Data fusion and multi-cue data matching using diffusion maps

Stéphane Lafon

Collaborators:
Raphy Coifman, Andreas Glaser, Yosi Keller, Steven Zucker
(Yale University)

Part of this work was supported by AFRL.
Motivation

Many applications involve mining high-dimensional data sets:

- Scientific data: biomedical/physics
- The web (web search, ads targeting, ads spam/fraud detection…)
- Military applications (intelligence, ATR, undersea mapping, situational awareness…)
- Corporate and financial data
- …
Typical “machine learning” tasks involved in processing these data:

• Clustering
• Visualization
• Classification
• Regression
• Pattern recognition
Challenges

• data sets are *high-dimensional*: need for dimension reduction
• data sets are *nonlinear*: need to learn these nonlinearities
• data coming from several sensors at the same time, sampled at different rates: need for *density-invariant data fusion* techniques.
• data likely to be *noisy*: need for robustness.
Main points of this talk:

Present the diffusion framework.

Show that it responds to the concerns of the previous slide:

1. Nonlinear dimension reduction: the intrinsic geometry of data sets is preserved.
2. Easy way to deal with sampling issues: density invariant embeddings.
3. Framework can be extended to a multisensor context.
From data sets to graphs

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a data set.
Construct a graph $(X, W)$ 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).
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 $P'$ are denoted $p_i(x, y)$. 
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.

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

If a problem, replace $W$ by $W^TW$ or $WW^T$. 
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

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

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 signal over 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:

$$\psi_t : x \mapsto \left( \begin{array}{c} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{n-1}^t \psi_{n-1}(x) \end{array} \right)$$
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 $w(x, y) = e^{-\|x-y\|/\epsilon}$ where $\|x-y\|$ is the $L^2$ distance between two images $x$ and $y$
Diffusion coordinates
Learning the constraints

Most of the time, the constraints are not known in advance and have to be learned from the data.

Goal of manifold learning: to find a system of coordinates that parameterize/describe the underlying manifold.

Note: Most manifolds are nonlinear, so can’t efficiently use global linear techniques like Principal Component Analysis (SVD).
Learning nonlinear structures

K-means in original space

K-means in diffusion space
Diffusion distance

Meaning of mapping $\Psi_t$: 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":

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

Two points are close in this metric if they are highly connected in the graph.
**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.
Generalization of spectral clustering

The diffusion distance can be viewed as generalizing the idea of computing an approximate solution to NCut via the first non constant eigenvector $\psi_1$ [Shi-Malik'97].
Dimension reduction

Since \( 1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq \ldots \), 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 \( \lambda_m^t \) is non-negligible.

The key to dimensionality reduction: decay of the spectrum of the powers of the transition matrix.
Density invariant diffusion maps

Suppose that the data set \( X = \{x_1, x_2, \ldots, x_n\} \) is sampled from a submanifold \( M \) of \( \mathbb{R}^d \) with density \( q \).

In many applications, one is only interested in the geometry of \( M \) (i.e., the set of constraints) and not in the distribution of the points (density \( q \)).

Therefore, we need the diffusion coordinates to be density-invariant.
Density invariant diffusion maps

Form the graph with weights $w_\varepsilon(x, y) = \frac{e^{-\|x-y\|^2 / \varepsilon}}{\hat{q}_\varepsilon(x)\hat{q}_\varepsilon(y)}$.

where $\hat{q}_\varepsilon(x) = \sum_{y \in X} e^{-\|x-y\|^2 / \varepsilon}$.

Again, form the transition matrix $P_\varepsilon$ corresponding to this graph.
Convergence to the Laplace-Beltrami operator

Define the normalized Graph Laplacian: \( \Delta_\varepsilon = \frac{I - P_\varepsilon}{\varepsilon} \)

Theorem: as the number of points \( n \to +\infty \) and the scale \( \varepsilon \to 0 \),
we have \( \Delta_\varepsilon \to c\Delta \) and \( P_\varepsilon^{t/\varepsilon} \to e^{-c\Delta t} \)

Therefore we can approximate the heat flow on manifolds.
We also obtain density-invariant diffusion coordinates.
Example: image analysis

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

Parametrization
Example: image analysis

Now, if we drop some patches…
Graph matching and data alignment

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 [Lafon-Keller'05].
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.
Alignment algorithm

