Tight Wavelet Frames on Multislice Graphs

Nora Leonardi, Dimitri Van de Ville

presented by Yusuf Yigit Pilavci

10/06/2020

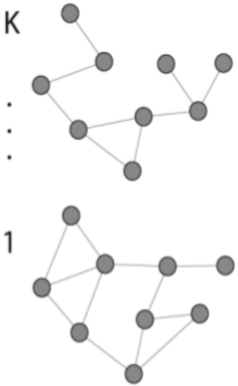
Outline

Introduction

- A collection of individual graphs with the same vertex set but different edge set.

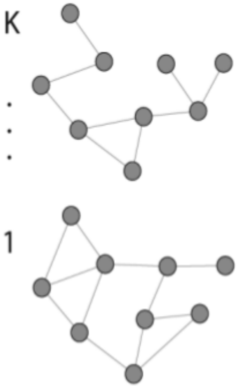
Introduction

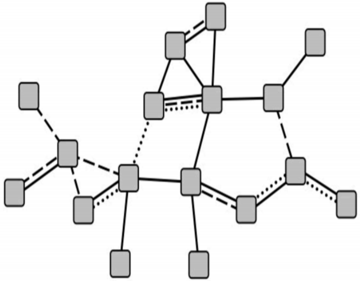
- A collection of individual graphs with the same vertex set but different edge set.
- graphs with varying interactions over time, *i.e.* Dynamic Graphs



Introduction

- A collection of individual graphs with the same vertex set but different edge set.
- graphs with varying interactions over time, *i.e.* Dynamic Graphs
- graphs with multiple type of connectivity, *i.e.* Multiplex Graphs





Taken from [SLT10]

Introduction

- Graph wavelets are powerful tools to for multi-scale analysis in static graphs.

Introduction

- Graph wavelets are powerful tools to for multi-scale analysis in static graphs.
- What about multislice graphs?

Introduction

- Graph wavelets are powerful tools to for multi-scale analysis in static graphs.
- What about multislice graphs?

Expectation

Multislice graph wavelets, which can be adapted to the varying graph topology, may give a better suited analysis than the analysis on a single slice.

Outline

A Quick Recap on SGWT

A Quick Recap on SGWT

Variable	Classical	Graph analogy
space variable	x	nodes
Fourier variable	w	λ_i
Fourier basis	e^{-jwx}	$\mathbf{u}_i$
Fourier transform	$\hat{f}(w) = \int\limits_{-\infty}^{\infty} f(x)e^{-jwx}dx$	$\hat{f}(\lambda_i) = \sum\limits_{j=1}^{N} f(j)\mathbf{u}_i(j)$
A scaled filter	$\hat{\psi}(sw)$	$g(s\lambda_i)$

A Quick Recap on SGWT

Variable	Classical	Graph analogy
space variable	x	nodes
Fourier variable	w	λ_i
Fourier basis	e^{-jwx}	u_i
Fourier transform	$\hat{f}(w) = \int\limits_{-\infty}^{\infty} f(x)e^{-jwx}dx$	$\hat{f}(\lambda_i) = \sum\limits_{j=1}^{N} f(j)\mathrm{u}_i(j)$
A scaled filter	$\hat{\psi}(sw)$	$g(s\lambda_i)$

- In classical signal processing, the wavelet $\psi_{s,a}(x)$ at scale s and location a for a given "mother wavelet" $\psi(x)$ are:

$$\psi_{s,a}(x) = \psi\left(\frac{x-a}{s}\right)$$

A Quick Recap on SGWT

Variable	Classical	Graph analogy
space variable	x	nodes
Fourier variable	w	λ_i
Fourier basis	e^{-jwx}	u_i
Fourier transform	$\hat{f}(w) = \int\limits_{-\infty}^{\infty} f(x)e^{-jwx}dx$	$\hat{f}(\lambda_i) = \sum\limits_{j=1}^{N} f(j)\mathrm{u}_i(j)$
A scaled filter	$\hat{\psi}(sw)$	$g(s\lambda_i)$

