Clustering in kernel embedding spaces and organization of documents

Stéphane Lafon

Google

Collaborators:

Raphy Coifman (Yale), Yosi Keller (Yale), Ioannis G. Kevrekidis (Princeton), Ann B. Lee (CMU), Boaz Nadler (Weizmann)

Motivation

Data everywhere:

- Scientific databases: biomedical/physics
- The web (web search, ads targeting, ads spam/fraud detection…)
- Books: Conventional libraries + {Amazon, Google Print…}
- Military applications (Intelligence, ATR, undersea mapping, situational awareness…)
- Corporate and financial data
- …

Motivation

Tremendous need for automated ways of

- Organizing this information (clustering, parametrization of data sets, adapted metrics on the data)
- Extracting knowledge (learning, clustering, pattern recognition, dimensionality reduction)
- Make it useful to end-users (dimensionality reduction)

All of this in the most efficient way.

Challenges

- data sets are *high-dimensional*: need for dimension reduction
- data sets are *nonlinear*: need to learn these nonlinearities
- data likely to be *noisy*: need for robustness.
- data sets are *massive*.

Highlights of this talk:

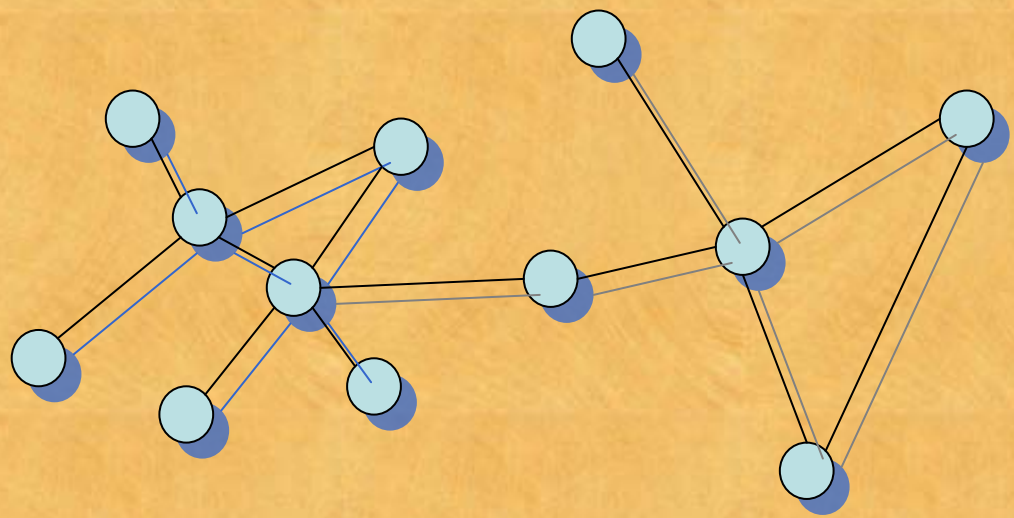
- Review the diffusion framework: parametrization of data sets, dimensionality reduction, diffusion metric
- Present a few results on clustering in the diffusion space
- Extend these results to non-symmetric graphs: beyond diffusions.

From data sets to graphs

Let $X = \{x_1, x_2, ..., x_n\}$ be a data set.

Construct a graph $G = (X, W)$ where

- to each point x_i corresponds a node,
- every two nodes are connected by an edge with a non-negative weight $w(x, y)$.



Choice of the weight matrix

The quantity $w(x, y)$ should reflect the degree of similarity or interaction between x and y.

The choice of the weight is crucial and application-driven.

Examples:

- Take a correlation matrix, and throw away values that are not close to 1.
- If we have a distance metric $d(x, y)$ on the data, consider using $e^{-d(x,y)^2/\varepsilon}$.
- Sometimes, the data already comes in the form of a graph (e.g. social networks).

In practise, difficult problem! Feature selection, dynamical data sets...

Markov chain on the data

Define the degree of node x as $d(x) = \sum_{z \in X} w(x, z)$.

Form the n by n matrix P with entries $p(x, y) = \frac{w(x, y)}{d(x)}$. In other words, $P = D^{-1}W$.

Because $\sum_{y \in X} p(x, y) = 1$ and $p(x, y) \geq 0$,

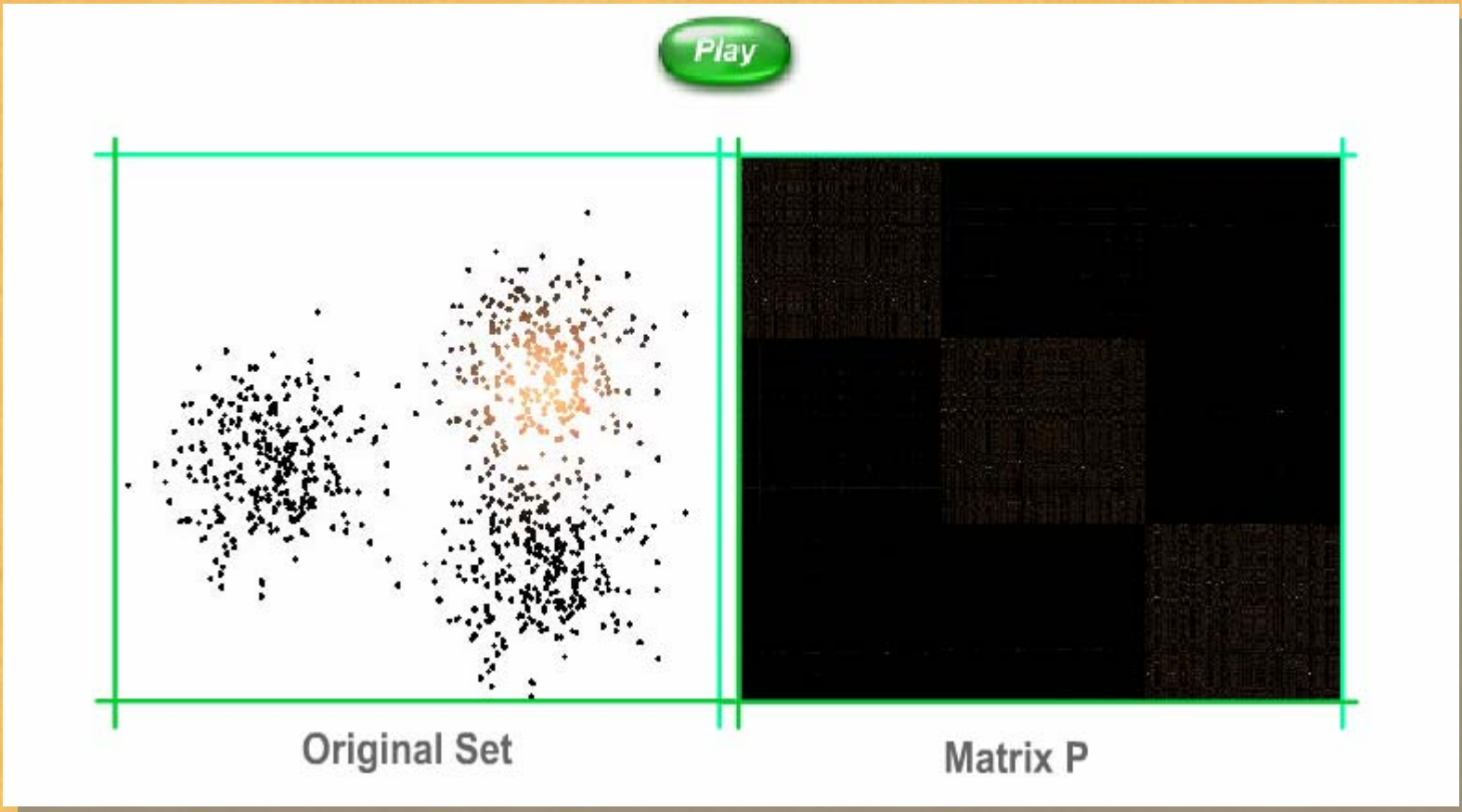
P is the transition matrix of a Markov chain on the graph of the data.

$I - P$ is called "normalized graph Laplacian" [Chung'97].

Notation: entries of P^t are denoted $p_t(x, y)$.

Powers of *P*

Main idea: the behavior of the Markov chain over different time scales will reveal geometric structures of the data.



Time parameter t

$p_t(x, y)$ is the probability of transition from x to y in t time steps.

Therefore, it is close to 1 if y is easily reachable from x in t steps.

This happens if there are many paths connecting these two points

t defines the granularity of the analysis.

Increasing the value of t is a way to integrate the local geometric information:

transitions are likely to happen between similar data points

and occur rarely otherwise.

Assumptions on the weights

Our object of interest: powers of P.

We now assume that the weight function
is symmetric, i.e., that the graph is symmetric.

$$w(x, y) = w(y, x).$$

Spectral decomposition of *P*

With these conditions, P has a sequence $|\lambda_0| \geq |\lambda_1| \geq \ldots \geq |\lambda_{n-1}|$ of eigenvalues and a collection $\{\psi_m\}$ of corresponding (right) eigenvectors

$$P^t \psi_m = \lambda_m^t \psi_m$$

In addition, it can be checked that $\lambda_0 = 1$ and $\psi_0 \equiv \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$.

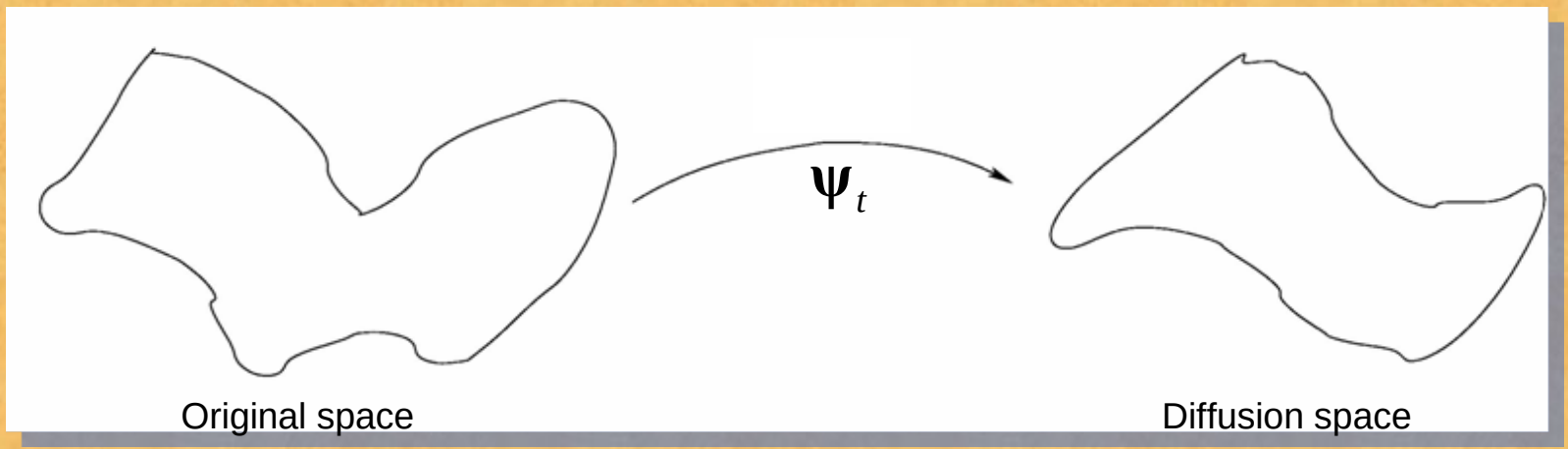
Diffusion coordinates

Each eigenvector is a function defined on the data points.

