Geometric Methods and Manifold Learning

Mikhail Belkin, Ohio State University,
prepared jointly with Partha Niyogi
When can we avoid the curse of dimensionality?

- **Smoothness**
  
  rate $\approx (1/n)^{\frac{\delta}{d}}$
  
  splines, kernel methods, $L_2$ regularization...

- **Sparsity**
  
  wavelets, $L_1$ regularization, LASSO, compressed sensing..

- **Geometry**
  
  graphs, simplicial complexes, laplacians, diffusions
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:** $\mathcal{M} \rightarrow \{1, \ldots, k\}$
  connected components, min cut

- **Classification:** $\mathcal{M} \rightarrow \{-1, +1\}$
  $P$ on $\mathcal{M} \times \{-1, +1\}$

- **Dimensionality Reduction:** $f : \mathcal{M} \rightarrow \mathbb{R}^n$, $n << N$

- $\mathcal{M}$ 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 \( x_1, \ldots, x_n \in \mathbb{R}^D \)

Find \( y_1, \ldots, y_n \in \mathbb{R} \) such that

\[
y_i = w \cdot x_i
\]

and

\[
\max_w \text{Variance}(\{y_i\}) = \sum_i y_i^2 = w^T \left( \sum_i x_ix_i^T \right) w
\]

\( w_* = \text{leading eigenvector of} \ \sum_i x_ix_i^T \)
Suppose data does not lie on a linear subspace.

Yet data has inherently one degree of freedom.
An Acoustic Example

\[ u(t) \quad l \quad s(t) \]
An Acoustic Example

One Dimensional Air Flow

\[
\begin{align*}
\frac{\partial V}{\partial x} &= -\frac{A}{\rho c^2} \frac{\partial P}{\partial t} \\
\frac{\partial P}{\partial x} &= -\frac{\rho}{A} \frac{\partial V}{\partial t}
\end{align*}
\]

\( V(x, t) = \text{volume velocity} \)

\( P(x, t) = \text{pressure} \)
\[ u(t) = \sum_{n=1}^{\infty} \alpha_n \sin(n\omega_0 t) \in l_2 \]

\[ s(t) = \sum_{n=1}^{\infty} \beta_n \sin(n\omega_0 t) \in l_2 \]
Vocal Tract modeled as a sequence of tubes. (e.g. Stevens, 1998)
Vision Example

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

\[ \mathcal{F} = \{ f | f(x, y) = v(x - t, y - r) \} \]
\[ g : S^2 \times S^2 \times S^2 \rightarrow \mathbb{R}^3 \]

\[ \langle (\theta_1, \phi_1), (\theta_2, \phi_2), (\theta_3, \phi_3) \rangle \rightarrow (x, y, z) \]
Learning when data $\sim \mathcal{M} \subset \mathbb{R}^N$

- **Clustering:** $\mathcal{M} \rightarrow \{1, \ldots, k\}$
  connected components, min cut

- **Classification/Regression:** $\mathcal{M} \rightarrow \{-1, +1\}$ or $\mathcal{M} \rightarrow \mathbb{R}$
  $P$ on $\mathcal{M} \times \{-1, +1\}$ or $P$ on $\mathcal{M} \times \mathbb{R}$

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

- $\mathcal{M}$ unknown: what can you learn about $\mathcal{M}$ from data?
  e.g. dimensionality, connected components
  holes, handles, homology
  curvature, geodesics
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.
Tangent space

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

$T_p\mathcal{M}^k \subset \mathbb{R}^N$
Tangent vectors and curves
Tangent vectors and curves

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

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

Tangent vectors \(\longrightarrow\) curves.
Tangent vectors as derivatives

\[ f : \mathcal{M}^k \rightarrow \mathbb{R} \]
Tangent vectors as derivatives

Tangent vectors as derivatives

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

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

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

\[ \frac{df}{dv} = \frac{df(\phi(t))}{dt} \bigg|_0 \]

Tangent vectors \(\longleftrightarrow\) Directional derivatives.
Riemannian geometry

Norms and angles in tangent space.

\[ \langle v, w \rangle, \|v\|, \|w\| \]
Length of curves and geodesics

Can measure length using norm in tangent space.

Geodesic — shortest curve between two points.
Gradient

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

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

Tangent vectors \(\xrightarrow{\quad}\) 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 \( \phi(t) \)

\[ \phi(0) = p, \quad \phi(\|v\|) = q \quad \frac{d\phi(t)}{dt} \bigg|_0 = v \]
Laplace-Beltrami operator

\[ \Delta f(p) \equiv \sum_i \frac{\partial^2 f(\exp_p(x))}{\partial x_i^2} \]

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

Neighborhood graph common to all methods.
1. Construct Neighborhood Graph.
2. Find **shortest path (geodesic)** distances.

\[ D_{ij} \text{ is } n \times n \]

Multidimensional Scaling

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

1. Inner product from distances:

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

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

Answer:

$$A = -\frac{1}{2} HDH \text{ where } H = I - \frac{1}{n}11^T$$

In general only an approximation.
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.
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:

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

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

\[
 = -\frac{1}{2} \sum x_i^t H x_i \approx -\text{tr} H = \Delta f
\]
Step 1 [Constructing the Graph]

\[ e_{ij} = 1 \iff x_i \text{ “close to” } x_j \]

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

\[ ||x_i - x_j||^2 < \epsilon \]

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 \( i \).
Step 2. [Choosing the weights].

