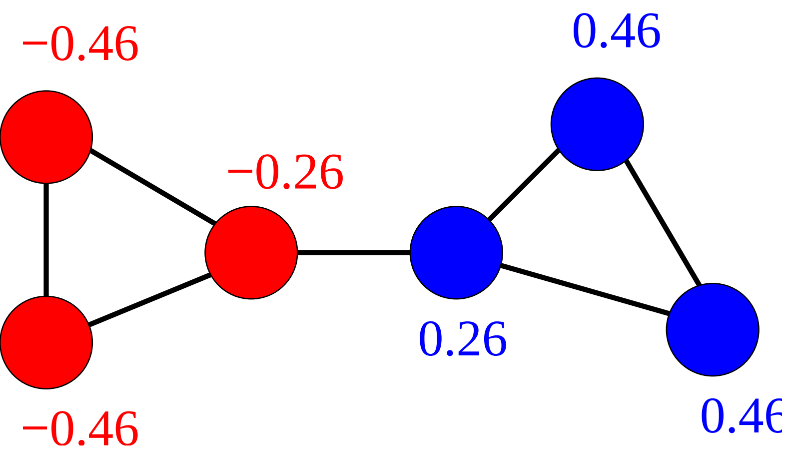


$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

Unnormalized clustering:

$$L\mathbf{e_1} = \lambda_1 \mathbf{e_1} \qquad \mathbf{e_1} = [-0.46, -0.46, -0.26, 0.26,\ 0.46,\ 0.46]$$

Spectral graph clustering



$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

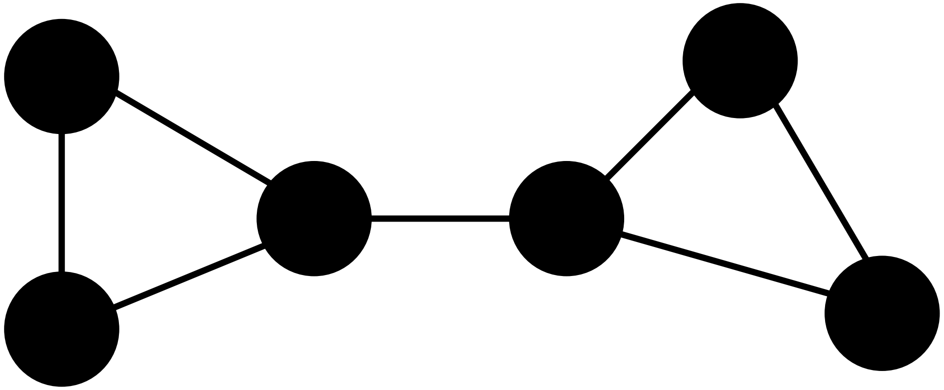
Unnormalized clustering:

$$L\mathbf{e_1} = \lambda_1 \mathbf{e_1} \qquad \mathbf{e_1} = [-0.46, -0.46, -0.26, 0.26,\ 0.46,\ 0.46]$$

Normalized clustering:

$$L\mathbf{e_1} = \lambda_1 D\mathbf{e_1} \qquad \mathbf{e_1} = [-0.31, -0.31, -0.18, 0.18,\ 0.31,\ 0.31]$$

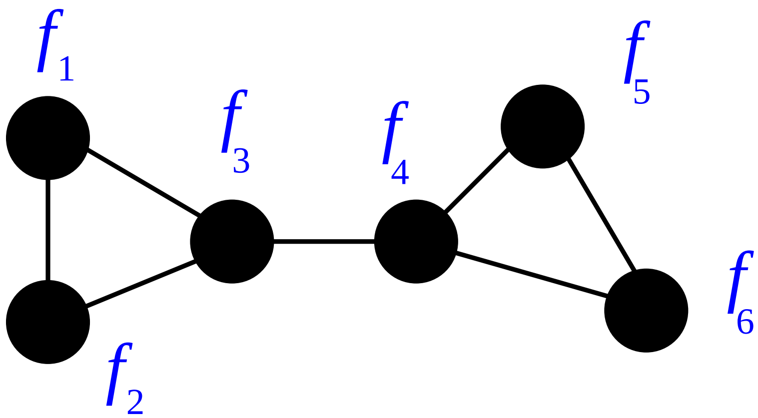
Graph Clustering: Mincut



Mincut: minimize the number (total weight) of edges cut).

