Constructing Laplace Operators from Data

Mikhail Belkin

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

Collaborators: Partha Niyogi, Jian Sun, Yusu Wang, Vikas Sindhwani

Machine Learning – high dimensional data (100-D).

Typical problems: clustering, classification, dimensionality reduction.

Machine Learning – high dimensional data (100-D).

Typical problems: clustering, classification, dimensionality reduction.

Graphics/CG – low-dimensional data (2-D).

Typical problems: identify, match and process surfaces.

Machine learning:

Probability Distribution	——	Data
Manifold	——	Graph

Machine learning:

Probability Distribution	——	Data
Manifold	——	Graph

Graphics/CG:

Underlying Spatial Object	——	Data
2-D Surface	——	Mesh

Plan of the talk

Laplacian is a powerful geometric analyzer.

Laplacian is a powerful geometric analyzer.
(As evidenced by Nobel prizes!)

Laplacian is a powerful geometric analyzer.
(As evidenced by Nobel prizes!)

Want practical algorithms with theorems.

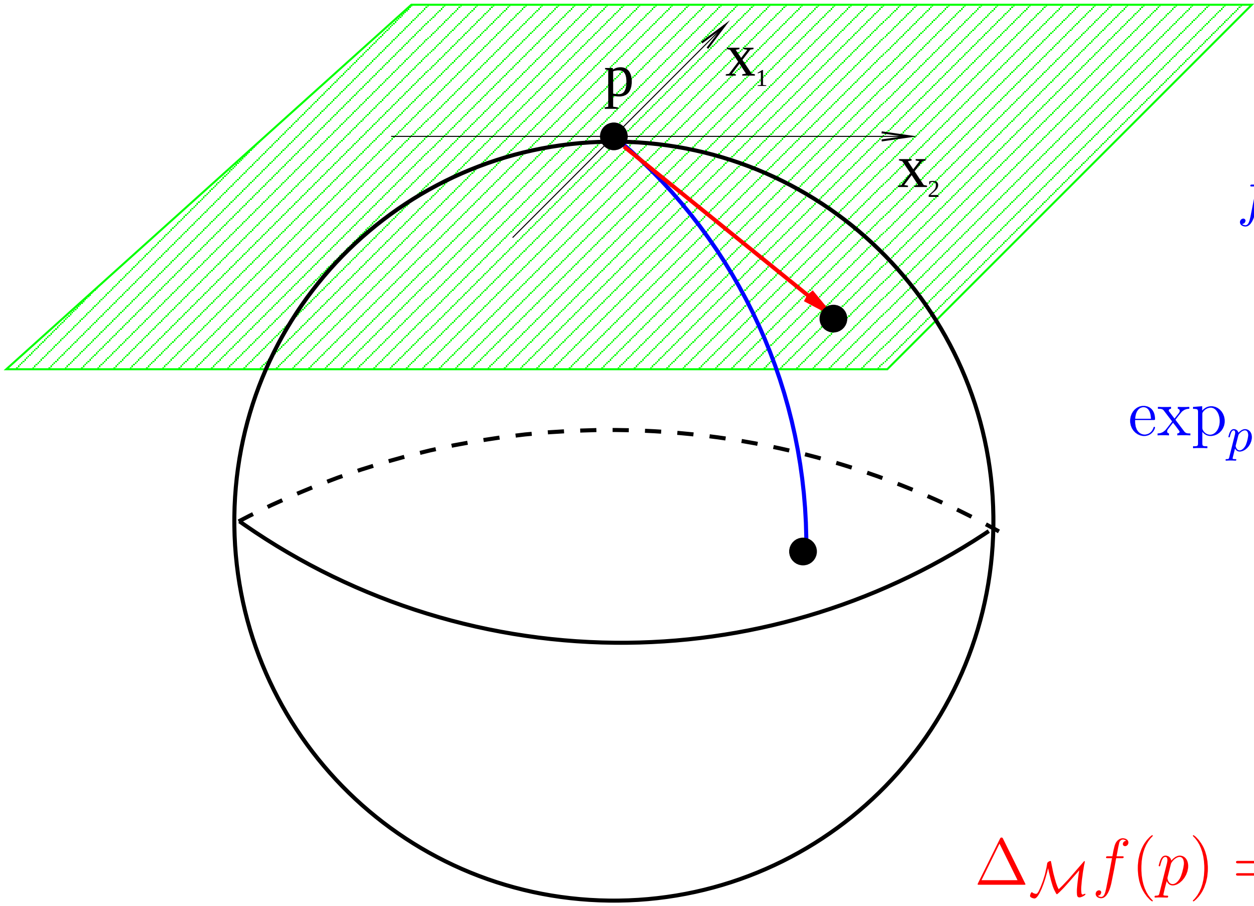
Plan of the talk

Laplacian is a powerful geometric analyzer.
(As evidenced by Nobel prizes!)

Want practical algorithms with theorems.

Will discuss algorithms/theory for ML/CG.

