

# Diffusion Maps

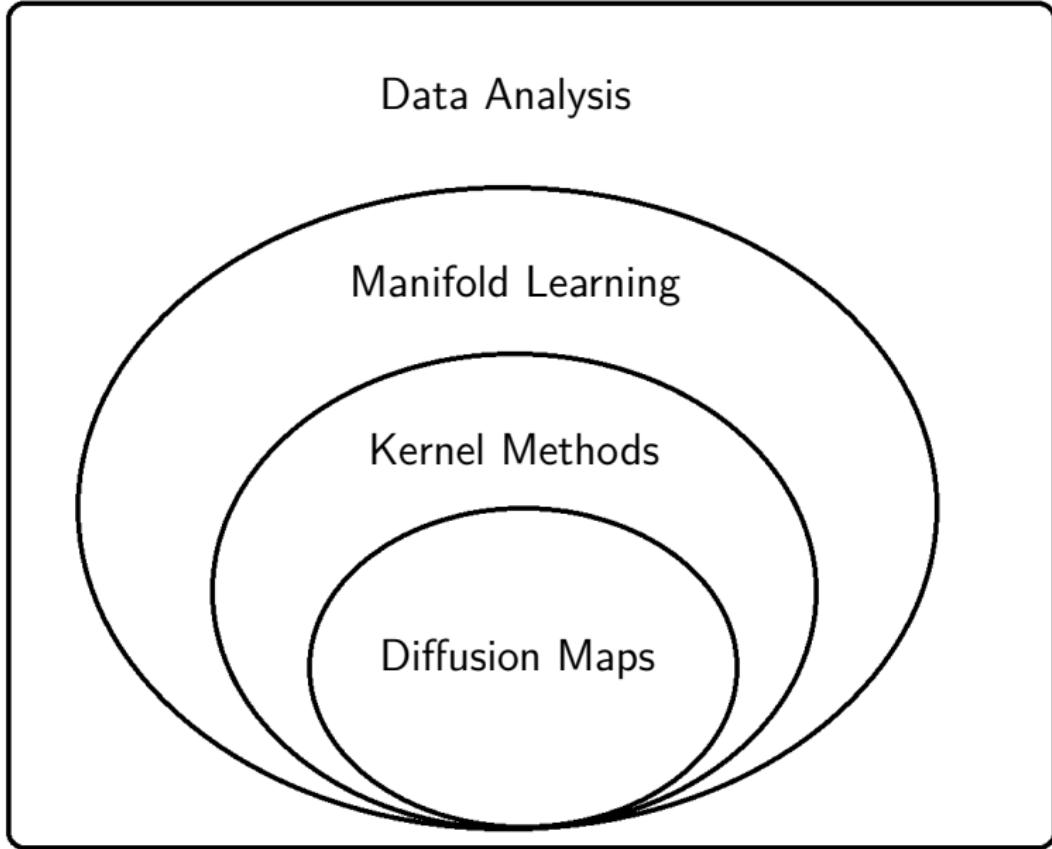
Aviv Rotbart

Tel Aviv University

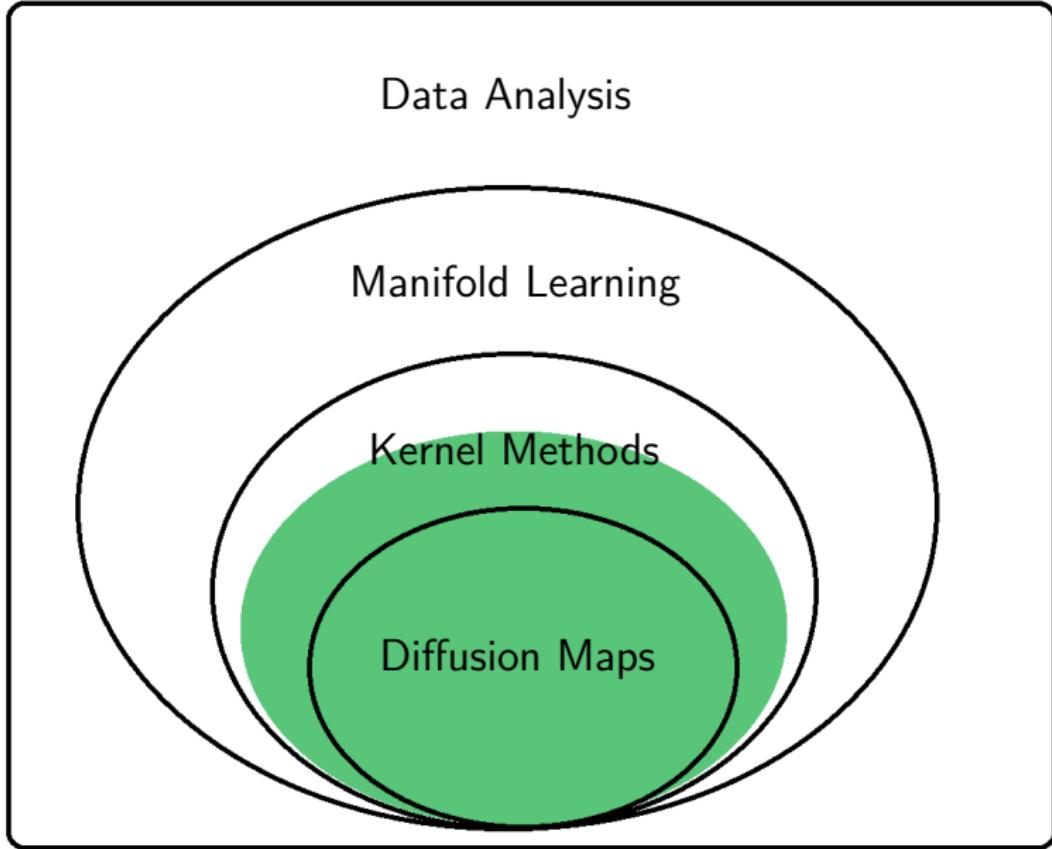
December 2012



# Data Analysis Scope



# Data Analysis Scope

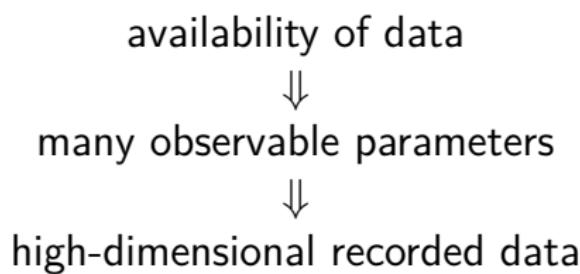


# Outline

## 1 Data Analysis Background

# Manifold learning

Why use manifolds?

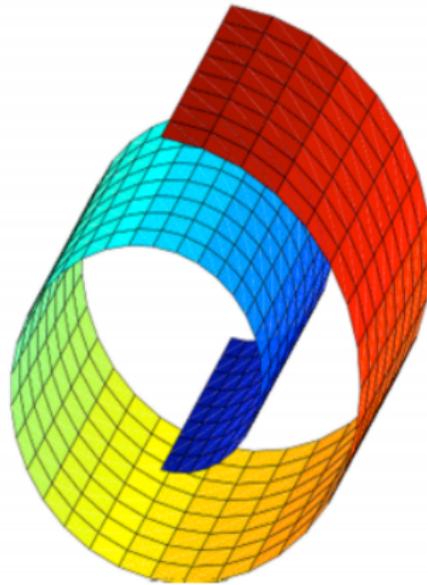


# parameters =  $O(10)$ ,  $O(100)$ , ...

- Data contain dependencies & redundancies
- Observable space = non-linear mapping of few underlying factors
- Underlying locally low-dimensional in high-dimensional ambient space