$$\operatorname*{argmin}_{S} \sum_{i \in S,\ j \in V-S} w_{ij}$$

Graph Laplacian

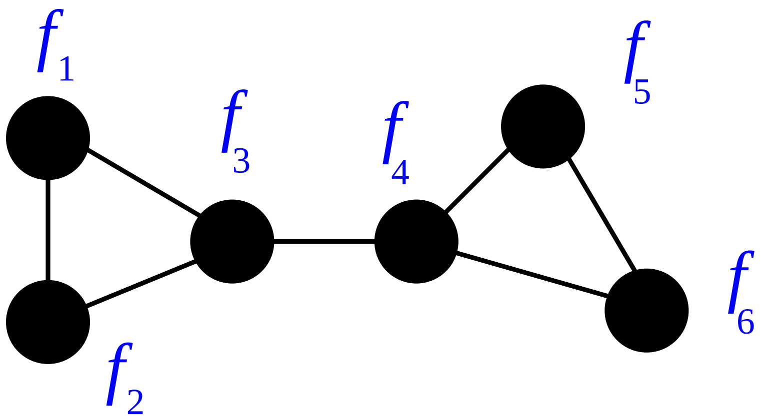


$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

Basic fact:

$$\sum_{i \sim j} (f_i - f_j)^2 w_{ij} = \frac{1}{2} \mathbf{f}^t \, \mathbf{L} \, \mathbf{f}$$

Graph Laplacian



$$\mathbf{L} = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$$\underset{S}{\operatorname{argmin}} \sum_{i \in S,\ j \in V-S} w_{ij} = \underset{f_i \in \{-1,1\}}{\operatorname{argmin}} \sum_{i \sim j} (f_i - f_j)^2 = \frac{1}{8} \underset{f_i \in \{-1,1\}}{\operatorname{argmin}} \mathbf{f}^t \, \mathbf{L} \, \mathbf{f}$$

Relaxation gives eigenvectors.

$$\mathbf{L}v = \lambda v$$

Consistency of spectral clustering

Limit behavior of spectral clustering.

$$\mathbf{x}_1, \ldots, \mathbf{x}_n \qquad\qquad n \to \infty$$

Sampled from probability distribution P on X.

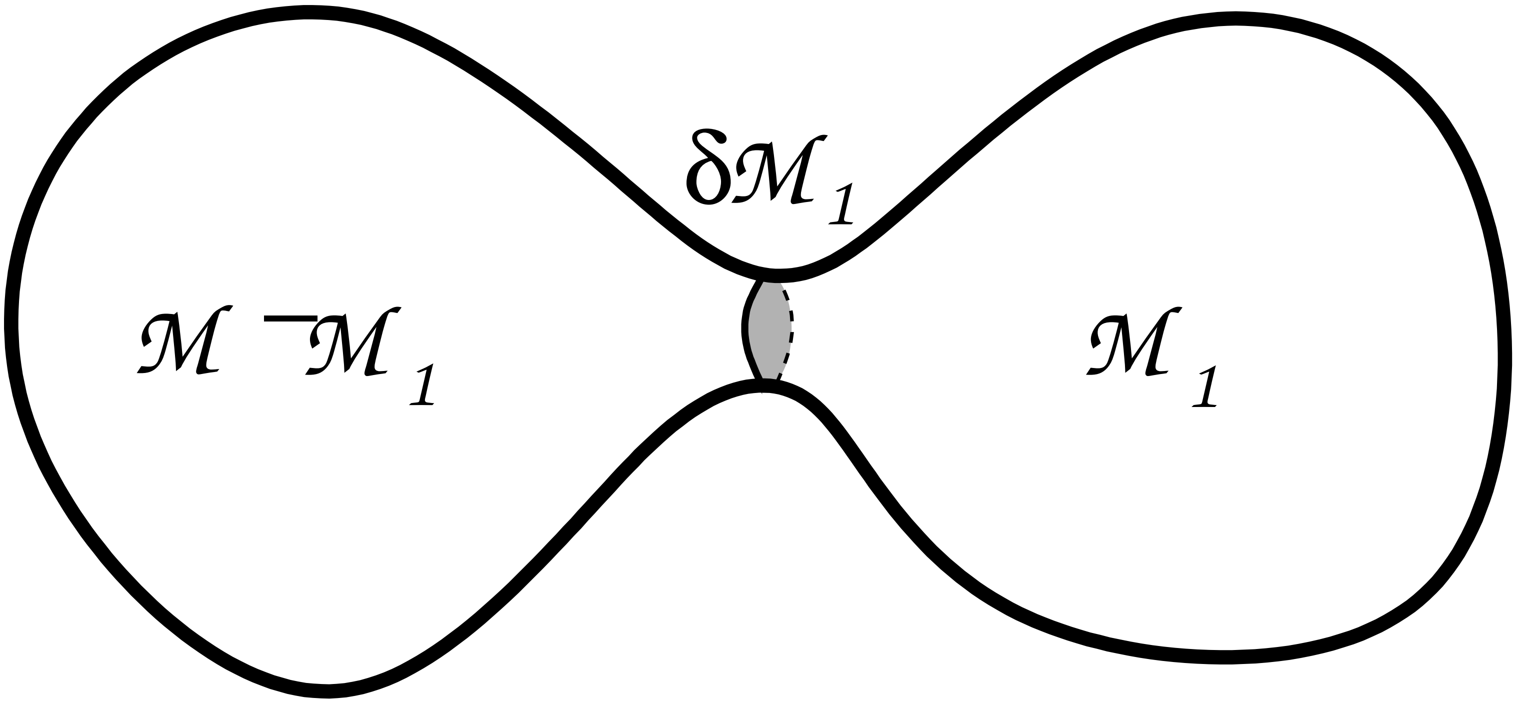
Theorem 1:
Normalized spectral clustering (bisectioning) is consistent.