Laplace-Beltrami operator



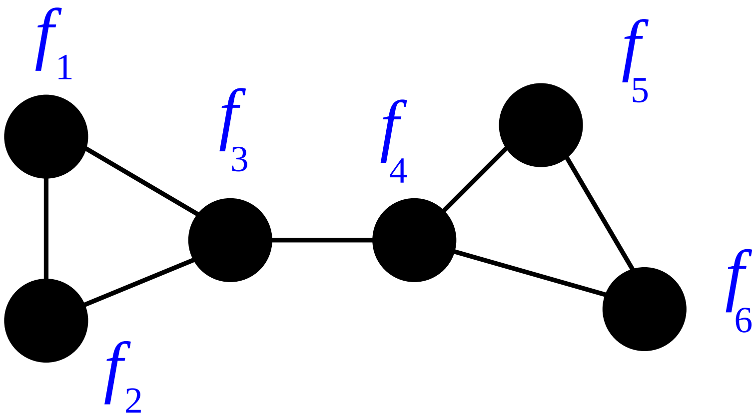
$$f : \mathcal{M}^k \to \mathbb{R}$$

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

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

Generalization of Fourier analysis.

Algorithmic framework: Laplacian



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

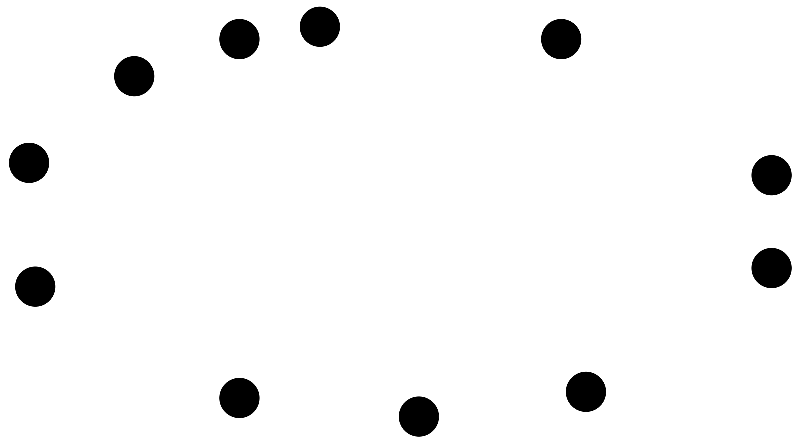
Natural smoothness functional (analogue of grad):

$$\mathcal{S}(\mathbf{f}) = (f_1 - f_2)^2 + (f_1 - f_3)^2 + (f_2 - f_3)^2 + (f_3 - f_4)^2 + (f_4 - f_5)^2 + (f_4 - f_5)^2 + (f_5 - f_6)^2$$

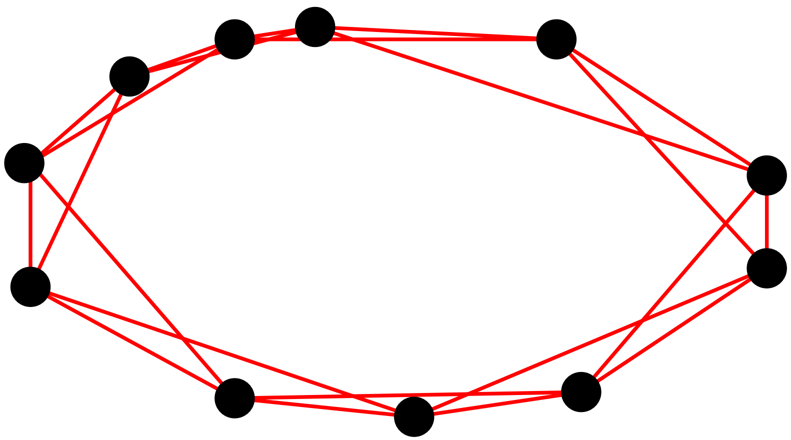
Basic fact:

