Matrix Data Analysis using Hierarchical Co-Clustering and Multiscale Basis Dictionaries on Graphs

Jeff Irion & Naoki Saito

Department of Mathematics
University of California, Davis

Workshop on Harmonic Analysis, Graphs and Learning
Hausdorff Research Institute for Mathematics, Bonn, Germany
March 14, 2016

Outline

Acknowledgment

- Support from Office of Naval Research grants: N00014-12-1-0177; N00014-16-1-2255
- Support from National Science Foundation grant: DMS-1418779
- Support for Jeff Irion from National Defense Science and Engineering Graduate Fellowship, 32 CFR 168a via AFOSR FA9550-11-C-0028
- The Science News dataset provided by Jeff Solka (George Mason Univ.) via Raphy Coifman (Yale) and Matan Gavish (Hebrew Univ.)

Outline

Motivations

Many modern data analysis tasks often involve large matrix-form datasets:

- Spatiotemporal data measured by sensor networks
  - Columns → sensors
  - Rows → time indices
  - a_{ij} → sensor j's temperature reading at the ith time sample
- Ratings/Reviews
  - Columns → movies
  - Rows → Netflix users
  - a_{ij} → user i's rating of movie j on a 1-5 scale
- Term-document databases
  - Columns → documents, articles
  - Rows → words, terms
  - a_{ij} → the relative frequency of occurrences of word i in document j

By utilizing graph-based techniques, we can discover and exploit underlying (often hidden) dependency and geometric structure in the data for a variety of tasks, e.g., compression, classification, regression, ...

Motivations . . .

A big difference between those datasets from usual images/photos.

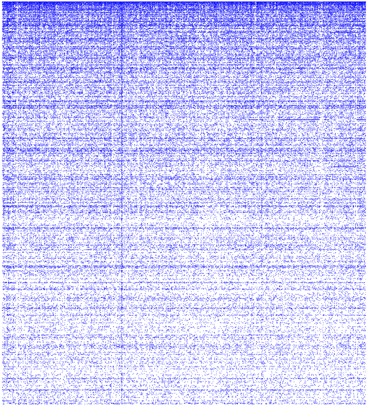


Figure: Science News database (1153 words × 1042 documents)

Motivations . . .

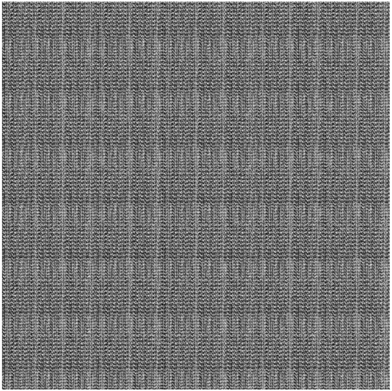
They are often more like shuffled and permuted images, i.e., possess no spatial smoothness or coherency in general:

(a) The original Barbara image

Motivations . . .

They are often more like shuffled and permuted images, i.e., possess no spatial smoothness or coherency in general:

(a) The original Barbara image



(b) The shuffled Barbara image

Outline

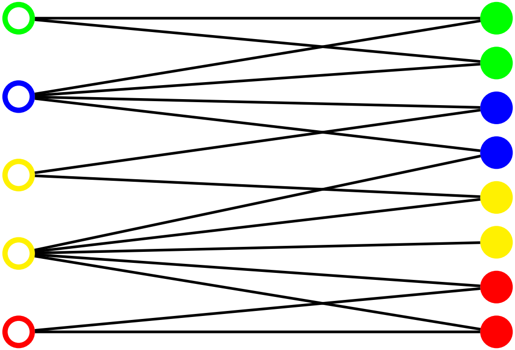
Spectral Co-Clustering (Dhillon, 2001)[1]

- Given a matrix $A \in \mathbb{R}_{\geq 0}^{N_r \times N_c}$ (e.g., a term-document matrix), the rows and columns are viewed as the two sets of nodes in a *bipartite* graph.
- a_{ij} denotes the edge weight between the ith row and the jth column.

[1] I. S. Dhillon: "Co-clustering documents and words using Bipartite Spectral Graph Partitioning," *Proc. 7th ACM SIGKDD*, pp. 269–274, 2001.

Spectral Co-Clustering (Dhillon, 2001)[1]

- Given a matrix $A \in \mathbb{R}_{\geq 0}^{N_r \times N_c}$ (e.g., a term-document matrix), the rows and columns are viewed as the two sets of nodes in a *bipartite* graph.
- a_{ij} denotes the edge weight between the ith row and the jth column.



[1] I. S. Dhillon: "Co-clustering documents and words using Bipartite Spectral Graph Partitioning," *Proc. 7th ACM SIGKDD*, pp. 269–274, 2001.

Spectral Co-Clustering ...

- Then, matrices associated with this bipartite graph can be written as:

$$W = \begin{bmatrix} O & A \\ A^{\mathsf{T}} & O \end{bmatrix}$$ weighted adjacency matrix

$$D = \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix} \quad \begin{aligned} D_r &:= \mathrm{diag}(A\mathbf{1}) \\ D_c &:= \mathrm{diag}(A^{\mathsf{T}}\mathbf{1}) \end{aligned}$$ degree matrix

$$L := D - W = \begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix}$$ (unnormalized) graph Laplacian

$$L_{\mathrm{rw}} := D^{-1}L = I - D^{-1}W$$ random-walk normalized graph Laplacian

Spectral Clustering of General Graphs

- Both L and L_{rw} are positive semidefinite and if the graph is *connected*, the smallest eigenvalue is 0 and the corresponding eigenvector $\boldsymbol{\phi}_0 \propto \mathbf{1}$.
- For graph partitioning and clustering, it is often useful to embed the nodes in the low dimensional Euclidean space formed by a few eigenvectors corresponding to the smallest positive eigenvalues.
- The eigenvectors of L are orthonormal in the usual sense while those of L_{rw} are orthonormal relative to $D^{1/2}$, i.e., $\boldsymbol{\phi}_k^{\mathsf{T}} D \boldsymbol{\phi}_l = \delta_{kl}$.
- Yet, the eigenvectors of L_{rw} are preferable to those of L for the purpose of graph partitioning and clustering because the former better reflects the influence of nodes via the weights w_{ij} than the latter[2]
- More precisely, . . .

[2] See, e.g., U. von Luxburg: "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in the *Fiedler vector*

Why? Using $\boldsymbol{\phi}_1$ of L to generate X and X^c yields an *approximate* minimizer of the RatioCut function[3]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|}, \quad \text{where } \mathrm{cut}(X, X^c) := \sum_{\substack{i \in X \\ j \in X^c}} w_{ij}$$

On the other hand, $\boldsymbol{\phi}_1$ of L_{rw} to cut a graph, which yield an *approximate* minimizer of the *Normalized Cut* (or *NCut*) function of Shi and Malik[4]:

$$\mathrm{NCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X)} + \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X^c)}, \quad \text{where } \mathrm{vol}(X) := \sum_{i \in X} d_i$$

---

[3] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[4] J. Shi & J. Malik: "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in the *Fiedler vector*

Why? Using $\boldsymbol{\phi}_1$ of L to generate X and X^c yields an *approximate* minimizer of the RatioCut function[3]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|}, \quad \text{where } \mathrm{cut}(X, X^c) := \sum_{\substack{i \in X \\ j \in X^c}} w_{ij}$$