- In classical signal processing, the wavelet $\psi_{s,a}(x)$ at scale s and location a for a given "mother wavelet" $\psi(x)$ are:

$$\psi_{s,a}(x) = \psi\left(\frac{x-a}{s}\right) \iff \psi_{s,a}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega x}\hat{\psi}(sw)e^{-j\omega a}d\omega$$

- Then, the graph analogous of $\psi_{s,a}(x)$ is:

$$\psi_{s,a}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega x}\hat{\psi}(sw)e^{-j\omega a}d\omega \rightarrow \psi_{s,a}(i) = \sum_{j=1}^{N} \mathrm{u}_j(i)g(s\lambda_j)\mathrm{u}_j(a)$$

- Then, the graph analogous of $\psi_{s,a}(x)$ is:

$$\psi_{s,a}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega x}\hat{\psi}(sw)e^{-j\omega a}d\omega \rightarrow \psi_{s,a}(i) = \sum_{j=1}^{N} \mathrm{u}_j(i)g(s\lambda_j)\mathrm{u}_j(a)$$

- In a matrix-vector product:

$$\psi_{s,a} = \mathrm{U}^T g(s\Lambda)\mathrm{U}\delta_a$$

- Then, the graph analogous of $\psi_{s,a}(x)$ is:

$$\psi_{s,a}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega x}\hat{\psi}(sw)e^{-j\omega a}d\omega \rightarrow \psi_{s,a}(i) = \sum_{j=1}^{N} \mathrm{u}_j(i)g(s\lambda_j)\mathrm{u}_j(a)$$

- In a matrix-vector product:

$$\psi_{s,a} = \mathrm{U}^T g(s\Lambda)\mathrm{U}\delta_a$$

- One needs the scaling function $h(\lambda_i)$ to capture the residual low-pass components:

$$\phi_a = \mathrm{U}^T h(\Lambda)\mathrm{U}\delta_a$$

- Then, the graph analogous of $\psi_{s,a}(x)$ is:

$$\psi_{s,a}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-j\omega x}\hat{\psi}(sw)e^{-j\omega a}d\omega \to \psi_{s,a}(i) = \sum_{j=1}^{N} \mathrm{u}_j(i)g(s\lambda_j)\mathrm{u}_j(a)$$

- In a matrix-vector product:

$$\psi_{s,a} = \mathrm{U}^T g(s\Lambda)\mathrm{U}\delta_a$$

- One needs the scaling function $h(\lambda_i)$ to capture the residual low-pass components:

$$\phi_a = \mathrm{U}^T h(\Lambda)\mathrm{U}\delta_a$$

- If we put them together, we obtain a set of vectors:

$$F = \{\phi_a\}_{a=1}^{n} \cup \{\psi_{a,s_j}\}_{a=1,j=1}^{n,J}$$

Tight Wavelet Frames

- F forms a frame of $l_2(\mathcal{V})$, if there exists frame bounds $A, B > 0$ such that

$$\forall \mathsf{v} \in l_2(\mathcal{V}), \quad A||f||^2 \leq \sum_{\mathsf{f} \in F} ||\langle \mathsf{f}, \mathsf{v} \rangle||^2 \leq B||f||^2$$

Tight Wavelet Frames

- F forms a frame of $l_2(\mathcal{V})$, if there exists frame bounds $A, B > 0$ such that

$$\forall \mathrm{v} \in l_2(\mathcal{V}), \quad A||f||^2 \leq \sum_{\mathrm{f} \in F} ||\langle \mathrm{f}, \mathrm{v} \rangle||^2 \leq B||f||^2$$

- Hammond *et.al.*provide:

$$A = \min_{\lambda \in [0, \lambda_N]} G(\lambda)$$
$$B = \max_{\lambda \in [0, \lambda_N]} G(\lambda)$$

where $G(\lambda) = h^2(\lambda) + \sum_{j=1}^{J} g^2(s_j \lambda)$.