Therefore, we can think of the (right) eigenvectors $\{\psi_m\}$ as forming a set of coordinates on the data set X.

For any choice of $t \geq 0$, define the mapping:

$$\Psi_t : x \mapsto \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{n-1}^t \psi_{n-1}(x) \end{pmatrix}$$



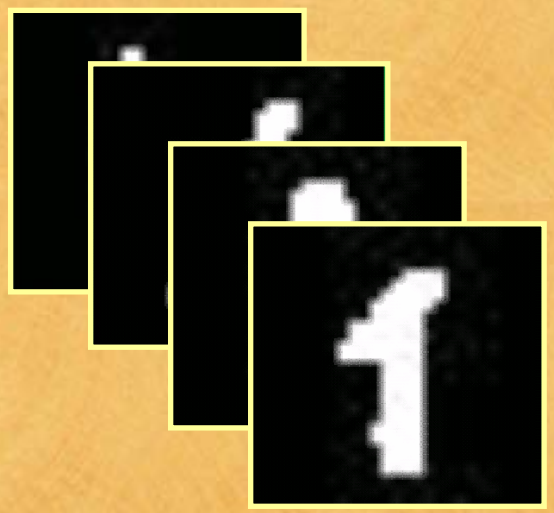
Over the past 5 years, new techniques have emerged for manifold learning

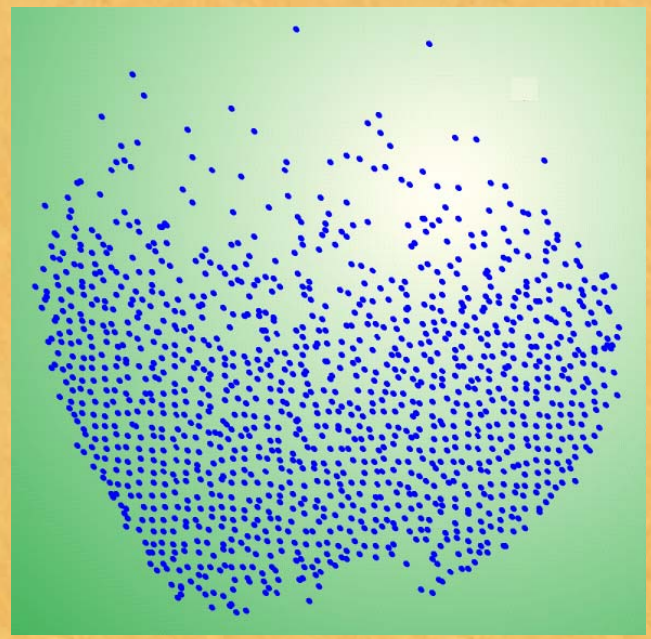
- Isomap [Tenenbaum-DeSilva-Langford'00]
- L.L.E. [Roweis-Saul'00]
- Laplacian eigenmaps [Belkin-Niyogi'01]
- Hessian eigenmaps [Donoho-Grimes'03]
- …

They all aim at finding coordinates on data sets by computing the eigenfunctions of a psd matrix.

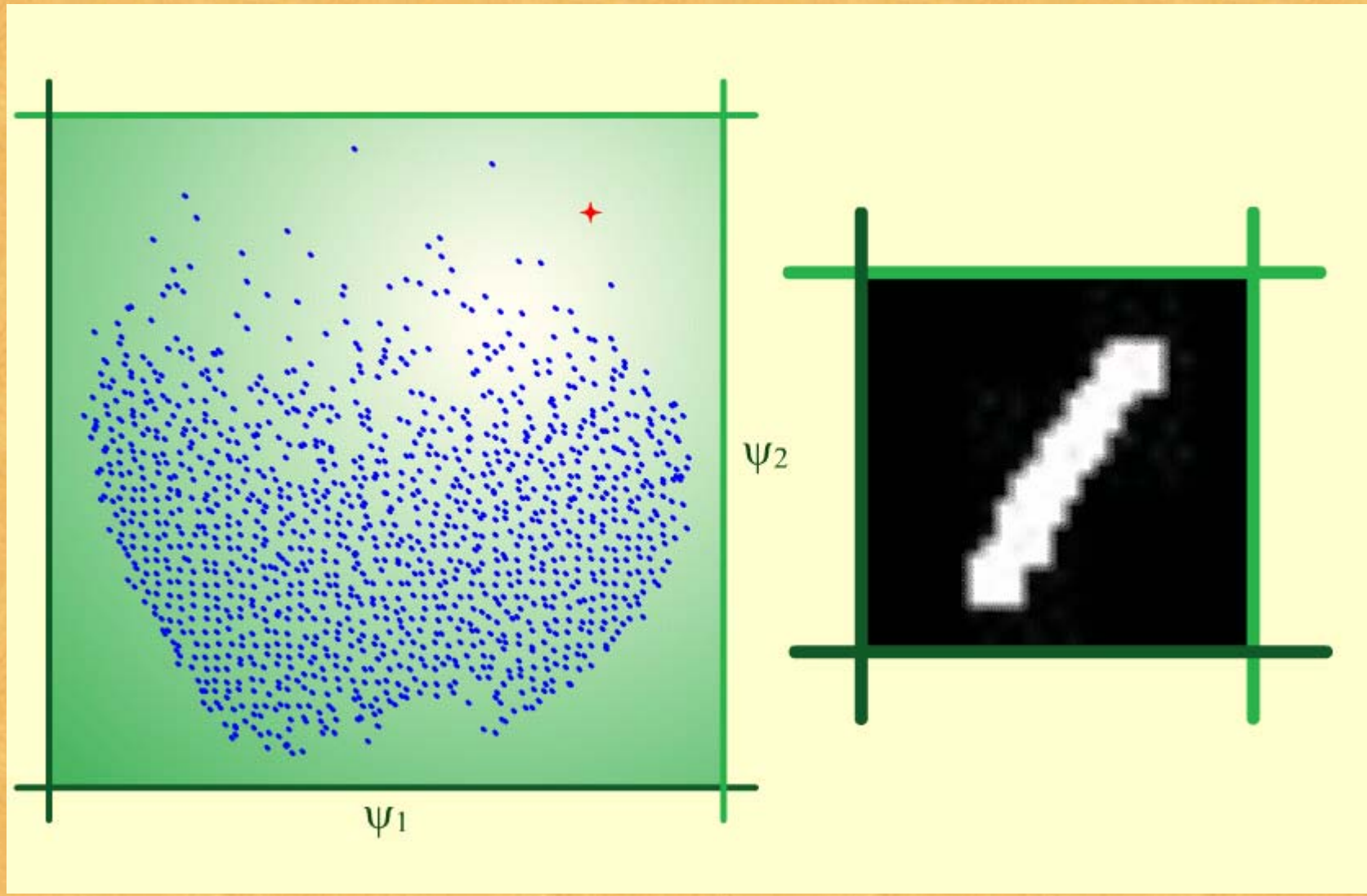
Diffusion coordinates

- Data set: *unordered* collection of images of handwritten digits "1"
- Weight kernel $w(x,y) = e^{-\|x-y\|^2/\varepsilon}$ where $\|x-y\|$ is the L^2 distance between two images x and y





Diffusion coordinates

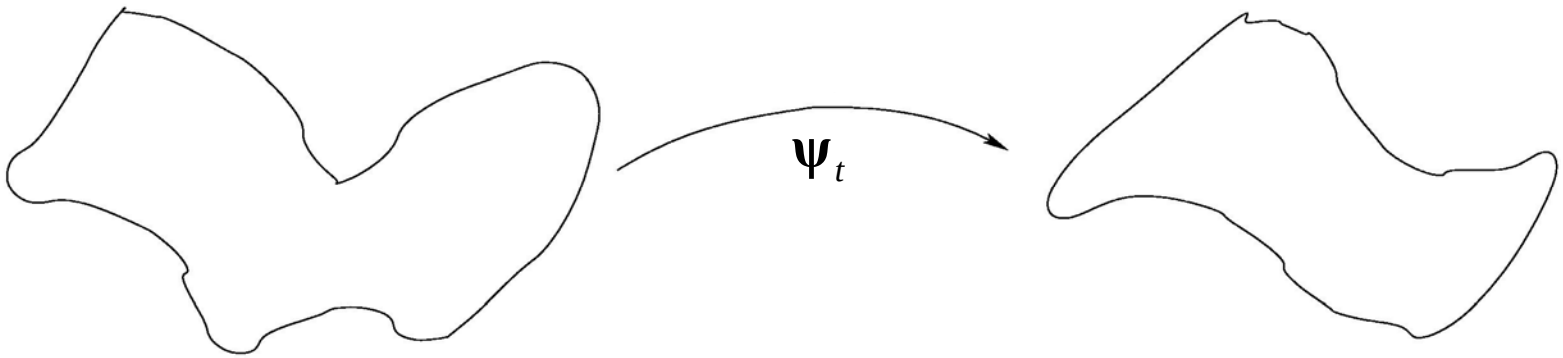


Diffusion distance

Meaning of mapping Ψ_t: two data points x and y are mapped as $\Psi_t(x)$ and $\Psi_t(y)$ so that the distance between them is equal to the so-called "diffusion distance":

$$\left\|\Psi_t(x)-\Psi_t(y)\right\| = D_t(x,y) = \left\|p_t(x,\bullet)-p_t(y,\bullet)\right\|_{L^2(X,1/\pi)}.$$

Two points are close in this metric if they are highly connected in the graph.