On the other hand, $\boldsymbol{\phi}_1$ of L_{rw} to cut a graph, which yield an *approximate* minimizer of the *Normalized Cut* (or *NCut*) function of Shi and Malik[4]:

$$\mathrm{NCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X)} + \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X^c)}, \quad \text{where } \mathrm{vol}(X) := \sum_{i \in X} d_i$$

---

[3] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[4] J. Shi & J. Malik: "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in the *Fiedler vector*

Why? Using $\boldsymbol{\phi}_1$ of L to generate X and X^c yields an *approximate* minimizer of the RatioCut function[3]:

$$\mathrm{RatioCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{|X|} + \frac{\mathrm{cut}(X, X^c)}{|X^c|}, \quad \text{where } \mathrm{cut}(X, X^c) := \sum_{\substack{i \in X \\ j \in X^c}} w_{ij}$$

On the other hand, $\boldsymbol{\phi}_1$ of L_{rw} to cut a graph, which yield an *approximate* minimizer of the *Normalized Cut* (or *NCut*) function of Shi and Malik[4]:

$$\mathrm{NCut}(X, X^c) := \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X)} + \frac{\mathrm{cut}(X, X^c)}{\mathrm{vol}(X^c)}, \quad \text{where } \mathrm{vol}(X) := \sum_{i \in X} d_i$$

[3] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[4] J. Shi & J. Malik: "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

Graph Partitioning via Spectral Clustering

Goal: Split the vertices V into two "good" subsets, X and X^c

Plan: Use the signs of the entries in the *Fiedler vector*

Why? Using $\boldsymbol{\phi}_1$ of L to generate X and X^c yields an *approximate* minimizer of the RatioCut function[3]:

$$\mathrm{RatioCut}(X,X^c) := \frac{\mathrm{cut}(X,X^c)}{|X|} + \frac{\mathrm{cut}(X,X^c)}{|X^c|}, \quad \text{where } \mathrm{cut}(X,X^c) := \sum_{\substack{i\in X\\ j\in X^c}} w_{ij}$$

On the other hand, $\boldsymbol{\phi}_1$ of L_{rw} to cut a graph, which yield an *approximate* minimizer of the *Normalized Cut* (or *NCut*) function of Shi and Malik[4]:

$$\mathrm{NCut}(X,X^c) := \frac{\mathrm{cut}(X,X^c)}{\mathrm{vol}(X)} + \frac{\mathrm{cut}(X,X^c)}{\mathrm{vol}(X^c)}, \quad \text{where } \mathrm{vol}(X) := \sum_{i\in X} d_i$$

---

[3] L. Hagen and A. B. Kahng: "New spectral methods for ratio cut partitioning and clustering," *IEEE Trans. Comput.-Aided Des.*, vol. 11, no. 9, pp. 1074-1085, 1992.

[4] J. Shi & J. Malik: "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

Graph Partitioning via Spectral Clustering . . .

The practice of using the Fiedler vector to partition a graph is supported by the following theory.

Definition (Weak Nodal Domain)

A positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.

Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\phi_1) = 2$.

Graph Partitioning via Spectral Clustering . . .

The practice of using the Fiedler vector to partition a graph is supported by the following theory.

Definition (Weak Nodal Domain)

A positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.

Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\phi_1) = 2$.

Graph Partitioning via Spectral Clustering ...

The practice of using the Fiedler vector to partition a graph is supported by the following theory.

Definition (Weak Nodal Domain)

A positive (or negative) weak nodal domain of f on $V(G)$ is a maximal connected induced subgraph of G on vertices $v \in V$ with $f(v) \geq 0$ (or $f(v) \leq 0$) that contains at least one nonzero vertex. The number of weak nodal domains of f is denoted by $\mathfrak{W}(f)$.

Corollary (Fiedler (1975))

If G is connected, then $\mathfrak{W}(\phi_1) = 2$.

Spectral Co-Clustering (Dhillon, 2001) ...

- Recall the matrices associated with a bipartite graph given $A \in \mathbb{R}_{\geq 0}^{N_r \times N_c}$:

$$W = \begin{bmatrix} O & A \\ A^{\mathsf{T}} & O \end{bmatrix}; D = \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix}; L := D - W = \begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix}; L_{\mathrm{rw}} := I - D^{-1} W$$

- Since $L_{\mathrm{rw}} \boldsymbol{\phi} = \lambda \boldsymbol{\phi} \Leftrightarrow L\boldsymbol{\phi} = \lambda D \boldsymbol{\phi}$, we have

$$\begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_r \\ \boldsymbol{\phi}_c \end{bmatrix} = \lambda \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_r \\ \boldsymbol{\phi}_c \end{bmatrix} \Leftrightarrow \begin{cases} A\boldsymbol{\phi}_c = (1-\lambda) D_r \boldsymbol{\phi}_r \\ A^{\mathsf{T}} \boldsymbol{\phi}_r = (1-\lambda) D_c \boldsymbol{\phi}_c \end{cases}$$

Then, setting $\boldsymbol{u} := D_r^{1/2} \boldsymbol{\phi}_r$, $\boldsymbol{v} := D_c^{1/2} \boldsymbol{\phi}_c$, we get

$$D_r^{-1/2} A D_c^{-1/2} \boldsymbol{v} = (1-\lambda) \boldsymbol{u}; \quad D_c^{-1/2} A^{\mathsf{T}} D_r^{-1/2} \boldsymbol{u} = (1-\lambda) \boldsymbol{v},$$

which precisely defines the *SVD* of $\tilde{A} := D_r^{-1/2} A D_c^{-1/2} \in \mathbb{R}^{N_r \times N_c}$; no need to compute the eigenvectors of $L, L_{\mathrm{rw}} \in \mathbb{R}^{(N_r+N_c)\times(N_r+N_c)}$

Spectral Co-Clustering (Dhillon, 2001) ...

- Recall the matrices associated with a bipartite graph given $A \in \mathbb{R}_{\geq 0}^{N_r \times N_c}$:

$$W = \begin{bmatrix} O & A \\ A^{\mathsf{T}} & O \end{bmatrix}; D = \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix}; L := D - W = \begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix}; L_{\mathrm{rw}} := I - D^{-1} W$$

- Since $L_{\mathrm{rw}} \boldsymbol{\phi} = \lambda \boldsymbol{\phi} \Leftrightarrow L \boldsymbol{\phi} = \lambda D \boldsymbol{\phi}$, we have

$$\begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_r \\ \boldsymbol{\phi}_c \end{bmatrix} = \lambda \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_r \\ \boldsymbol{\phi}_c \end{bmatrix} \Leftrightarrow \begin{cases} A \boldsymbol{\phi}_c = (1-\lambda) D_r \boldsymbol{\phi}_r \\ A^{\mathsf{T}} \boldsymbol{\phi}_r = (1-\lambda) D_c \boldsymbol{\phi}_c \end{cases}$$

Then, setting $\boldsymbol{u} := D_r^{1/2} \boldsymbol{\phi}_r$, $\boldsymbol{v} := D_c^{1/2} \boldsymbol{\phi}_c$, we get

$$D_r^{-1/2} A D_c^{-1/2} \boldsymbol{v} = (1-\lambda) \boldsymbol{u}; \quad D_c^{-1/2} A^{\mathsf{T}} D_r^{-1/2} \boldsymbol{u} = (1-\lambda) \boldsymbol{v},$$

which precisely defines the *SVD* of $\tilde{A} := D_r^{-1/2} A D_c^{-1/2} \in \mathbb{R}^{N_r \times N_c}$; no need to compute the eigenvectors of $L, L_{\mathrm{rw}} \in \mathbb{R}^{(N_r+N_c) \times (N_r+N_c)}$

Spectral Co-Clustering (Dhillon, 2001) ...

