Geometry Learning for Multimodal Signal Processing

Ronen Talmon

Technion-IIT, Haifa, Israel

Joint work with Roy R. Lederman

40th SIPL day, July 2015
Multimodal data analysis: sleep stage identification

Figure: Sleep polysomnography [Source: NIH]
Multimodal data analysis: sleep stage identification

**Our goal:** Based on the 3 multimodal respiratory signals, in a purely data-driven manner, we would like to recover the underlying process related to the breathing/sleep
Problem definition

Setting:

- Three *hidden* high-dimensional random variables, \( X, Y \) and \( Z \), such that given \( X \), the variables \( Y \) and \( Z \) are independent.

\[
(X, Y, Z) \sim \pi_{x,y,z}(X, Y, Z)
\]

\[
\pi_{x,y,z}(X, Y, Z) = \pi_x(X) \pi_{y|x}(Y|X) \pi_{z|x}(Z|X)
\]

- Two measured *observable* random variables: \( S^{(1)} = g(X, Y) \) and \( S^{(2)} = h(X, Z) \), where \( g \) and \( h \) are bilipschitz functions.

Goal:

- Given \( n \) pairs of measurements \( \left\{ (s_i^{(1)}, s_i^{(2)}) \right\}_{i=1}^{n} \), for \( n \) realizations of the hidden variables \( \left\{ (x_i, y_i, z_i) \right\}_{i=1}^{n} \).
- Recover a parametrization of the common variable \( X \).
A toy example

Figure: Experimental setup
A toy example

Figure: Two views of the three rotating figures

Our goal: Based on these 2 signals, we would like to recover the rotation angle of the common figure (bulldog)
Organizing data – the diffusion maps way

**Algorithm 1 Diffusion maps [Coifman & Lafon, 06’]**

1. Calculate an affinity matrix $W_{ij} = \exp \left( -\frac{\|s_i - s_j\|^2}{\varepsilon} \right)$.

2. Compute diffusion operators $K_{ij} = \frac{W_{ij}}{\sum_{l=1}^{n} W_{lj}}$.

3. Compute the diffusion distance at time $m$ between each two points:

   $$d_m(i, j) = \| (K^m)_{.,i} - (K^m)_{.,j} \|_2.$$

4. Find a low dimensional embedding consistent with the diffusion distance (usually implemented through the solution of an eigenvector problem).
Multiple variables

Figure: Each variable is the rotation angle of one of the objects.
Diffusion maps

Construct a random walk kernel $K$ on the graph of samples.
Construct a sequence of probability distributions $p_{i,t} = K^t p_{i,0}$.

figure: Diffusion sequence for two variables.
How to build the embedding

We would like to find a general solution to the following diffusion equation:

\[ p_{i,t+1} = K p_{i,t} \]

It can be shown that when \( n \to \infty \) and \( \epsilon \to 0 \) [Coifman & Lafon, 06']:

\[ K \to \mathcal{I} - \mathcal{L} \]

where \( \mathcal{L} \) is a continuous Fokker-Flanck/Laplace operator.
Consider the following simple example:

\[ u_t = \mathcal{L}u = u_{xx} \]
\[ u(x, 0) = f(x), \forall x \in [0, 1] \]
\[ u(0, t) = u(1, t) = 0, \forall t > 0 \]

Analogy to the rotating figures example (single figure case):

- \( x \) - the rotation angle of the figure.
- \( t \) - the “diffusion time”.
- \( u(x, t) \) - the propagating distributions.

**Note:** we do not know or have access to \( x \). We only have access to a high dimensional mapping of the angle (the images).
How to build the embedding

A caricature of the solution I:

\[ u(x, t) = X(x)T(t) \]
\[ \frac{\dot{T}(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\lambda \]
\[ X''(x) = -\lambda X(x) \]
\[ X_k(x) = \sin(\sqrt{\lambda_k} x), \cos(\sqrt{\lambda_k} x) \]
A caricature of the solution II:

\[ u_t = \mathcal{L}u = u_{xx} \]
\[ \mathcal{L}\psi_n = \lambda_n \psi_n \]
\[ X''(x) = -\lambda X(x) \]
Diffusion maps: multiple variables

Compute the diffusion distance: $d_t(i, j) = \|p_{i,t} - p_{j,t}\|_M$, and a low dimensional embedding consistent with this distance:

![Diagram](https://via.placeholder.com/150)

**Figure**: The embedding captures the two sources of variability, but does not separate the sources of variability.
Our approach: alternating diffusion

Construct the diffusion operators $K^{(1)}$ and $K^{(2)}$ for Sensor 1 and Sensor 2, respectively. $K^{(1)}$ alone would generate the diffusion for Sensor 1, $K^{(2)}$ alone would generate the diffusion for Sensor 2.

