LEARNING SPARSITY AND STRUCTURE OF MATRICES WITH MULTISCALE GRAPH BASIS DICTIONARIES

Jeff Irion[*] *and Naoki Saito*[†]

Department of Mathematics, University of California, Davis

ABSTRACT

Many modern data analysis tasks often require one to efficiently handle and analyze large matrix-form datasets such as term-document matrices and spatiotemporal measurements made via sensor networks. Since such matrices are often shuffled and scrambled, they do not have spatial coherency and smoothness that usual images and photographs possess, and consequently, the conventional wavelets and their relatives cannot be used in practice. Instead we propose to use our multiscale basis dictionaries for graphs, i.e., the Generalized Haar-Walsh Transform. In particular, we build such dictionaries for columns and rows separately, extract the column best basis and the row best basis from the basis dictionaries, and construct the tensor product of such best bases, which turns out to reveal hidden dependency and underlying geometric structure in the given matrix data. Finally, we will demonstrate the effectiveness of our approach using the Science News database.

Index Terms— Multiscale basis dictionaries on graphs, Haar-Walsh wavelet packets, adaptive best basis algorithm, spectral co-clustering, term-document matrices

1. INTRODUCTION

Many modern data analysis tasks often involve large matrix-form datasets. For example, spatiotemporal data measured by sensor networks may be represented as a matrix whose columns represent sensors while the rows represent time indices. Another important example is ratings or reviews of commercial products by their users; this leads to a matrix in which columns represent products, rows represent users, and matrix entry a_{ij} represent user i's rating of product j, say, on a 1–5 scale. Such matrices are quite different from usual images and photographs. In fact, they are often more like shuffled and permuted images, possessing no spatial smoothness or coherency in general. Yet, the rows and columns of such a matrix are interrelated, and thus the rows of the matrix can tell us about the underlying structure of the columns, and vice versa. By considering the interplay between the rows and columns, we can learn more about the matrix than if we treat them separately. Moreover, by utilizing multiscale basis dictionaries on graphs, we can discover, learn, and exploit underlying (often hidden) dependency and geometric structure in the matrix data for a variety of tasks, e.g., compression, classification, etc.

2. SPECTRAL CO-CLUSTERING FOR ORGANIZING ROWS AND COLUMNS

In order to discover the structure of a given matrix and hierarchically organize its rows and columns, as required by our tool, we use the spectral co-clustering method of Dhillon [1]. Given a data matrix $A \in \mathbb{R}^{N_r \times N_c}$, this method views the rows and columns as the two sets of nodes in an undirected *bipartite* graph, where matrix entry a_{ij} is the edge weight between the node for row i and the node for column j. Let us order the nodes of this bipartite graph such that the first N_r nodes correspond to the rows and the last N_c correspond to the columns. Then the associated $(N_r + N_c) \times (N_r + N_c)$ weight matrix becomes of the form

$$W = \begin{bmatrix} O & A \\ A^{\mathsf{T}} & O \end{bmatrix}.$$

Accordingly, the degree and Laplacian matrices are

$$D = \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix}, \quad L := D - W = \begin{bmatrix} D_r & -A \\ -A^{\mathsf{T}} & D_c \end{bmatrix}, \tag{1}$$

where $D_r := \operatorname{diag}(A\mathbf{1}_{N_c})$ and $D_c := \operatorname{diag}(A^{\mathsf{T}}\mathbf{1}_{N_r})$; $\mathbf{1}_{N_*}$ is the vector of all ones of length N_*. A common means of bipartitioning the graph is by using the *Fiedler vector*, i.e., the eigenvector corresponding to the smallest positive eigenvalue of the random-walk Laplacian $L_{\mathrm{rw}} := D^{-1}L$; see, e.g., [2] for the details on why the eigenvectors of L_{rw} are preferred to those of L. For the sake of computational efficiency, Dhillon's method computes the second left and right singular vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ of $D_r^{-1/2} A D_c^{-1/2}$ and then forms the Fiedler vector as

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

[*] Current address: Bosch Research and Technology Center, 4009 Miranda Avenue, Palo Alto, CA 94304. Partially supported by National Defense Science and Engineering Graduate Fellowship, 32 CFR 168a via AFOSR FA9550-11-C-0028 and the National Science Foundation grant DMS-1418779.

