

---

# ***Graphs, Geometry and Semi-supervised Learning***

**Mikhail Belkin**

The Ohio State University,  
Dept of Computer Science and Engineering and Dept of  
Statistics

Collaborators: Partha Niyogi, Vikas Sindhwani

# *Ubiquity of manifolds*

---

- ▶ In many domains (e.g., speech, some vision problems) data **explicitly** lies on a manifold.

# *Ubiquity of manifolds*

---

- ▶ In many domains (e.g., speech, some vision problems) data **explicitly** lies on a manifold.
- ▶ For **all sources** of high-dimensional data, true dimensionality is much lower than the number of features.

# *Ubiquity of manifolds*

---

- ▶ In many domains (e.g., speech, some vision problems) data **explicitly** lies on a manifold.
- ▶ For **all sources** of high-dimensional data, true dimensionality is much lower than the number of features.
- ▶ Much of the data is highly nonlinear.

Important point: **only small distances are meaningful.**  
In fact, all large distances are (almost) the same.

Important point: **only small distances are meaningful.**  
In fact, all large distances are (almost) the same.

Manifolds (Riemannian manifolds with a measure + noise) provide a natural mathematical language for thinking about **high-dimensional data.**

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

- ▶ **Clustering:**  $\mathcal{M} \rightarrow \{1, \dots, k\}$   
connected components, min cut, normalized 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:**  $f : \mathcal{M} \rightarrow \mathbb{R}^n$      $n \ll N$
- ▶  $\mathcal{M}$  unknown: what can you learn about  $\mathcal{M}$  from data?  
e.g. dimensionality, connected components  
holes, handles, homology  
curvature, geodesics

# *Graph-based methods*

---

Data — Probability Distribution

Graph — Manifold

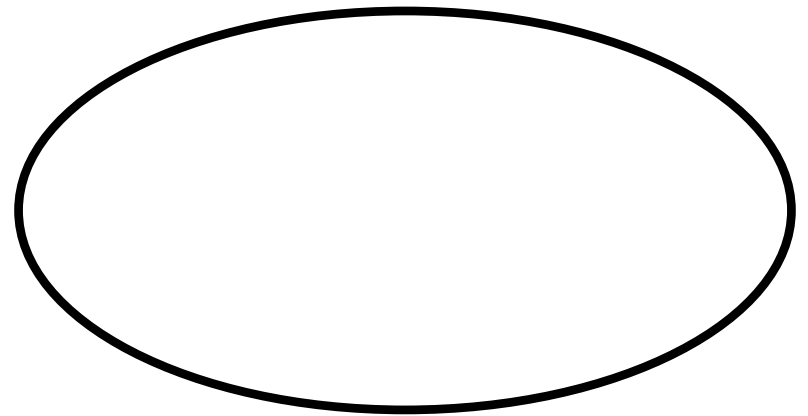
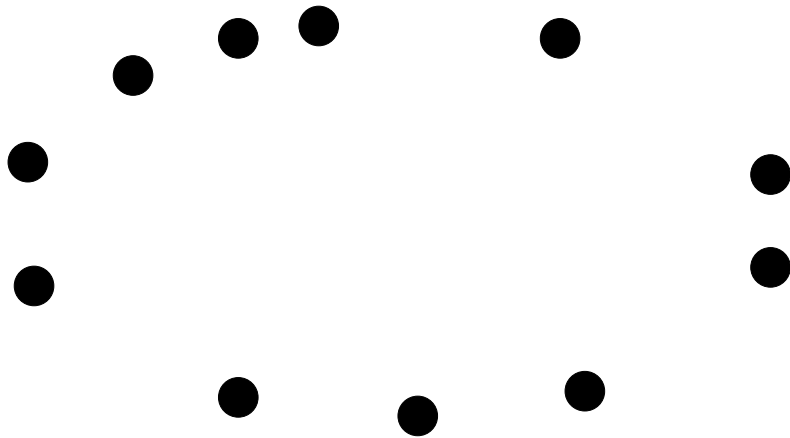


# *Graph-based methods*

---

Data ——— Probability Distribution

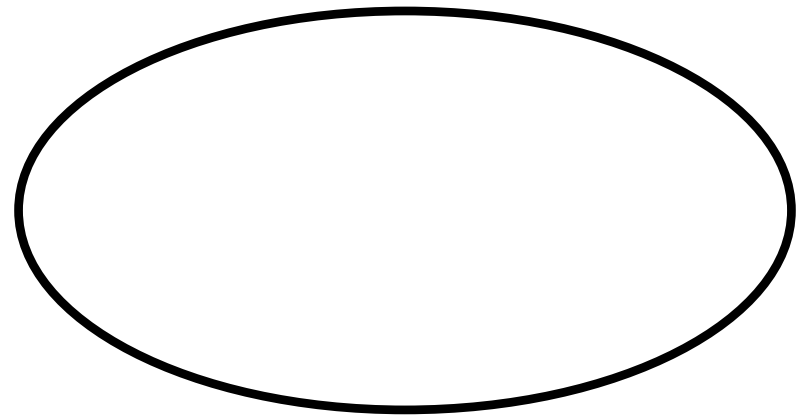
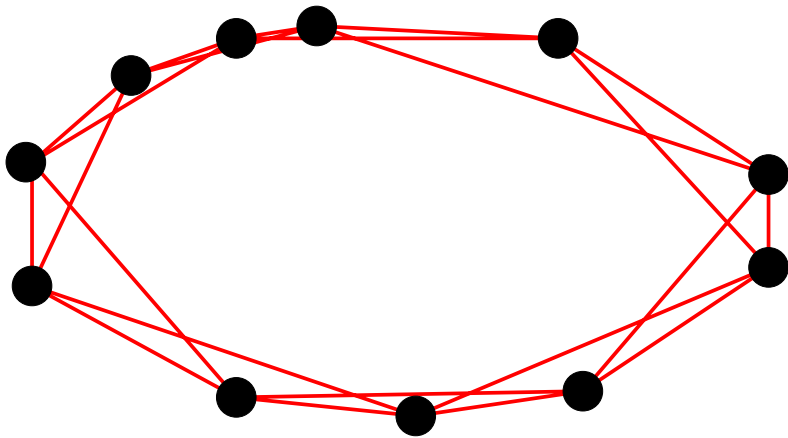
Graph ——— Manifold



# *Graph-based methods*

Data ——— Probability Distribution

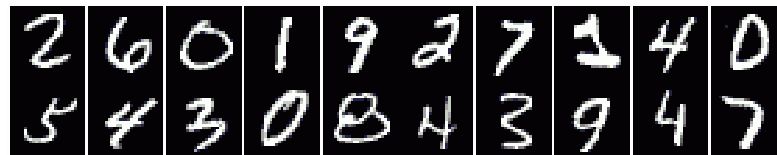
Graph ——— Manifold



Graph extracts underlying geometric structure.