# Manifold learning

The goal



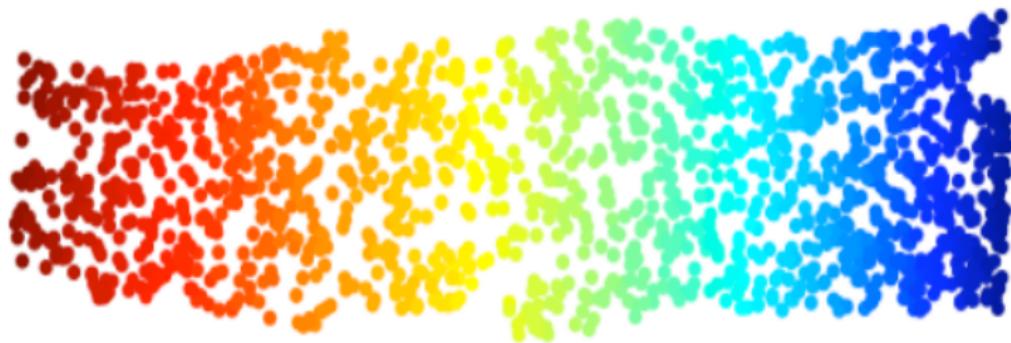
# Manifold learning

The goal



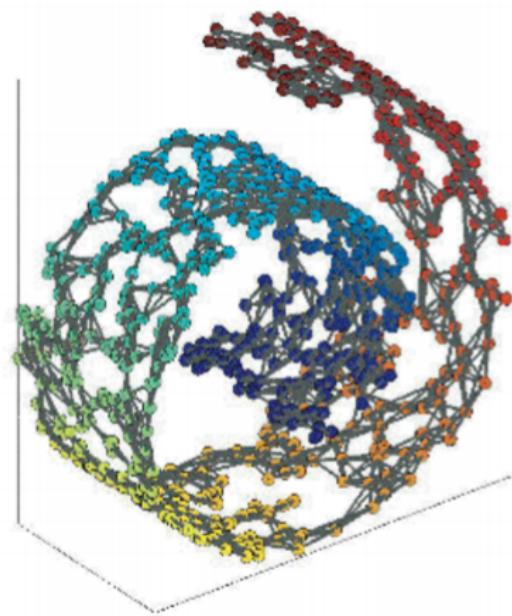
# Manifold learning

The goal



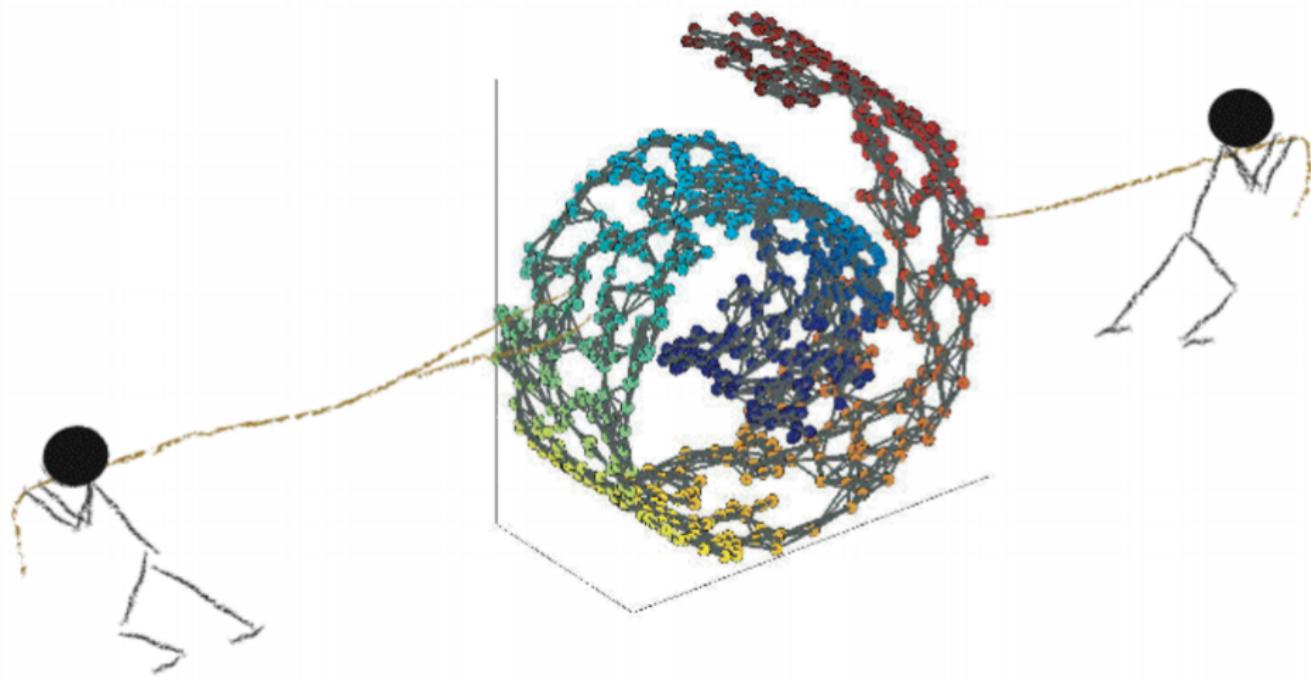
# Manifold learning

## Kernel methods



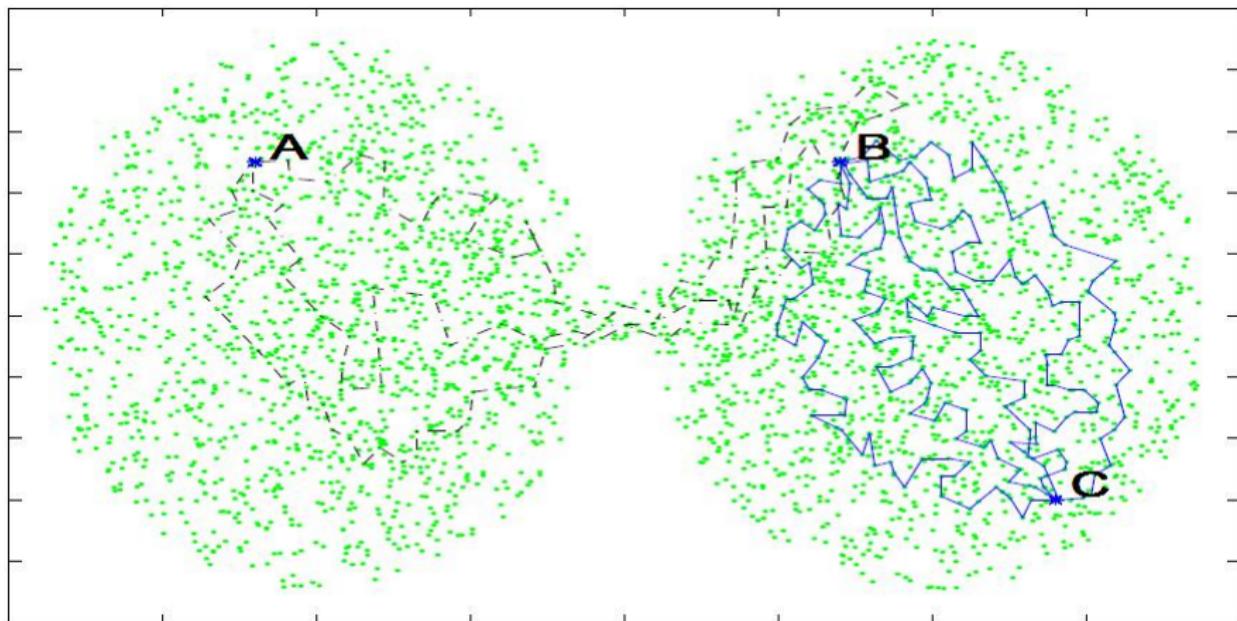
# Manifold learning

## Kernel methods



# Manifold learning

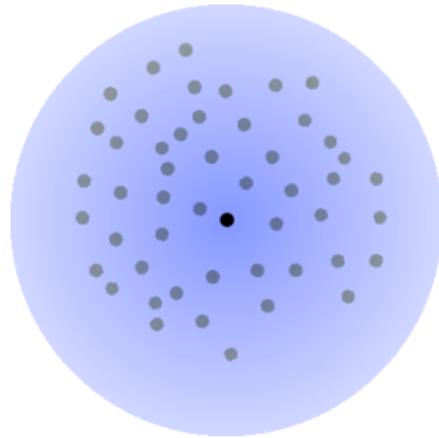
## Diffusion distances



# Diffusion maps<sup>1</sup>

Diffusion process & affinities

- Gaussian kernel:  
 $k(x, y) \triangleq e^{-\frac{\|x-y\|}{\varepsilon}}$



- Degrees:  $q(x) \triangleq \sum k(x, y)$
- Transition probabilities:

$$p(x, y) \triangleq \frac{k(x, y)}{q(x)}$$

- Diffusion affinities:

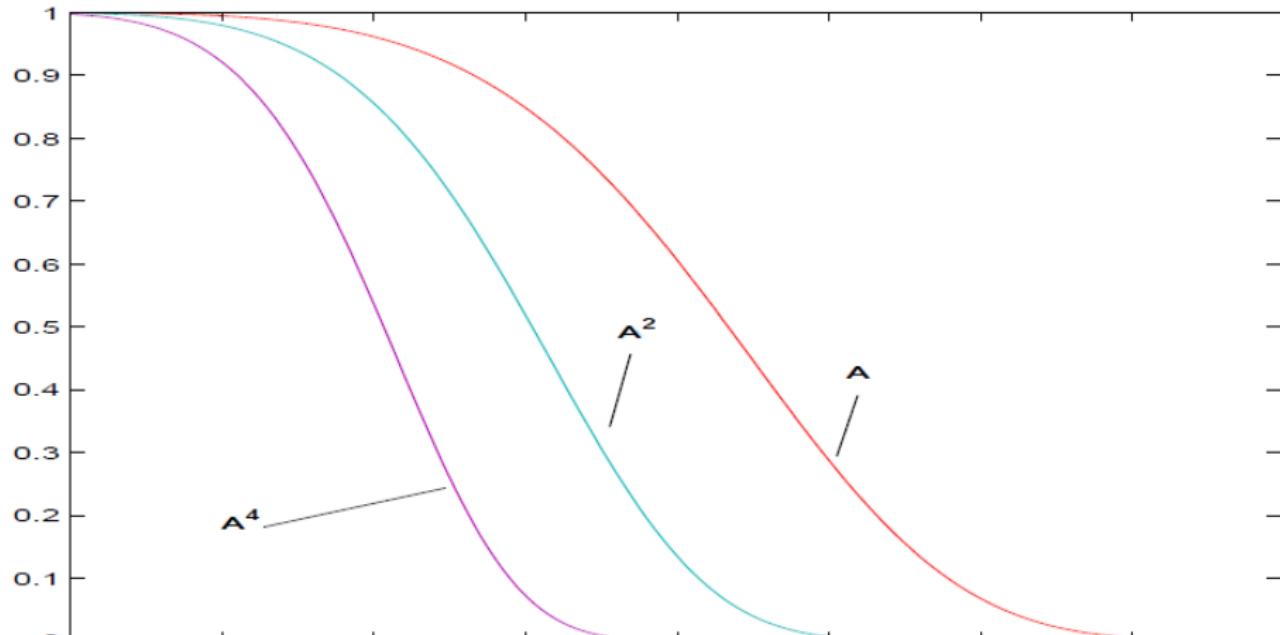
$$\begin{aligned} a(x, y) &\triangleq \frac{k(x, y)}{\sqrt{q(x)} \sqrt{q(y)}} \\ &= q^{1/2}(x) p(x, y) q^{-1/2}(y) \end{aligned}$$

---

<sup>1</sup>R.R. Coifman and S. Lafon. "Diffusion Maps". In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 5–30.