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

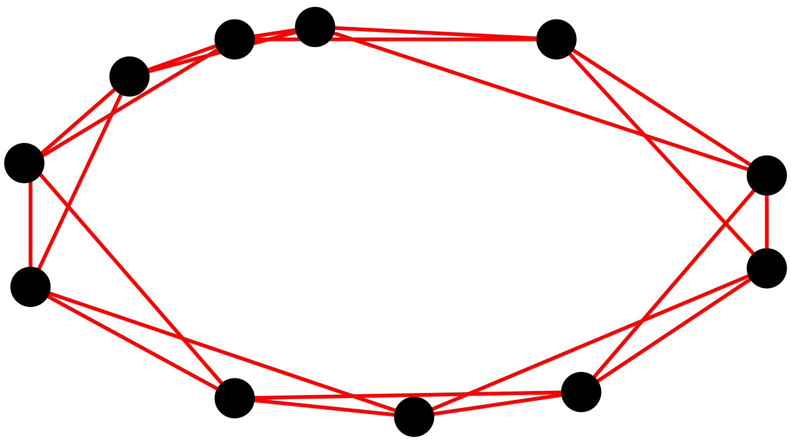
Algorithmic framework

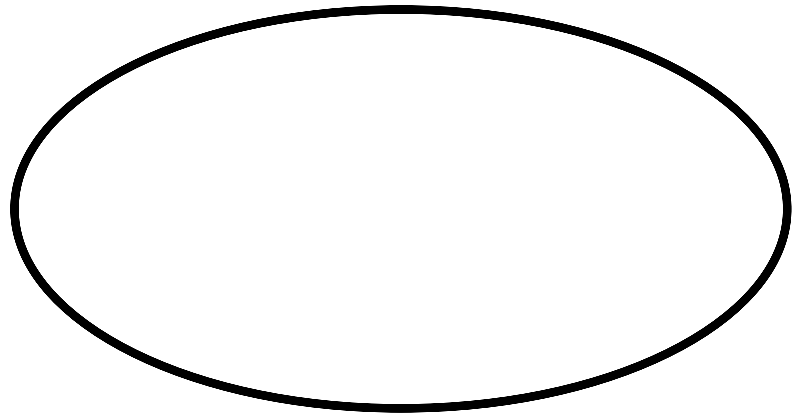


Algorithmic framework



Algorithmic framework

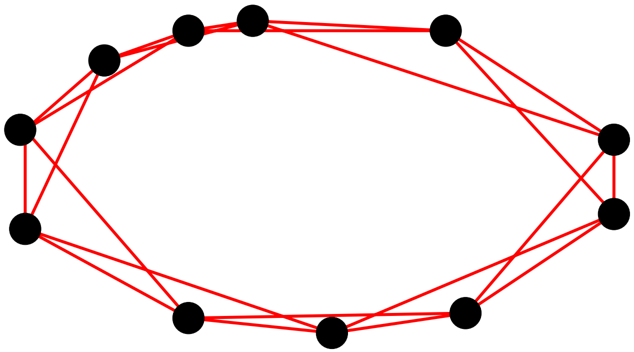




$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

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

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



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

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

Preserve adjacency.

Solution: $Lf = \lambda f$ (slightly better $Lf = \lambda D f$)
Lowest eigenfunctions of L ($\tilde{L}$).

Laplacian Eigenmaps

Belkin Niyogi 01

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

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

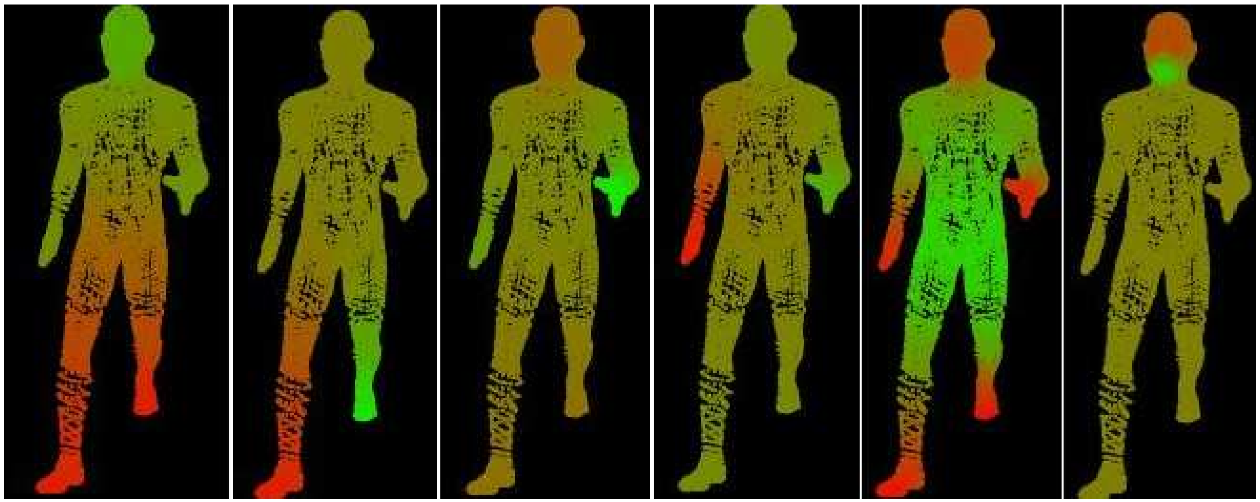
Laplacian Eigenmaps

- 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

Learning from labeled and unlabeled data.

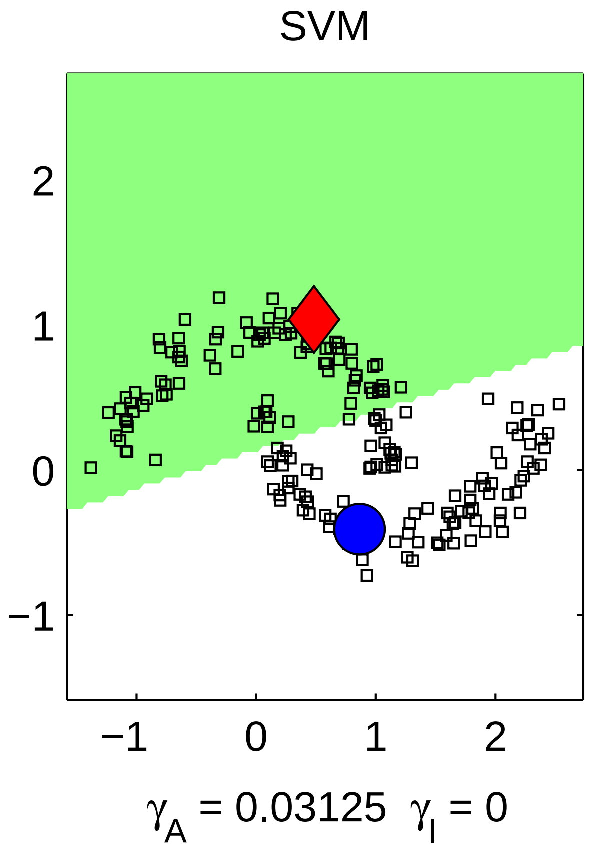
- Unlabeled data is everywhere. Need to use it.
- Natural learning is semi-supervised.