Diffusion metric in original space

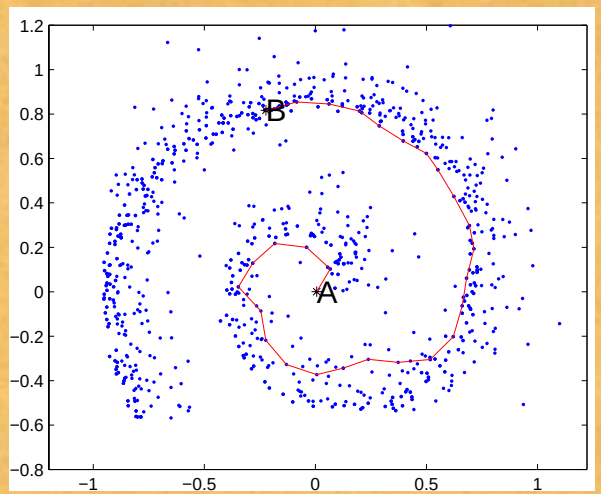
Euclidean distance in diffusion space

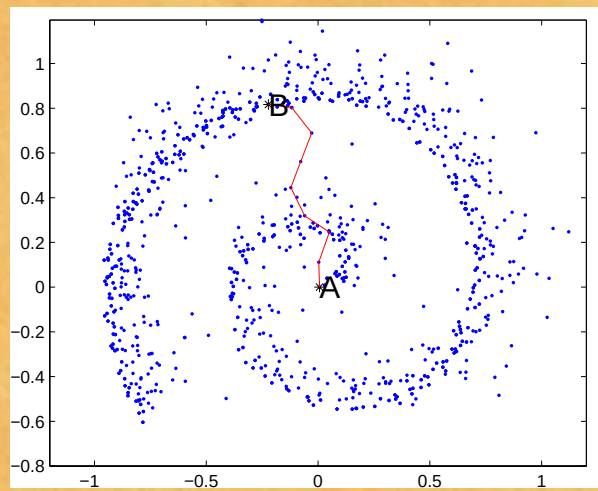
Diffusion distance

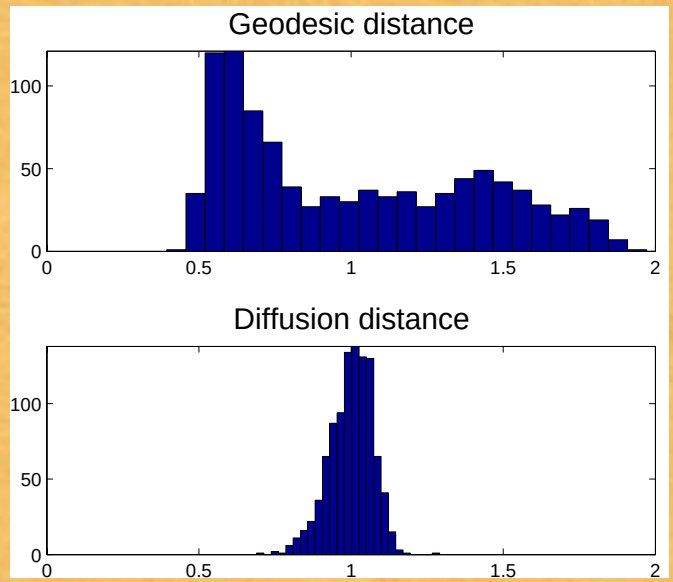
The diffusion metric measures proximity in terms of connectivity in the graph.

In particular,

- It is useful to detect and characterize clusters.
- It allows to develop learning algorithms based on the preponderance of evidences.
- It is very robust to noise, unlike the geodesic distance.







Resistance distance, mean commute time, Green's function…

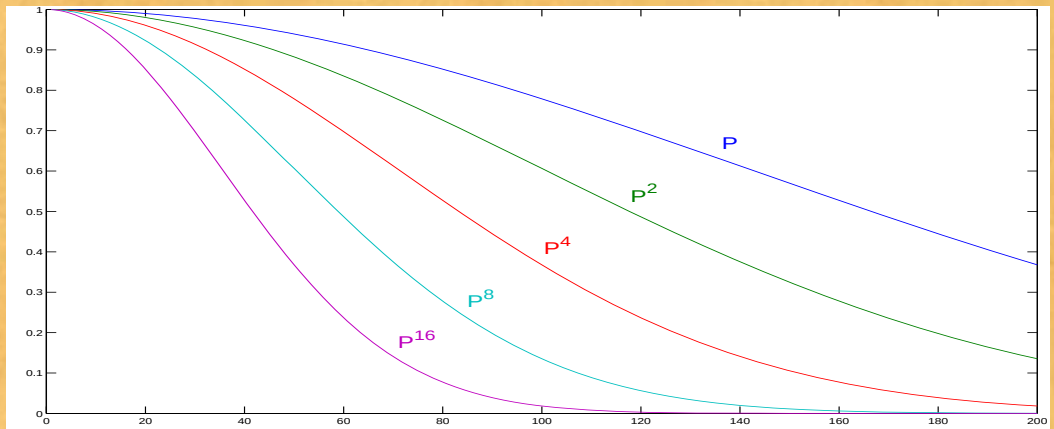
Dimension reduction

Since $1 \geq |\lambda_1| \geq |\lambda_2| \geq \ldots$, not all terms are numerically significant in

$$\Psi_t : x \mapsto \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{n-1}^t \psi_{n-1}(x) \end{pmatrix}$$

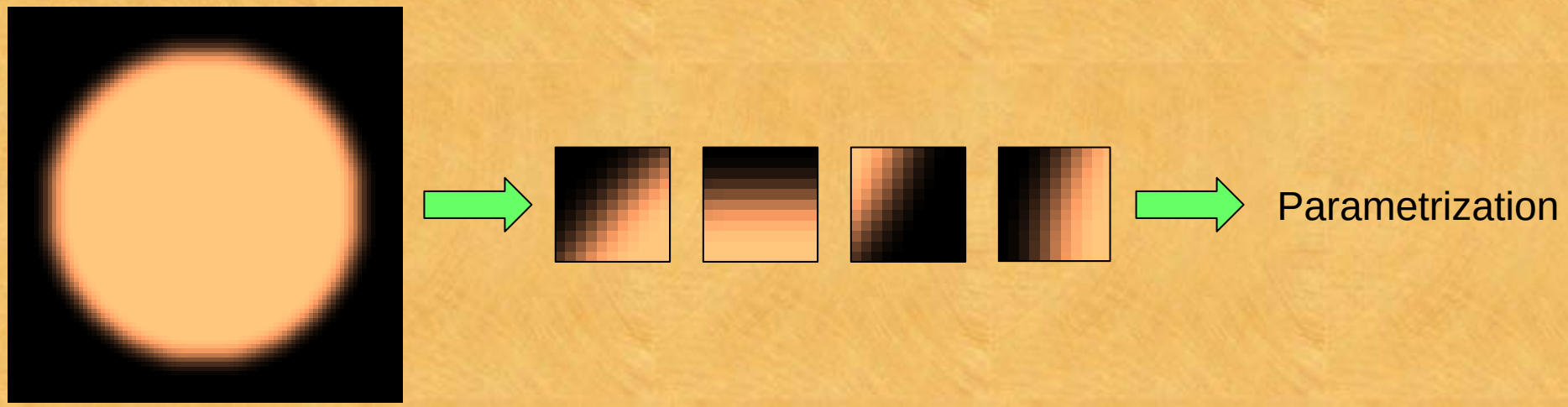
Therefore, for a given value of $t \geq 0$, one only needs to embed using coordinates for which λ_m^t is non-negligible.

The key to dimensionality reduction: decay of the spectrum of the powers of the transition matrix.

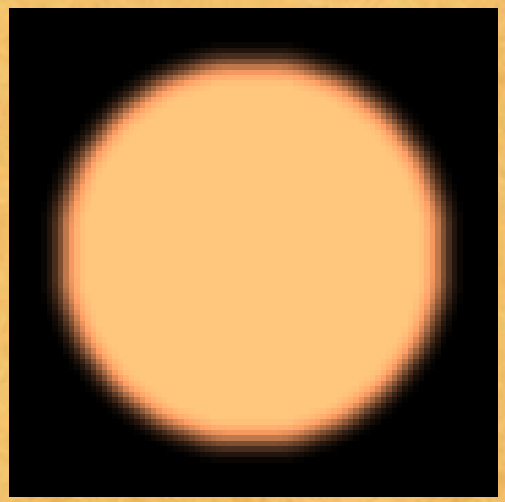


Example: image analysis

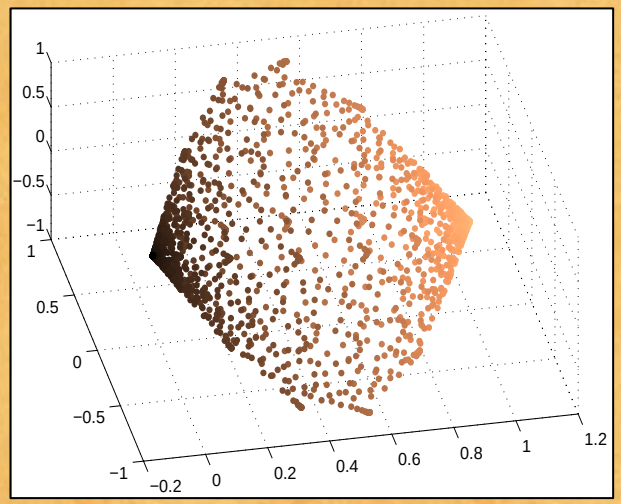
1. Take an image
2. Form the set of all 7x7 patches
3. Compute the diffusion maps