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

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

2. **Simple-minded.** [No parameters]. $W_{ij} = 1$ if and only if vertices $i$ and $j$ are connected by an edge.
Step 3. [Eigenmaps] Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

\[ Lf = \lambda Df \]

\( D \) is diagonal matrix where

\[ D_{ii} = \sum_j W_{ij} \]

\[ L = D - W \]

Let \( f_0, \ldots, f_{k-1} \) be eigenvectors.

Leave out the eigenvector \( f_0 \) and use the next \( m \) lowest eigenvectors for embedding in an \( m \)-dimensional Euclidean space.
Heat diffusion operator $H^t$.

$\delta_x$ and $\delta_y$ 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:

\[ x \rightarrow (e^{-\lambda_1 t}f_1(x), e^{-\lambda_2 t}f_2(x), \ldots) \]

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

Closely related to random walks on graphs.
Find $y_1, \ldots, y_n \in \mathbb{R}$

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

Tries to preserve locality
A Fundamental Identity

But

\[ \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 W_{ij} = y^T Ly \]

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

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

\[ = 2y^T Ly \]
Embedding

\[ \lambda = 0 \rightarrow y = 1 \]

\[ \min_{y^T 1 = 0} y^T L y \]

Let \( Y = [y_1 \, y_2 \, \ldots \, y_m] \)

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

subject to \( Y^T Y = I \).

Use eigenvectors of \( L \) to embed.
PCA versus Laplacian Eigenmaps
smooth map $f : \mathcal{M} \rightarrow \mathbb{R}$

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

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

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial z_1} \\ \frac{\partial f}{\partial z_2} \\ \vdots \\ \frac{\partial f}{\partial z_k} \end{bmatrix}$$
Curves on Manifolds

Consider a curve on $\mathcal{M}$

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

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

$$|f(0) - f(\tau)| \leq d_G(p, q)\|\nabla_M f(p)\|$$
Stokes Theorem

A Basic Fact

\[ \int_{\mathcal{M}} \| \nabla \mathcal{M} f \|^2 = \int f \cdot \Delta \mathcal{M} f \]

This is like

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

where

\[ \Delta \mathcal{M} f \] is the manifold Laplacian
Recall ordinary Laplacian in $\mathbb{R}^k$

This maps 

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

Manifold Laplacian is the same on the tangent space.
Properties of Laplacian

Eigensystem

\[ \Delta_M f = \lambda_i \phi_i \]

\[ \lambda_i \geq 0 \text{ and } \lambda_i \to \infty \]

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

\[ \int \| \nabla_M \phi_i \|^2 = \lambda_i \]
The Circle: An Example

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

Eigenvalues are

\[ \lambda_n = n^2 \]

Eigenfunctions are

\[ \sin(nt), \cos(nt) \]
From graphs to manifolds

\[ f : \mathcal{M} \rightarrow \mathbb{R} \quad x \in \mathcal{M} \quad x_1, \ldots, x_n \in \mathcal{M} \]

Graph Laplacian:

\[
L_t^n(f)(x) = f(x) \sum_j e^{-\frac{\|x-x_j\|^2}{t}} - \sum_j f(x_j)e^{-\frac{\|x-x_j\|^2}{t}}
\]

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

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

Belkin 03, Lafon Coifman 04, Belkin Niyogi 05, Hein et al 05
Theorem [convergence of eigenfunctions]

\[
\lim_{t \to 0, n \to \infty} Eig[L_n^t] \rightarrow Eig[\Delta M]
\]

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
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.
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 $t \rightarrow 0$ 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 arise for manifolds:

- Do not know distances.
- Do not know the heat kernel.
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:
\[ \mu = \mu_M + \mu_{\mathbb{R}^N} \]

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

- 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.
- 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.
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

![Graph showing the error rate for k-NN and Geodesic k-NN algorithms as a function of the number of labeled points. The graph demonstrates a decrease in error rate as the number of labeled points increases, with Geodesic k-NN generally performing better than k-NN.](image-url)
Cluster assumption
Cluster assumption
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 $X \rightarrow Y$. Probability distribution $P$ on $X \times Y$.

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

\[
\int_X \|\nabla f\|^2 p(x) = \langle f, \Delta_p f \rangle_X
\]
Function $f : X \rightarrow \mathbb{R}$. 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

SVM

\[ \gamma_A = 0.03125 \quad \gamma_I = 0 \]
Example

SVM

\[ \gamma_A = 0.03125 \quad \gamma_I = 0 \]

Laplacian SVM

\[ \gamma_A = 0.03125 \quad \gamma_I = 0.01 \]

Laplacian SVM

\[ \gamma_A = 0.03125 \quad \gamma_I = 1 \]
Regularization

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

Data: $(x_1, y_1), \ldots, (x_l, y_l)$

Regularized least squares (hinge loss for SVM):

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{l} \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \lambda \|f\|^2_K$$

fit to data + smoothness penalty

$\|f\|_K$ incorporates our smoothness assumptions.