[†] Partially supported by the Office of Naval Research grants N00014-12-1-0177, N00014-16-1-2255, and the National Science Foundation grant DMS-1418779.

To appear in *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, Sep. 13–16, 2016, Salerno, Italy ©2016 IEEE

This is because $L_{\mathrm{rw}}\boldsymbol{\phi} = \lambda\boldsymbol{\phi}$ is equivalent to $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$, and using Eq. (1), this further leads to

$$A\boldsymbol{\phi}_c = (1-\lambda)D_r\boldsymbol{\phi}_r; \quad A^{\mathsf{T}}\boldsymbol{\phi}_r = (1-\lambda)D_c\boldsymbol{\phi}_c,$$

where $\boldsymbol{\phi}_r \in \mathbb{R}^{N_r}$ and $\boldsymbol{\phi}_c \in \mathbb{R}^{N_c}$ are the first N_r entries and the next N_c entries of $\boldsymbol{\phi} \in \mathbb{R}^{N_r+N_c}$. Then, setting $\boldsymbol{u} := D_r^{1/2}\boldsymbol{\phi}_r$, $\boldsymbol{v} := D_c^{1/2}\boldsymbol{\phi}_c$, we get

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

which precisely defines the *SVD* of $D_r^{-1/2}AD_c^{-1/2} \in \mathbb{R}^{N_r \times N_c}$; we do not need to compute the eigenvector of the larger matrix $L_{\mathrm{rw}} \in \mathbb{R}^{(N_r+N_c)\times(N_r+N_c)}$. Using this Fiedler vector, the rows and the columns of A can be partitioned *simultaneously*.

3. THE GENERALIZED HAAR-WALSH TRANSFORM

The *Generalized Haar-Walsh Transform* (GHWT) [3, 4, 5] is a true generalization of the classical Haar-Walsh wavelet packet transform [6] to the graph setting, and it generates a *dictionary* (i.e., a redundant set) of basis vectors that are multiscale, oscillatory, and piecewise-constant on their support. In this section, we briefly review the GHWT dictionary construction.

Let $G = G(V, E)$ be an undirected and connected graph with $|V| = N$ nodes and $|E| = M$ edges. To construct the GHWT, we first need a *hierarchical bipartition tree* of G, i.e., a set of tree-structured subgraphs of G constructed by recursively bipartitioning G. This bipartitioning operation ideally splits each subgraph into two smaller subgraphs that are roughly equal in size while keeping tightly-connected nodes grouped together. Any reasonable graph partitioning method can be used for this operation, but in this paper, we use the Fiedler vector of the random-walk Laplacian matrix of each subgraph for the next level of bipartitioning. Let j be a level index of the hierarchical bipartition tree, with $j = 0$ denoting the coarsest level and $j = j_{\max}$ denoting the finest level. We use K^j to denote the number of sets of nodes on level j of the tree, and we use $k \in [0, K^j)$ to index these sets. We use V_k^j to denote the kth set of nodes on level j, and set $N_k^j := |V_k^j|$. Let G_k^j be the subgraph of G formed by restricting to the nodes in V_k^j and the edges between them. We often use the term “region” to refer to a subgraph.

We impose the following requirements for a hierarchical bipartition tree of G:

i. The coarsest level is the entire graph: $G_0^0 = G$, $V_0^0 = V$, $N_0^0 = N$, $K^0 = 1$.

ii. At the finest level $j = j_{\max}$, each region is a single node: $N_k^{j_{\max}} = 1$ for $0 \le k < K^{j_{\max}} = N$.

iii. All regions on a given level are disjoint: $V_k^j \cap V_{\tilde{k}}^j = \emptyset$ if $k \neq \tilde{k}$.

iv. Each region on level $j < j_{\max}$ containing two or more nodes is partitioned into exactly two regions on level $j+1$.