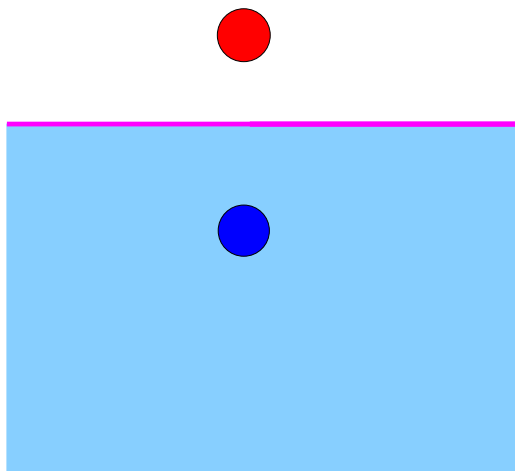
# Problems of machine learning

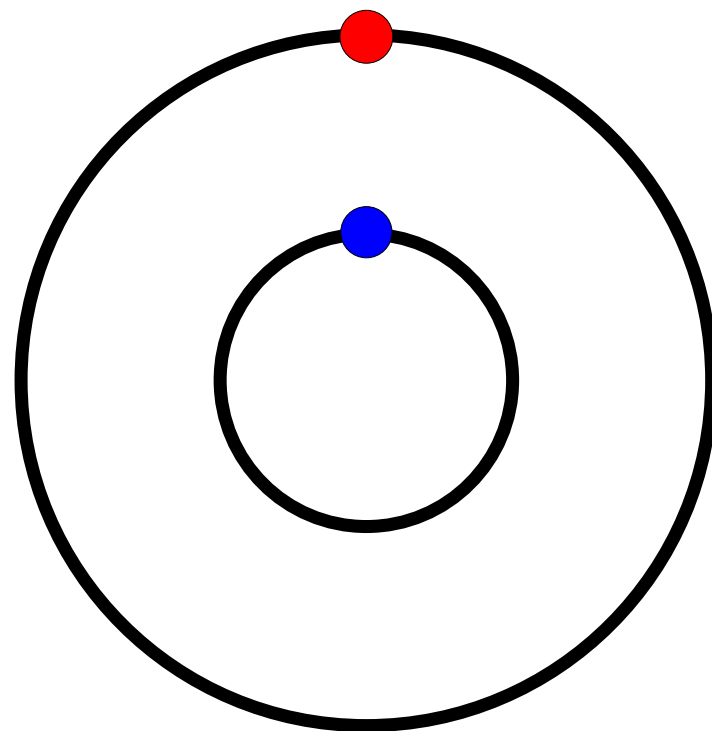
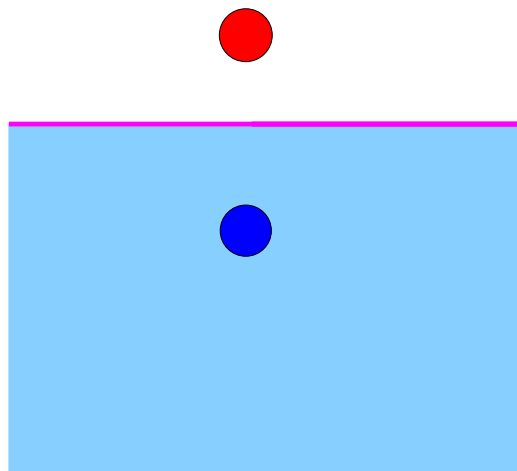
- ▶ Classification / regression.
- ▶ Data representation / dimensionality reduction.
- ▶ Clustering.

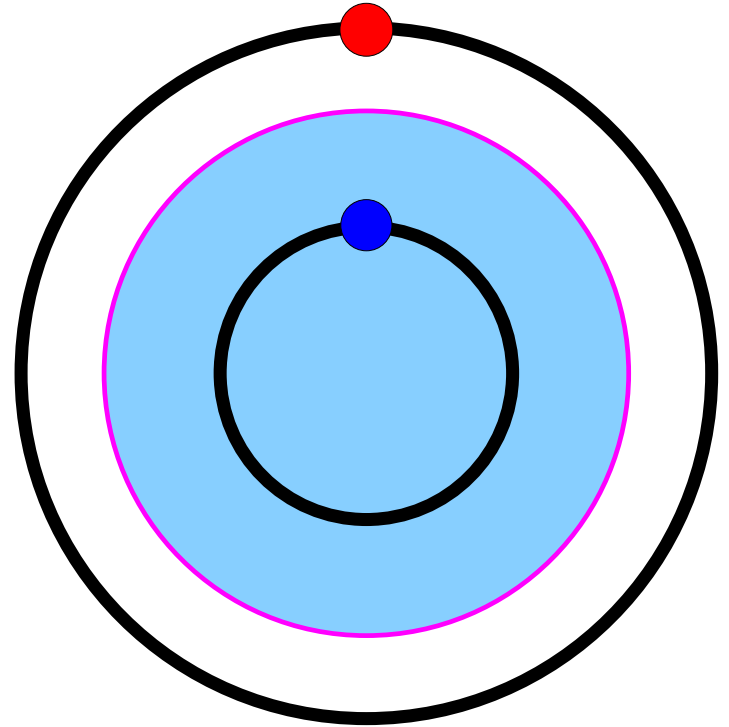
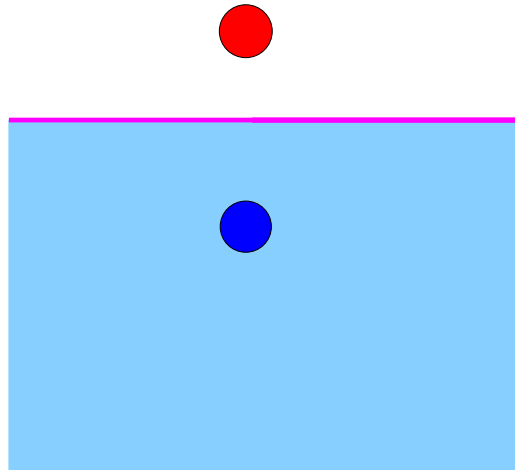