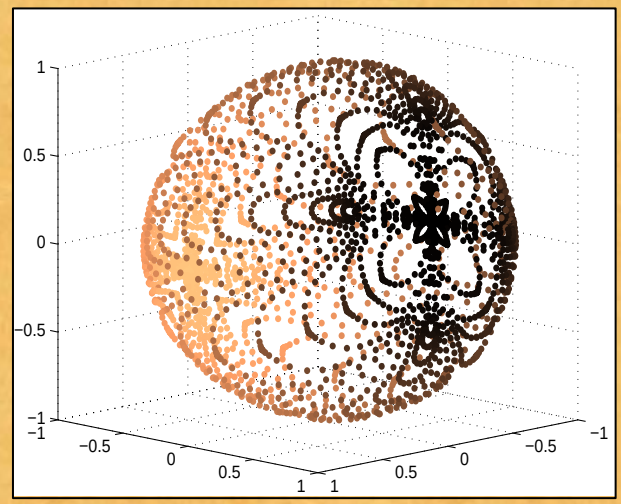
Example: image analysis



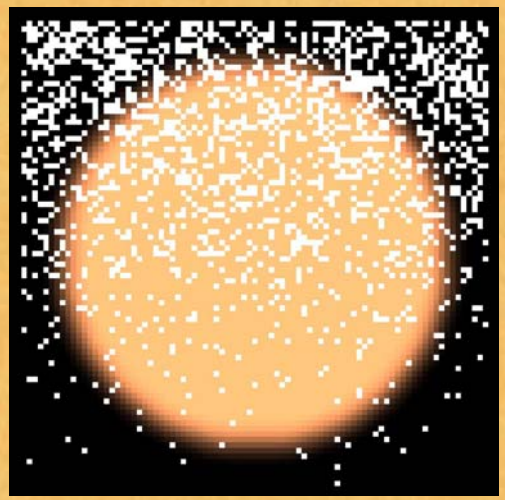
Regular diffusion maps

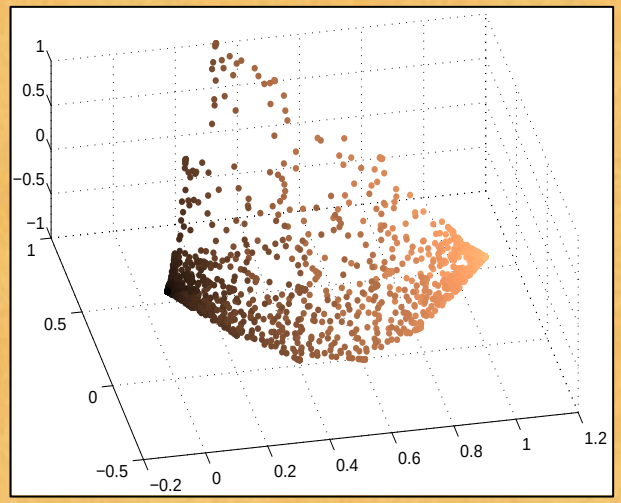


Density-normalized diffusion maps

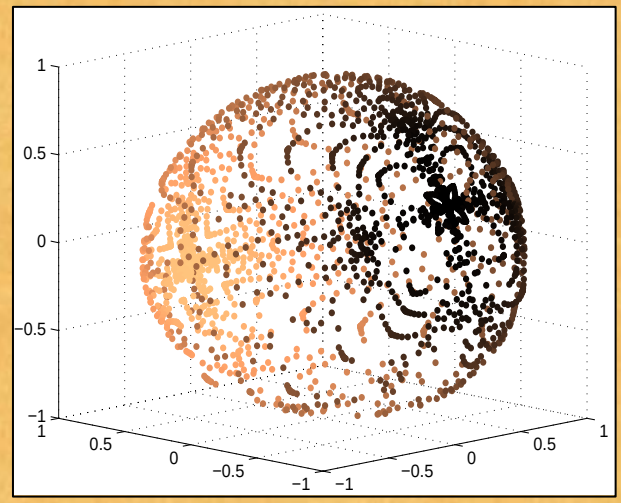


Now, if we drop some patches…





Regular diffusion maps



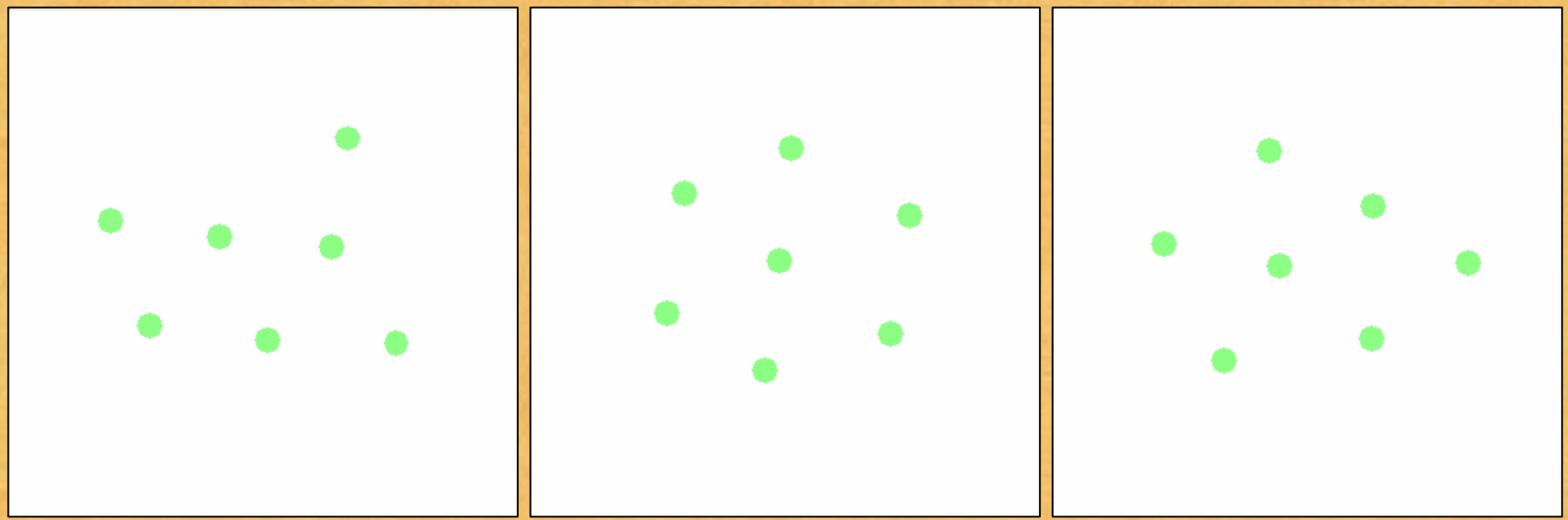
Density-normalized diffusion maps

Example: Molecular dynamics

(with Raphy R. Coifman, Ioannis Kevrekidis, Mauro Maggioni and Boaz Nadler)

Output of the simulation of a molecule of 7 atoms in a solvent. The molecule evolves in a potential, and is also subject to interactions with the solvent.

Data set: millions of snapshots of the molecule (coordinates of atoms).



Graph matching and data alignment

(with Yosi Keller)

Suppose we have two datasets X and Y with approximately the same geometry.

Question: how can we match/align/register X and Y?

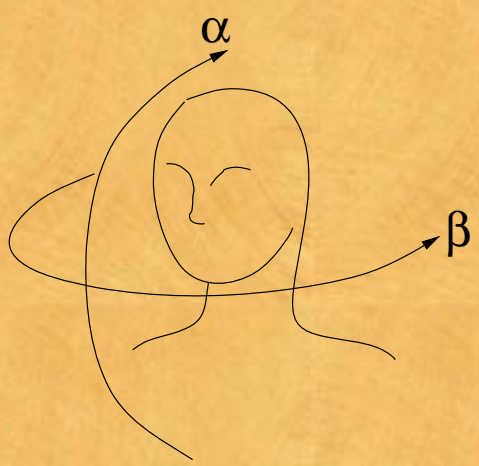
Since we have coordinates for X and Y, we can embed both sets,

and align them in diffusion space.

Graph matching and data alignment

Illustration: moving heads.

The heads of 3 subjects wearing strange masks are recorded in 3 movies.

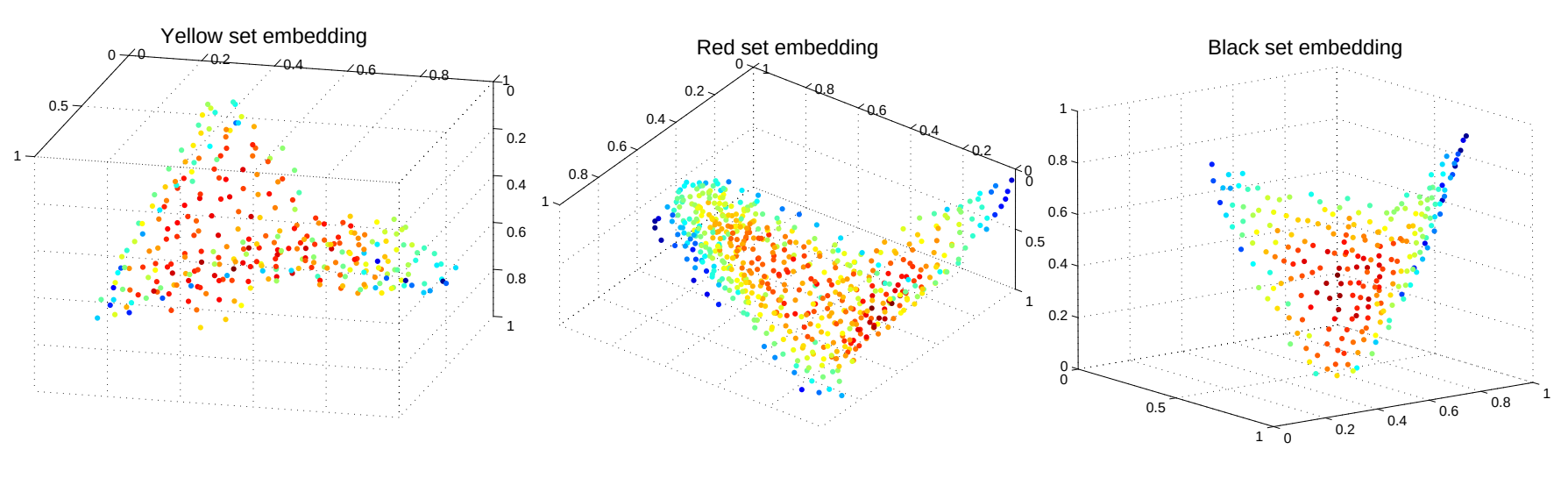


We obtain 3 data sets where each frame is a data point.

Because this is a constrained system (head-neck mechanical articulation), all 3 sets exhibit approximately the same geometry.