Choice of $\|f\|_K$ is important.
Algorithm: RLS/SVM

Solve: \[ f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{l} \sum_{i=1}^{l} (f(x_i) - y_i)^2 + \lambda \|f\|^2_K \]

\[ \|f\|_K \] is a Reproducing Kernel Hilbert Space norm with kernel \( K(x, y) \).

Can solve explicitly (via Representer theorem):

\[ f^*(\cdot) = \sum_{i=1}^{l} \alpha_i K(x_i, \cdot) \]

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

\[ (K)_{ij} = K(x_i, x_j) \]
Estimate $f : \mathbb{R}^N \rightarrow \mathbb{R}$

Labeled data: $(x_1, y_1), \ldots, (x_l, y_l)$
Unlabeled data: $x_{l+1}, \ldots, x_{l+u}$

$$f^* = \arg\min_{f \in \mathcal{H}} \frac{1}{l} \sum (f(x_i) - y_i)^2 + \lambda_A \|f\|_K^2 + \lambda_I \|f\|_I^2$$

fit to data + extrinsic smoothness + intrinsic smoothness

Empirical estimate:

$$\|f\|_I^2 = \frac{1}{(l + u)^2} [f(x_1), \ldots, f(x_{l+u})]^t L [f(x_1), \ldots, f(x_{l+u})]^t$$
Representer theorem (discrete case):

\[ f^*(\cdot) = \sum_{i=1}^{l+u} \alpha_i K(x_i, \cdot) \]

Explicit solution for quadratic loss:

\[ \tilde{\alpha} = (JK + \lambda_A l I + \frac{\lambda_I l^2}{(u + l)^2} LK)^{-1}[y_1, \ldots, y_l, 0, \ldots, 0]^t \]

\[ (K)_{ij} = K(x_i, x_j), \quad J = \text{diag}(1, \ldots, 1, 0, \ldots, 0) \]
Experimental results: USPS

- **RLS vs LapRLS**
- **SVM vs LapSVM**
- **TSVM vs LapSVM**

Error Rates

- **Out-of-Sample Extension**
- **Std Deviation of Error Rates**

- SVM (○), TSVM (x) Std Dev
Experimental comparisons

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

Unnormalized clustering:

\[ L e_1 = \lambda_1 e_1 \quad e_1 = [-0.46, -0.46, -0.26, 0.26, 0.46, 0.46] \]
Spectral graph clustering

Unnormalized clustering:

\[ L e_1 = \lambda_1 e_1 \quad e_1 = [-0.46, -0.46, -0.26, 0.26, 0.46, 0.46] \]

Normalized clustering:

\[ L e_1 = \lambda_1 D e_1 \quad e_1 = [-0.31, -0.31, -0.18, 0.18, 0.31, 0.31] \]
Mincut: minimize the number (total weight) of edges cut).

\[ \arg \min_S \sum_{i \in S, \ j \in V - S} w_{ij} \]
Graph Laplacian

Basic fact:

\[ \sum_{i \sim j} (f_i - f_j)^2 w_{ij} = \frac{1}{2} f^t L f \]
Graph Laplacian

\[
L = \begin{pmatrix}
2 & -1 & -1 & 0 & 0 & 0 \\
-1 & 2 & -1 & 0 & 0 & 0 \\
-1 & -1 & 3 & -1 & 0 & 0 \\
0 & 0 & -1 & 3 & -1 & -1 \\
0 & 0 & 0 & -1 & 2 & -1 \\
0 & 0 & 0 & -1 & -1 & 2
\end{pmatrix}
\]

\[
\argmin_S \sum_{i \in S, j \in V - S} w_{ij} = \argmin_{f_i \in \{-1, 1\}} \sum_{i \sim j} (f_i - f_j)^2 = \frac{1}{8} \argmin_{f_i \in \{-1, 1\}} f^t L f
\]

Relaxation gives eigenvectors.

\[
L v = \lambda v
\]
Consistency of spectral clustering

Limit behavior of spectral clustering.

\[ x_1, \ldots, x_n \quad n \to \infty \]

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.

\[ h = \inf \frac{\text{vol}^{n-1}(\delta M_1)}{\min(\text{vol}^n(M_1), \text{vol}^n(M - M_1))} \]
Continuous spectral clustering

Laplacian eigenfunction as a relaxation of the isoperimetric problem.

\[ h = \inf \frac{\text{vol}^{n-1}(\delta M_1)}{\min(\text{vol}^n(M_1), \text{vol}^n(M - M_1))} \]

\[ 0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \]

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

[Cheeger]
Theorem:

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

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

\[ d_i = \sum_j w_{ij} \]

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

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