Common intuition – **similar objects have similar labels.**





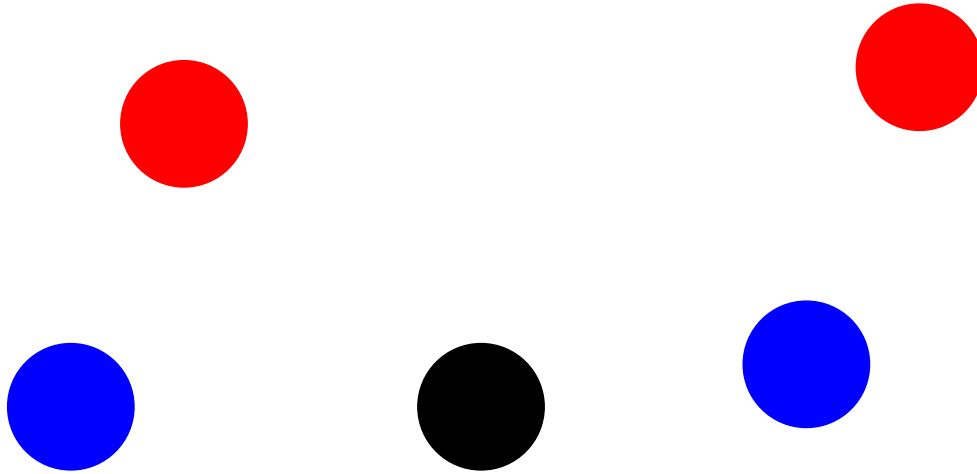




Geometry of data changes our notion of similarity.

# *Manifold assumption*

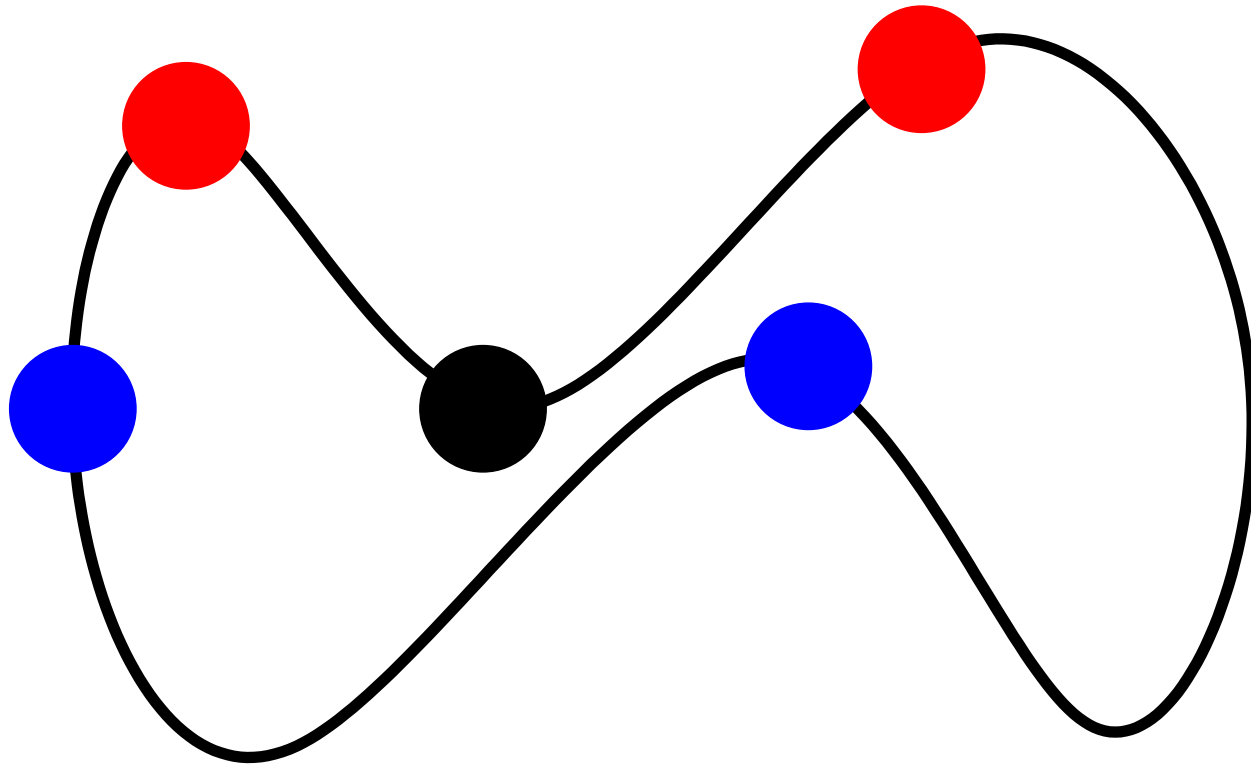
---





# *Manifold assumption*

---



Geometry is important.

# *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  $X \rightarrow Y$ . Probability distribution  $P$  on  $X \times Y$ .

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

## **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  $x \in X$ :

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

Total penalty – Laplace operator:

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

# What is smooth?

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

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

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.

Laplace operator is a fundamental geometric object.