Tight Wavelet Frames

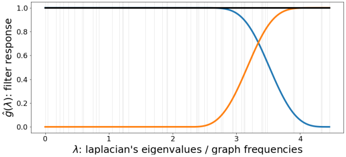
- A tight frame satisfies $A = B = G(\lambda)$ for all λ's.

Tight Wavelet Frames

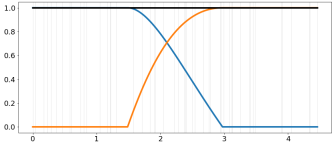
- A tight frame satisfies $A = B = G(\lambda)$ for all λ's.
- A Parseval frame is a normalized tight frame with $G(\lambda) = A = B = 1$ for all λ's.

Tight Wavelet Frames

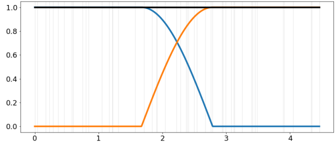
- A tight frame satisfies $A = B = G(\lambda)$ for all λ's.
- A Parseval frame is a normalized tight frame with $G(\lambda) = A = B = 1$ for all λ's.
- Various Parseval wavelet frames have been adapted to graphs:



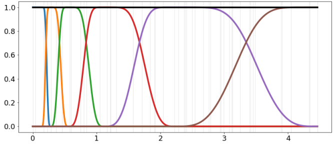
Meyer



Simoncelli



Papadakis



Meyer with more scales

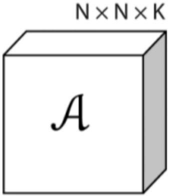
Outline

Tensors and Multislice Graphs

- A tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots I_d}$ is an algebraic object that can interpret d dimensional data.

Tensors and Multislice Graphs

- A tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots I_d}$ is an algebraic object that can interpret d dimensional data.
- A three dimensional tensor $\mathcal{A} \in \mathbb{R}^{N \times N \times K}$ can interpret a multislice graph.



- Each frontal slice $A_{::k}$ is an adjacency matrix.

Tensor Operations

Tensor Operations

- Multiplication between a 3D tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ and a matrix $\mathrm{B} \in \mathbb{R}^{M \times I}$ is defined as:

$$(\mathcal{A} \times_1 \mathrm{B})_{jkm} = \sum_{i=1}^{I} \mathcal{A}_{ijk} \mathrm{B}_{mi}$$

Tensor Operations

- Multiplication between a 3D tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ and a matrix $\mathrm{B} \in \mathbb{R}^{M \times I}$ is defined as:

$$(\mathcal{A} \times_1 \mathrm{B})_{jkm} = \sum_{i=1}^{I} \mathcal{A}_{ijk} \mathrm{B}_{mi}$$

- Matricization or Unfolding $\mathrm{A}_{(n)}$ is a reordered concatenation of the slices in the dimension n:

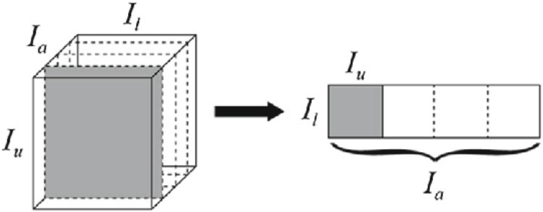


Figure: *Taken from [STK+15]*

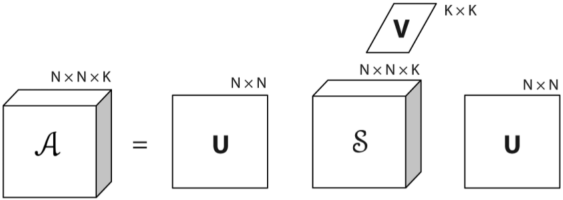
Tensor Analysis

Proposition

A tensor $\mathcal{A} \in \mathbb{R}^{N\times N\times K}$ *with* $\mathrm{A}_{::k} = \mathrm{A}_{::k}^T$ *for all* $k = 1, \dots, K$ *has the the higher order singular value decomposition (HOSVD) decomposition:*