Learning from labeled and unlabeled data.

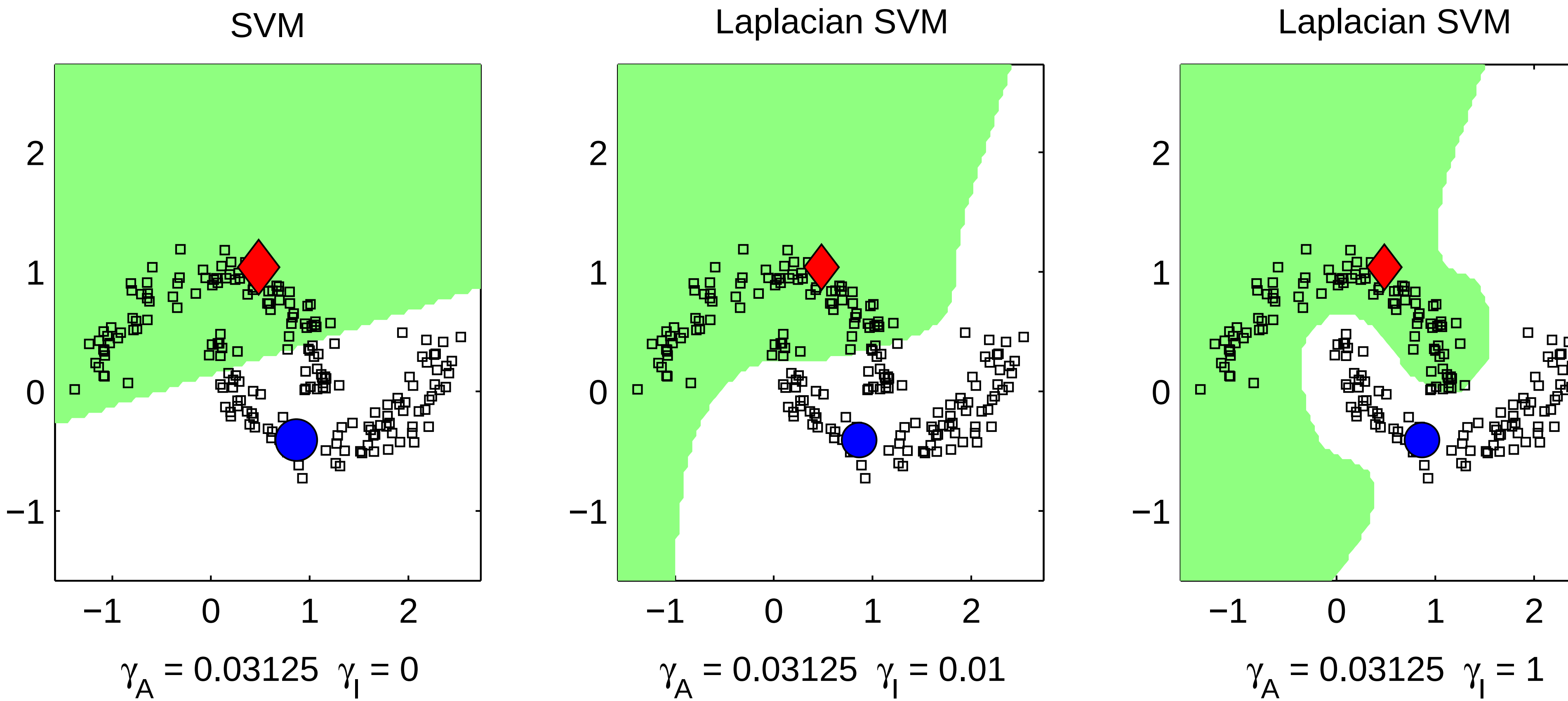
- Unlabeled data is everywhere. Need to use it.
- Natural learning is semi-supervised.

Idea:
construct the Laplace operator using unlabeled data.

Fit eigenfunctions using labeled data.


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

Toy example



SVM
Laplacian SVM
Laplacian SVM
2
1
0
−1
−1
0
1
2
γ_A = 0.03125 γ_I = 0
γ_A = 0.03125 γ_I = 0.01
γ_A = 0.03125 γ_I = 1

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**

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 $\quad \overset{?}{\longleftrightarrow} \quad$ Eigenfunctions of $\Delta_{\mathcal{M}}$

Theorem [convergence of eigenfunctions]

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

(Convergence in probability)

number of data points $n \to \infty$
width of the Gaussian $t_n \to 0$

Previous work. Point-wise convergence.