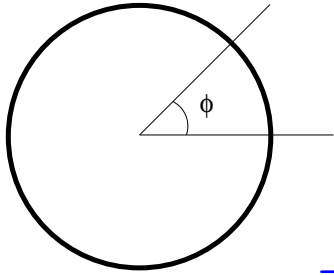
$$\Delta f = - \sum_{i=1}^k \frac{\partial^2 f}{\partial x_i^2}$$

The only differential operator invariant under translations and rotations.

Heat, Wave, Schroedinger equations.

Fourier analysis.

# Laplacian on the circle



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

Same as in  $\mathbb{R}$  with periodic boundary conditions.

Eigenvalues:

$$\lambda_n = n^2$$

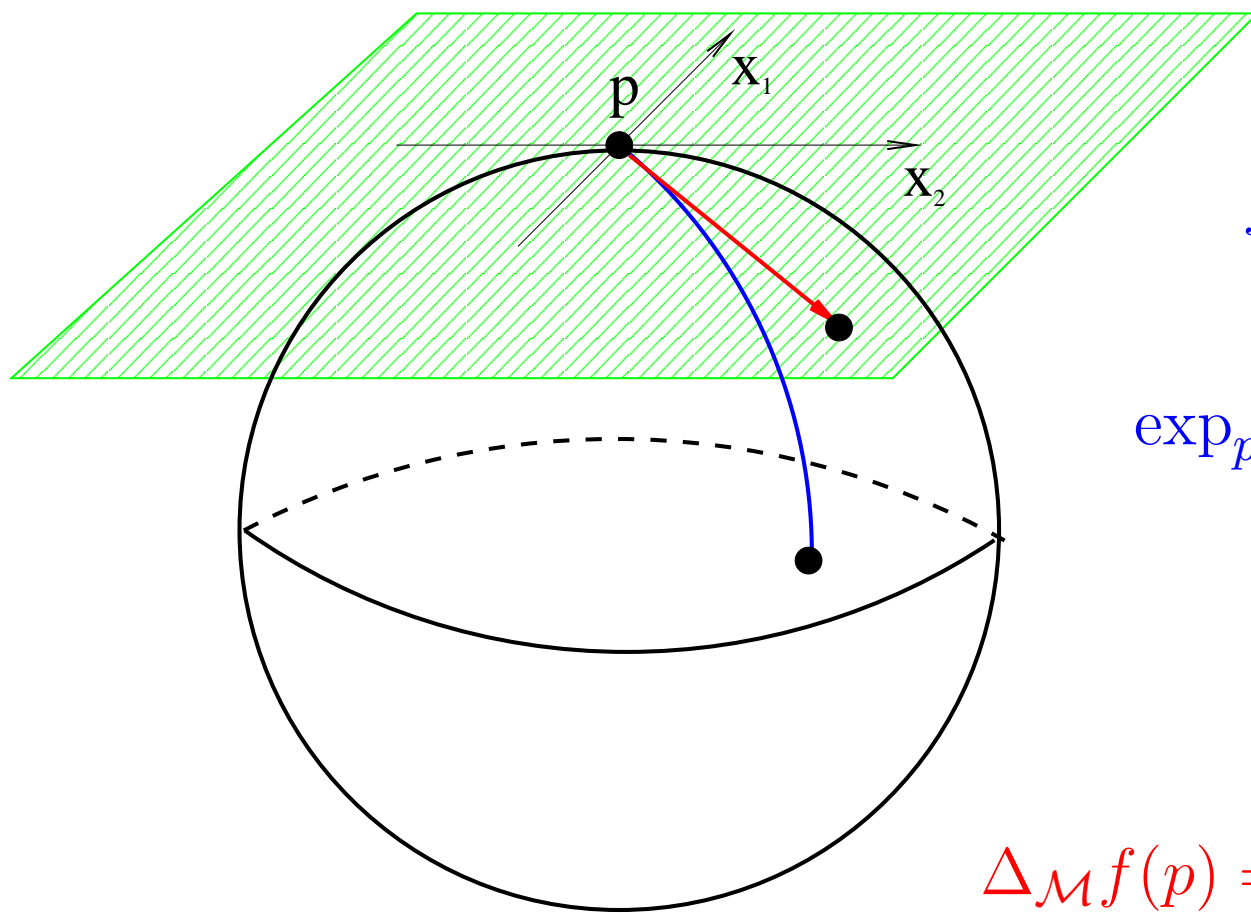
Eigenfunctions:

$$\sin(n\phi), \cos(n\phi)$$

Fourier analysis.



# Laplace-Beltrami operator



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

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

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

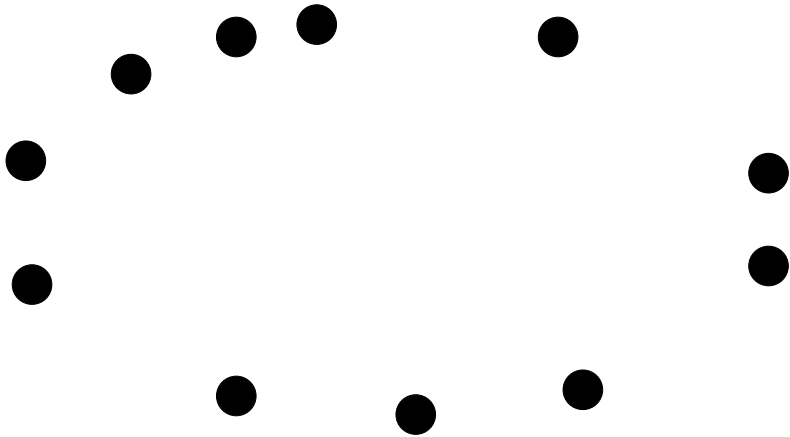
Generalization of Fourier analysis.

Machine learning: manifold is **unknown**.