$$\mathcal{A} = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \times_3 \mathrm{V}$$

with $\mathcal{S} \in \mathbb{R}^{N\times N\times K}$ *and* $\mathrm{U} \in \mathbb{R}^{N\times N}$ *and* $\mathrm{V} \in \mathbb{R}^{K\times K}$ *are orthonormal matrices i.e.* $\mathrm{V}^T\mathrm{V} = \mathrm{I}$, $\mathrm{U}^T\mathrm{U} = \mathrm{I}$.



Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathsf{U} \times_2 \mathsf{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathsf{V}^T$$

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathsf{U} \times_2 \mathsf{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathsf{V}^T$$

- Each slice $\mathsf{S}'_{::k} = \mathsf{U}\mathsf{S}_{::k}\mathsf{U}^T$, called as eigennetwork, are

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathrm{V}^T$$

- Each slice $\mathrm{S}'_{::k} = \mathrm{U}\mathrm{S}_{::k}\mathrm{U}^T$, called as eigennetwork, are
  - Symmetric *i.e.* $(\mathrm{S}'_{::k})^T = \mathrm{S}'_{::k}$

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathrm{V}^T$$

- Each slice $\mathrm{S}'_{::k} = \mathrm{U}\mathrm{S}_{::k}\mathrm{U}^T$, called as eigennetwork, are
  - Symmetric *i.e.* $(\mathrm{S}'_{::k})^T = \mathrm{S}'_{::k}$
  - Orthogonal in the sense of scalar product *i.e.* $\langle \mathrm{S}'_{::a}, \mathrm{S}'_{::b} \rangle = 0$ for $i \neq j$.

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathrm{V}^T$$

- Each slice $\mathrm{S}'_{::k} = \mathrm{U}\mathrm{S}_{::k}\mathrm{U}^T$, called as eigennetwork, are
  - Symmetric *i.e.* $(\mathrm{S}'_{::k})^T = \mathrm{S}'_{::k}$
  - Orthogonal in the sense of scalar product *i.e.* $\langle \mathrm{S}'_{::a}, \mathrm{S}'_{::b} \rangle = 0$ for $i \neq j$.

What do they mean?

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathrm{V}^T$$

- Each slice $\mathrm{S}'_{::k} = \mathrm{U}\mathrm{S}_{::k}\mathrm{U}^T$, called as eigennetwork, are
  - Symmetric *i.e.* $(\mathrm{S}'_{::k})^T = \mathrm{S}'_{::k}$
  - Orthogonal in the sense of scalar product *i.e.* $\langle \mathrm{S}'_{::a}, \mathrm{S}'_{::b} \rangle = 0$ for $i \neq j$.

What do they mean?

- The first eigennetwork $\mathrm{S}'_{::1}$ turns to be the average of all adjacency matrices.

Eigennetworks

- An *eigennetwork* tensor $\mathcal{S}' \in \mathbb{R}^{N \times N \times K}$ is defined as:

$$\mathcal{S}' = \mathcal{S} \times_1 \mathrm{U} \times_2 \mathrm{U} \text{ or } \mathcal{S}' = \mathcal{A} \times_3 \mathrm{V}^T$$

- Each slice $\mathrm{S}'_{::k} = \mathrm{U}\mathrm{S}_{::k}\mathrm{U}^T$, called as eigennetwork, are
  - Symmetric *i.e.* $(\mathrm{S}'_{::k})^T = \mathrm{S}'_{::k}$
  - Orthogonal in the sense of scalar product *i.e.* $\langle \mathrm{S}'_{::a}, \mathrm{S}'_{::b} \rangle = 0$ for $i \neq j$.

What do they mean?