Given a hierarchical bipartition tree of G, the GHWT generates an overcomplete dictionary whose basis vectors have their supports ranging from a single node to the entire graph. See [3, 4] and [5, Chap. 5] for the details of the algorithm to generate the GHWT dictionary for a given graph. We use $\boldsymbol{\psi}_{k,l}^j \in \mathbb{R}^N$ to denote the GHWT basis vectors supported on V_k^j, and $d_{k,l}^j := \langle \boldsymbol{f}, \boldsymbol{\psi}_{k,l}^j \rangle$ to denote the corresponding expansion coefficient of an input graph signal $\boldsymbol{f} \in \mathbb{R}^N$. The index $l \in [0, N_k^j)$ represents the *tag* of a basis vector/coefficient, and it assumes N_k^j distinct values within the range $[0, 2^{j_{\max}-j})$. We refer to coefficients with tag $l = 0$ as *scaling coefficients*, those with tag $l = 1$ as *Haar coefficients*, and those with tag $l \ge 2$ as *Walsh coefficients*. The total number of the expansion coefficients in this GHWT dictionary is $N \times (j_{\max} + 1)$, and given a hierarchical bipartition tree of G with $O(\log_2 N)$ levels, the computational cost of generating all these coefficients is $O(N \log_2 N)$.

One of the key features of the GHWT is that we can arrange the coefficients in two ways. On each level j, we can group them by their k index, yielding the *coarse-to-fine dictionary*. Alternatively, we can group them by their tag l to obtain the *fine-to-coarse dictionary*, the significance of which is that it affords us more bases from which to choose. Generally speaking, for a graph with N nodes, both of the GHWT dictionaries contain $> 2^{\lfloor N/2 \rfloor}$ choosable bases. We note, however, that exceptions can occur when the recursive bipartitioning is highly imbalanced [5, Chap. 5].

For the task of selecting one basis from the immense number of choosable bases, we have generalized the best basis algorithm of Coifman and Wickerhauser [6] for the GHWT transforms. The algorithm requires a user-specified cost functional, and the search starts at the bottom level of the dictionary and proceeds upwards, comparing the cost of the children coefficients to the cost of the parent coefficients; see [5, Chap. 6] for the details.

4. MATRIX DATA ANALYSIS USING THE GHWT

Our basic strategy for matrix data analysis is the following:

1. Use the matrix data and the spectral co-clustering to recursively bipartition the rows and the columns

2. Analyze row vectors of the input matrix using the GHWT dictionaries based on the column partitions and extract the best basis for organizing columns, which we call the *column* best basis

3. Analyze column vectors of the input matrix using the GHWT dictionaries based on the row partitions and

extract the best basis for organizing rows, which we call the *row* best basis

4. Expand the input matrix w.r.t. the *tensor product* of the column and row best bases

5. Analyze the expansion coefficients for a variety of tasks, e.g., compression, classification, etc.

More detailed descriptions are in order. In Step 1, the spectral co-clustering discussed in Section 2 is recursively applied to the given matrix in order to yield hierarchical bipartition trees for the rows and columns. In addition to its ability to simultaneously partition the rows and columns, another advantage of this method is that we do not need to construct edge weight matrices explicitly, which would require defining a weight function and specifying a means of constructing a graph (e.g., k-nearest neighbor, ϵ-neighborhood, or complete) of rows and columns: the given matrix data automatically define the bipartite graph with its edge weights as long as a_{ij}'s are all non-negative.

In Step 2, using the hierarchical bipartition tree of the columns, we apply the GHWT to each row vector of the matrix, which generates an array of the expansion coefficients of size $N_r \times N_c \times (j_{\max}^{\text{col}}+1)$, where $j_{\max}^{\text{col}}$ is the maximum level in the recursive bipartitioning of the columns. Therefore, our next step is to "flatten" this 3-dimensional array to a 2-dimensional matrix of size $N_c \times (j_{\max}^{\text{col}}+1)$, from which we can select a column best basis. There are various ways in which we can do this, but typically we take the ℓ^1-norm along the dimension corresponding to the rows. Thus, the entries in the resulting matrix reflect the average ℓ^1-norm over the rows relative to the column GHWT dictionary coefficients. We then apply the GHWT best basis algorithm to this flattened $N_c \times (j_{\max}^{\text{col}}+1)$ matrix using a cost functional of our choice (e.g., the ℓ^p-(quasi)norm, $0 < p \le 1$) to obtain the column best basis. We note that what we obtain here is the *specification* of the column best basis without generating its basis vectors.