How to do Fourier analysis/reconstruct  
Laplace operator on an **unknown  
manifold?**

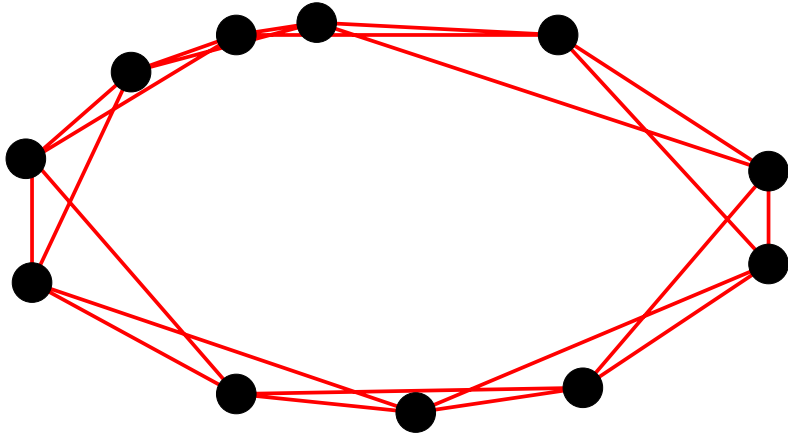
# *Algorithmic framework*

---

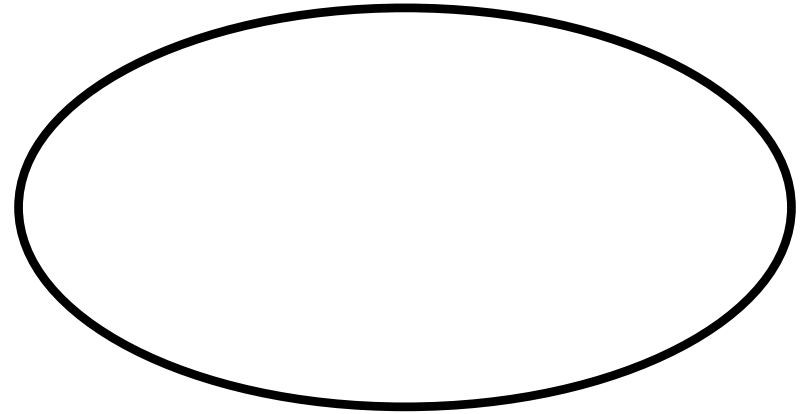
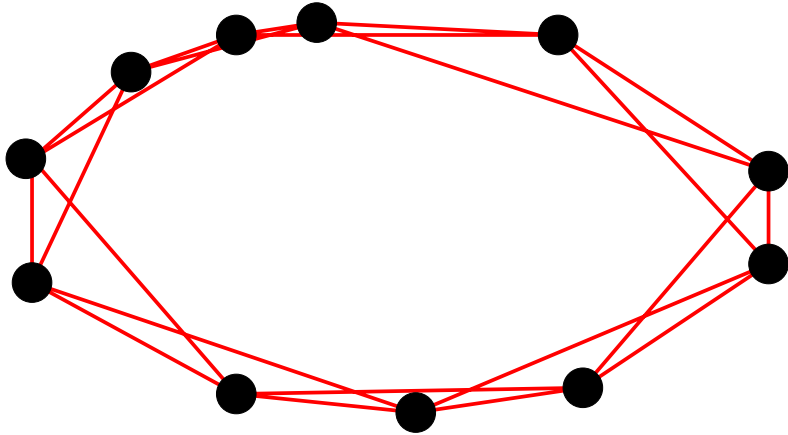


# *Algorithmic framework*

---



# Algorithmic framework

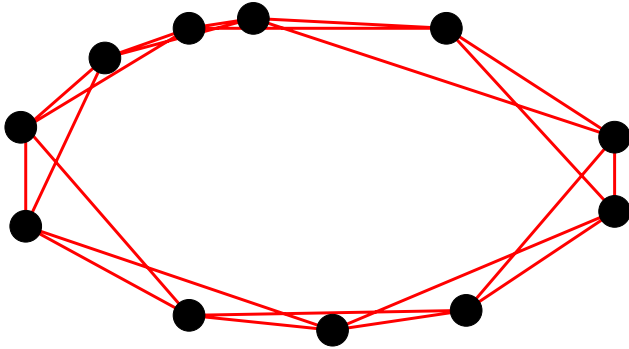


$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

[justification: **heat equation**]

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

$$\mathbf{f}^t \mathbf{L} \mathbf{f} = 2 \sum_{i,j} e^{-\frac{\|x_i - x_j\|^2}{t}} (f_i - f_j)^2$$



$$f : G \rightarrow \mathbb{R}$$

$$\text{Minimize } \sum_{i \sim j} w_{ij} (f_i - f_j)^2$$

Preserve adjacency.

Solution:  $Lf = \lambda f$  (slightly better  $Lf = \lambda Df$ )

Lowest eigenfunctions of  $L$  ( $\tilde{L}$ ).

## Laplacian Eigenmaps

Related work: LLE: Roweis, Saul 00; Isomap: Tenenbaum, De Silva, Langford 00

Hessian Eigenmaps: Donoho, Grimes, 03; Diffusion Maps: Coifman, et al, 04

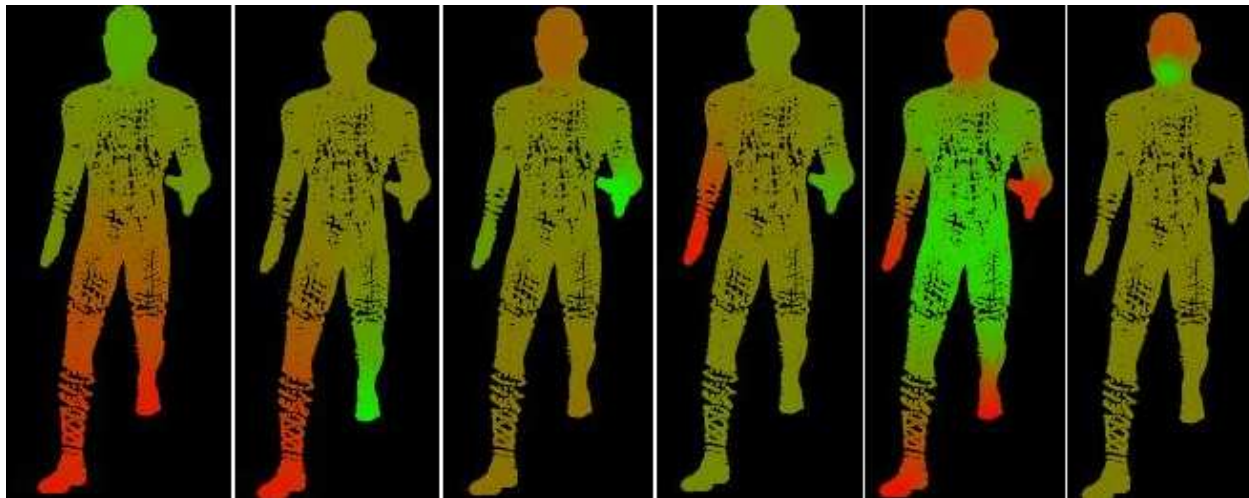
# Laplacian Eigenmaps

- ▶ Visualizing spaces of digits and sounds.