Theorem 2:
Unnormalized spectral clustering may not converge depending on the spectrum of L and P.

von Luxburg Belkin Bousquet 04

Continuous Cheeger clustering

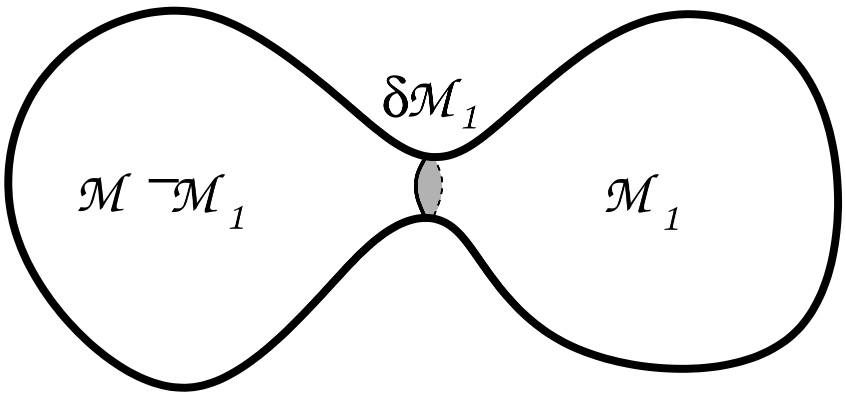
Isoperimetric problem. Cheeger constant.



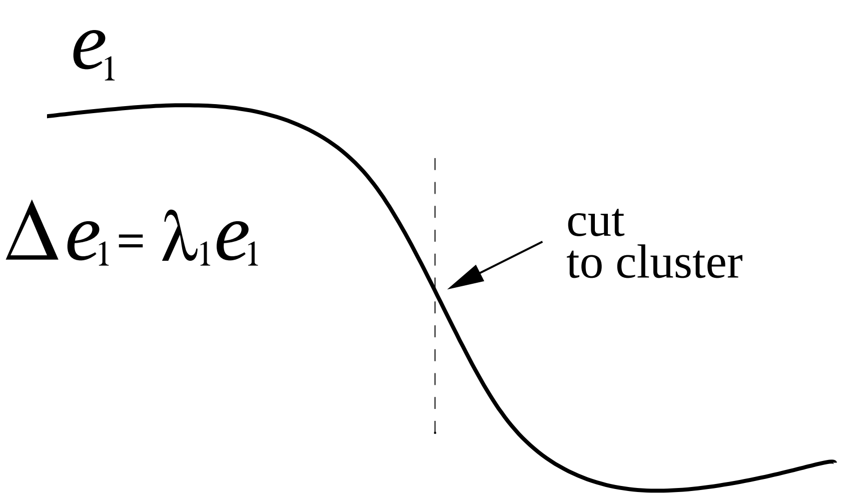
$$h = \inf \frac{\mathrm{vol}^{n-1}(\delta \mathcal{M}_1)}{\min\left(\mathrm{vol}^n(\mathcal{M}_1), \mathrm{vol}^n(\mathcal{M} - \mathcal{M}_1)\right)}$$

Continuous spectral clustering

Laplacian eigenfunction as a relaxation of the isoperimetric problem.