- The first eigennetwork $\mathrm{S}'_{::1}$ turns to be the average of all adjacency matrices.
- Each eigennetwork captures a component of the variation in the edge weights across the networks.

How to compute?

- A quick calculation shows:

$$\mathcal{S}' = \mathcal{A} \times_3 \mathsf{V}^T \iff \mathsf{S}'_{::k} = \sum_{t=1}^{T} \mathsf{V}_{tk} \mathsf{A}_{::t}$$

How to compute?

- A quick calculation shows:

$$\mathcal{S}' = \mathcal{A} \times_3 \mathsf{V}^T \iff \mathsf{S}'_{::k} = \sum_{t=1}^{T} \mathsf{V}_{tk}\mathsf{A}_{::t}$$

- With the unfolding in third dimension $\mathsf{A}_{(3)} = \mathsf{V}\Sigma\mathsf{W}^T$, for $K << N$,one can efficiently compute V by decomposing $\mathsf{A}_{(3)}\mathsf{A}_{(3)}{}^T = \mathsf{V}\Sigma^2\mathsf{V}^T$.

Outline

SGWT meets with Multislice Graphs

- Each eigennetwork captures a component of the variation.

SGWT meets with Multislice Graphs

- Each eigennetwork captures a component of the variation.
- Also, by definition, one has

$$\mathsf{A}_{::k} = \sum_{t=1}^{K} \mathsf{V}_{kt} \mathsf{S}'_{::t}$$

SGWT meets with Multislice Graphs

- Each eigennetwork captures a component of the variation.
- Also, by definition, one has

$$\mathsf{A}_{::k} = \sum_{t=1}^{K} \mathsf{V}_{kt} \mathsf{S}'_{::t}$$

- A new graph can be obtained by combining eigennetworks:

$$\mathsf{A}' = \sum_{t=1}^{K} \alpha_t \mathsf{S}'_{::t} \quad \text{s.t. } \mathsf{A}' \geq 0$$

SGWT meets with Multislice Graphs

- Each eigennetwork captures a component of the variation.
- Also, by definition, one has

$$\mathsf{A}_{::k} = \sum_{t=1}^{K} \mathsf{V}_{kt}\mathsf{S}'_{::t}$$

- A new graph can be obtained by combining eigennetworks:

$$\mathsf{A}' = \sum_{t=1}^{K} \alpha_t \mathsf{S}'_{::t} \quad \text{s.t. } \mathsf{A}' \geq 0$$

- Depending on α_t's and $\mathsf{S}_{::t}$, the edge weights associated to *relevant* variation components are emphasized in the new network.

SGWT meets with Multislice Graphs

- Each eigennetwork captures a component of the variation.
- Also, by definition, one has

$$\mathsf{A}_{::k} = \sum_{t=1}^{K} \mathsf{V}_{kt}\mathsf{S}'_{::t}$$

- A new graph can be obtained by combining eigennetworks:

$$\mathsf{A}' = \sum_{t=1}^{K} \alpha_t \mathsf{S}'_{::t} \quad \text{s.t. } \mathsf{A}' \geq 0$$

- Depending on α_t's and $\mathsf{S}_{::t}$, the edge weights associated to *relevant* variation components are emphasized in the new network.
- From the new graph Laplacian $\mathsf{L}' = \mathsf{D}' - \mathsf{A}'$,one has a new SGWT frame on it.