Partiview, Ndaona, Surendran 04

- ▶ Machine vision: inferring joint angles.

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



Isometrically invariant representation. [[link](#)]

- ▶ Reinforcement Learning: value function approximation. Mahadevan, Maggioni, 05

# *Semi-supervised learning*

---

Learning from labeled and unlabeled data.

- ▶ Unlabeled data is everywhere. Need to use it.
- ▶ Natural learning is semi-supervised.



# *Semi-supervised learning*

---

Learning from labeled and unlabeled data.

- ▶ Unlabeled data is everywhere. Need to use it.
- ▶ Natural learning is semi-supervised.

Labeled data:  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l) \in \mathbb{R}^N \times \mathbb{R}$

Unlabeled data:  $\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u} \in \mathbb{R}^N$

Need to reconstruct

$$f_{L,U} : \mathbb{R}^N \rightarrow \mathbb{R}$$

A lot of recent work. Here are a few early papers:

- ▶ Partially labeled classification with Markov random walks.

Martin Szummer, Tommi Jaakkola, 01.

- ▶ Learning from Labeled and Unlabeled Data using Graph Mincuts.

A Blum, S Chawla, 01.

- ▶ Cluster kernels for semi-supervised learning.

O. Chapelle, J. Weston, and B. Schoelkopf, 02.

- ▶ Using Manifold Structure for Partially Labelled Classification.

M. Belkin, P. Niyogi, 02.

- ▶ Diffusion Kernels on Graphs and Other Discrete Input Spaces.

R. Kondor, J. Lafferty, 02.

- ▶ Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions.

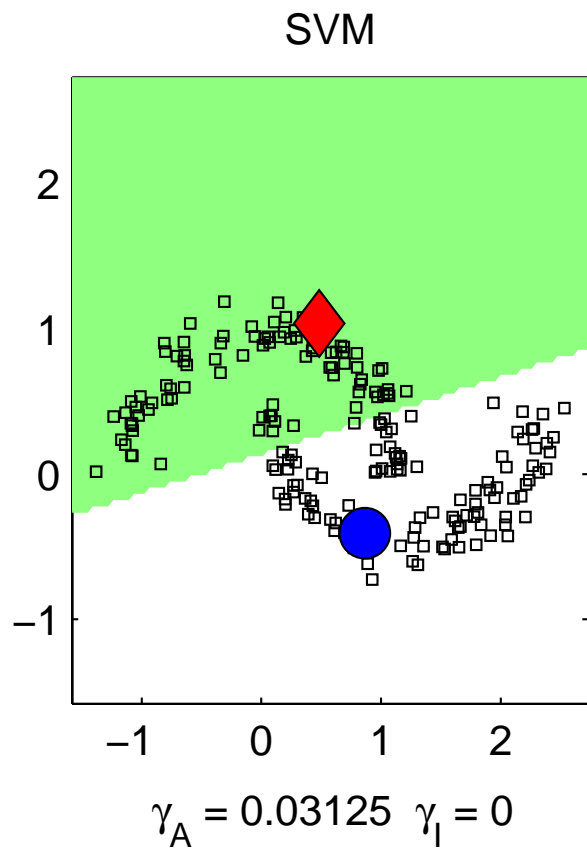
Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, 03

- ▶ Transductive Learning via Spectral Graph Partitioning.

T. Joachims, 03.

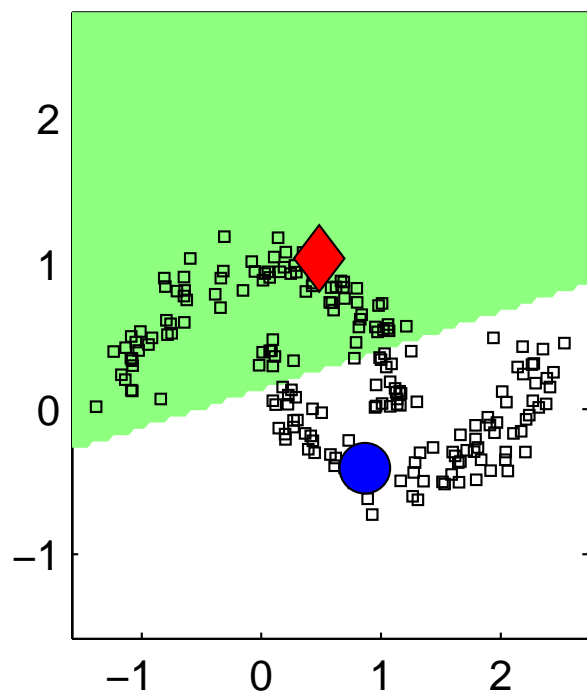
Will discuss Manifold Regularization framework.

- ▶ Extends SVM/RLS for unlabeled data. Standard SVM is a special case of the framework.
- ▶ Provides natural out-of-sample extension.