Similarly in Step 3, we apply the GHWT to each column vector of the original matrix, which yields an array of size $N_r \times N_c \times (j_{\max}^{\text{row}}+1)$. As we did for the columns in Step 2, we "flatten" this to a matrix of size $N_r \times (j_{\max}^{\text{row}}+1)$ and then use the best basis algorithm to find the row best basis.

At the beginning of Step 4, we have the specifications of the column and row best bases. Now, for each column, we select the coefficients corresponding to the row best basis from the $N_r \times N_c \times (j_{\max}^{\text{row}}+1)$ array, and the result is a row-transformed $N_r \times N_c$ matrix, which is a collection of the expansion coefficients of each column vector w.r.t. the row best basis. We now use the existing hierarchical bipartition tree of the columns to apply the GHWT to each row vector of this transformed matrix, once again yielding an array of size $N_r \times N_c \times (j_{\max}^{\text{col}}+1)$. We then extract the coefficients corresponding to the column best basis, yielding the final result of our analysis: a row- and column-transformed $N_r \times N_c$ matrix of GHWT expansion coefficients of A. The computational cost of transforming such a matrix relative to these best bases is $O(N_r N_c \log_2(N_r N_c))$.

Although in our description we transform the rows and extract the best basis, then transform the columns and extract the best basis, it does not matter whether we analyze the rows or the columns first. To see why, let $\Psi_{\text{rows}} \in \mathbb{R}^{N_r \times N_r}$ and $\Psi_{\text{cols}} \in \mathbb{R}^{N_c \times N_c}$ denote the orthogonal matrices whose columns are the row and column best basis vectors. Although we do not form these matrices for the sake of computational efficiency, our matrix transform is equivalent to computing $\Psi_{\text{rows}}^{\mathsf{T}} A \Psi_{\text{cols}}$, and thus it is not impacted by which dimension is transformed first.

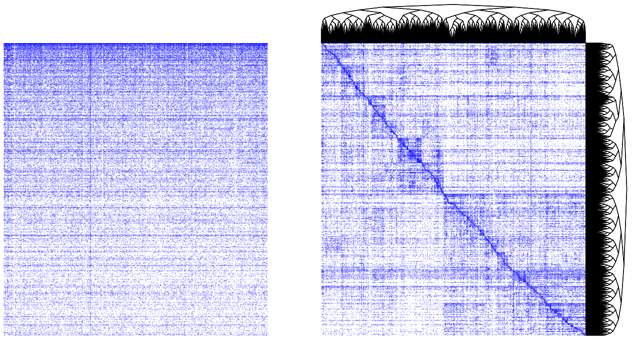
We note here that Bremer [7, Chap's. 3, 4] also developed a matrix tiling algorithm using the idea of the Coifman-Wickerhauser best basis algorithm; however, he did not construct bases or dictionaries to analyze a given matrix, and his main concern was to reorganize rows and columns to reveal low rank structure (if any) of an input matrix.

5. ANALYSIS OF A SPECIFIC TERM-DOCUMENT MATRIX: THE SCIENCE NEWS DATASET

As an example, we use the Science News database, in particular, the term-document matrix consisting of 1042 columns representing documents obtained from the Science News website and 1153 rows representing preselected most relevant words out of 10906 meaningful words. This list of words was initially generated by J. Solka [8] followed by the effort of M. Maggioni and M. Gavish [9], and was finalized by N. Saito by removing five identical documents. These documents are already classified/labeled into eight different categories: Anthropology; Astronomy; Behavioral Sciences; Earth Sciences; Life Sciences; Math/CS; Medicine; Physics. The ijth entry of this matrix, a_{ij}, represents the *relative* frequency with which word i appears in document j, and consequently, the each column sum is 1.