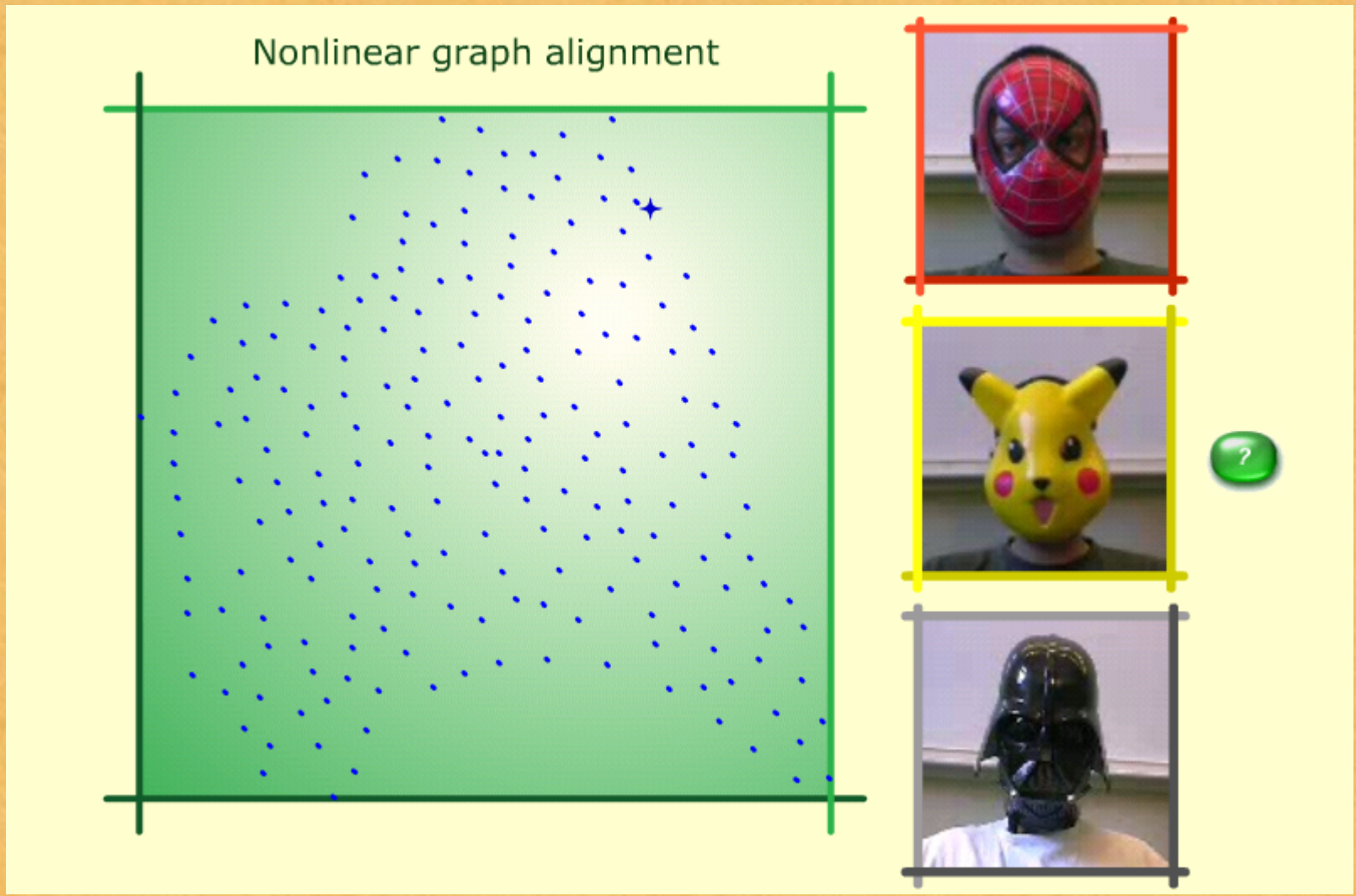
Graph matching and data alignment

We compute 3 density-invariant embeddings.



We then align the 3 data sets in diffusion space
using a limited number of landmarks.

Graph matching and data alignment



Science News articles (reloaded ☺)

- Data set: collection of documents from Science News
- Data modeling : we form a mutual information feature matrix Q.

For each document i and word j, we compute the information shared by i and j :

$$Q(i,j) = \log\left(\frac{N_{i,j}}{\tilde{N}_i N_j}\right) \text{ where } \begin{cases} N_{i,j} : \text{number of words } j \text{ in document } i \\ \tilde{N}_i : \text{ total number of words in document } i \\ N_j : \text{ total number of words } j \text{ in all documents} \end{cases}$$

- We can work on the rows of Q in order to organize documents

or on its columns to organize words.

Word j

$\downarrow$

Document $i \rightarrow \begin{pmatrix} \\ Q(i,j) \\ \\ \end{pmatrix}$

Clustering and data subsampling

Organizing documents: we work on rows of Q.

Automatic extraction of themes,
organization of documents according to these themes.
Can work at different levels of granularity by changing
the number of diffusion coordinates considered.

[TODO: Matlab demo]

Clustering of documents in diffusion space

(with Ann B. Lee)

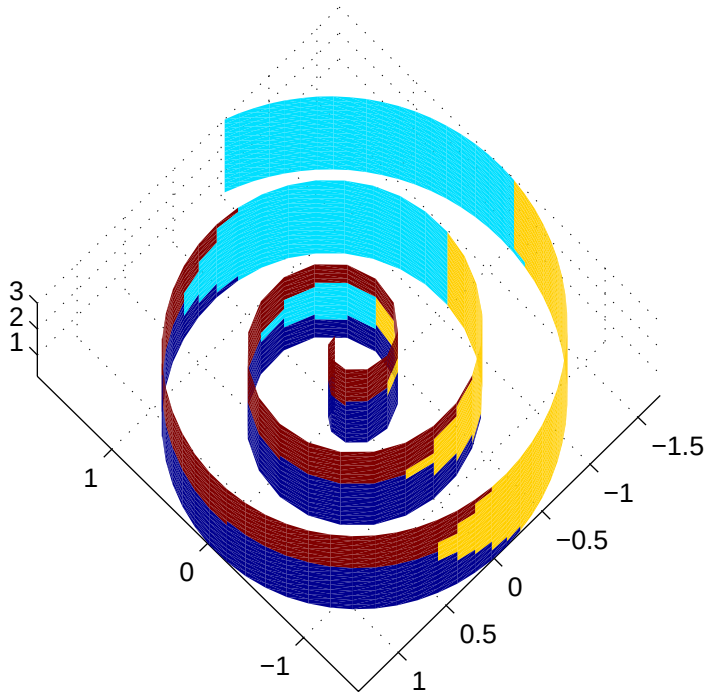
In the diffusion space, we can use classical geometric objects:

- can use hyperplanes
- can use Euclidean distance
- can use balls
- can apply usual geometric algorithms: k-means, coverings…

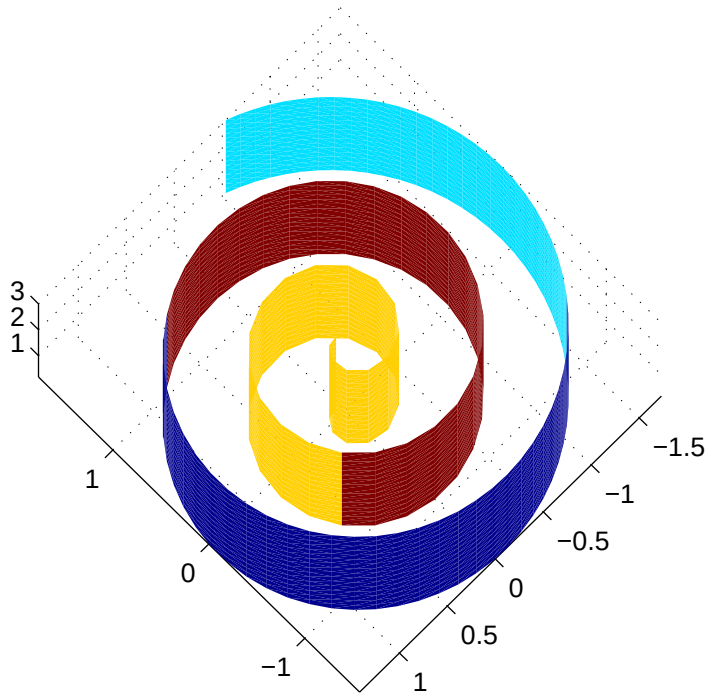
We now show that clustering in the diffusion space is equivalent to compressing the diffusion matrix *P*.

Learning nonlinear structures

K-means in original space



K-means in diffusion space



Coarsening of the graph and Markov chain lumping

Given an arbitrary partition of the nodes $\{S_i\}_{1\le i\le k}$ into k subsets, we coarsen the graph G into a new graph $\tilde{G}$ with k meta-nodes.

- Each meta-node is identified with a subset S_i
- The new edge-weight function is defined by

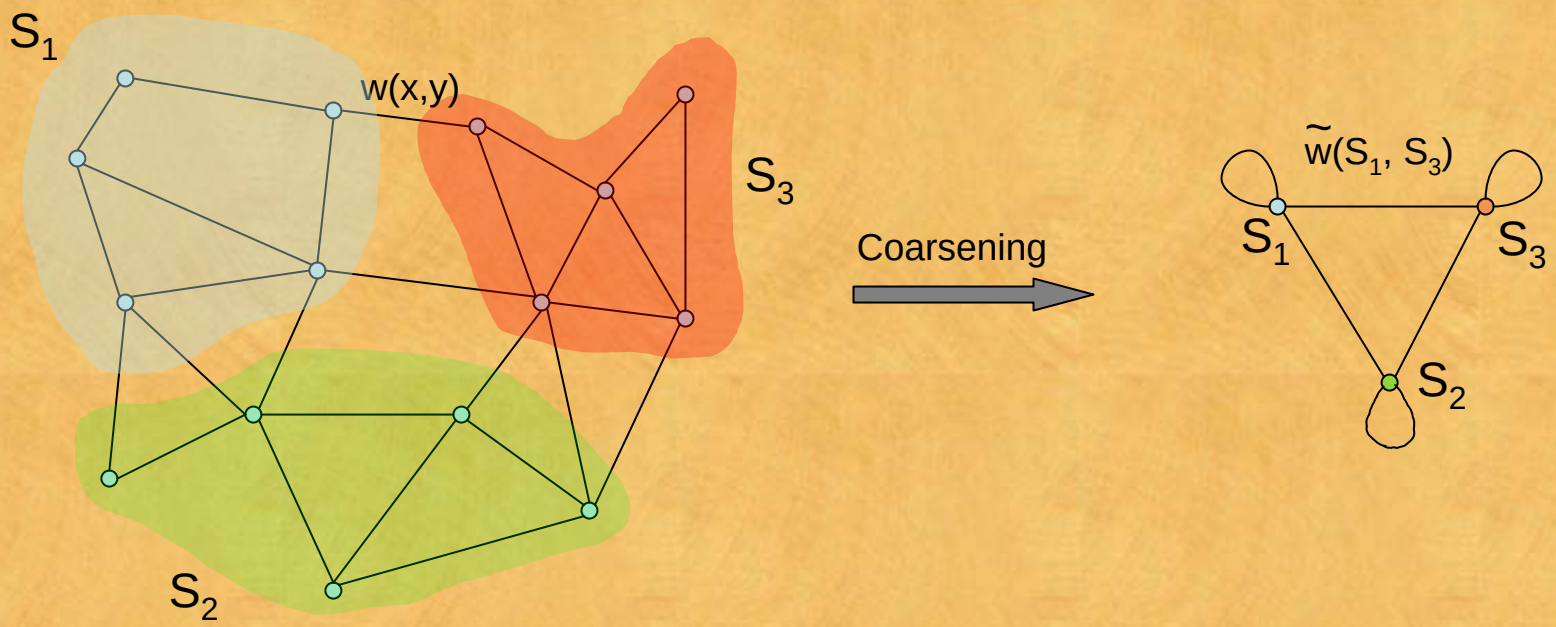
$$\tilde{w}(S_i, S_j) = \sum_{x\in S_i}\sum_{y\in S_j} w(x,y)$$