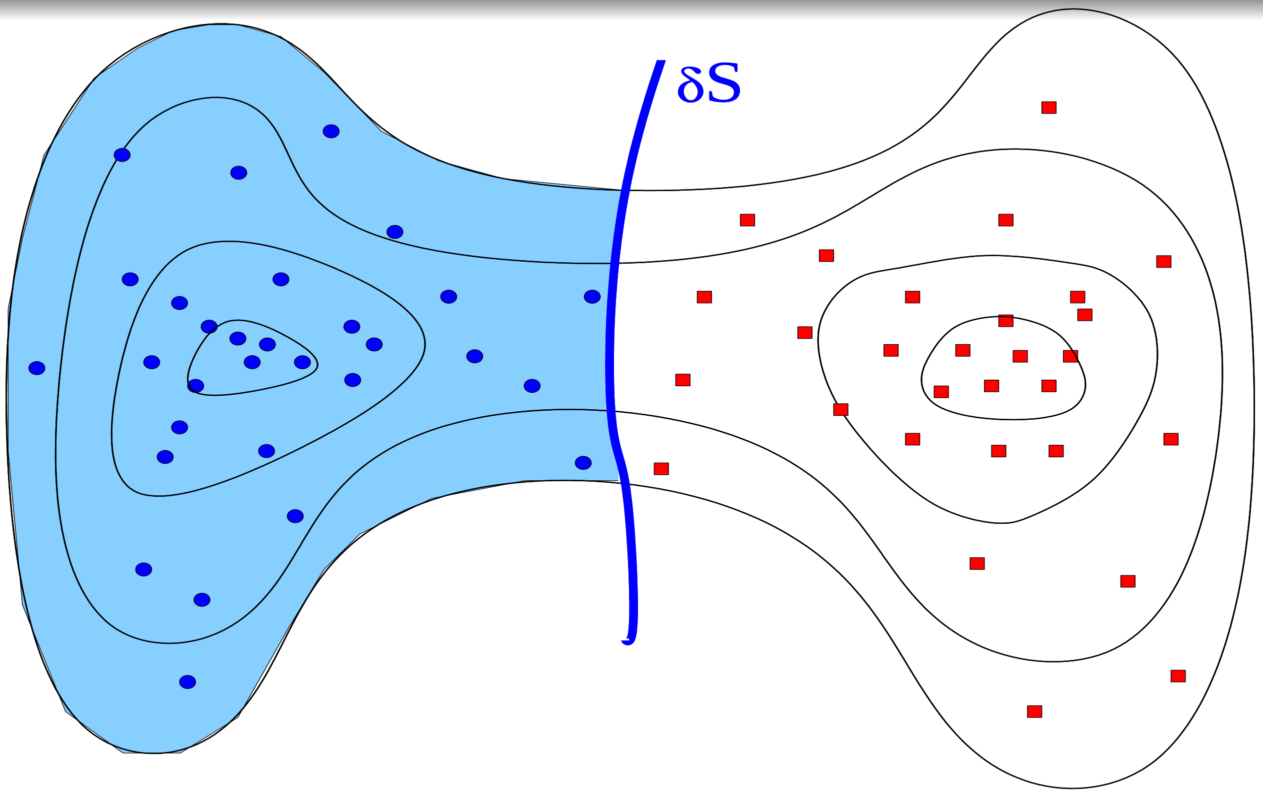
$$h = \inf \frac{\mathrm{vol}^{n-1}(\delta \mathcal{M}_1)}{\min\left(\mathrm{vol}^n(\mathcal{M}_1), \mathrm{vol}^n(\mathcal{M} - \mathcal{M}_1)\right)}$$



$$0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots$$

$$h \leq \frac{\sqrt{\lambda_1}}{2}$$ [Cheeger]

Estimating volumes of cuts



$$\sum_{i \in \text{blue}} \sum_{j \in \text{red}} \frac{w_{ij}}{\sqrt{d_j d_j}}$$

$$w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{4t}}$$

$$d_i = \sum_j w_{ij}$$

Theorem:

$$\text{vol}(\delta S) \approx \frac{2}{N} \frac{1}{(4\pi t)^{n/2}} \sqrt{\frac{\pi}{t}}\, \mathbf{1}_S^t \, L \, \mathbf{1}_S$$

L is the normalized graph Laplacian and $\mathbf{1}_S$ is the indicator vector of points in S. (Narayanan Belkin Niyogi, 06)

Clustering

- Clustering is all about geometry of unlabeled data (no labeled data!).
- Need to combine probability density with the geometry of the total space.

Future Directions

- Machine Learning
  - Scaling Up
  - Multi-scale
  - Geometry of Natural Data
  - Geometry of Structured Data
- Algorithmic Nash embedding
- Graphics / Non-randomly sampled data
- Random Hodge Theory
- Partial Differential Equations
- Algorithms