- Hence, the Fiedler vector of L_{rw} bipartitions the bipartite graph:

$$\boldsymbol{\phi}_1 = \begin{bmatrix} D_r^{-1/2}\boldsymbol{u}_1 \\ D_c^{-1/2}\boldsymbol{v}_1 \end{bmatrix},$$

  where $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the second left and right singular vectors of $\tilde{A} = D_r^{-1/2} A D_c^{-1/2}$.
- The rows and the columns are partitioned *simultaneously*.
- This also allows the analysis of rows and columns *on an equal footing*, i.e., we can see not only which columns are similar but also which rows are closely related to a specific group of columns, etc.

Spectral Co-Clustering (Dhillon, 2001) . . .

- Hence, the Fiedler vector of L_{rw} bipartitions the bipartite graph:

$$\boldsymbol{\phi}_1 = \begin{bmatrix} D_r^{-1/2}\boldsymbol{u}_1 \\ D_c^{-1/2}\boldsymbol{v}_1 \end{bmatrix},$$

  where $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the second left and right singular vectors of $\tilde{A} = D_r^{-1/2} A D_c^{-1/2}$.
- The rows and the columns are partitioned *simultaneously*.
- This also allows the analysis of rows and columns *on an equal footing*, i.e., we can see not only which columns are similar but also which rows are closely related to a specific group of columns, etc.

Spectral Co-Clustering (Dhillon, 2001) ...

- Hence, the Fiedler vector of L_{rw} bipartitions the bipartite graph:

$$\boldsymbol{\phi}_1 = \begin{bmatrix} D_r^{-1/2}\boldsymbol{u}_1 \\ D_c^{-1/2}\boldsymbol{v}_1 \end{bmatrix},$$

  where $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are the second left and right singular vectors of $\tilde{A} = D_r^{-1/2} A D_c^{-1/2}$.
- The rows and the columns are partitioned *simultaneously*.
- This also allows the analysis of rows and columns *on an equal footing*, i.e., we can see not only which columns are similar but also which rows are closely related to a specific group of columns, etc.

An Example: Science News Dataset

Dataset: the Science News database (1153×1042)

- Rows → preselected words
- Columns → articles from 8 fields: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math & CS; Medicine; Physics
- a_{ij} → the relative frequency of word i appears in article j ⇒ all column sums are 1

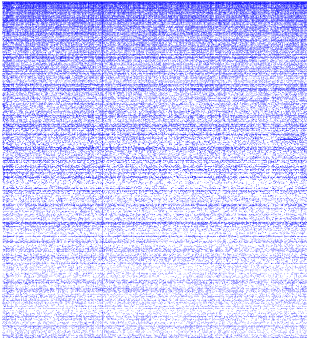


Figure: Science News database (original order)

An Example: Science News Dataset . . .

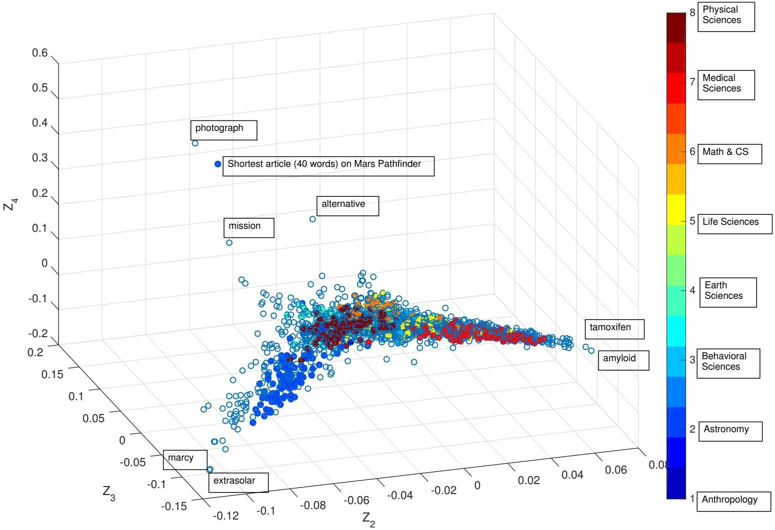


Figure: Words and articles embedded in $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3\}$ at the top level.

Outline

Generalized Haar-Walsh Transform (GHWT)

The *Generalized Haar-Walsh Transform* (GHWT) is a true generalization of the classical Haar-Walsh Wavelet Packet Transform, and it generates a *dictionary* (i.e., a redundant set) of basis vectors that are *piecewise-constant on their support.*

The algorithm using the Fiedler vectors can be summarized as follows although any other graph partitioning algorithm can be used . . .

1. Generate a full recursive bipartitioning of the graph using *Fiedler vectors* $\boldsymbol{\phi}^j_{k,1}$ of $L_{\mathrm{rw}}(G^j_k)$, where $k = 0, \ldots, K^j - 1$ indicates a region, $j = 0, \ldots, j_{\max}$ indicates a level (or scale), $V = V^0_0 = V^1_0 \cup V^1_1 = \cdots$
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max} - 1 \Rightarrow$ ***scaling*** and *Haar* vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j - 1 \Rightarrow$ ***scaling***, *Haar*, and *Walsh* vectors

1. Generate a full recursive bipartitioning of the graph using *Fiedler vectors* $\boldsymbol{\phi}^{j}_{k,1}$ of $L_{\mathrm{rw}}(G^{j}_{k})$, where $k = 0, \ldots, K^{j} - 1$ indicates a region, $j = 0, \ldots, j_{\max}$ indicates a level (or scale), $V = V^{0}_{0} = V^{1}_{0} \cup V^{1}_{1} = \cdots$
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max} - 1 \Rightarrow$ *scaling* and *Haar* vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j - 1 \Rightarrow$ *scaling*, *Haar*, and *Walsh* vectors

$$\left[\,\boldsymbol{\psi}^{j_{\max}}_{0,0}\,\right] \quad \left[\,\boldsymbol{\psi}^{j_{\max}}_{1,0}\,\right] \quad \left[\,\boldsymbol{\psi}^{j_{\max}}_{2,0}\,\right] \quad \left[\,\boldsymbol{\psi}^{j_{\max}}_{3,0}\,\right] \cdots \left[\,\boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-2,0}\,\right] \quad \left[\,\boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-1,0}\,\right]$$

1. Generate a full recursive bipartitioning of the graph using *Fiedler vectors* $\boldsymbol{\phi}^{j}_{k,1}$ of $L_{\mathrm{rw}}(G^{j}_{k})$, where $k = 0, \ldots, K^{j} - 1$ indicates a region, $j = 0, \ldots, j_{\max}$ indicates a level (or scale), $V = V^{0}_{0} = V^{1}_{0} \cup V^{1}_{1} = \cdots$
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max} - 1 \Rightarrow$ ***scaling*** and ***Haar*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j-1 \Rightarrow$ *scaling*, *Haar*, and *Walsh* vectors

$$\left[\ \boldsymbol{\psi}^{j_{\max}-1}_{0,0}\quad \boldsymbol{\psi}^{j_{\max}-1}_{0,1}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{1,0}\quad \boldsymbol{\psi}^{j_{\max}-1}_{1,1}\ \right]\ \cdots\ \left[\ \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,0}\quad \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,1}\ \right]$$

$$\left[\ \boldsymbol{\psi}^{j_{\max}}_{0,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{1,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{2,0}\ \right]\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{3,0}\ \right]\ \cdots\ \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-2,0}\ \right]\quad \left[\ \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-1,0}\ \right]$$