- We form a new Markov chain on this new graph:

$$\tilde{d}(S_i) = \sum_{1\le j\le k} \tilde{w}(S_i, S_j)$$

$$\tilde{p}(S_i, S_j) = \frac{\tilde{w}(S_i, S_j)}{\tilde{d}(S_i)}$$

Coarsening of the graph and Markov chain lumping



Equivalence with matrix approximation problem

The new Markov matrix $\tilde{P}$ is $k \times k$, and it is an approximation of P.

Are the spectral properties of P preserved by the coarsening operation?

Define the approximate eigenvectors of $\tilde{P}$ by

$$\tilde{\psi}_l(S_i) = \frac{\sum_{x \in S_i} \pi(x) \psi_l(x)}{\sum_{x \in S_i} \pi(x)}$$

These are not actual eigenvectors of $\tilde{P}$, but only approximations.

The quality of approximation depends on the choice of the partition $\{S_i\}$.

Error of approximation

Proposition: We have, for all $0 \le l \le k-1$,

$$\left\| \tilde{P}\tilde{\psi}_l - \lambda_l \tilde{\psi}_l \right\|_\pi \le D$$

where D is the distortion in the diffusion space:

$$D = \left(\sum_{1 \le i \le k} \tilde{\pi}(S_i) \sum_{x \in S_i} \sum_{z \in S_i} \frac{\pi(x)}{\tilde{\pi}(S_i)} \frac{\pi(z)}{\tilde{\pi}(S_i)} \left\| \Psi_1(x) - \Psi_1(z) \right\| \right)^{\frac{1}{2}}$$

This is the quantization distortion induced by $\{S_i\}$, where all the points have a mass distribution π.

This results justifies the use of k-means in diffusion space (MeilaShi) and coverings for data clustering.

Any clustering scheme that aims at minimizing this distortion preserves the spectral properties of the Markov chain.

Clustering of words: automatic lexicon organization

Organizing words: we work on columns of Q.

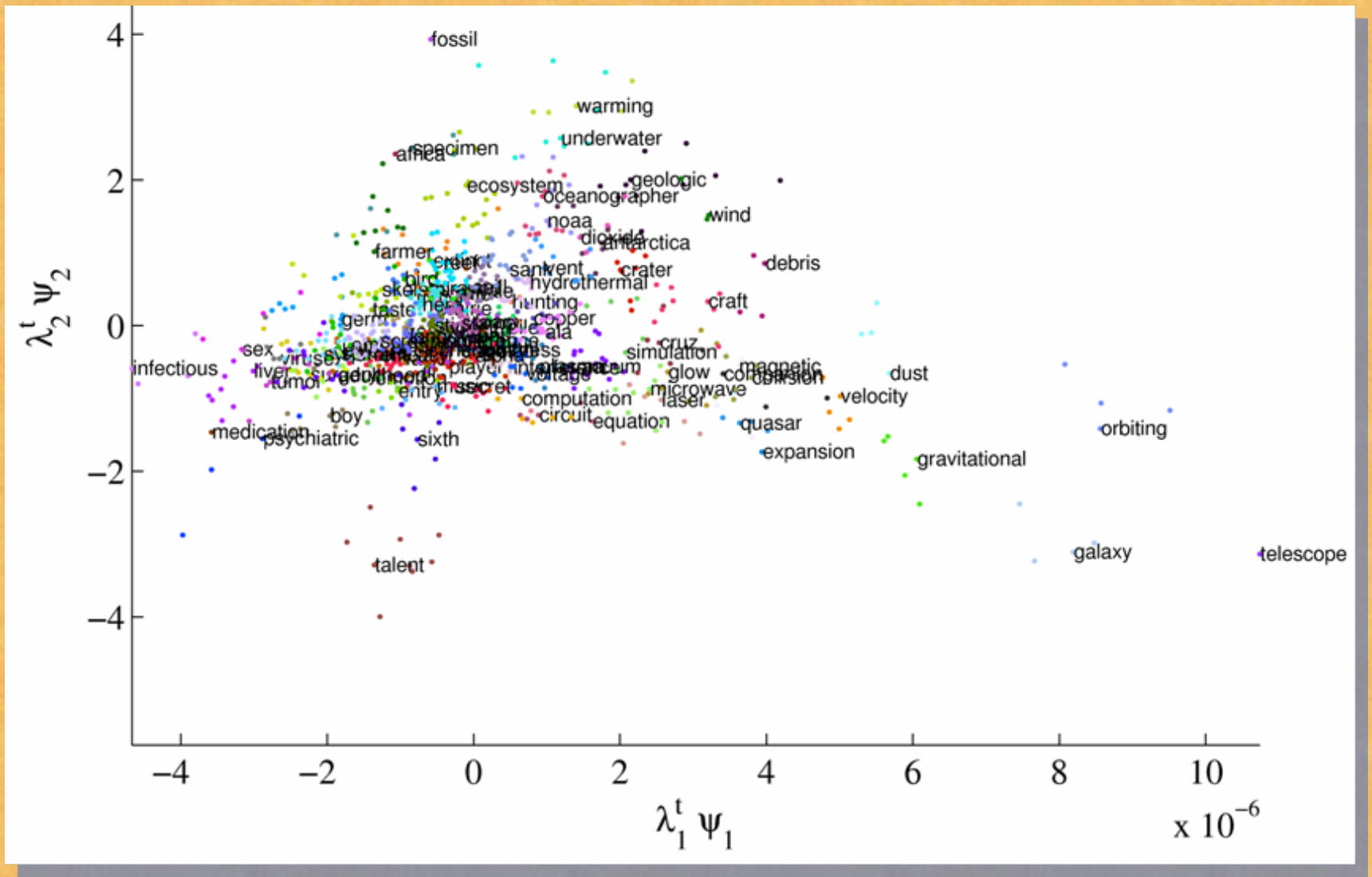
We can subsample the word collection by quantizing the diffusion space.

Clusters can be interpretated as "wordlets", or semantic units.

Centers of the clusters are representative of a concept.

Automatic extraction of concepts from a collection of documents.