In this experiment, we use the ℓ^1-norm as a cost functional in the best basis algorithm. The total number of orthonormal bases searched via the best basis algorithm exceeds 10^{370}. Fig. 2 compares the approximation performance of our GHWT best basis with that of the Haar and Walsh bases that can also be extracted from the GHWT dictionaries. From this figure, we see that 62.3% of the Haar coefficients and 100% of the Walsh coefficients must be kept to achieve perfect reconstruction, compared to 10.15% using the GHWT best basis, which turns out to be almost the canonical basis in this case. Note that the sparsity of the original matrix is 10.13%. In fact, if we use the ℓ^p-quasinorm with $0 < p \lesssim 0.00979$, the resulting best basis exactly becomes the canonical basis.

Since the ℓ^1 cost functional promotes sparsity, the resulting best basis does in fact sparsify the input matrix



(a) Original order (b) Rows/cols reordered

Fig. 1. The sparsity patterns of the Science News dataset before and after the reordering based on the hierarchical spectral co-clustering.

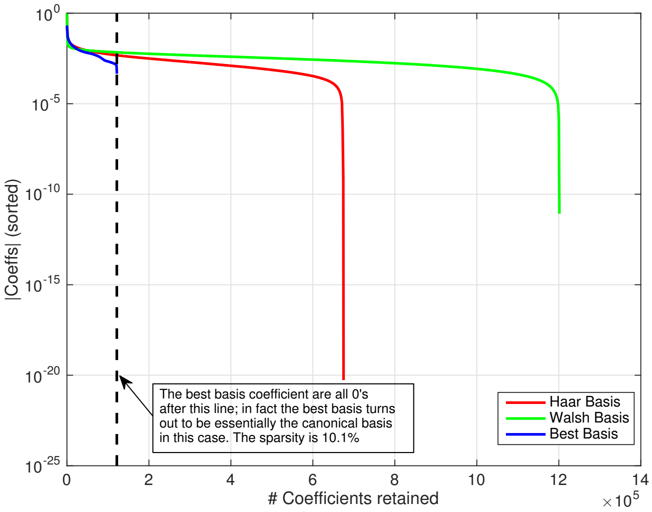


Fig. 2. Decay of the expansion coefficients w.r.t. Haar basis, Walsh basis, and GHWT best basis. The vertical line denotes the percentage of nonzero entries in the matrix (**10.1%**).

nicely, which is reassuring on the one hand because the best basis does the expected job. But on the other hand, the fine scale information is too much emphasized in our algorithm with the ℓ^p-(quasi)norm cost functional with $0 < p \leq 1$, 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 documents (columns). To do so, one possibility is to *weight* the coefficients in the GHWT dictionaries as follows:

$$d^j_{k,l} \leftarrow d^j_{k,l} \cdot (|V^0_0|/|V^j_k|)^\beta = d^j_{k,l} \cdot (N/N^j_k)^\beta, \tag{2}$$

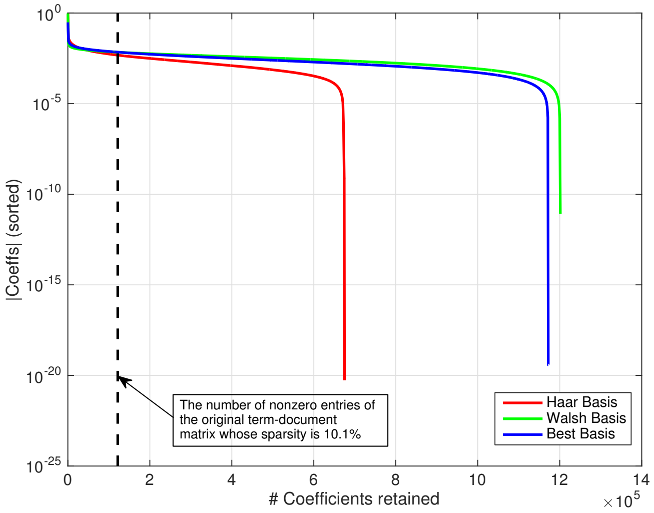
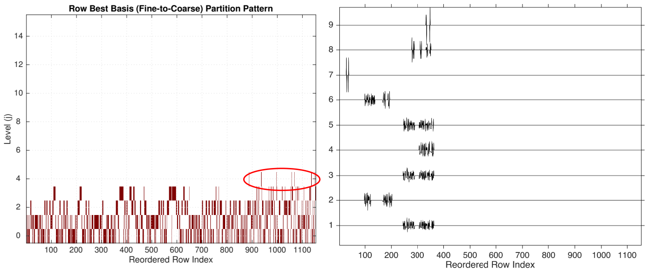