1. Generate a full recursive bipartitioning of the graph using *Fiedler vectors* $\boldsymbol{\phi}^j_{k,1}$ of $L_{\mathrm{rw}}(G^j_k)$, where $k = 0, \dots, K^j - 1$ indicates a region, $j = 0, \dots, j_{\max}$ indicates a level (or scale), $V = V^0_0 = V^1_0 \cup V^1_1 = \cdots$
2. Generate an orthonormal basis for level $j_{\max}$ (the finest level) $\Rightarrow$ ***scaling vectors*** on the single-node regions
3. Using the basis for level $j_{\max}$, generate an orthonormal basis for level $j_{\max} - 1 \Rightarrow$ ***scaling*** and ***Haar*** vectors
4. Repeat... Using the basis for level j, generate an orthonormal basis for level $j - 1 \Rightarrow$ ***scaling***, ***Haar***, and ***Walsh*** vectors

$$\begin{bmatrix} \boldsymbol{\psi}^0_{0,0} & \boldsymbol{\psi}^0_{0,1} & \boldsymbol{\psi}^0_{0,2} & \boldsymbol{\psi}^0_{0,3} & \cdots & \boldsymbol{\psi}^0_{0,n-2} & \boldsymbol{\psi}^0_{0,n-1} \end{bmatrix}$$

$$\vdots$$

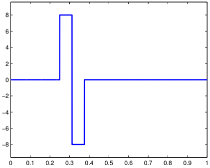
$$\begin{bmatrix} \boldsymbol{\psi}^{j_{\max}-1}_{0,0} & \boldsymbol{\psi}^{j_{\max}-1}_{0,1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}-1}_{1,0} & \boldsymbol{\psi}^{j_{\max}-1}_{1,1} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,0} & \boldsymbol{\psi}^{j_{\max}-1}_{K^{j_{\max}-1}-1,1} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{0,0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{1,0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{2,0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{3,0} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-2,0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}^{j_{\max}}_{K^{j_{\max}}-1,0} \end{bmatrix}$$

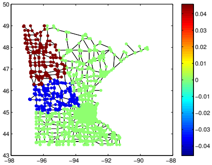
Basis Vector & Coefficient Notation

GHWT basis vectors and coefficients are written as $\boldsymbol{\psi}^{j}_{k,\ell}$ and $c^{j}_{k,\ell}$, respectively, where j and k correspond to level and region and ℓ is the **tag**.

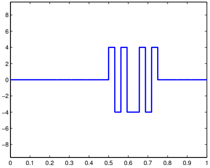
- $\ell = 0 \Rightarrow$ **scaling coefficient/basis vector**
- $\ell = 1 \Rightarrow$ **Haar coefficient/basis vector**
- $\ell \geq 2 \Rightarrow$ **Walsh coefficient/basis vector**



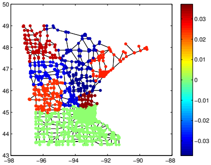
(a) Haar function on $\mathbb{R}$



(b) Haar vector $\boldsymbol{\psi}^{2}_{0,1}$



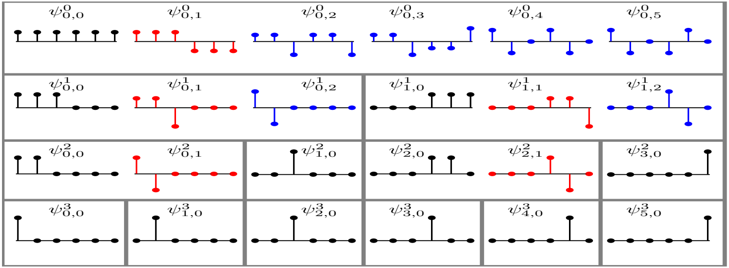
(c) Haar-Walsh wavelet packet on $\mathbb{R}$



(d) Walsh vector $\boldsymbol{\psi}^{1}_{0,5}$

Remarks

- For an unweighted path graph, this yields a dictionary of Haar-Walsh wavelet packets.
- Recursive Partitioning (RP) via Fiedler vectors costs $O(N^2)$ in general.
- Given a recursive partitioning with $O(\log N)$ levels, the computational cost of expanding an input data into the GHWT is $O(N \log N)$.
- We can select an orthonormal basis for the entire graph by taking the union of orthonormal bases on disjoint regions.



Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

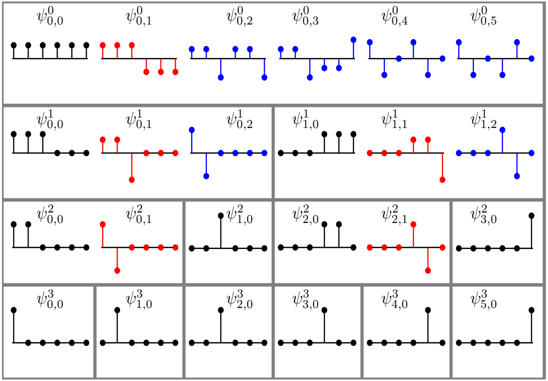


Figure: Default dictionary; i.e., coarse-to-fine

Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

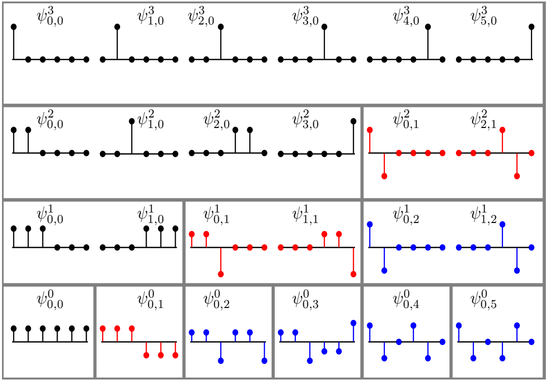


Figure: Reordered & regrouped dictionary; i.e., fine-to-coarse

Remarks . . .

- We can also reorder and regroup the vectors on each level of the GHWT dictionary according to their type (**scaling**, **Haar**, or **Walsh**).

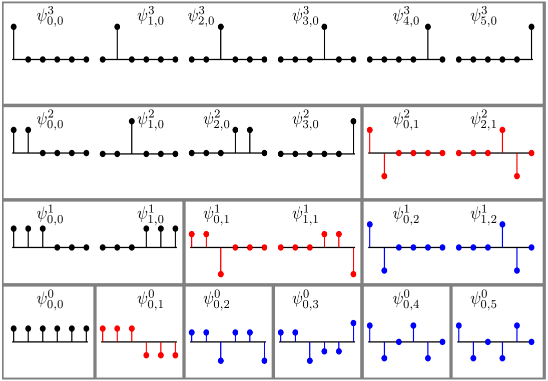


Figure: Reordered & regrouped dictionary; i.e., fine-to-coarse

- This reorganization gives us *more options for choosing a good basis*.

Related Work

The following articles (and perhaps many more) also discussed the Haar-like transform on graphs, but *not the Haar-Walsh Wavelet Packets* on them:

1. A. D. Szlam, M. Maggioni, R. R. Coifman, and J. C. Bremer, Jr., "Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions," in *Wavelets XI* (M. Papadakis et al. eds.), *Proc. SPIE 5914*, Paper # 59141D, 2005.
2. F. Murtagh, "The Haar wavelet transform of a dendrogram," *J. Classification*, vol. 24, pp. 3–32, 2007.
3. A. Lee, B. Nadler, and L. Wasserman, "Treelets–an adaptive multi-scale basis for sparse unordered data," *Ann. Appl. Stat.*, vol. 2, pp. 435–471, 2008.
4. M. Gavish, B. Nadler, and R. Coifman, "Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning," in *Proc. 27th Intern. Conf. Machine Learning*, pp. 367–374, 2010.
5. R. Coifman and M. Gavish, "Harmonic analysis of digital data bases," in *Wavelets and Multiscale Analysis: Theory and Applications* (J. Cohen and A. I. Zayed, eds.), pp. 161–197, Birkhäuser, 2011.