# Diffusion maps

## Spectral embedding



Spectrum (eigenvalues) of the diffusion affinity  $A$  and its powers

# Diffusion maps

Spectral embedding

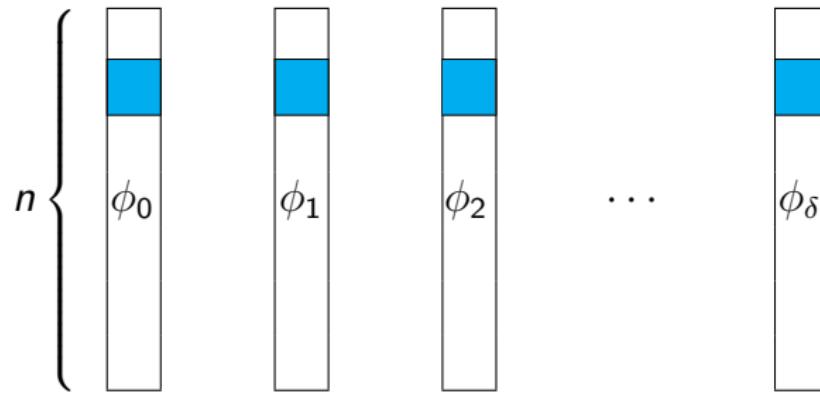
$$1 = \boxed{\lambda_0} \geq \boxed{\lambda_1} \geq \boxed{\lambda_2} \geq \dots \geq \boxed{\lambda_\delta} > 0$$

$$n \left\{ \begin{array}{ccccccc} \phi_0 & & \phi_1 & & \phi_2 & & \dots \\ | & & | & & | & & | \\ \hline \end{array} \right. \quad \begin{array}{c} \phi_\delta \end{array}$$

# Diffusion maps

## Spectral embedding

$$1 = \boxed{\lambda_0} \geq \boxed{\lambda_1} \geq \boxed{\lambda_2} \geq \dots \geq \boxed{\lambda_\delta} > 0$$

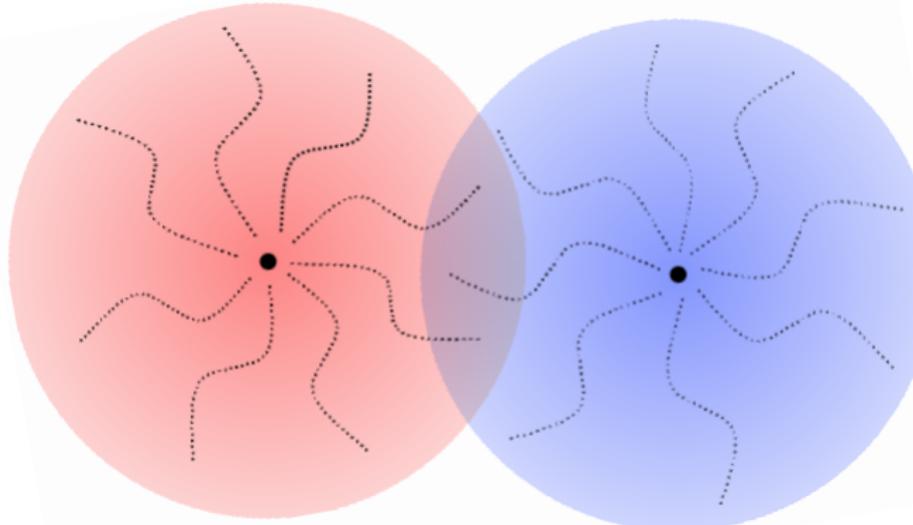


$$x \mapsto \Phi(x) \triangleq [\lambda_0\phi_0(x), \lambda_1\phi_1(x), \lambda_2\phi_2(x), \dots, \lambda_\delta\phi_\delta(x)]^T$$

# Diffusion maps

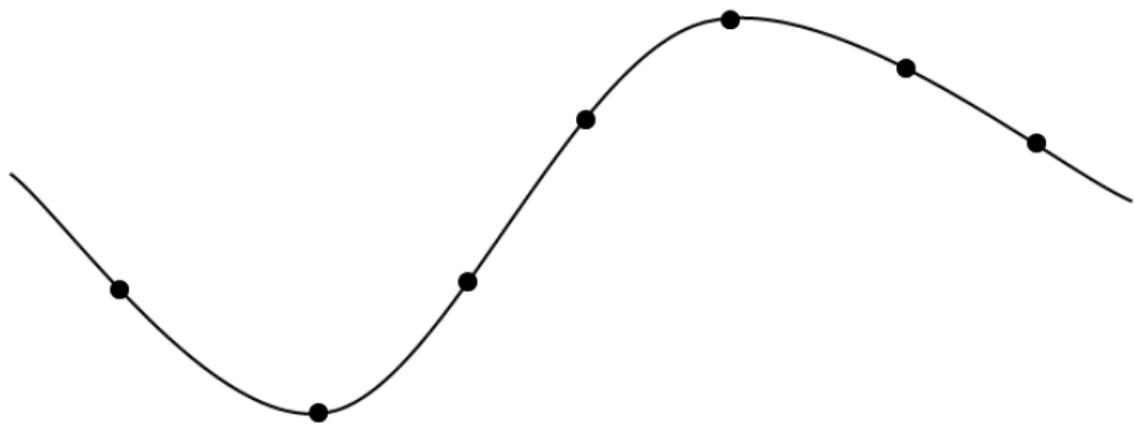
## Embedded distances

$$\overbrace{\|\Phi(x) - \Phi(y)\|}^{\text{embedded distance}} = \overbrace{\|a(x, \cdot) - a(y, \cdot)\|}^{\text{diffusion distance}}$$



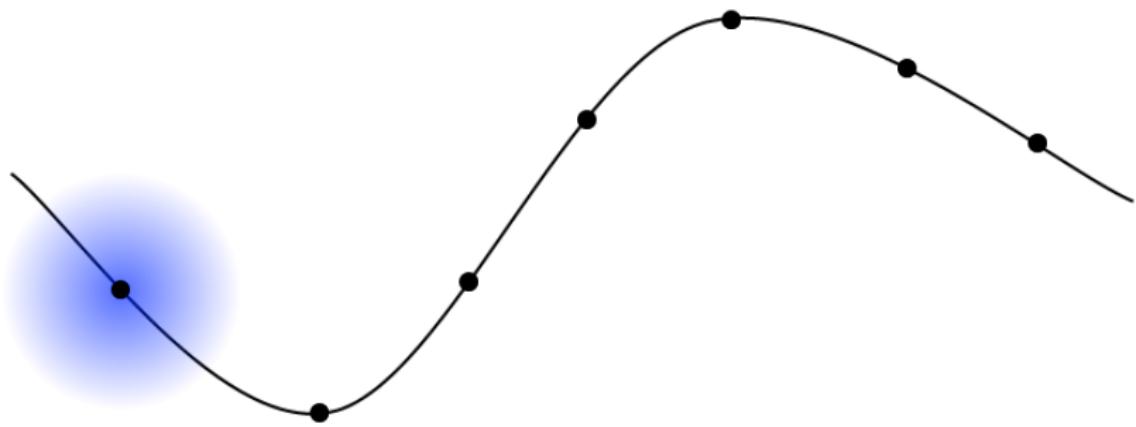
# Gaussian-based diffusion maps

Gaussian kernel:



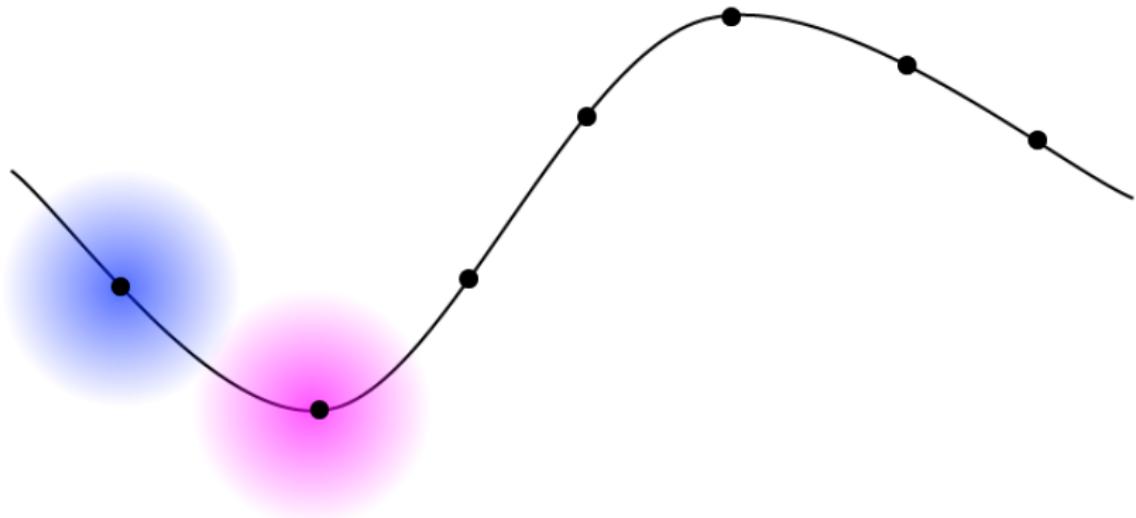
# Gaussian-based diffusion maps

Gaussian kernel:



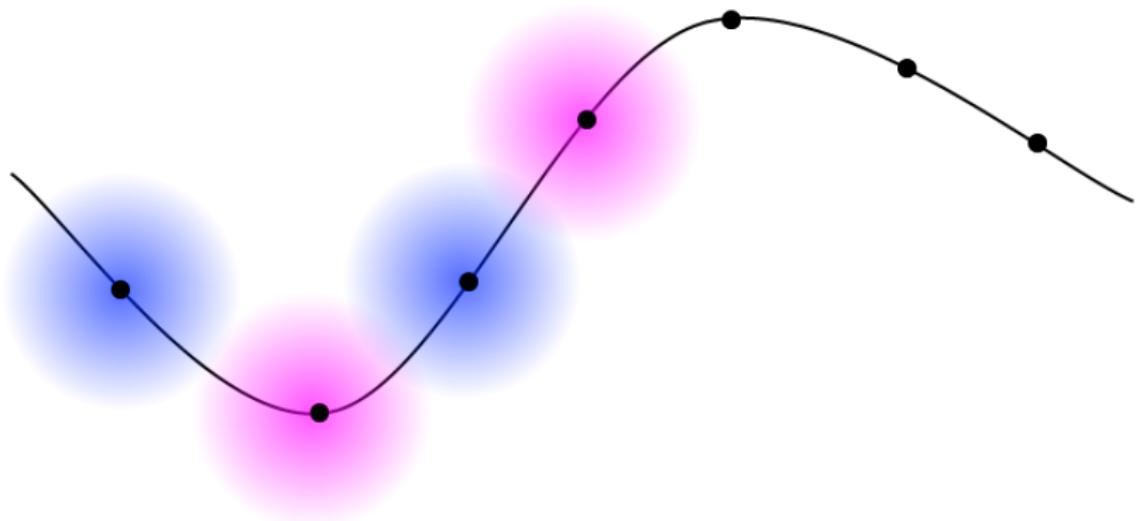
# Gaussian-based diffusion maps

Gaussian kernel:



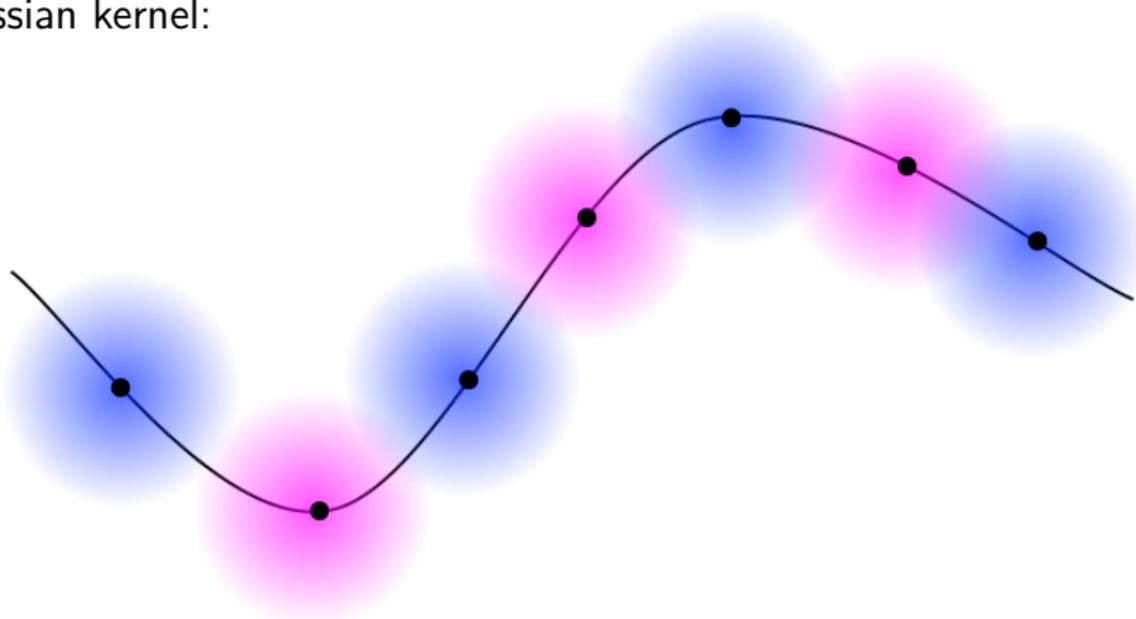
# Gaussian-based diffusion maps

Gaussian kernel:



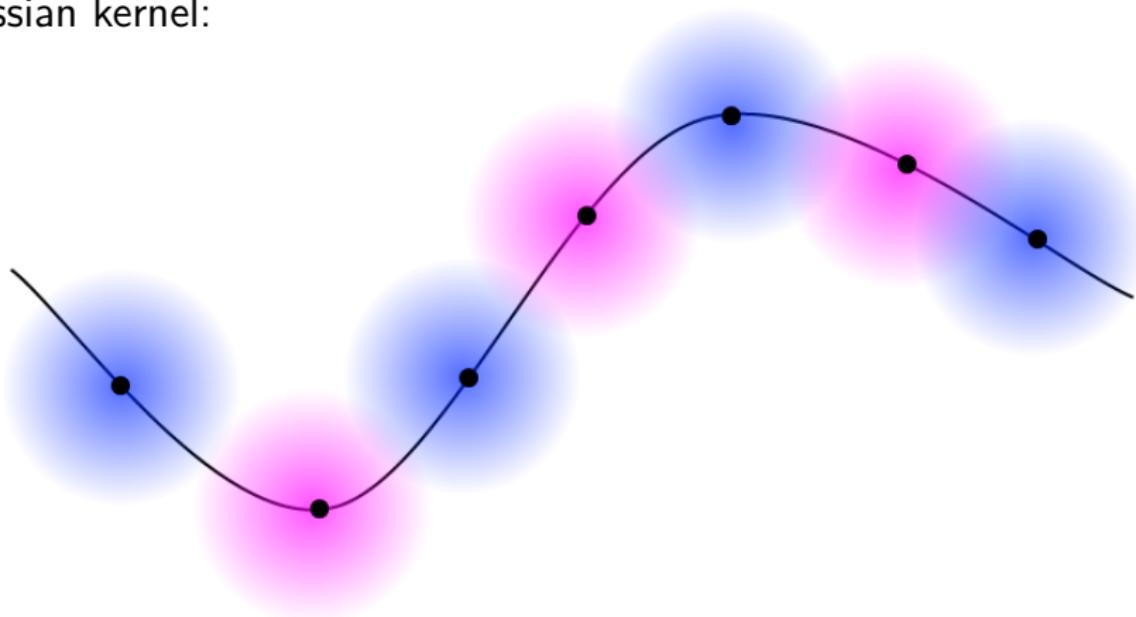
# Gaussian-based diffusion maps

Gaussian kernel:



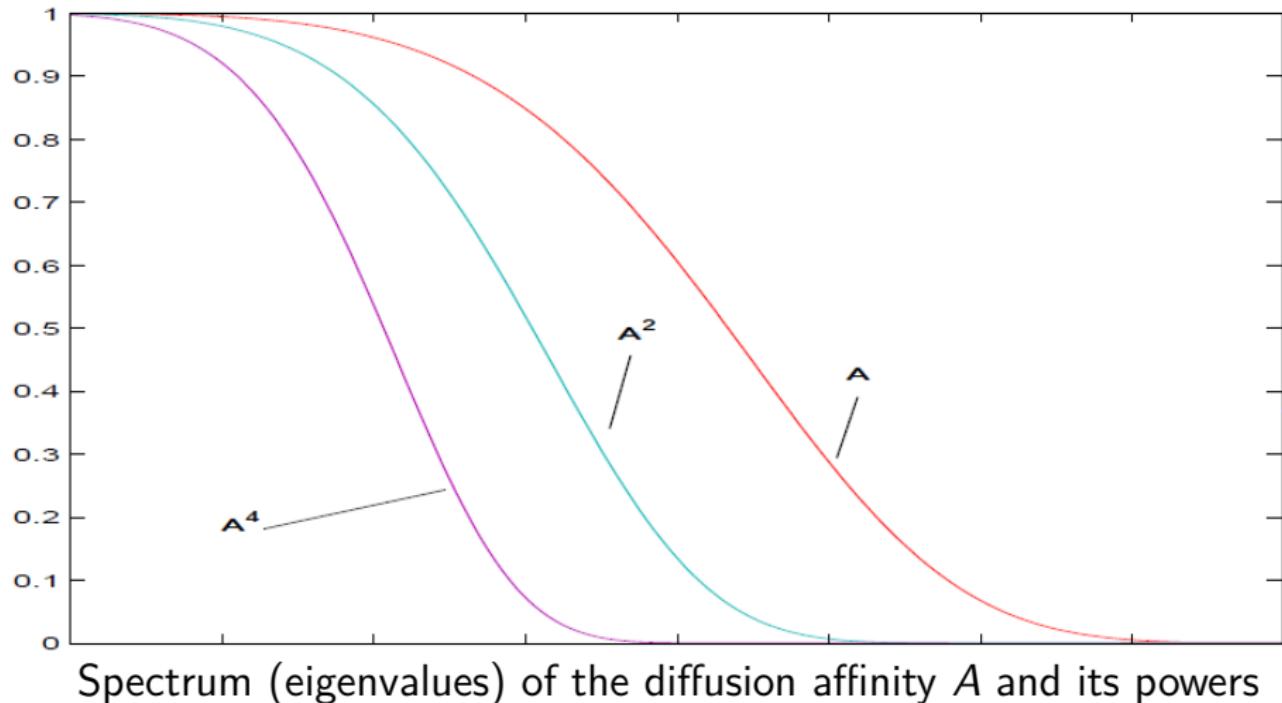
# Gaussian-based diffusion maps

Gaussian kernel:

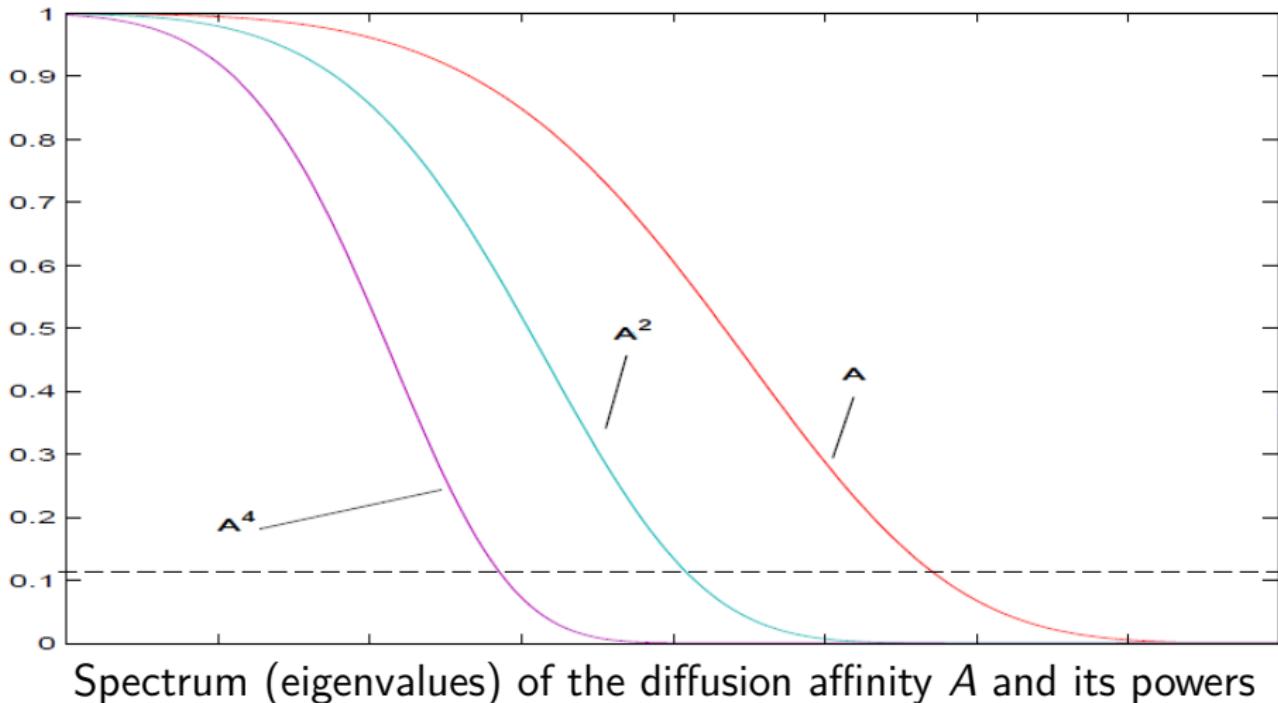


- Normalization  $\implies$  diffusion kernel
- Spectral analysis  $\implies$  map from  $\mathcal{M} \subseteq \mathbb{R}^m$  to  $\mathbb{R}^{\delta \ll m}$