Start with two data sets $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_m\}$, and a set of $k$ examples (landmarks) $\{(x_{\sigma(1)}, y_{\tau(1)}), ..., (x_{\sigma(k)}, y_{\tau(k)})\}$

- Embed each set in the diffusion space with $k - 1$ diffusion coordinates. We obtain 2 clouds of points, $\overline{X} = \{\overline{x}_1, ..., \overline{x}_n\}$ and $\overline{Y} = \{\overline{y}_1, ..., \overline{y}_m\}$, in $\mathbb{R}^{k-1}$.
- In the diffusion space, compute the affine function $f : \mathbb{R}^{k-1} \rightarrow \mathbb{R}^{k-1}$ that satisfies $f(\overline{x}_{\sigma(1)}) = \overline{y}_{\tau(1)}, ..., f(\overline{x}_{\sigma(k)}) = \overline{y}_{\tau(k)}$
- Define the correspondence between $X$ and $Y$ to be $g(x_i) = \arg\min_{y \in Y} \| f(x_i) - y \|$
Graph matching and data alignment

Nonlinear graph alignment
Out-of-sample extension

In real-life situation, it is necessary to be able to extend the diffusion coordinates learned offline to new data points. This can be done efficiently using special sets of functions termed geometric harmonics [Coifman-Lafon’04].

The extension procedure is a multiscale version of the Nyström extension, in which the complexity of the empirical function determines the distance to which it can be extended off the training set.

When the domain (training set) is a manifold, complexity is measured in terms of the decay of the intrinsic spectrum.
Data fusion and pattern recognition

We often need to integrate data streams coming from sources (e.g., audio + video). Major issues:

• scaling of each variable, non-commensurate sensors
• integration of the various streams into a unified representation

We propose the following algorithm:
Multisensor integration: Lip reading + speech recognition

We recorded the voice and a movie of the lips of a subject reading some text. In this example we have 2 streams of data:

- audio track of speaker
- video frames of lips moving

We learned the intrinsic geometry of both data streams independently, and then we combined them into a unique representation.
1. We first learned the geometry of speech data from the recording (audio + video) of the speaker reading an article.

2. We then had the subject repeat several instances of each digit (“0”, “1”,…, “9”)
Spoken digits recognition

3. Each digit is viewed as a trajectory in the diffusion space.

4. We trained a (trajectory) nearest-neighbor classifier in this space
### Results

<table>
<thead>
<tr>
<th>Channel type</th>
<th>“0”</th>
<th>“1”</th>
<th>“2”</th>
<th>“3”</th>
<th>“4”</th>
<th>“5”</th>
<th>“6”</th>
<th>“7”</th>
<th>“8”</th>
<th>“9”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>0.75</td>
<td>0.94</td>
<td>0.87</td>
<td>0.90</td>
<td>0.96</td>
<td>0.86</td>
<td>0.93</td>
<td>0.81</td>
<td>0.80</td>
<td>0.92</td>
</tr>
<tr>
<td>Visual</td>
<td>0.90</td>
<td>0.99</td>
<td>0.90</td>
<td>0.94</td>
<td>0.93</td>
<td>0.81</td>
<td>0.87</td>
<td>0.74</td>
<td>0.75</td>
<td>0.82</td>
</tr>
<tr>
<td>Combined</td>
<td>0.90</td>
<td>0.99</td>
<td>0.96</td>
<td>0.99</td>
<td>0.96</td>
<td>0.97</td>
<td>0.90</td>
<td>0.93</td>
<td>0.95</td>
<td>0.96</td>
</tr>
</tbody>
</table>

- Best performance achieved when combining both information streams
- The combined diffusion metric gets “the best” of the audio and visual diffusion metrics by adding their discriminative powers.
Multi-cue image segmentation

- RGB features
- Interleaving contours features
Multi-cue image segmentation

RGB features

IC features

Combined sensors
Computational issues

We need to compute eigenvectors of a very large matrix.

Note however that:

1. This matrix is usually sparse
2. It is conjugate to a symmetric matrix
3. We don’t need all eigenvectors, but just the top ones
4. Computation is parallelizable
5. There exist fast algorithms, in particular using diffusion wavelets [CoifmanMaggioni04].
Conclusion

• Very flexible framework providing coordinates on data sets
• Diffusion distance relevant to data mining and machine learning applications
• Naturally extends to a multisensor context

Thank you!

Other material and demos at http://www.yale.edu/~sl349