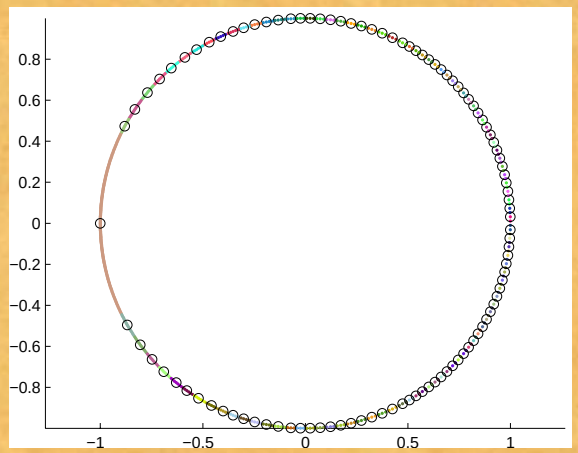
Clustering and data subsampling

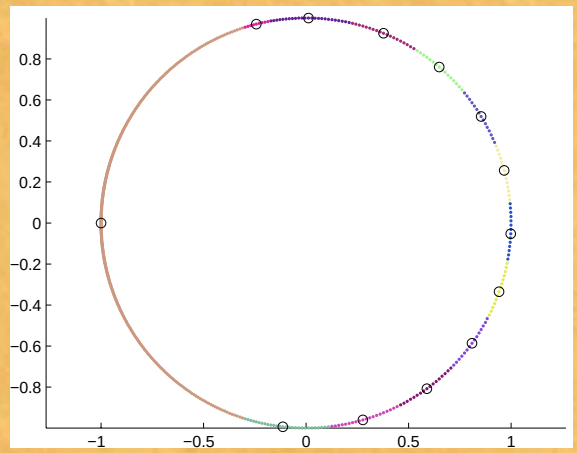


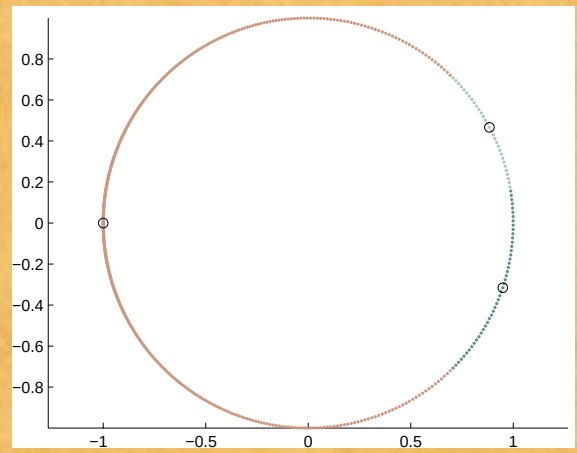
Clustering and data subsampling

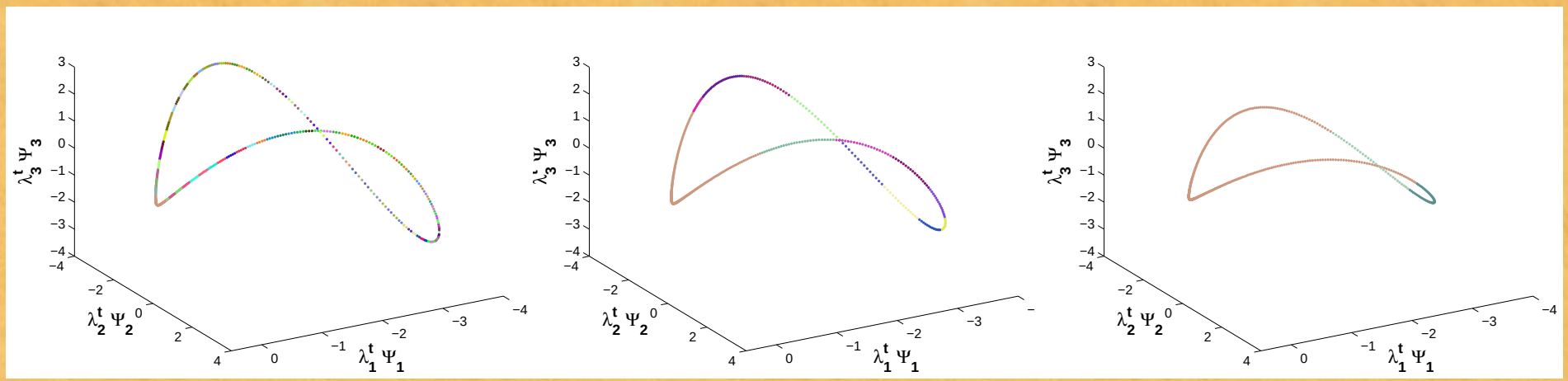
Diffusion center	Words in cluster
infectious	antibody, infected, virus
psychiatric	depression, psychiatrist, psychologist
talent	award, competition, finalist, intel, prize, scholarship, student, winner
laser	beam, nanometer, photon, pulse, quantum
velocity	detector, emit, infrared, ultraviolet
gravitational	bang, cosmo, gravity, hubble
orbiting	jupiter, orbit, solar
geologic	beneath, crust, depth, earthquake, ice, km, plate, seismic, trapped, volcanic
warming	climate, el, nino, pacific, weather
ecosystem	algae, drought, dry, ecologist, extinction, forest, gulf, lake, pollution, river,
farmer	carolina, crop, fish, florida, insect, nutrient, pesticide, pollutant,
	soil, tree, tropical, wash, wood

Going multiscale









t=1

t=16

t=64

Beyond diffusions

The previous ideas can be generalized to the wider setting of a *directed* graph, with *signed* edge-weights $w(\bullet,\bullet)$, and a set of positive node-weights $\pi(\bullet)$.

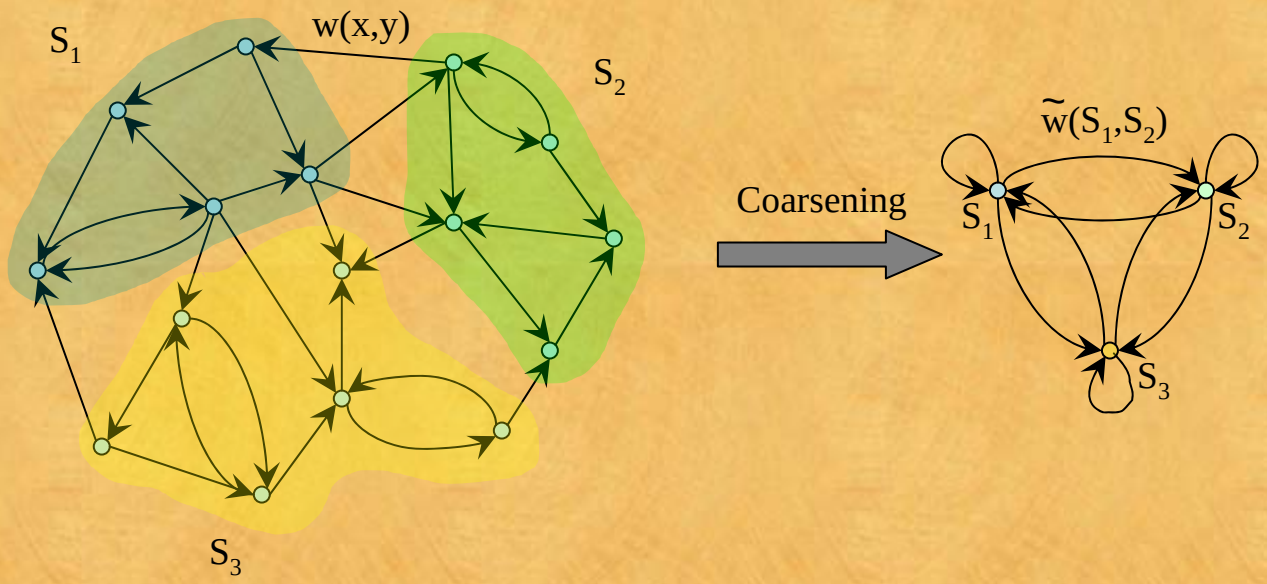
- This time, we consider oriented interactions between data points (e.g. source-target). No spectral theory available!
- The node-weight can be used to rescale the relative importance of each data point (e.g. density estimate).
- This framework contains diffusion matrices as in this case the node-weight is equal to the out-degree of each node.

Coarsening

This time we have no spectral embedding.

However, for a given partitioning $\{S_i\}$ of the nodes,

the same coarsening procedure can still be applied to the graph.



$$\tilde{w}(S_i, S_j) = \sum_{x \in S_i} \sum_{y \in S_j} w(x, y) \text{ and } \tilde{\pi}(S_i) = \sum_{x \in S_i} \pi(x)$$

Beyond spectral embeddings

Idea: form the matrix $A = \Pi^{-1} W$ and view each row $a(x,\bullet)$ as a feature vector representing x. The graph G now lives as a cloud of points in $\mathbb{R}^n$.

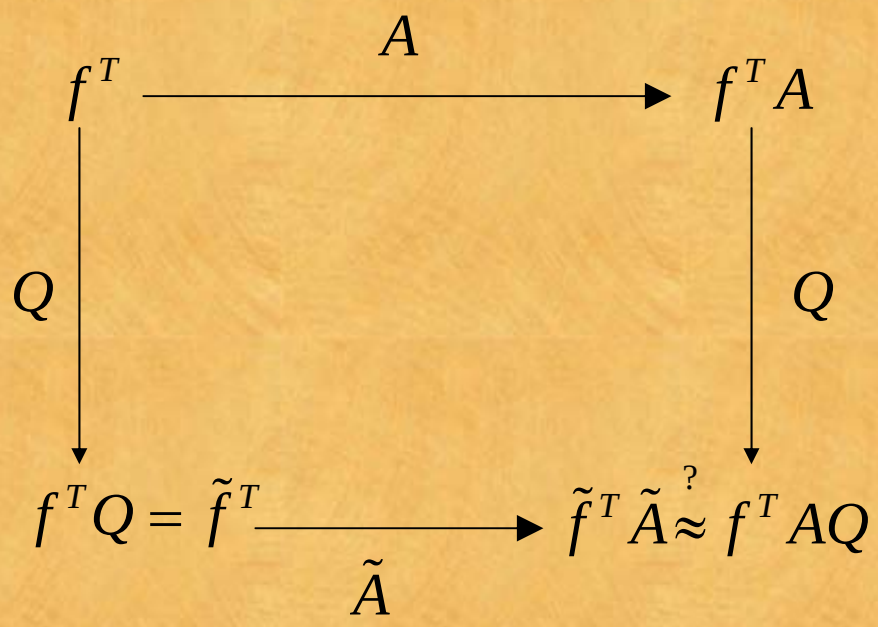
Similarly for the coarser graph $\tilde{G}$, form $\tilde{A} = \tilde{\Pi}^{-1} \tilde{W}$ and obtain a cloud of points in $\mathbb{R}^k$.

Question: how do these pointsets compare ?

Comparison between graphs

A is $n \times n$, whereas $\tilde{A}$ is $k \times k$.

They can be related by the $n \times k$ matrix Q whose rows are the indicators of the partitioning.



In other words, this diagram should approximately commute:

$$Q\tilde{A} \approx AQ$$

Error of approximation

We now have a similar result as before:

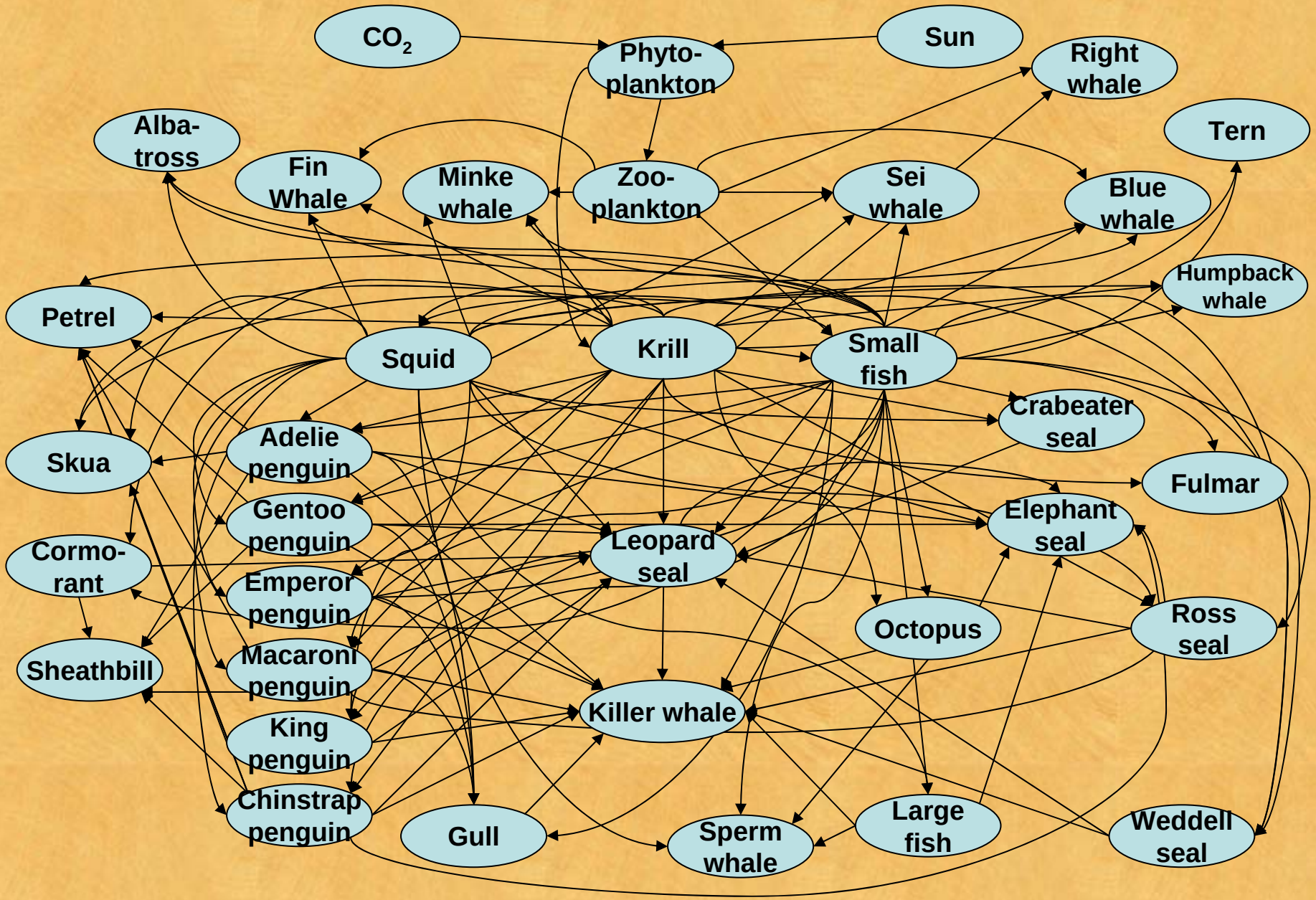
Proposition:

$$\sup_{f \in \mathbb{R}^n : \|f\|_{1/\pi} = 1} \left\| f^T \left(Q\tilde{A} - AQ \right) \right\|_{1/\tilde{\pi}} \leq D$$

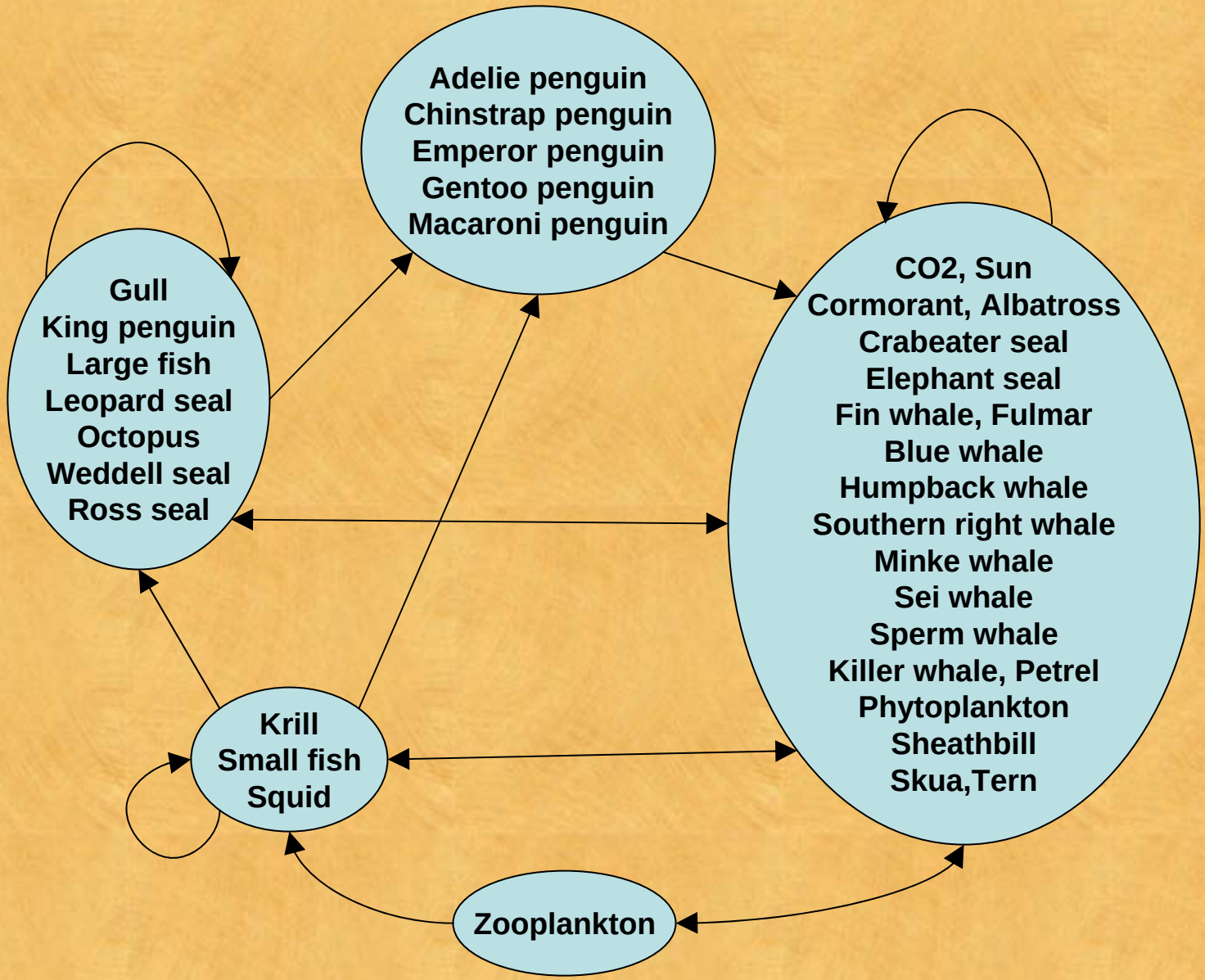
where $D = \left(\sum_{1 \leq i \leq k} \tilde{\pi}(S_i) \sum_{x \in S_i} \sum_{z \in S_i} \frac{\pi(x)}{\tilde{\pi}(S_i)} \frac{\pi(z)}{\tilde{\pi}(S_i)} \left\| a(x,\bullet) - a(z,\bullet) \right\|_{\pi} \right)^{\frac{1}{2}}$.

D is the distortion resulting from quantizing the set of rows of A, with mass distribution π

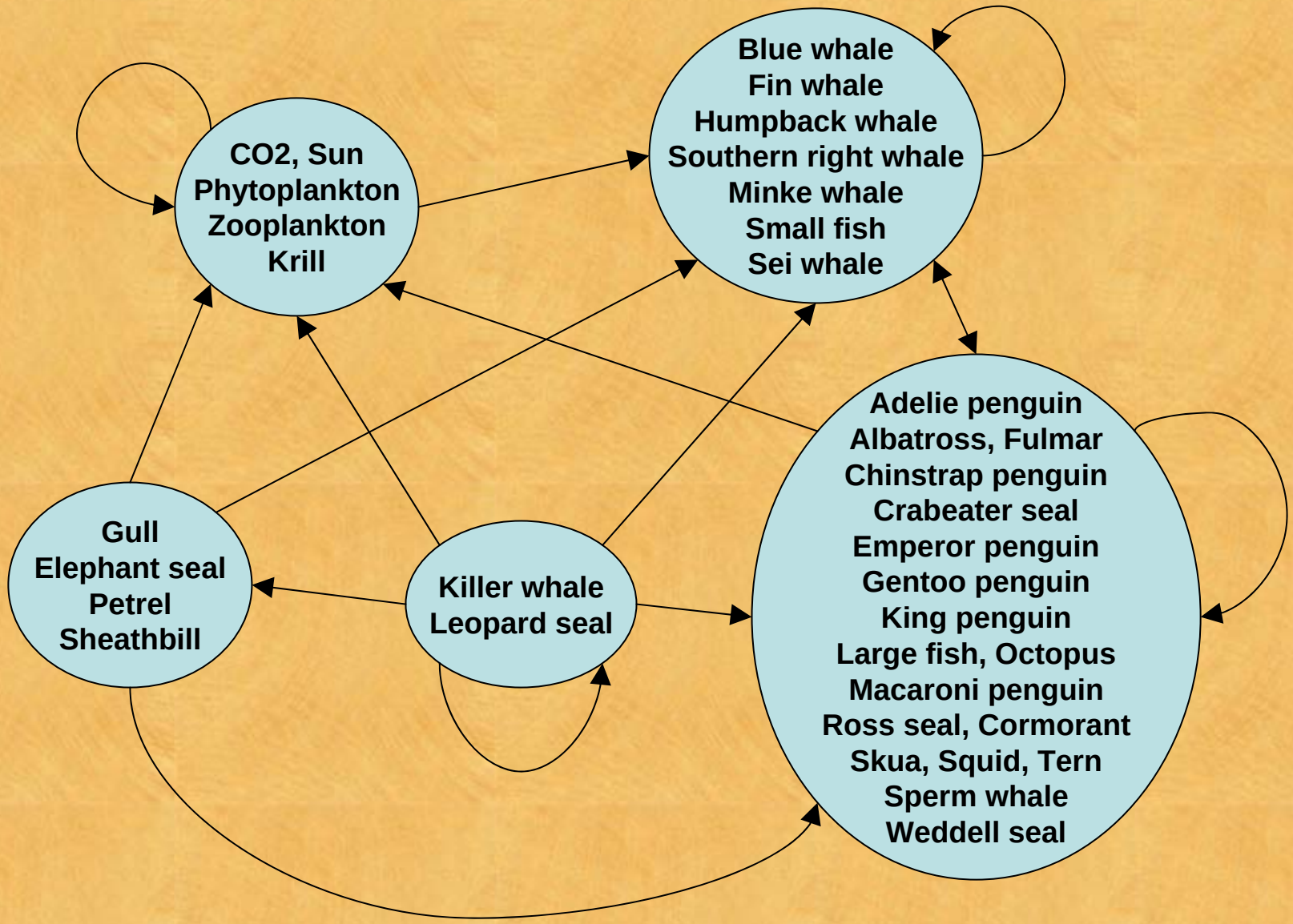
Antarctic food web



Partitioning the forward graph



Partitioning the reverse graph



Conclusion

- Very flexible framework providing coordinates on data sets
- Diffusion distance relevant to data mining and machine learning applications
- Clustering meaningful in this embedding space
- Can be extended to directed graphs
- Multiscale bases and analyses on graphs

Thank you!