Fig. 3. Decay of the expansion coefficients w.r.t. Haar basis, Walsh basis, and GHWT best basis using the weighted coefficients via Eq. (2).

where $\beta \geq 0$ is chosen empirically to make the magnitude of the finer coefficients bigger, which discourages the best basis algorithm from selecting fine scale subgraphs. See also [10, 11] for similar weighting scheme and its relation to the *Earth Mover's Distance*. Here, we set $\beta^{\text{row}} = 1.0, \beta^{\text{col}} = 0.15$ empirically after repeated experiments. As Fig. 3 shows, this best basis sparsifies A less than before, and is somewhat between the Haar and the Walsh bases, yet well captures information on intermediate scales as we discuss below. Fig. 4a shows the row best basis: the horizontal axis indicates the row indices reordered by the hierarchical bipartition tree followed by the best basis algorithm whereas the vertical axis indicates the scale (or level) information j from the top ($j = j^{\text{row}}_{\max} = 16$: the finest) to the bottom ($j = 0$: the coarsest); the colored blocks in Fig. 4a represent the scale information of the row best basis coefficients at the reordered row indices. One can see that many of the coefficients are at rather coarse scales, i.e., $0 \leq j \leq 4$ due to the weighting scheme Eq. (2). Fig. 4b shows the nine row best basis vectors corresponding to the coefficients indicated by the ellipse in Fig. 4a, all of which are on level $j = 4$. We clearly see that these vectors try to analyze the documents using specific groups of words (reordered rows). For example, the positive components of the sixth basis vector in Fig. 4b cor-



(a) The best basis partition (b) The basis vectors on $j = 4$

Fig. 4. The *row* (word) best basis (fine-to-coarse) respond 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 document is in Category 4 (Earth Sciences). Fig. 5 demonstrates the potential usefulness of these basis vectors. For each of these basis vectors $\boldsymbol{\psi}_{k,l}^{4,\mathrm{row}} \in \mathbb{R}^{N_r}$, we compute the corresponding expansion coefficients for the whole set of documents , i.e., $(\boldsymbol{\psi}_{k,l}^{4,\mathrm{row}})^{\mathsf{T}} A$, and we select the documents whose expansion coefficients are greater than 0.01. Then, we create the histogram of the document categories of those selected documents. For example, Fig. 5 demonstrates that the sixth basis vector clearly contributes to single out Category 4 (Earth Sciences) documents most with small spillover to Categories 5 (Life Sciences) and 6 (Math/CS) while the ninth basis vector does not (the words corresponding to its nonzero components are: *device, industry, electrical, electric, fluid*).