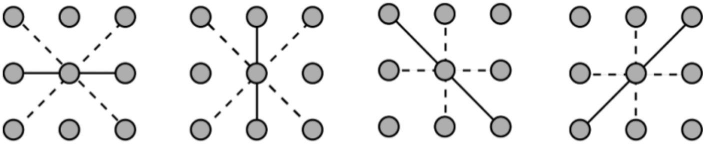
Outline

Experiments

- Two illustrations for this framework is given:
  - Multiplex Grid Graph on Images
  - Dynamic Brain Graphs

Experiments

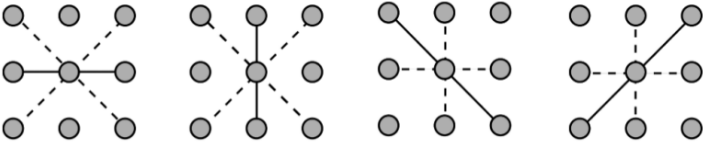
- Two illustrations for this framework is given:
  - Multiplex Grid Graph on Images
  - Dynamic Brain Graphs
- In the first one, we have four underlying 2D grid graphs for an image:



(a) (b) (c) (d)

Experiments

- Two illustrations for this framework is given:
  - Multiplex Grid Graph on Images
  - Dynamic Brain Graphs
- In the first one, we have four underlying 2D grid graphs for an image:



(a) (b) (c) (d)

- The resultant decomposition gives:

$$V = \begin{bmatrix} -0.47 & 0.73 & 0 & 0.59 \\ -0.49 & -0.69 & 0 & 0.54 \\ -0.52 & -0.01 & 0.71 & -0.47 \\ -0.52 & -0.01 & -0.71 & -0.49 \end{bmatrix} \text{ with } S'_{::k} = \sum_{t=1}^{4} V_{tk} A_{::t} \qquad (1)$$

Experiments

- A network set is created from these eigennetworks and a localized filter illustrated on them:

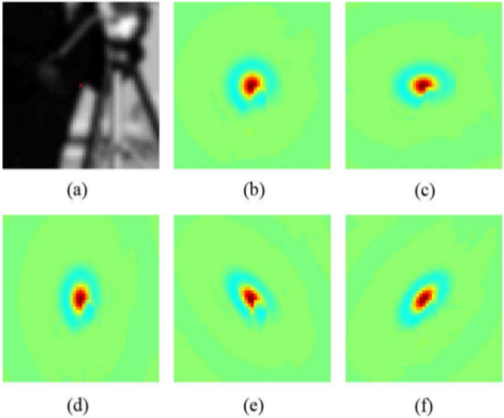


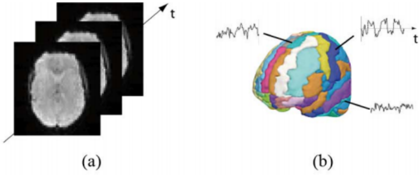
Figure: *(a)filter location,(b)* $A'_1 = -0.5S'_{::1}$, *(c)*$A'_2 = -0.5S'_{::1} + 0.7S'_{::2}$,*(d)*$A'_3 = -0.5S'_{::1} - 0.7S'_{::2}$,*(e)*$A'_4 = -0.5S'_{::1} + 0.8S'_{::3}$,*(f)*$A'_5 = -0.5S'_{::1} - 0.8S'_{::3}$

Experiments: Dynamic Brain Graphs

- In the experiments, 15 healthy subjects were periodically shown a short movie excerpt followed by a *resting* period.

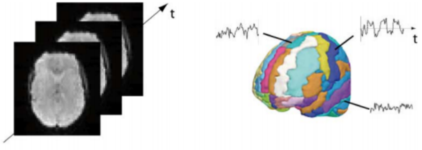
Experiments: Dynamic Brain Graphs

- In the experiments, 15 healthy subjects were periodically shown a short movie excerpt followed by a *resting* period.
- The collected fMRI data is transformed to the regional mean activity and averaged across subjects.



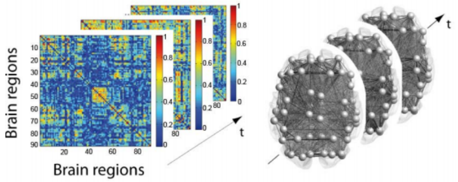
Experiments: Dynamic Brain Graphs

- In the experiments, 15 healthy subjects were periodically shown a short movie excerpt followed by a *resting* period.
- The collected fMRI data is transformed to the regional mean activity and averaged across subjects.