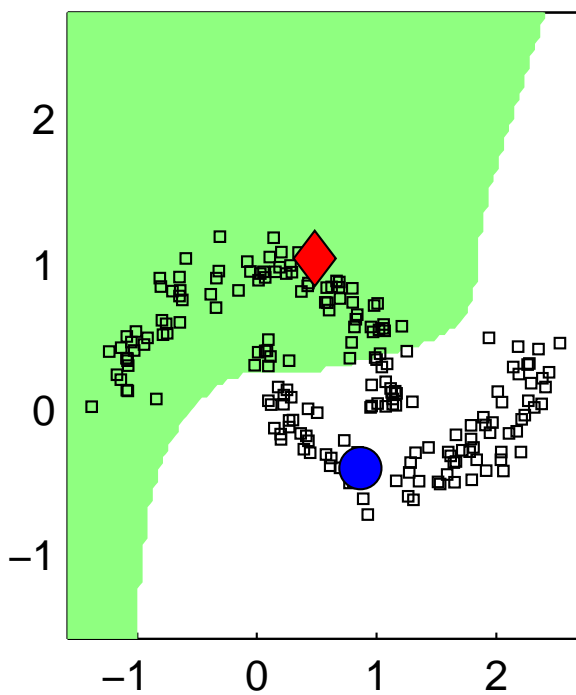
# Example

SVM



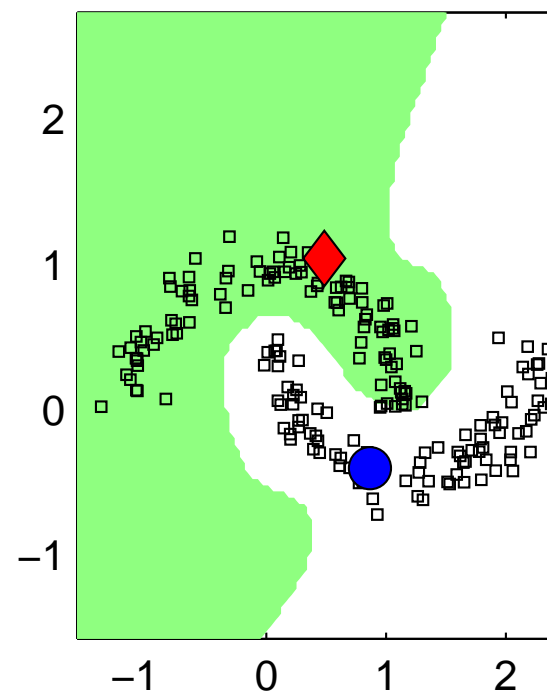
$$\gamma_A = 0.03125 \quad \gamma_I = 0$$

Laplacian SVM



$$\gamma_A = 0.03125 \quad \gamma_I = 0.01$$

Laplacian SVM



$$\gamma_A = 0.03125 \quad \gamma_I = 1$$

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

Data:  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)$

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  $\| \cdot \|_K$  is **important**.

# Algorithm: RLS/SVM

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

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

Can solve explicitly (via Representer theorem):

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

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

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

# Manifold regularization

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

Labeled data:  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)$

Unlabeled data:  $\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}$

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

fit to data + extrinsic smoothness + intrinsic smoothness

Empirical estimate:

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



Representer theorem (discrete case):