Outline

Best-Basis Algorithms for GHWT

- Coifman and Wickerhauser (1992) developed the best-basis algorithm as a means of selecting the basis from a dictionary of wavelet packets that is "best" for approximation/compression.
- We generalize this approach, developing and implementing an algorithm for selecting the basis from the GHWT dictionary in the *bottom-up* manner that is "best" for approximation and compression.
- We require an appropriate cost functional $\mathscr{J}$. For example:

$$\mathscr{J}\left(\boldsymbol{c}_k^j\right) = \left\|\boldsymbol{c}_k^j\right\|_p := \left(\sum_{\ell=0}^{N_k^j-1} \left|c_{k,\ell}^j\right|^p\right)^{1/p} \qquad 0 < p \leq 1$$

- For other tasks, e.g., classification and regression, see the work of N.S. on *Local Discriminant Basis*, *Local Regression Basis*, *Least Statistically-Dependent Basis*, ..., all of which use different cost functionals and can also be used in the graph setting.

Outline

Method

1. Use the matrix data and the spectral co-clustering to recursively partition the rows and the columns
2. Analyze column vectors of the input matrix using the GHWT dictionary based on the row partitions and extract the best basis for handling columns as a whole, which we call the *row* best basis
3. Analyze row vectors of the input matrix using the GHWT dictionary based on the column partitions and extract the best basis for handling rows as a whole, which we call the *column* best basis
4. Expand the input matrix w.r.t. the *tensor product* of the row and column best bases
5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, regression, etc.

Method

1. Use the matrix data and the spectral co-clustering to recursively partition the rows and the columns
2. Analyze column vectors of the input matrix using the GHWT dictionary based on the row partitions and extract the best basis for handling columns as a whole, which we call the *row* best basis
3. Analyze row vectors of the input matrix using the GHWT dictionary based on the column partitions and extract the best basis for handling rows as a whole, which we call the *column* best basis
4. Expand the input matrix w.r.t. the *tensor product* of the row and column best bases
5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, regression, etc.

Method

1. Use the matrix data and the spectral co-clustering to recursively partition the rows and the columns
2. Analyze column vectors of the input matrix using the GHWT dictionary based on the row partitions and extract the best basis for handling columns as a whole, which we call the *row* best basis
3. Analyze row vectors of the input matrix using the GHWT dictionary based on the column partitions and extract the best basis for handling rows as a whole, which we call the *column* best basis
4. Expand the input matrix w.r.t. the *tensor product* of the row and column best bases
5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, regression, etc.

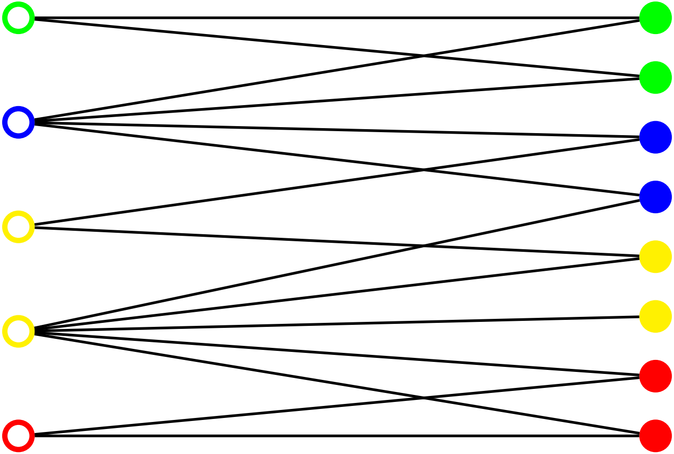
Method

1. Use the matrix data and the spectral co-clustering to recursively partition the rows and the columns
2. Analyze column vectors of the input matrix using the GHWT dictionary based on the row partitions and extract the best basis for handling columns as a whole, which we call the *row* best basis
3. Analyze row vectors of the input matrix using the GHWT dictionary based on the column partitions and extract the best basis for handling rows as a whole, which we call the *column* best basis
4. Expand the input matrix w.r.t. the *tensor product* of the row and column best bases
5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, regression, etc.

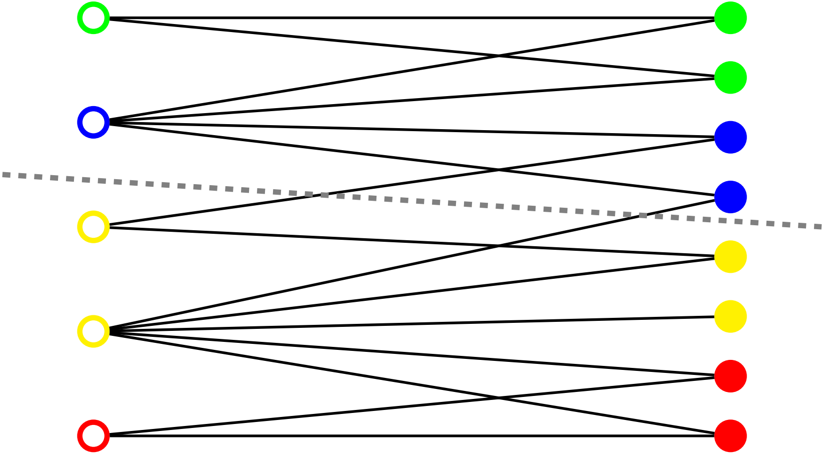
Method

1. Use the matrix data and the spectral co-clustering to recursively partition the rows and the columns
2. Analyze column vectors of the input matrix using the GHWT dictionary based on the row partitions and extract the best basis for handling columns as a whole, which we call the *row* best basis
3. Analyze row vectors of the input matrix using the GHWT dictionary based on the column partitions and extract the best basis for handling rows as a whole, which we call the *column* best basis
4. Expand the input matrix w.r.t. the *tensor product* of the row and column best bases
5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, regression, etc.

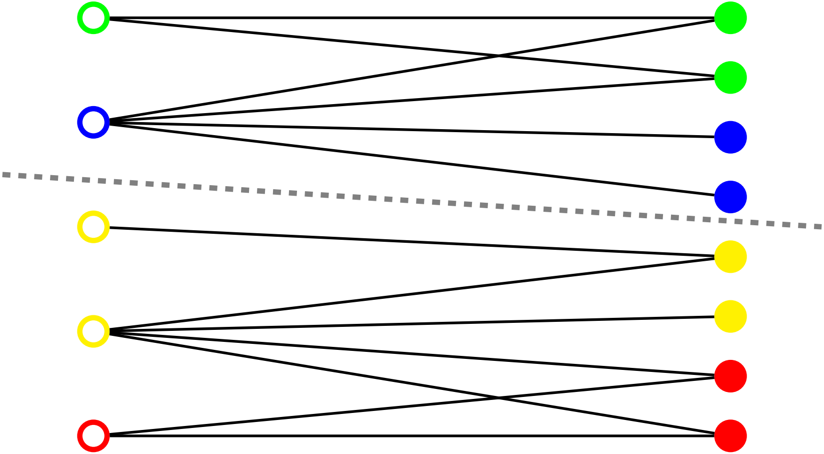
Matrix Partitioning à la Dhillon (2001)