# Numerical rank

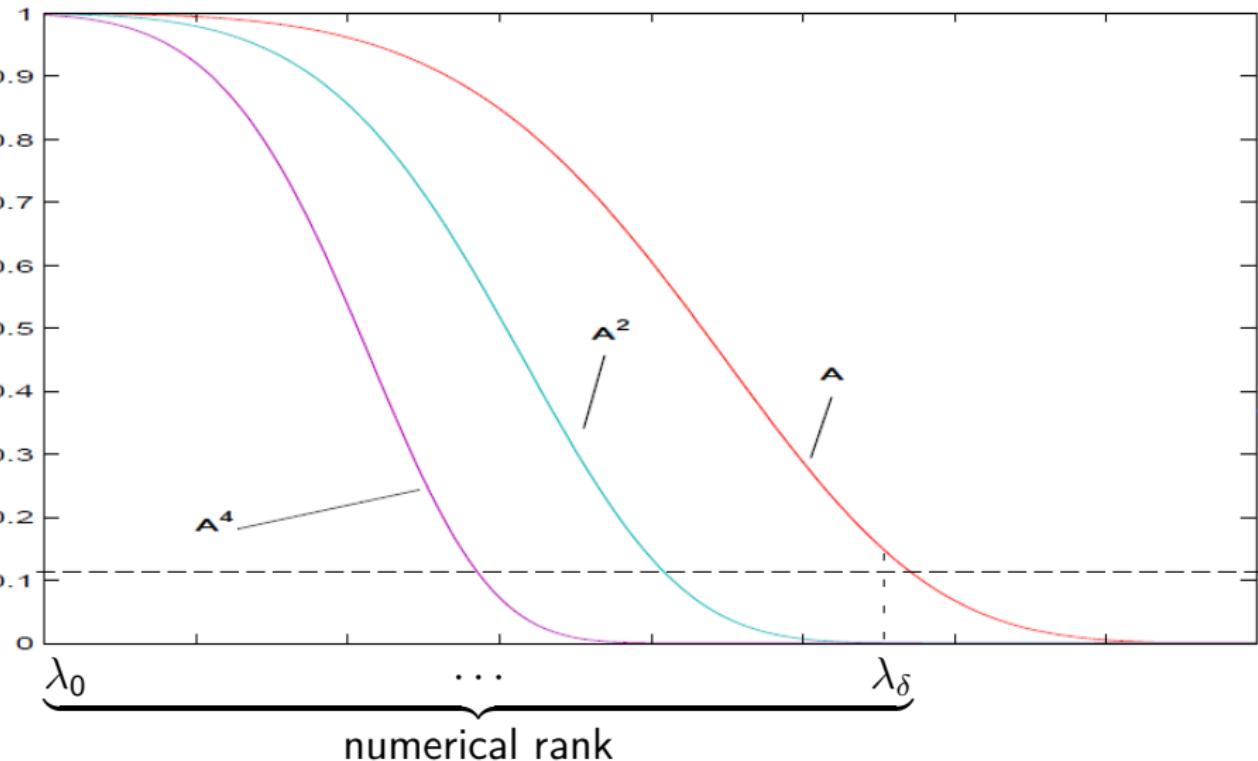


# Numerical rank



Spectrum (eigenvalues) of the diffusion affinity  $A$  and its powers

# Numerical rank



# Typical Scenario: Anomaly Detection



# Typical Scenario: Anomaly Detection



100 system parameters



# Typical Scenario: Anomaly Detection



100 system parameters



Training Phase

# Typical Scenario: Anomaly Detection



100 system parameters



Training Phase



# Typical Scenario: Anomaly Detection



100 system parameters



Training Phase

# Typical Scenario: Anomaly Detection



100 system parameters



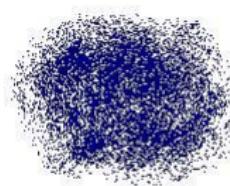
$$\subseteq \mathbb{R}^{100}$$

Training Phase

# Typical Scenario: Anomaly Detection



$$\subseteq \mathbb{R}^{100}$$



$$\subseteq \mathbb{R}^3$$

Dim. reduction

Training Phase

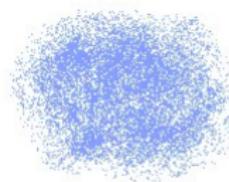
# Typical Scenario: Anomaly Detection



every few minutes



$\in \mathbb{R}^{100}$



Testing Phase

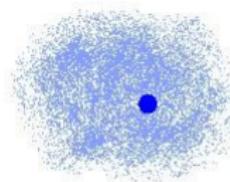


# Typical Scenario: Anomaly Detection



$$\boxed{\quad} \in \mathbb{R}^{100}$$

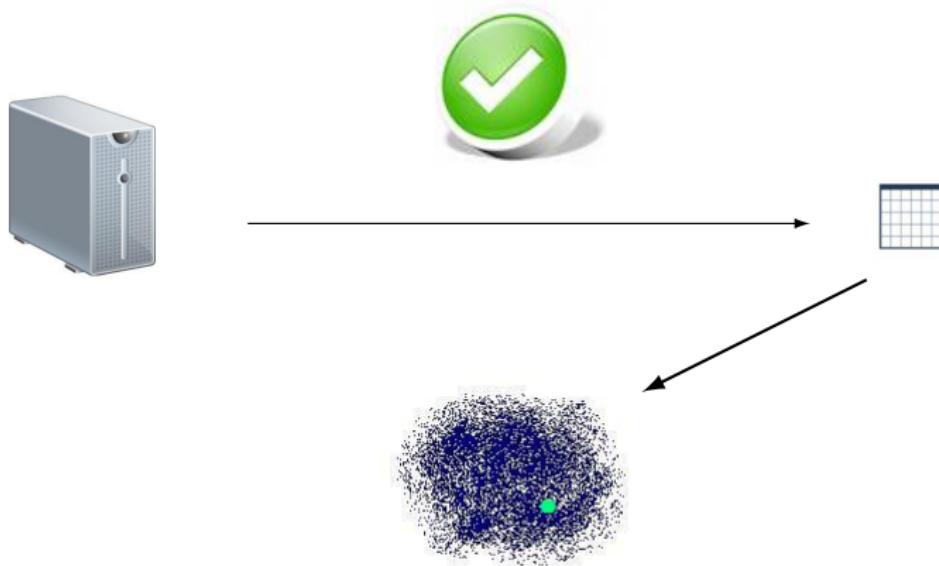
Out-of-sample ext.



$$\in \mathbb{R}^3$$

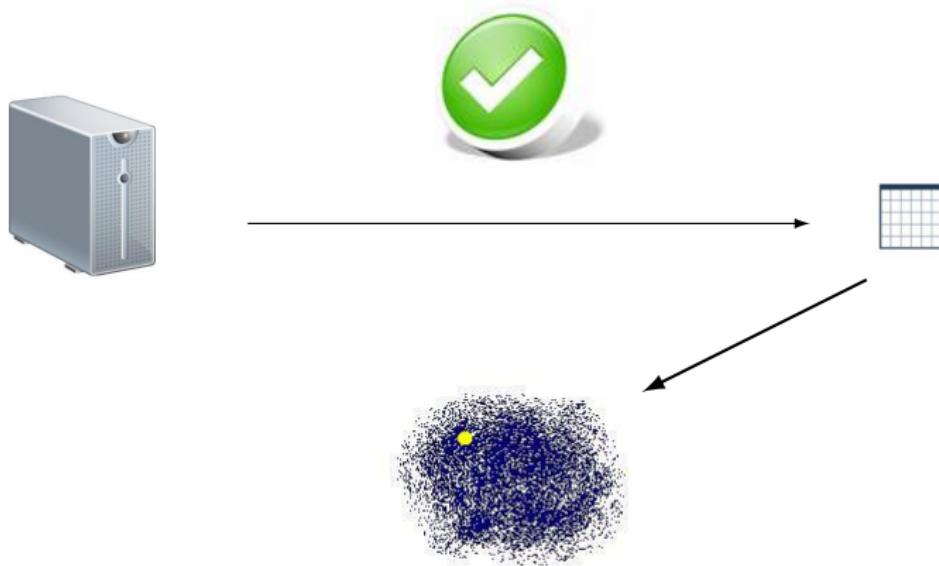
Testing Phase

# Typical Scenario: Anomaly Detection



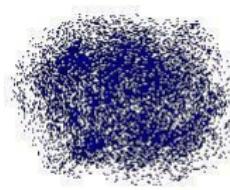
Testing Phase

# Typical Scenario: Anomaly Detection



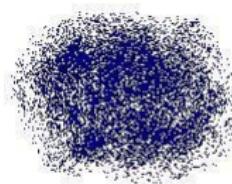
Testing Phase

# Typical Scenario: Anomaly Detection



Testing Phase

# Typical Scenario: Anomaly Detection



Testing Phase

Parameter	Significance
CPU %	High
Process #	High
⋮	⋮
Net I/O	Low
HD I/O	Low

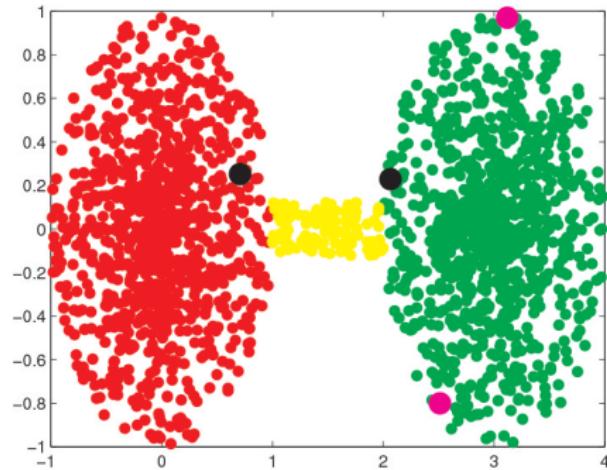
# Diffusion Maps

The Training stage is based on **diffusion maps**.

- Diffusion maps generate efficient representations of complex geometric structures in lower dimensional spaces.
- Use the eigenfunctions of Markov matrices.
- Revels global geometric information from local structures.
- Nonlinear reduction of dimensionality.

# Diffusion Maps - example I

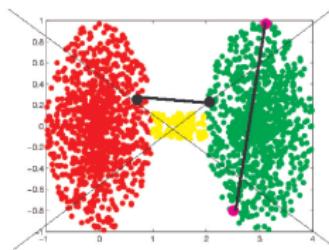
Let's take a look at this dataset and the two pairs of points on it.



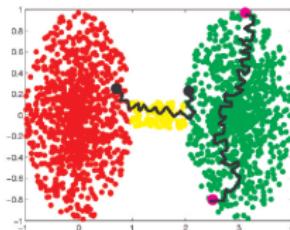
Is it right to measure the distance between the points using euclidian distance?

# Diffusion maps and distance

The distance between two points  $x_i$  and  $x_j$  can be defined as the probability to land in  $x_j$  after  $N$  random steps that start at  $x_i$ .



(a) Euclidean distance



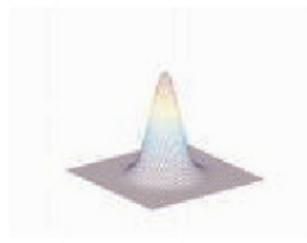
(b) Random walk

This distance measures the connectivity between  $x_i$  and  $x_j$  in the data, while taking into account all possible paths between  $x_i$  and  $x_j$ .

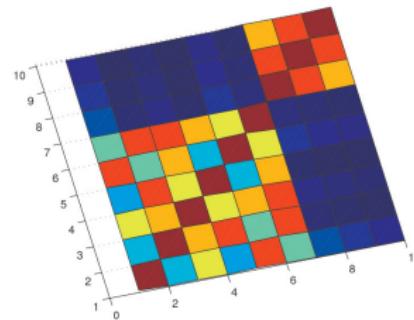
# Diffusion Maps

- Let  $\Gamma = \{x_1, \dots, x_m\}$  be a set of points in  $\mathbb{R}^n$ .
- Construct a non-negative symmetric kernel on the data:

$$w_\epsilon(x, y) = e^{-\frac{\|x-y\|^2}{2\epsilon}}.$$



(c) Gaussian



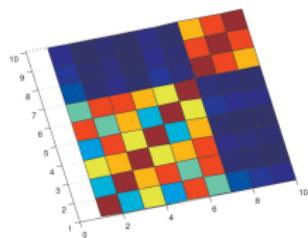
(d) Gaussian Kernel

# Diffusion Maps

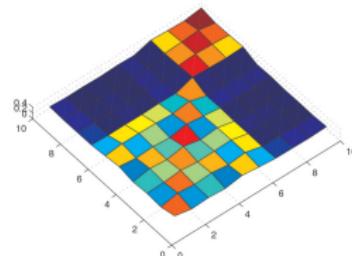
- A general form of a kernel, with a parameter  $\alpha$  that controls the normalization type, given by

$$w_{\varepsilon}^{(\alpha)}(x, y) = \frac{w_{\varepsilon}(x, y)}{q^{\alpha}(x) q^{\alpha}(y)}, \quad q(x) = \sum_{y \in \Gamma} w_{\varepsilon}(x, y) \quad (1)$$

If we set  $\alpha$  to 0.5, the normalized kernel will look like:



(e) Gaussian Kernel  
 $w_{\varepsilon}(x, y)$



(f)  $w_{\varepsilon}^{(0.5)}(x, y)$

# Diffusion Maps

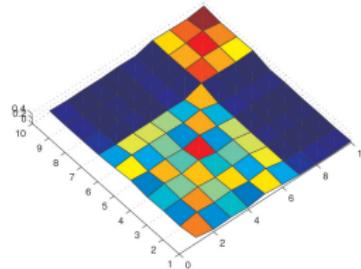
- The transition matrix is defined by

$$p_{\varepsilon}^{\alpha}(x, y) = \frac{w_{\varepsilon}^{(\alpha)}(x, y)}{d_{\varepsilon}^{(\alpha)}(y)} \quad (2)$$

where

$$d_{\varepsilon}^{(\alpha)}(y) = \sum_{x \in \Gamma} w_{\varepsilon}^{(\alpha)}(x, y). \quad (3)$$

The transition matrix  $p_{\varepsilon}^{0.5}(x, y)$  look like this:



# Diffusion Maps

- The asymptotic behavior,  $\varepsilon \rightarrow 0$ , generates different infinitesimal operator for different values of  $\alpha$ :
  - ①  $\alpha = 0$  is the classical normalized graph Laplacian.
  - ②  $\alpha = 1$  approximates the Laplace-Beltrami operator.
  - ③  $\alpha = \frac{1}{2}$  approximates the diffusion of the Fokker-Planck equation.