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

Convergence of eigenfunctions for a fixed$t$:

Kolchniskii Gine 00, Luxburg Belkin Bousquet 04

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 $t \rightarrow 0$ and writing the derivative:

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

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

Proof idea (pointwise convergence)

Functional approximation:

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

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

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

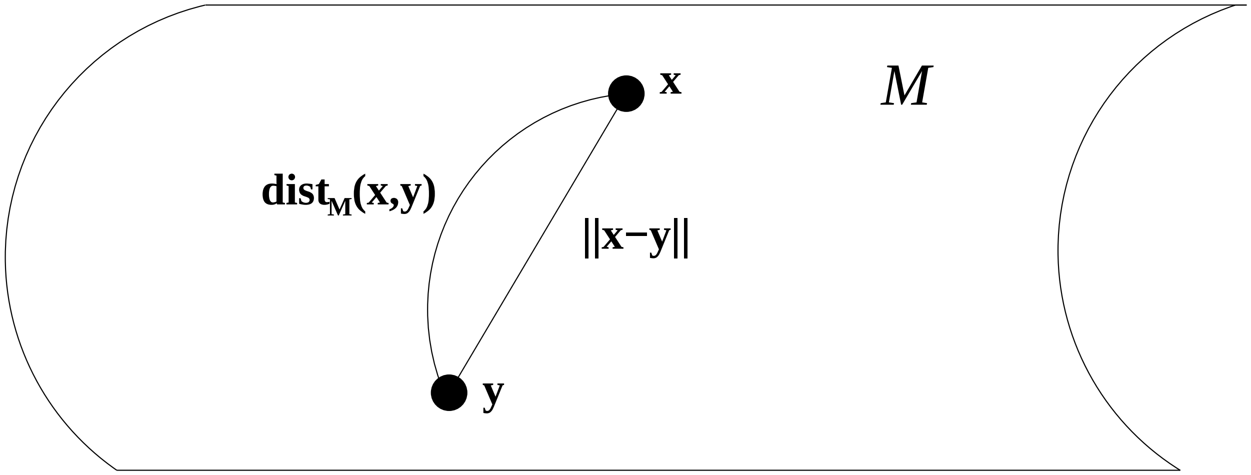
Empirical approximation:

Integral can be estimated from empirical data.

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

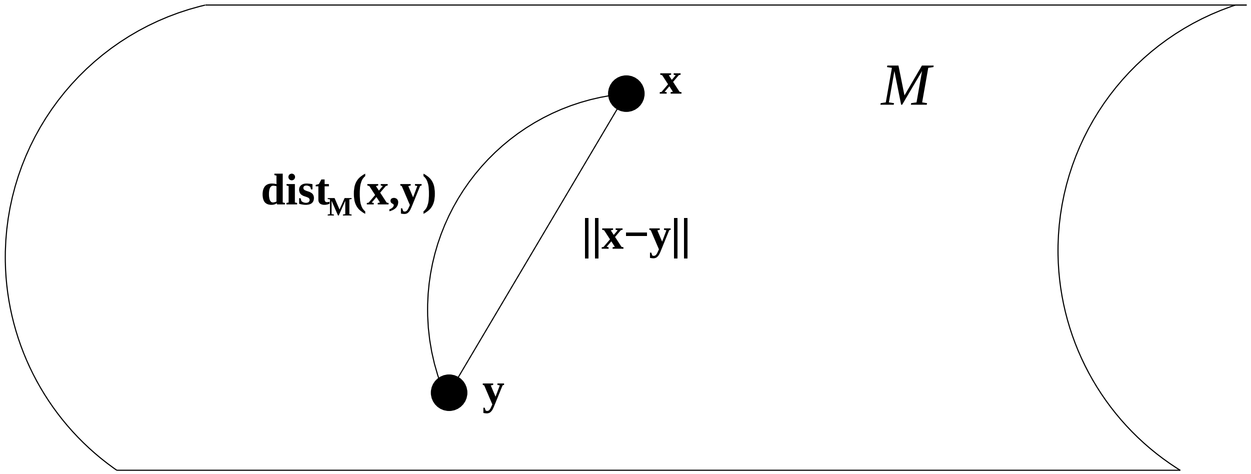
Some difficulties arise for manifolds:

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

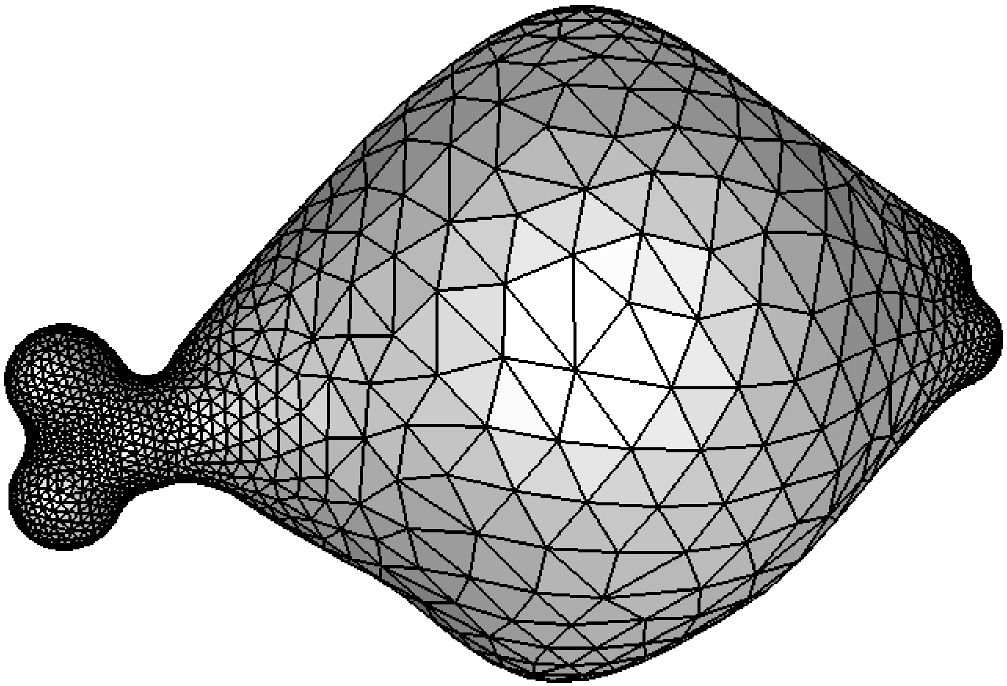


Some difficulties arise for manifolds:

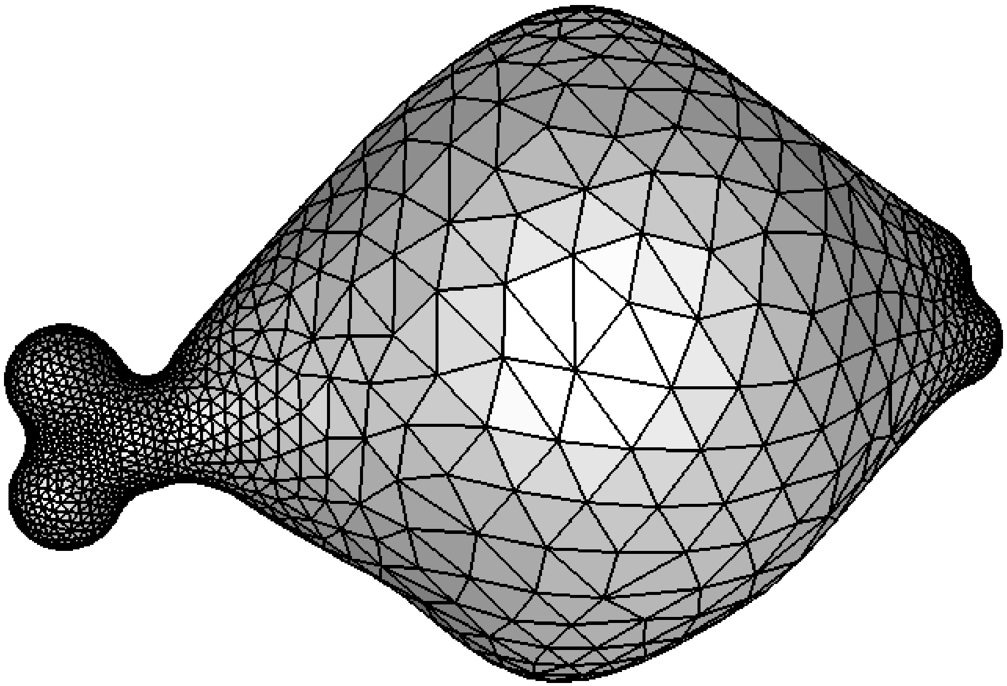
- Do not know distances.
- Do not know the heat kernel.



Careful analysis required.



Mesh Laplacian



Mesh K.

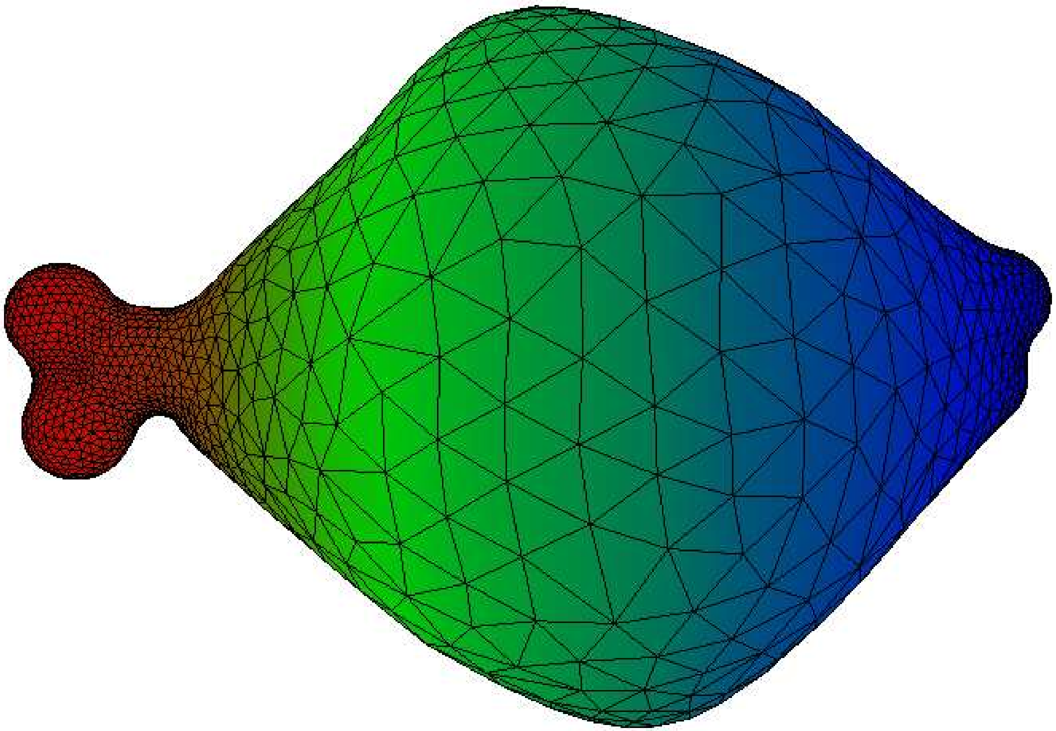
Triangle t. Area $A(t)$.