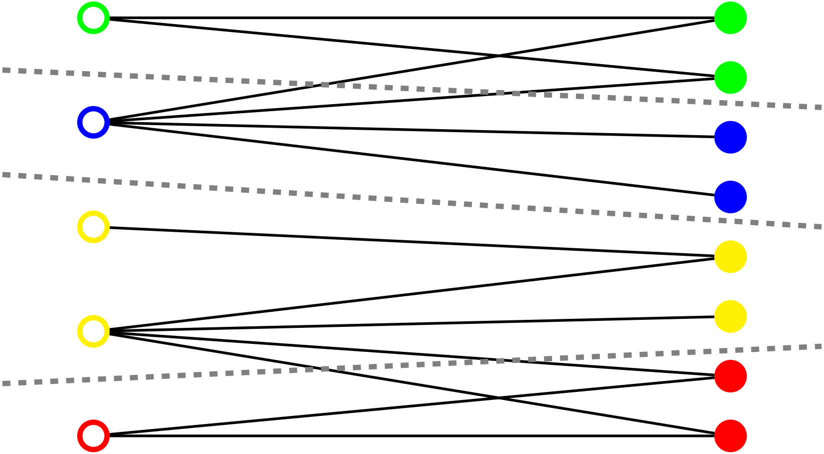
Matrix Partitioning à la Dhillon (2001)



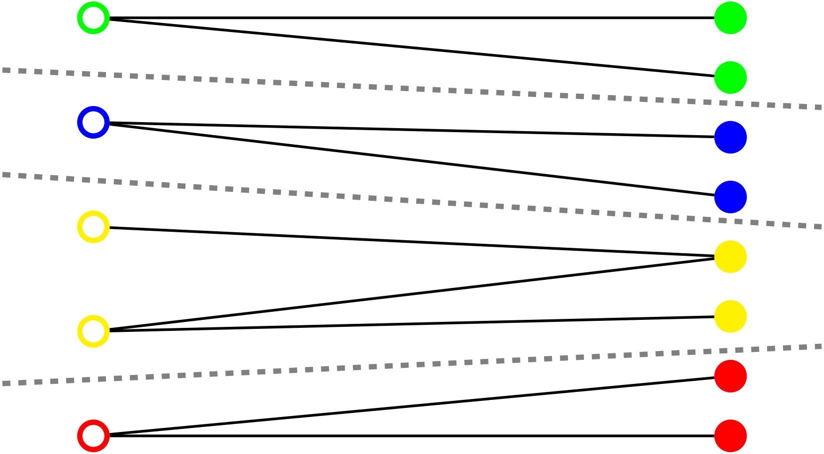
Matrix Partitioning à la Dhillon (2001)



Matrix Partitioning à la Dhillon (2001)



Matrix Partitioning à la Dhillon (2001)



Example 1: Science News Dataset

Dataset: the Science News database (1153×1042)

- Rows $\rightarrow$ preselected words
- Columns $\rightarrow$ articles from 8 fields: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math & CS; Medicine; Physics
- a_{ij} $\rightarrow$ the relative frequency of word i appears in article j $\Rightarrow$ all column sums are 1

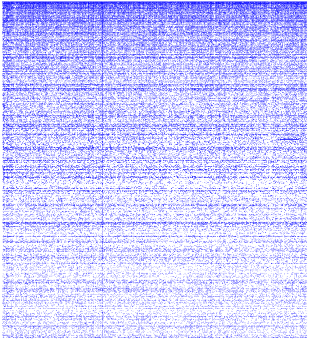


Figure: Science News database (original order)

Example 1: Science News Dataset

Dataset: the Science News database (1153×1042)

- Rows → preselected words
- Columns → articles from 8 fields: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math & CS; Medicine; Physics
- a_{ij} → the relative frequency of word i appears in article j ⇒ all column sums are 1

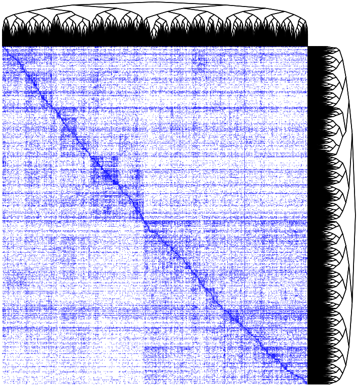


Figure: Science News database (reordered rows and columns)

Example 1: Science News Dataset

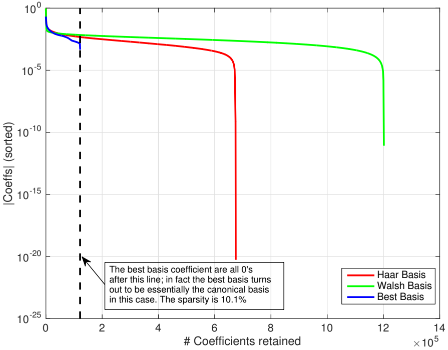


Figure: Decay of the expansion coefficients w.r.t. Haar basis, Walsh basis, and GHWT best basis. The vertical line denotes the percentage of nonzero entries in the matrix (**10.1%**).

- Cost functional: 1-norm
- Total number of orthonormal bases searched: $> 10^{370}$
- **62.3%** of the Haar coefficients and **100%** of the Walsh coefficients must be kept to achieve perfect reconstruction, compared to **10.1%** for the GHWT best basis

$\Rightarrow$ The Haar and Walsh bases could not efficiently capture the underlying structure of this Science News dataset under the current matrix partitioning strategy!

Example 1: Science News Dataset

- Since the sparsity was used as the cost functional, the best basis is in fact almost the canonical basis; the fine scale information was too much emphasized, which may be sensitive to 'noise'.
- We are interested in the *medium scale* information in this database, e.g., clustering structures both in words (rows) and articles (cols).
- Hence, we *weight* the coefficients in the GHWT dictionary as follows:

$$\begin{aligned} c^j_{k,l} &\leftarrow c^j_{k,l} \cdot 2^{j\alpha} \cdot \left(\mathrm{supp}(G^0_0) / \mathrm{supp}(G^j_k) \right)^\beta \\ &= c^j_{k,l} \cdot 2^{j\alpha} \cdot (N/N^j_k)^\beta \end{aligned}$$

  where $\alpha \geq 0$, $\beta \geq 0$, are chosen empirically to make the magnitude of the finer coefficients bigger, which discourages the best-basis algorithm to select fine scale subgraphs.
- See also Coifman-Leeb's technical report (2013) and Ankenman's Ph.D. dissertation (2014) for such weighting scheme and its relation to the *Earth Mover's Distance*.

Example 1: Science News Dataset

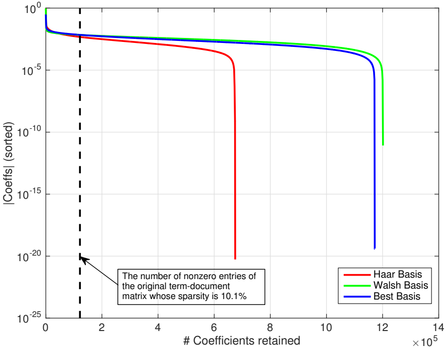


Figure: Decay of the expansion coefficients w.r.t. Haar basis, Walsh basis, and GHWT best basis. The vertical line denotes the percentage of nonzero entries in the matrix (**10.1%**).