**Figure:** Embedding of samples from Sensor 1

**Figure:** Embedding of samples from Sensor 2
Our approach: intuition
Our approach: intuition
Our approach: intuition
Alternating diffusion

Figure: Diffusion: $p_{i,0}$
Figure: Diffusion: $p_{i,1} = K^{(1)} p_{i,0}$
Alternating diffusion

Figure: Diffusion: $p_{i,2} = K^{(2)} p_{i,1}$
Alternating diffusion

Figure: Diffusion: $p_{i,3} = K^{(1)}p_{i,2}$
Alternating diffusion

Figure: Diffusion: $p_{i,4} = K^{(2)} p_{i,3}$
Alternating diffusion

Figure: Diffusion: $p_{i,5} = K^{(1)} p_{i,4}$
Alternating diffusion

Figure: Diffusion: $p_{i,6} = K^{(2)} p_{i,5}$
Alternating diffusion

Figure: diffusion sequence $p_{i,t}(x,y,z)$, projected on $X \times Y$ and $X \times Z$
Alternating diffusion

Compute the diffusion distance: \( d_t(i, j) = \| p_{i,t} - p_{j,t} \|_\pi \), and a low dimensional embedding consistent with this distance:

\[ \tilde{x}_{i,1}, \tilde{x}_{i,2} \]

**Figure:** alternating diffusion captures the geometry of the *common* variable.
Alternating diffusion embedding

Compute the diffusion distance: \( d_t(i, j) = \| p_{i,t} - p_{j,t} \|_\pi \), and a low dimensional embedding consistent with this distance:

**Figure:** alternating diffusion captures the geometry of the *common* variable and ignores the sensor-specific variables.
Algorithm 2 Alternating-diffusion

1. Calculate two affinity matrices
\[ W_{ij}^{(1)} = \exp \left( -\frac{\| s_i^{(1)} - s_j^{(1)} \|^2}{\epsilon^{(1)}} \right); \quad W_{ij}^{(2)} = \exp \left( -\frac{\| s_i^{(2)} - s_j^{(2)} \|^2}{\epsilon^{(2)}} \right). \]

2. Compute diffusion operators
\[ K_{ij}^{(1)} = \frac{W_{ij}^{(1)}}{\sum_{l=1}^{n} W_{lj}^{(1)} }; \quad K_{ij}^{(2)} = \frac{W_{ij}^{(2)}}{\sum_{l=1}^{n} W_{lj}^{(2)}}. \]

3. Compute the alternating diffusion operator
\[ K = K^{(2)} K^{(1)}. \]

4. Compute the diffusion distance at time $2m$ between each two points:
\[ d_{2m}(i, j) = \| (K^m)_{.,i} - (K^m)_{.,j} \|_2. \]

5. (Optionally:) Refine using a standard diffusion maps algorithm.
Main results

We define the effective (“marginal”) functions \( p_{i,t}^{(e)}(x) \),

\[
p_{i,t}^{(e)}(x) = \int p_{i,t}(x, y, z) \pi_{y,z|x}(y, z) dydz
\]

**Theorem 1**

*The sequence of effective functions \( p_{i,t}^{(e)}(x) \) is a diffusion sequence with an appropriate diffusion operator \( D^{(e)} \).*
Main results

The effective functions $p_{i,t}^{(e)}(x)$ cannot be measured directly. We can show that their pairwise distances can be computed from the available diffusion sequences $p_{i,t}(x, y, z)$.

**Theorem 2**

$$d_t(i, j) = \left\| p_{i,t}^{(e)}(x) - p_{j,t}^{(e)}(x) \right\|_M =$$

$$= \left\| p_{i,t}(x, y, z) - p_{j,t}(x, y, z) \right\|_\pi,$$

*with the appropriate norms $\|f(x)\|_M$ and $\|f(x, y, z)\|_\pi$.***
Our goal: Based on the 3 multimodal respiratory signals, in a purely data-driven manner, we would like to recover the underlying process related to the breathing/sleep.
Sleep stage identification

[Lederman et al., ICASSP 15’]

Figure: Common variable in respiration signals. The embedding of the airflow, chest belt, and abdominal belt signals colored by sleep stage.
Sleep stage identification

Figure: the sensitivity and the specificity of the sleep stage identification per individual.

Gray: single channels identification (light gray - airflow, dark gray - abdominal motion).
Blue: alternating diffusion identification.
Related work

- Canonical Correlation Analysis (CCA) [Hotelling, 1936]
- Kernel CCA [Lai, 2000; Bach, Jordan, 2003]
- Two-Manifold Problems [Boots, Gordon, 2012]
Conclusions

- We don’t really care about puppets...
- The problem is general:
  - The common variable is hidden nonlinearly in each sensor.
  - Multimodal sensors.
  - The sensor-specific distortions are nonlinear and cannot be suppressed simply by averaging.
  - Unknown model.
- Our solution to the problem is alternating-diffusion.
- We show that alternating-diffusion is diffusion on the common variable.
Alternating diffusion - random projection of images

**Figure:** alternating diffusion captures the geometry of the *common* variable and ignores the sensor-specific variables.
Thank you!

More information (Matlab code and toy example data):

http://roy.lederman.name/alternating-diffusion/