Vertices v, w.

$$L_K f(w) = \frac{1}{4\pi t^2} \sum_{t \in K} \frac{A(t)}{3} \sum_{v \in t} e^{-\frac{\|p-w\|^2}{4t}} (f(v) - f($$

Belkin Sun Wang, 07

Mesh Laplacian



Mesh K.

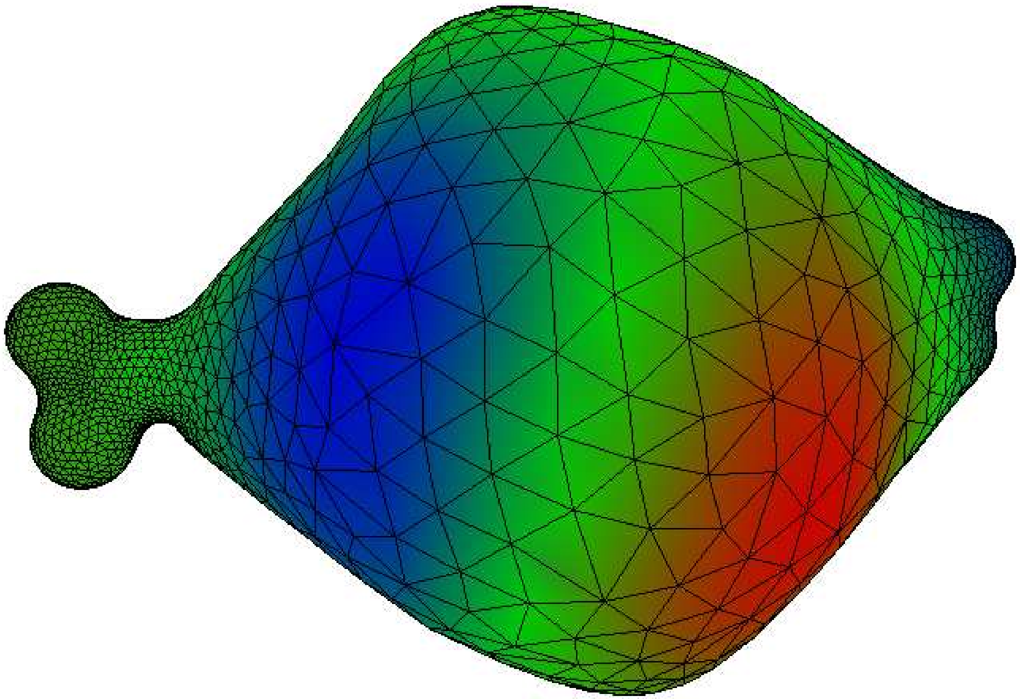
Triangle t. Area $A(t)$.

Vertices v, w.

$$L_K f(w) = \frac{1}{4\pi t^2} \sum_{t \in K} \frac{A(t)}{3} \sum_{v \in t} e^{-\frac{\|p-w\|^2}{4t}} (f(v) - f($$

Belkin Sun Wang, 07

Mesh Laplacian



Mesh K.

Triangle t. Area $A(t)$.

Vertices v, w.

$$L_K f(w) = \frac{1}{4\pi t^2} \sum_{t \in K} \frac{A(t)}{3} \sum_{v \in t} e^{-\frac{\|p-w\|^2}{4t}} (f(v) - f($$

Belkin Sun Wang, 07

K is a "nice" mesh (does not fold onto itself).

Theorem:

$$L_K f \to \Delta_{\mathcal{M}} f$$

as mesh size ϵ (biggest triangle) tends to zero.

$t = \epsilon^{\frac{1}{2.5+0.001}}$.