- Cost functional: 1-norm
- $\alpha^{\text{row}} = \alpha^{\text{col}} = 0$
- $\beta^{\text{row}} = 1.0, \beta^{\text{col}} = 0.15$
- This best basis is less sparse than before, and is between the Haar and the Walsh bases, i.e., well captures information on intermediate scales.

Example 1: Science News Dataset

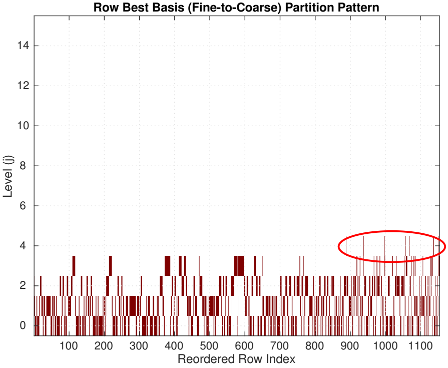


Figure: The *row* best basis partition pattern. This is a fine-to-coarse basis.

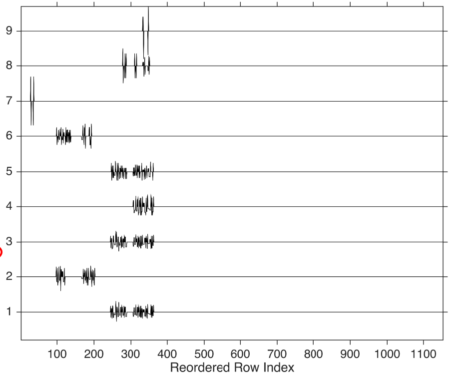


Figure: The *row* best basis vectors at $j = 4$.

Example 1: Science News Dataset

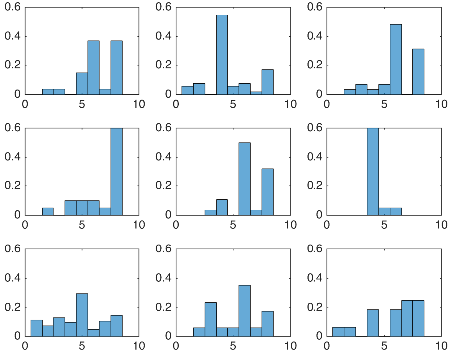
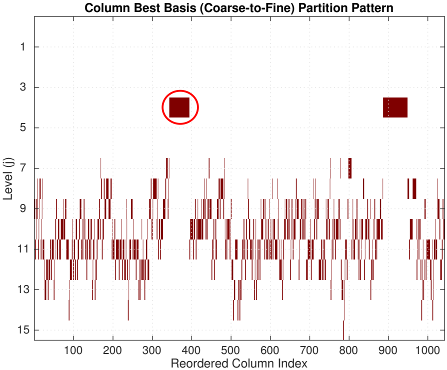


Figure: The histograms of the article categories (1 to 8) of the expansion coefficients of column vectors w.r.t. those 9 row best basis vectors.

- For example, the positive components of the 6th basis vector correspond to the following words: `earthquake, down, california, dioxide, deep, warm, el, southern, crust, valley, once, geologist, bottom, tsunami, oxide, fault, antarctica, warning, tsunamis, prediction, greenhouse`
- On the other hand, the negative components of that vector correspond to: `temperature, ice, sea, layer, flow, around, survey, coast, warming, quake, past, nino, global, seismologist, cycle, cold, slow, recent, plate, thickness, meter, japan, forecast`
- Clearly, this basis vector is checking if a given article is in Category 4 (Earth Sciences).

Example 1: Science News Dataset



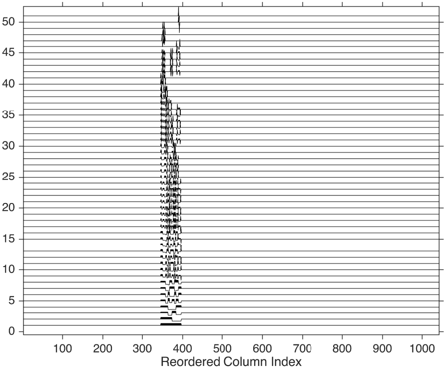


Figure: The *column* best basis partition pattern. This is a coarse-to-fine basis. The block indicated by a red circle corresponding to $(j,k) = (4,5)$.

Figure: The *column* best basis vectors with $(j,k) = (4,5)$ whose supports are 51 articles; 48 among 51 indicate 'Astronomy'.

Example 1: Science News Dataset

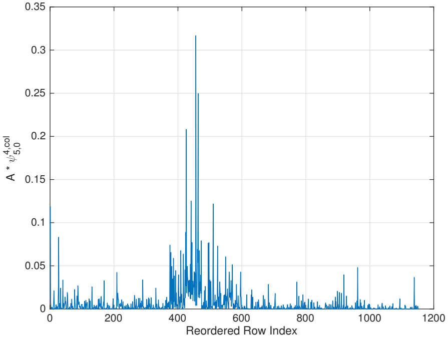


Figure: The expansion coefficients of row vectors w.r.t. the column best basis vector $\boldsymbol{\psi}_{5,0}^{4,\mathrm{col}}$ = the indicator vector of 51 articles.

- The 3 nonzero components in $\boldsymbol{\psi}_{5,0}^{4,\mathrm{col}}$ that are not in 'Astronomy' correspond to the following articles:
  - `"Old Glory, New Glory: The Star-Spangled Banner gets some tender loving care"` (Anthropology: on the preservation of the Star-Spangled Banner (flag) using the space-age technology);
  - `"Snouts: A star is born in a very odd way"` (Life Sciences: on star-nosed moles);
  - `"Gravity tugs at the center of a priority battle"` (Math & CS: on the priority war on the discovery of gravity between Newton, Halley, and Hooke).
- The expansion coefficients > 0.05 in the left figure correspond to the following words: `year, university, time, team, system, light, earth, star, planet, finding, astronomer, universe, galaxy, object, ray, telescope, orbit, mass, hole, dust, black, distance, disk, infrared`

Example 1: Science News Dataset

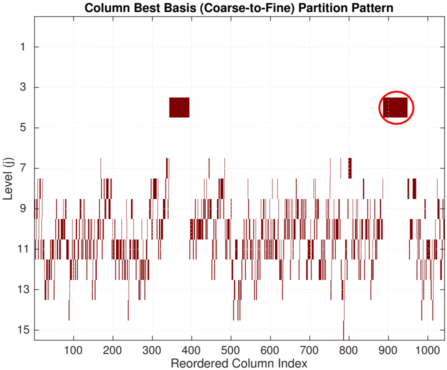


Figure: The *column* best basis partition pattern. This is a coarse-to-fine basis. The block indicated by a red circle corresponding to $(j,k) = (4,14)$.

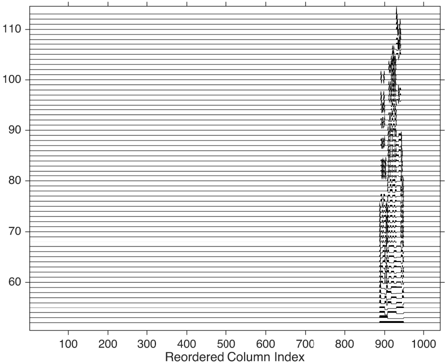


Figure: The *column* best basis vectors with $(j,k) = (4,14)$ whose supports are 62; 56 among 62 indicate 'Medical Sciences'.