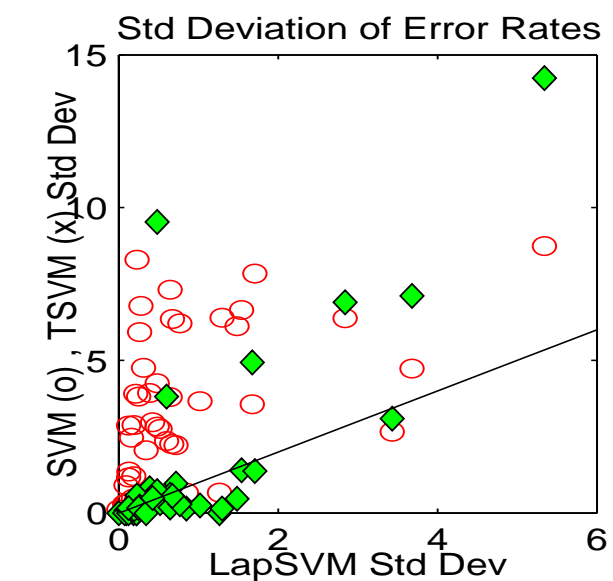
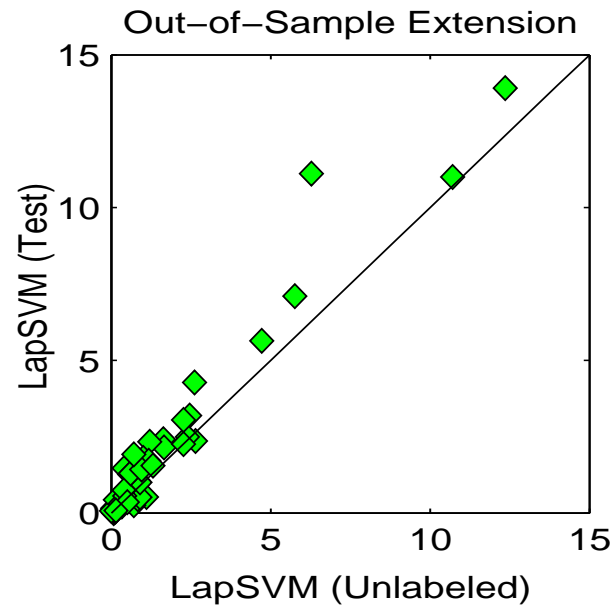
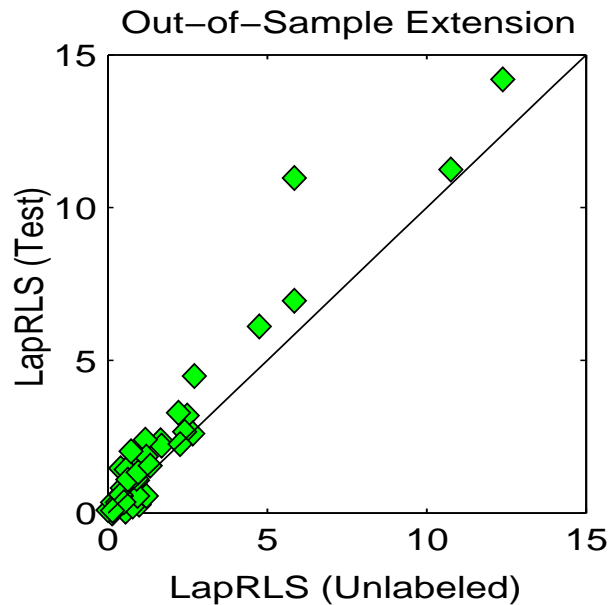
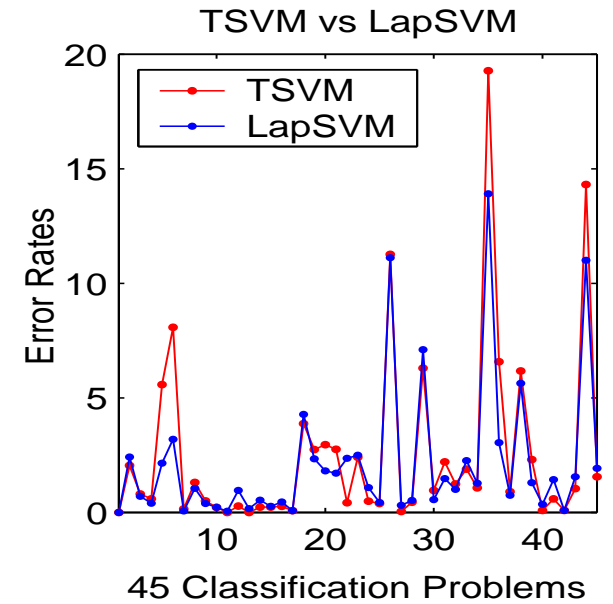
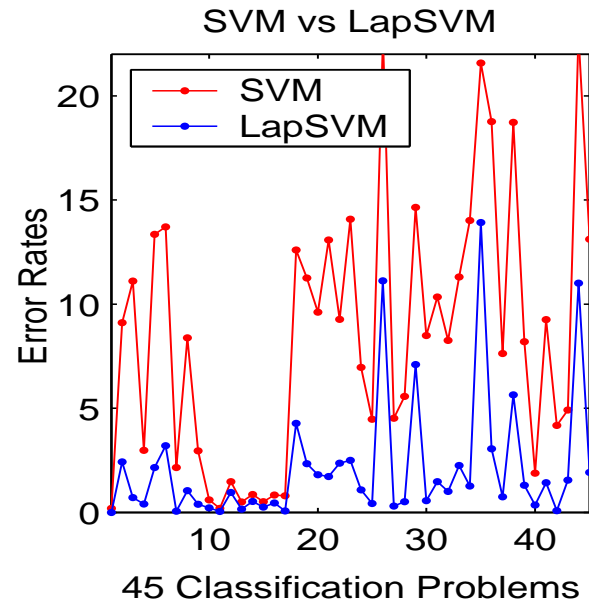
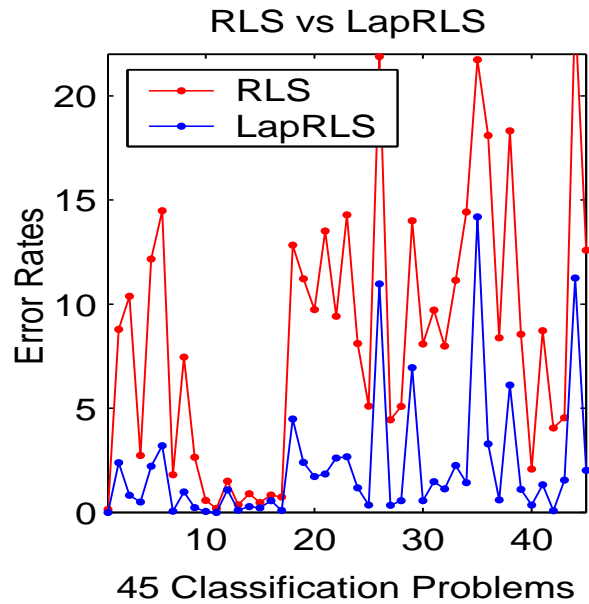
$$f^*(\cdot) = \sum_{i=1}^{l+u} \alpha_i K(\mathbf{x}_i, \cdot)$$

Explicit solution for quadratic loss:

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

$$(\mathbf{K})_{ij} = K(\mathbf{x}_i, \mathbf{x}_j), \quad J = \text{diag}(\underbrace{1, \dots, 1}_l, \underbrace{0, \dots, 0}_u)$$

# 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	-	-	-
<b>LapSVM</b>	<b>5.44</b>	<b>3.66</b>	<b>12.67</b>	<b>10.41</b>	<b>18.1</b>	<b>10.5</b>	<b>6.4</b>

# *Key theoretical question*

---

What is the **connection** between point-cloud Laplacian  $L$  and Laplace-Beltrami operator  $\Delta_{\mathcal{M}}$ ?

Analysis of algorithms:

**Eigenvectors** of  $L$   $\overset{?}{\longleftrightarrow}$  **Eigenfunctions** of  $\Delta_{\mathcal{M}}$

## Theorem [convergence of eigenfunctions]

$$\text{Eig}[L_n^{t_n}] \rightarrow \text{Eig}[\Delta_{\mathcal{M}}]$$

(Convergence in probability)

number of data points  $n \rightarrow \infty$

width of the Gaussian  $t_n \rightarrow 0$

**Previous work. Point-wise convergence.**

Belkin, 03 Belkin, Niyogi 05,06; Lafon Coifman 04,06; Hein Audibert Luxburg, 05; Gine Kolchinskii, 06

**Convergence of eigenfunctions for a fixed  $t$ :**

Kolchinskii Gine 00, Luxburg Belkin Bousquet 04

1. **Geometry** controls many aspects of inference.

1. **Geometry** controls many aspects of inference.
2. Our methods should adapt to geometry.  
**Graph**-based representation of data is good at that.

1. **Geometry** controls many aspects of inference.
2. Our methods should adapt to geometry.  
**Graph**-based representation of data is good at that.
3. **Laplace operator – graph Laplacian** is a key object for various inferential tasks.