# Diffusion Maps

- The transition matrix  $P$  is conjugate to a symmetric matrix  $A$

$$a(x, y) = \sqrt{d(x)} p(x, y) \frac{1}{\sqrt{d(y)}}. \quad (4)$$

$$A \triangleq D^{\frac{1}{2}} P D^{-\frac{1}{2}}.$$

- The symmetric matrix  $A$  has the following spectral decomposition:

$$a(x, y) = \sum_{k \geq 0} \lambda_k v_k(x) v_k(y). \quad (5)$$



# Diffusion Maps

- Since  $P$  is conjugate to  $A$

$$\phi_k = D^{\frac{1}{2}} v_k, \quad \psi_k = D^{-\frac{1}{2}} v_k. \quad (6)$$

where  $\{\phi_k\}$  and  $\{\psi_k\}$  are the left and right eigenvectors of  $P$ .

- From the orthonormality of  $\{v_i\}$  and Eq. 6 it follows that  $\{\phi_k\}$  and  $\{\psi_k\}$  are biorthonormal which means  $\langle \phi_l, \psi_k \rangle = \delta_{lk}$ .
- This leads to the following eigendecomposition:

$$p_t(x, y) = \sum_{k \geq 0} \lambda_k^t \psi_k(x) \phi_k(y). \quad (7)$$

- Because of the fast decay of the spectrum, only a few terms are required to achieve sufficient accuracy in the sum.



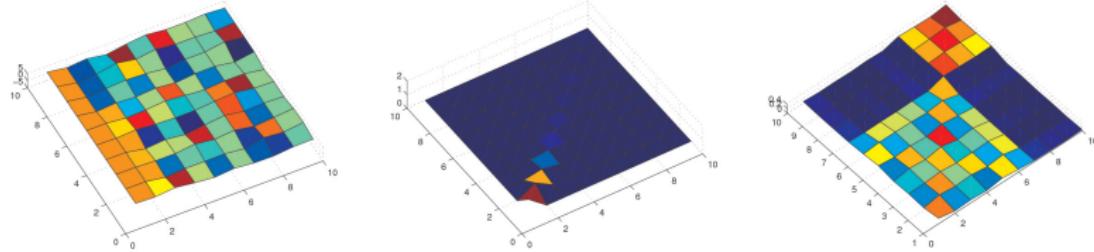
# Diffusion Maps

- The family of diffusion maps  $\{\Psi_t\}$ , is defined by

$$\Psi_t(x) = (\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \lambda_3^t \psi_3(x), \dots), \quad (8)$$

- Diffusion maps embeds the dataset into a Euclidean space.

# Diffusion Maps



**Figure:** Top Left: The eigenvectors:  $\psi_k(x)$ . Top Right: The eigenvalues  $\lambda_k^t$ . Bottom: The transition matrix.

# Diffusion Maps

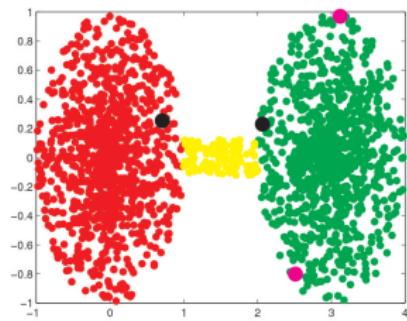
- The diffusion distance between two data points  $x_i$  and  $x_j$  is the weighted  $L^2$  distance

$$D_t^2(x_i, x_j) = \sum_{x_l \in \Gamma} \frac{(p_t(x_i, x_l) - p_t(x_l, x_j))^2}{\phi_0(z)}. \quad (9)$$

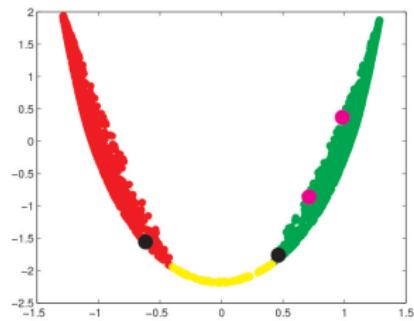
- The diffusion distance can be expressed by the eigenvalues and eigenvector in the following way:

$$D_t^2(x_i, x_j) = \sum_{k \geq 1} \lambda_k^{2t} (\psi_k(x_i) - \psi_k(x_j))^2. \quad (10)$$

# Diffusion Maps



(a) Original dataset



(b) Embedded dataset

# Diffusion Maps: A simple example

These are a few pictures of a vehicle that belong to a little guy I know:



Rand1.JPG



Rand2.JPG



Rand3.JPG



Rand4.JPG



Rand5.JPG



Rand6.JPG



Rand7.JPG



Rand8.JPG



Rand9.JPG



Rand10.JPG



Rand11.JPG



Rand12.JPG



Rand13.JPG



Rand14.JPG



Rand15.JPG



Rand16.JPG



Rand17.JPG

# Diffusion Maps: A simple example

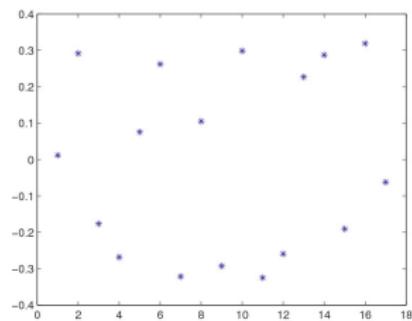
- What is the main difference between the pictures?
- Will the Diffusion Maps algorithm find this?

Now apply the Diffusion Maps algorithm to the picture set in following way:

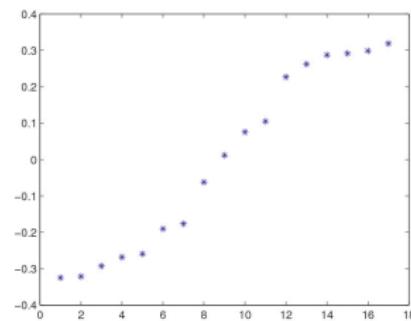
- Reshaped each picture to be a vector in  $R^{600*800*3}$ .
- Built a Gaussian kernel on this high dimensional data.
- Normalized it, and looked at the values of the first eigenvector.

# The truck

These are the values of the first diffusion maps coordinate.



(c) Before sorting



(d) After sorting

# The truck

Pictures sorted by the value on the first Diffusion map coordinate:



Q.JPG



P.JPG



O.JPG



N.JPG



M.JPG



L.JPG



K.JPG



J.JPG



I.JPG



H.JPG



G.JPG



F.JPG



E.JPG



D.JPG



C.JPG



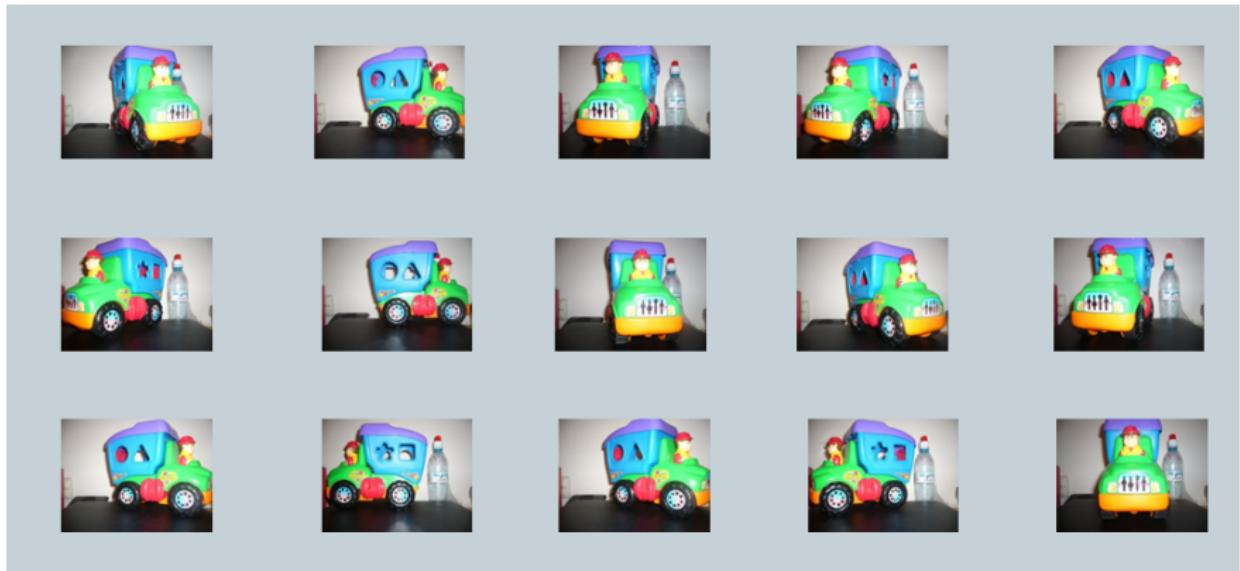
B.JPG



A.JPG

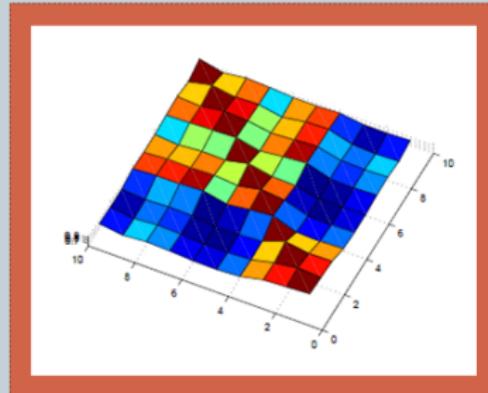
# Toy Example

unordered



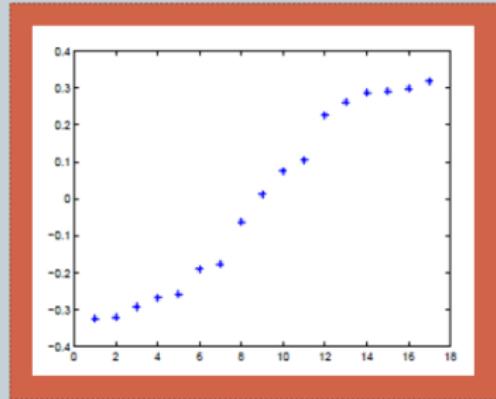
# Toy Example

Applying DM



Construct a transition matrix  
from a Gaussian kernel.

Spectral decomposition



The first non-trivial  
eigenvector, sorted.

# Toy Example

ordered



Images sorted according to their value in the first eigenvector.



# Embedded Space

Given the points  $\Gamma = \{x_1, \dots, x_n\}$  in  $R^m$

Construct a normalized kernel on the data  $P \propto W = e^{-\|x_i - x_j\|^2 / 2\varepsilon}$ .

Use the spectral decomposition of P:  $p(x_i, x_j) = \sum_{k \geq 0} \lambda_k \psi_k(x_i) \phi_k(x_j)$

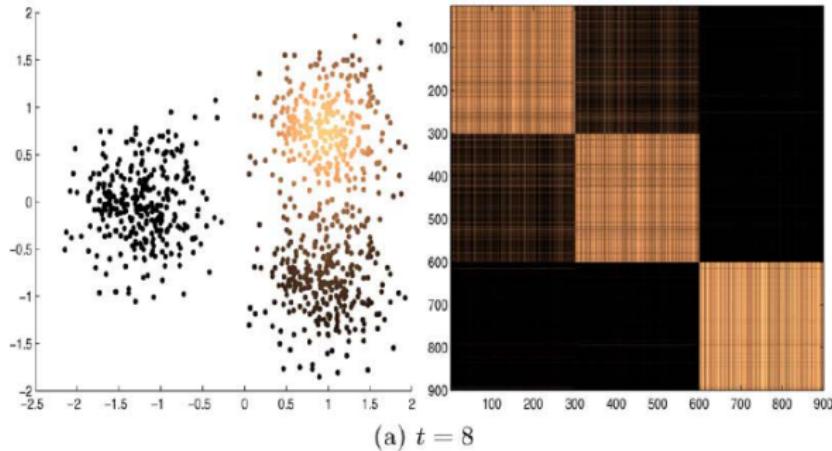
to construct the family of diffusion maps

$$\Psi(x_i) = (\lambda_1 \psi_1(x_i), \lambda_2 \psi_2(x_i), \lambda_3 \psi_3(x_i), \dots)$$

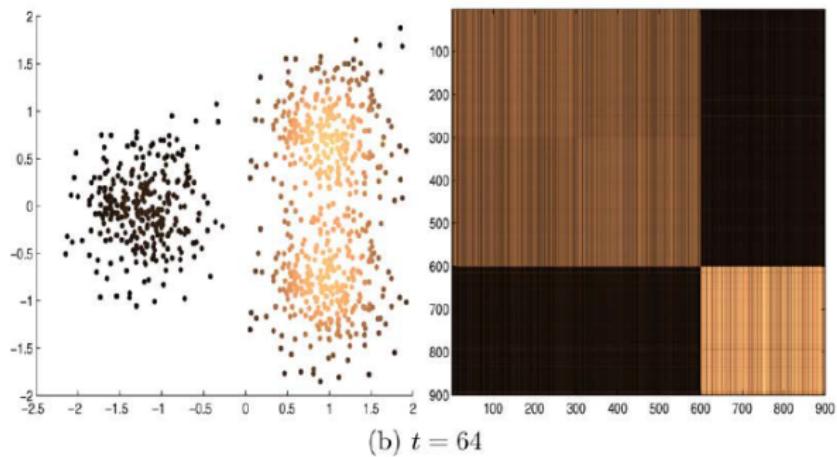
These embed the data to Euclidean space.



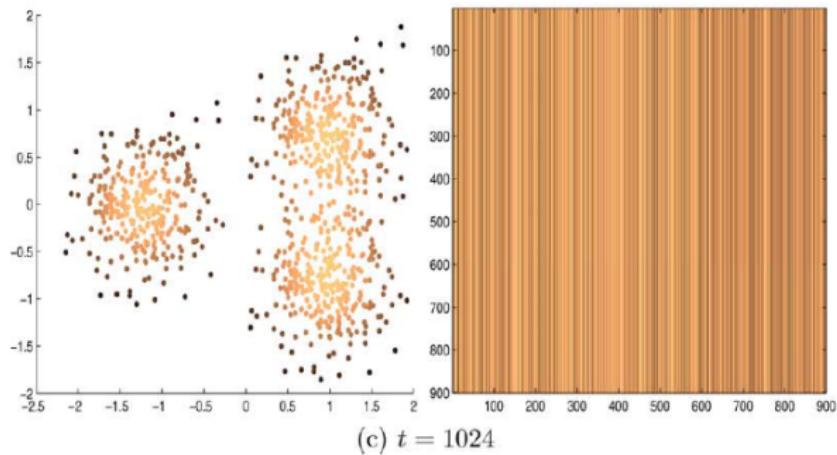
# Embedded Space



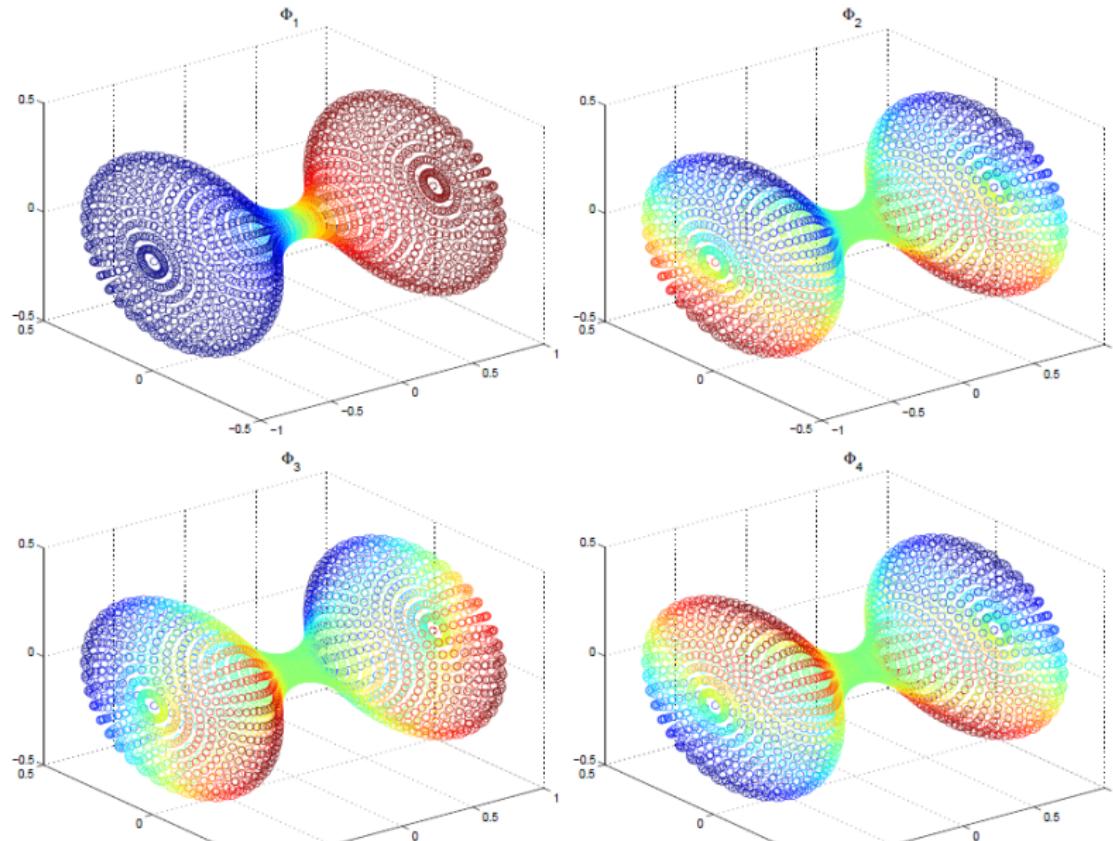
# Embedded Space



# Embedded Space



# Embedded Space



# Acknowledgement

Thanks to Guy Wolf and Neta Rabin for the slides

# Questions?