Example 1: Science News Dataset

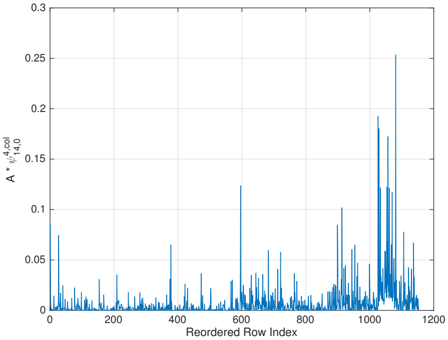
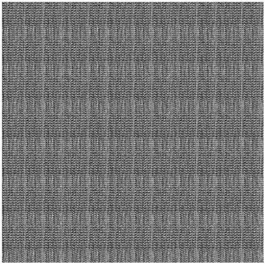


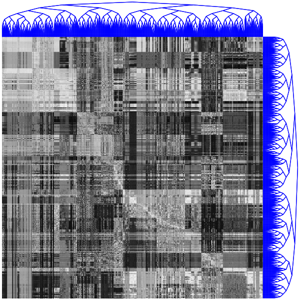
Figure: The expansion coefficients of row vectors w.r.t. the column basis vector $\boldsymbol{\psi}^{4,\text{col}}_{14,0}$ = the indicator vector of 62 articles.

- Out of these 6 anomalies, 3 are in 'Life Sciences', i.e., not really surprising. The remaining 3 anomalies are:
  - ``In Silico Medicine: Computer simulations aid drug development and medical care'' (Math & CS);
  - ``Beyond Virtual Vaccinations: Developing a digital immune system in bits and bytes'' (Math & CS);
  - ``Paleopathological Puzzles: Researchers unearth ancient medical secrets'' (Anthropology).
- The expansion coefficients > 0.05 in the left figure correspond to the following words: `year, university, study, scientist, people, cell, group, disease, system, drug, protein, brain, human, blood, patient, test, immune, virus, strain, infection, vaccine, antibody, hiv, infected, aids, amyloid`

Example 2: The Shuffled Barbara Image

Dataset: the 512×512 "Barbara" image with the rows and columns shuffled.





- **Left:** the original Barbara image
- **Middle:** the shuffled Barbara image
- **Right:** the shuffled image reordered according to the recursive partitioning

Example 2: The Shuffled Barbara Image

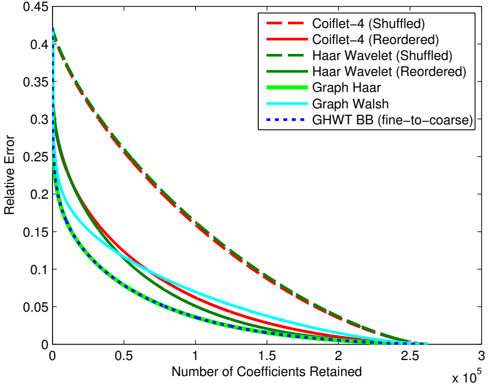


Figure: Approximation results. The "shuffled" and "reordered" results are for the cases that the shuffled image (middle figure on previous page) and reordered image (figure on the right) was analyzed, respectively.

- Cost functional: 1-norm
- Total number of ONBs searched: $> 6.37 \times 10^{173}$
- The GHWT BB nearly matches the graph Haar basis and performs better than the graph Walsh basis
- The GHWT BB performs much better than the Coiflet and Haar bases directly applied on the image, which are fixed and therefore cannot account for *nondyadic* geometry of the data

Example 2: The Shuffled Barbara Image

We can also use the GHWT and best basis algorithm to ascertain information about the spatial structure of the matrix data.

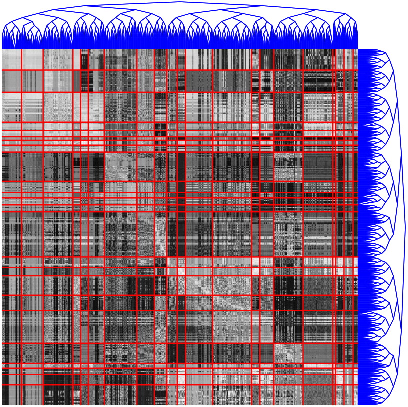


Figure: The coarse-to-fine row and column best bases for "Barbara" using the 0.1-quasinorm as our cost functional.

Example 2: The Shuffled Barbara Image

We can obtain different results by using a different cost functional.

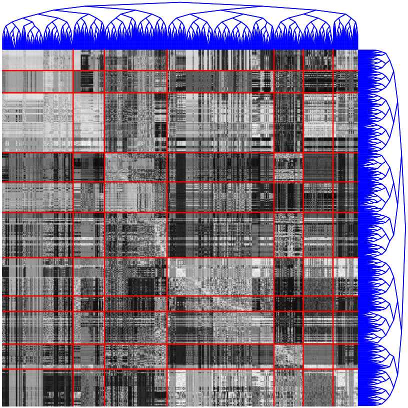


Figure: The coarse-to-fine row and column best bases for "Barbara" using the 0.5-quasinorm as our cost functional.

Example 2: The Shuffled Barbara Image

Another option is to not consider regions with fewer than $N_{\min}$ nodes.

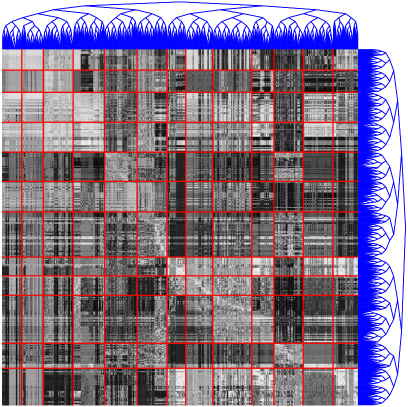


Figure: The coarse-to-fine row and column best bases for "Barbara" using the 0.1-quasinorm as our cost functional; regions with fewer than $[N_r/20] = [N_c/20] = 26$ nodes were not considered in the best basis search.

Outline

Summary

- Combining the spectral co-clustering and GHWT leads to a powerful matrix data analysis tool.
- The GHWT best-basis algorithm searches over an immense number of orthonormal bases, including the graph Haar/Walsh bases.
- When selected using an appropriate cost functional, the GHWT best basis equals or outperforms the graph Haar/Walsh bases.
- This demonstrates the importance/advantage of a *data-adaptive basis dictionary* from which one can select the most suitable basis for one's task at hand!
- Appropriately weighting the expansion coefficients dependent on *scales* leads to a more *meaningful* basis at the cost of sparsity.
- Should explore different cost functionals than the sparsity $\Longrightarrow$ Local Regression Basis (LRB) of Saito and Coifman
- What to do if your input data is of *tensor* form, i.e., $A = (a_{ijk}) \in \mathbb{R}^{I \times J \times K}$? $\Longrightarrow$ a *tripartite* graph (a.k.a. 3-uniform *hypergraph*)!

Outline

References

The following articles (and the other related ones) are available at http://www.math.ucdavis.edu/~saito/publications/

- J. Irion & N. Saito: "Efficient approximation and denoising of graph signals using the multiscale basis dictionaries," submitted for publication, 2016.
- J. Irion & N. Saito: "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, Paper # 95971F, 2015.
- J. Irion & N. Saito: "The generalized Haar-Walsh transform," *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, pp. 488-491, 2014.
- J. Irion & N. Saito: "Hierarchical graph Laplacian eigen transforms," *JSIAM Letters*, vol. 6, pp. 21–24, 2014.

Jeff Irion disseminates the codes for HGLET/GHWT and his Ph.D. dissertation at https://github.com/JeffLIrion/MTSG_Toolbox.

Thank you very much for your attention!