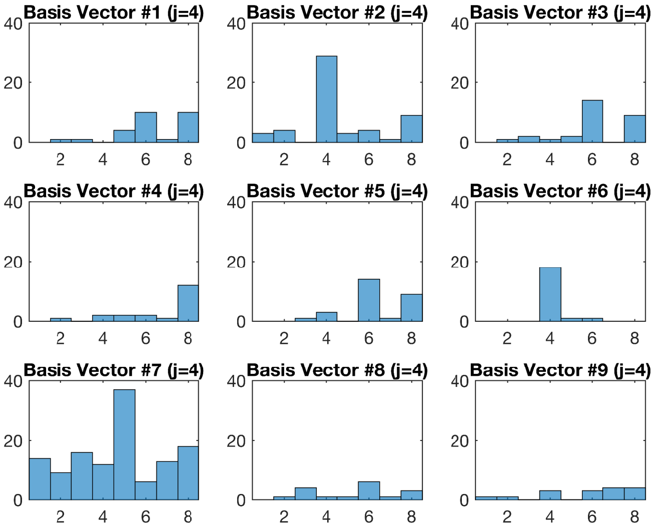
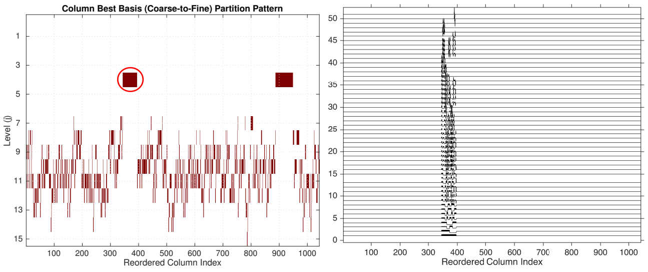


Fig. 5. The document category distributions based on the absolute value of the expansion coefficients of those nine basis vectors in Fig. 4b; the horizontal axis indicates the category number (1–8) while the vertical axis represents the number of the documents belonging to those categories.

As for the column best basis, Fig. 6a shows its partition pattern, which is selected from the coarse-to-fine GHWT dictionary: the vertical axis indicates the scale (or level) information j from the top ($j = 0$: the coarsest) to the bottom ($j_{\max}^{\mathrm{col}} = 15$: the finest) whereas the horizontal axis indicates the reordered column indices. One can see that many of the coefficients are at rather fine scales, i.e., $7 \le j \le 13$, which is quite a contrast to the row best basis pattern shown in Fig. 4a. It is interesting to see that the column best basis tends to group the documents rather finely despite the fact that the weight is less severe ($\beta^{\mathrm{col}} = 0.15$; see Fig. 6a), whereas the row best basis groups the words more coarsely ($\beta^{\mathrm{row}} = 1.0$; see Fig. 4a). In this figure, two coarse scale blocks stand out; they correspond to the node set V_5^4 and V_{14}^4. Fig. 6b shows the column best basis vectors corresponding to the coefficients marked by the circle in Fig. 6a, whose support is V_5^4 with $|V_5^4| = 51$ documents. Out of these 51 documents, 48 belong to Category 2 (Astronomy). The remaining three turn out to be the following:

- *"Old Glory, New Glory: The Star-Spangled Banner gets some tender loving care"* (Category 1: 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"* (Category 5: Life Sciences; on star-nosed moles)
- *"Gravity tugs at the center of a priority battle"* (Category 6: Math/CS; on the priority war on the discovery of gravity between Newton, Halley, and Hooke)

It is not surprising why these three documents are picked up by the basis vectors supported on V_5^4: they contain at least a few astronomical terms. In order to check how these column best basis vectors relate to the words, we compute $A\boldsymbol{\psi}_{5,0}^{4,\mathrm{col}}$, i.e., the expansion coefficients of all the row vectors of A w.r.t. $\boldsymbol{\psi}_{5,0}^{4,\mathrm{col}}$, which is the indicator vector of these 51 documents. Fig. 7 displays these expansion coefficients, which indicate how these 51 documents are related to those coefficients. The expansion coefficients exceeding 0.05 in this 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*. Clearly, the majority of them are highly relevant terms in astronomy! A similar experiment on another standout block V_{14}^4 in Fig. 6a, where $|V_{14}^4| = 62$, results in the following observation. 56 documents among these 62 indicate Category 7 (Medicine).