(a) (b)

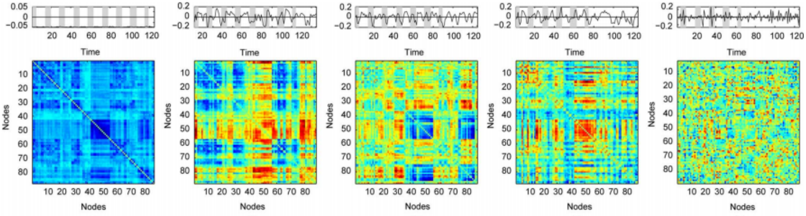
- By a sliding window approach, the correlations of different regions are computed and used as edge weights.



(c) (d)

Experiments:Dynamic Brain Graph

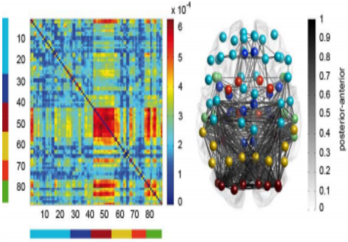
- The eigennetworks $S'_{::k}$'s and v_k's:

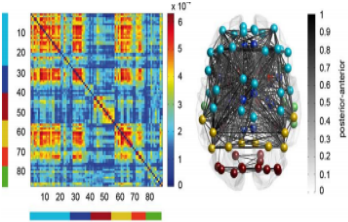


(a) (b) (c) (d) (e)

Experiments: Dynamic Brain Graph

- Two adjacency matrices $A'_1 = -0.1S'_1 + 0.2S'_{::2}$ and $A'_2 = -0.1S'_1 - 0.3S'_{::2}$ are generated:





Experiments:Dynamic Brain Graph

- In the end, two SGWT transforms are obtained.

Experiments:Dynamic Brain Graph

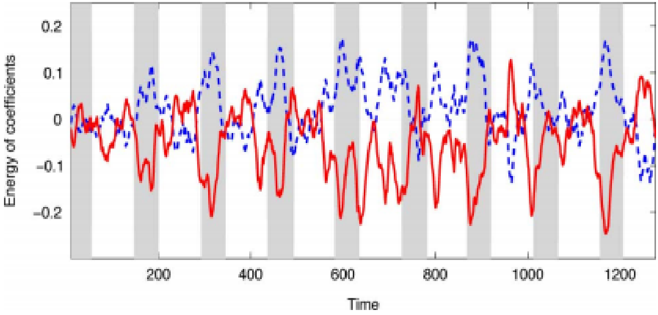
- In the end, two SGWT transforms are obtained.
- They are applied to the regional activity signal.

Experiments:Dynamic Brain Graph

- In the end, two SGWT transforms are obtained.
- They are applied to the regional activity signal.
- The energy of scaling and wavelet coefficients are computed in both frame.

Experiments:Dynamic Brain Graph

- In the end, two SGWT transforms are obtained.
- They are applied to the regional activity signal.
- The energy of scaling and wavelet coefficients are computed in both frame.
- Finally, the difference is plotted:



Outline

Conclusion

Conclusion

- An extension of SGWT on multislice graphs is presented.

Conclusion

- An extension of SGWT on multislice graphs is presented.
- This extension allows us to capture the variation across the graphs.

Conclusion

- An extension of SGWT on multislice graphs is presented.
- This extension allows us to capture the variation across the graphs.
- It can be used for different GSP tools.

Michael Szell, Renaud Lambiotte, and Stefan Thurner, *Multirelational organization of large-scale social networks in an online world*, Proceedings of the National Academy of Sciences **107** (2010), no. 31, 13636–13641.

Masoud Sattari, Ismail Hakki Toroslu, Pinar Karagoz, Panagiotis Symeonidis, and Yannis Manolopoulos, *Extended feature combination model for recommendations in location-based mobile services*, Knowledge and Information Systems **44** (2015), no. 3, 629–661.