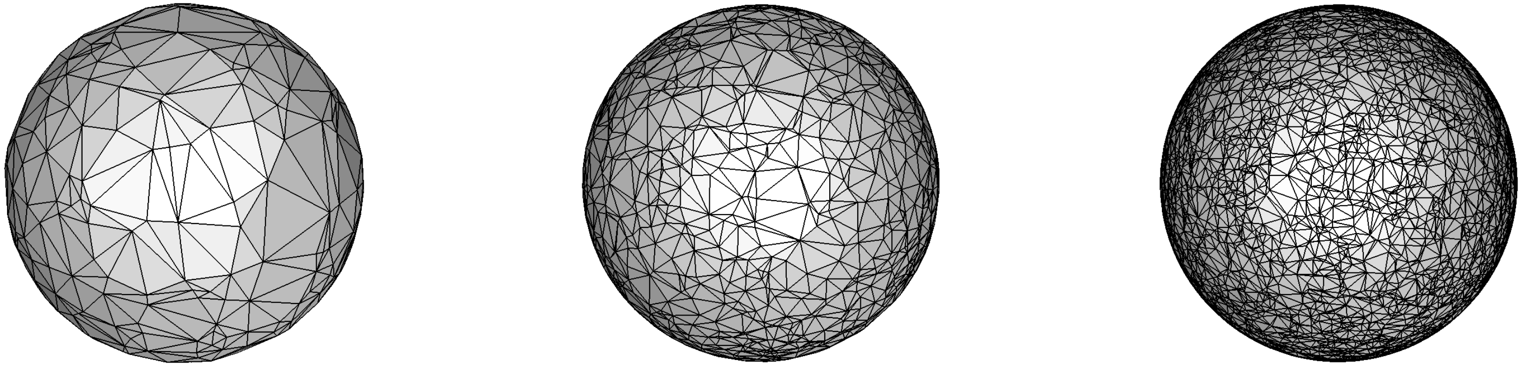
K is a “nice” mesh (does not fold onto itself).

Theorem:

$$L_K f \to \Delta_{\mathcal{M}} f$$

as mesh size ϵ (biggest triangle) tends to zero.

$t = \epsilon^{\frac{1}{2.5+0.001}}$.



	500 pts	2000 pts	8000 pts
Eig #1	0.004	0.003	0.0009
Eig #7	0.020	0.008	0.007
Eig #50	0.112	0.029	0.010

Existing work: several methods, including Desbrun, et al 99, Meyer, et al 02, Xu 04. Numerous applications (smoothing, quadrangulations, deformations, etc).

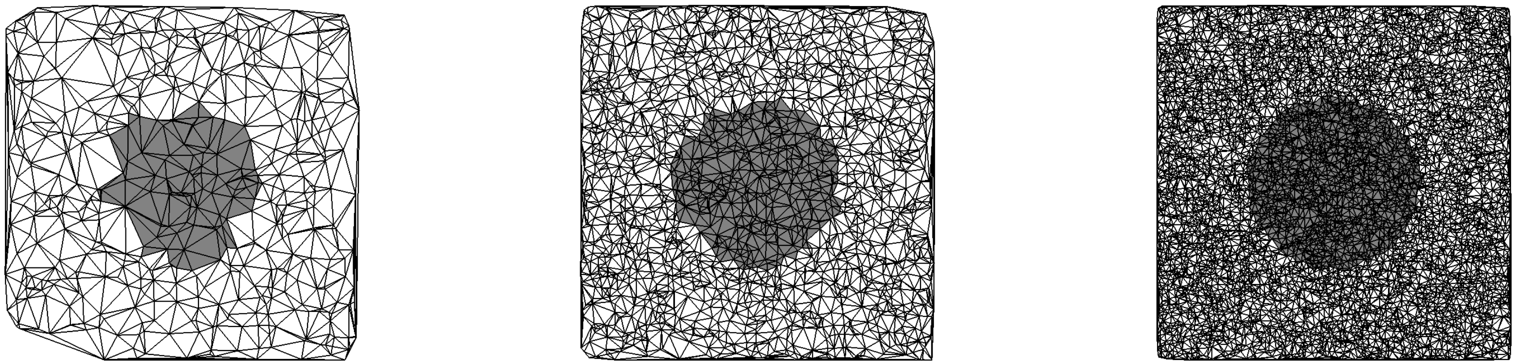
Existing work: several methods, including Desbrun, et al 99, Meyer, et al 02, Xu 04. Numerous applications (smoothing, quadrangulations, deformations, etc).

None converge even in $\mathbb{R}^2$.

Mesh Laplacian

Existing work: several methods, including Desbrun, et al 99, Meyer, et al 02, Xu 04. Numerous applications (smoothing, quadrangulations, deformations, etc).

None converge even in $\mathbb{R}^2$.



$f = x^2$	500 pts	2000 pts	8000 pts
Meyer	0.481	0.256	0.357
Xu	0.220	0.173	0.197
MLap	0.069	0.017	0.004

Conclusions

- Laplacian carries extensive geometric information.

- Laplacian carries extensive geometric information.
- Theoretically grounded and practical methods.

Conclusions

- Laplacian carries extensive geometric information.
- Theoretically grounded and practical methods.
  (Yes, it is possible!)

Conclusions

- Laplacian carries extensive geometric information.
- Theoretically grounded and practical methods.
(Yes, it is possible!)
- Interesting connections between machine learning and graphics.

Conclusions

- Laplacian carries extensive geometric information.
- Theoretically grounded and practical methods.
  (Yes, it is possible!)
- Interesting connections between machine learning and graphics.
- More things should be possible now.