(a) The best basis partition (b) The basis vectors on V_5^4

Fig. 6. The *column* (document) best basis (coarse-to-fine)

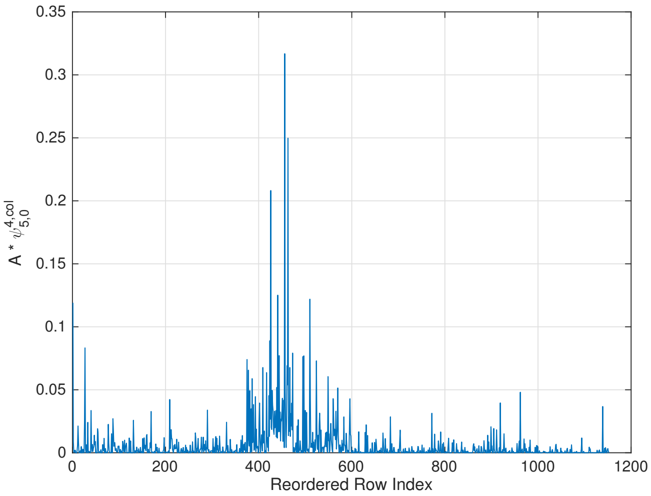


Fig. 7. The expansion coefficients of row vectors w.r.t. the column best basis vector $\boldsymbol{\psi}^{4,\mathrm{col}}_{5,0}$ = the indicator vector of 51 documents.

Out of these six anomalies, three are in Category 5 (Life Sciences), which is not quite surprising. The remaining three documents are:

- *"In Silico Medicine: Computer simulations aid drug development and medical care"* (Category 6);
- *"Beyond Virtual Vaccinations: Developing a digital immune system in bits and bytes"* (Category 6);
- *"Paleopathological Puzzles: Researchers unearth ancient medical secrets"* (Category 1).

In our opinion, these could have been categorized as Category 7. The significant expansion coefficients of all the row vectors w.r.t. the indicator vector $\boldsymbol{\psi}^{4,\mathrm{col}}_{14,0}$ 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.* Again, the majority of them are clearly related to medical sciences.

6. CONCLUSION

In this paper, we proposed a method to learn the sparsity of a given matrix and the interrelationship between its rows and columns using our GHWT dictionaries, and demonstrated their potential usefulness for matrix data analysis using the Science News term-document matrix. We are currently investigating cost functionals other than ℓ^p-(quasi)norms; a more extensive best basis algorithm; and basis vector selection algorithms that are fundamentally different from the best basis algorithm, and we hope to report our results at a later date.

7. REFERENCES

[1] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th International Conference on Knowledge Discovery and Data Mining*. ACM, 2001, pp. 269–274.

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

[3] J. Irion and N. Saito, "The generalized Haar-Walsh tranform," in *Proc. 2014 IEEE Workshop on Statistical Signal Processing*, 2014, pp. 472–475.

[4] J. Irion and N. Saito, "Applied and computational harmonic analysis on graphs and networks," in *Wavelets and Sparsity XVI, Proc. SPIE 9597*, M. Papadakis, V. K. Goyal, and D. Van De Ville, Eds., 2015, Paper # 95971F.

[5] J. L. Irion, *Multiscale Transforms for Signals on Graphs: Methods and Applications*, Ph.D. thesis, Appl. Math., Univ. California, Davis, Dec. 2015.

[6] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–719, 1992.

[7] J. C. Bremer, Jr., *Adaptive Multiscale Analysis of Graphs and Applications*, Ph.D. thesis, Yale Univ., May 2007.

[8] C. E. Priebe, D. J. Marchette, Y. Park, E. J. Wegman, J. L. Solka, D. A. Socolinsky, D. Karakos, K. W. Church, R. Guglielmi, R. R. Coifman, D. Lin, D. M. Healy, M. Q. Jacobs, and A. Tsao, "Iterative denoising for cross-corpus discovery," in *COMPSTAT 2004—Proceedings in Computational Statistics*, pp. 381–392. Physica-Verlag HD, 2004.

[9] R. 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., Boston, MA, 2011, Applied and Numerical Harmonic Analysis, pp. 161–197, Birkhäuser.

[10] R. R. Coifman and W. Leeb, "Earth mover's distance and equivalent metrics for spaces with hierarchical partition trees," Tech. Rep. YALEU/DCS/TR-1482, Yale Univ., 2013.

[11] J. I. Ankenman, *Geometry and Analysis of Dual Networks on Questionnaires*, Ph.D. thesis, Yale Univ., May 2014.