

Notes

April 9, 2018

1 Things to fix:

1. (fixed) Undefined function 'findpeaks' for input arguments of type 'double'.
Error in mcsfb_design_filter_bank_no_fourier (line 31) [, idx] = findpeaks(inverted);
2. (fixed) Warning: Matrix is close to singular or badly scaled (Reconstruction using eigenvalue and eigenvectors)
3. filter design function - does not perform well

2 Notations

Consider a weighted, undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$, where \mathcal{V} is the set of N vertices, \mathcal{E} is the set of edges and W is the associated weighted adjacency matrix. Let D be the diagonal matrix of vertex degrees. Denote \mathcal{L} as the graph Laplacian, where $\mathcal{L} = D - W$. Since \mathcal{L} is real, symmetric and semi-definite, we can diagonalize it as $\mathcal{L} = U\Lambda U^*$, where Λ is the diagonal matrix of its real eigenvalues $\lambda_0, \lambda_1, \dots, \lambda_{N-1}$ and the columns u_0, u_1, \dots, u_{N-1} of U are corresponding orthonormal eigenvectors of \mathcal{L} . We use $U_{\mathcal{R}}$ to denote the submatrix formed by taking the columns of U associated with the Laplacian eigenvalues indexed by $\mathcal{R} \subseteq \{0, 1, \dots, N-1\}$. And we use $U_{\mathcal{S}, \mathcal{R}}$ to denote the submatrix formed by taking the rows of $U_{\mathcal{R}}$ associated with the vertices indexed by the set $\mathcal{S} \subseteq \{1, 2, \dots, N\}$. For $i \in \{1, \dots, n\}$, $\delta_i \in \mathbb{R}^n$ denotes the i th column of the identity matrix $I \in \mathbb{R}^{n \times n}$. We consider graph signals $f \in \mathbb{R}^N$ residing on a graph \mathcal{G} . A signal $f \in \mathbb{R}^n$ defined on the nodes of the graph \mathcal{G} is \mathcal{R} -concentrated if $f \in \text{span}(U_{\mathcal{R}})$. The graph Fourier transform of a signal is $\hat{f} = U * f$ and $\hat{g}(\mathcal{L})f = U\hat{g}(\Lambda)U * f$ filters a graph signal by $\hat{g}(\cdot)$. Signal $r \in \mathbb{R}^N$ is a random vector such that each entry is randomly draw from a standard gaussian distribution.

3 Definitions

Definition. A signal $f \in \mathbb{R}^n$ defined on the nodes of the graph \mathcal{G} is \mathcal{R} -concentrated if $f \in \text{span}(U_{\mathcal{R}})$.

Definition. Let $p \in \mathbb{R}^n$ represent a sampling distribution on $\{1, 2, \dots, n\}$. The graph weighted coherence of order $|\mathcal{R}|$ for the pair (\mathcal{G}, p) is

$$v_p^k = \max_{1 \leq i \leq n} \{p_i^{-1/2} \|U_{\mathcal{R}}^T \delta_i\|_2\}$$

4 Theorems

Theorem (Orthogonality Proof). Let $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_M\}$ be M partitions of the graph Laplacian eigenvalue indices $\{0, 1, \dots, N\}$ and $\{g_1, g_2, \dots, g_M\}$ be filters defined on each of the M bands. Then, the subspace spanned by $g_i(\mathcal{L})$ is orthogonal to the subspace spanned by $g_j(\mathcal{L})$ for $i \neq j$

Proof. Notice that the (i, j) entry of matrix $g(\mathcal{L})$ can be represented as

$$g(\mathcal{L})(i, j) = [Ug(\Lambda)U^T](i, j) = \sum_{k=1}^N g(\lambda_k) U_k(i) U_k(j)$$

Now consider two filters g_1 and g_2 . To show that two subspaces spanned are orthogonal, we show that the dot product of any two vectors is 0.

$$\begin{aligned} g_1(\mathcal{L})(\cdot, k) \cdot g_2(\mathcal{L})(\cdot, l) &= \sum_{j=1}^N \left[\sum_{k=1}^N g_1(\lambda_k) U_k(i) U_k(j) \right] \left(\sum_{l=1}^N g_2(\lambda_l) U_l(i) U_l(j) \right) \\ &= \sum_{k=1}^N \sum_{l=1}^N g_1(\lambda_k) g_2(\lambda_l) U_k(i) U_l(i) \left(\sum_{j=1}^N U_k(j) U_l(j) \right) \\ &= \sum_{k=1}^N g_1(\lambda_k) g_2(\lambda_k) U_k(i) U_k(i) \left(\sum_{j=1}^N U_k(j) U_k(j) \right) \end{aligned}$$

Notice that $g_1(\lambda_k)g_2(\lambda_k)$ is always equal to 0, since g_1 and g_2 are defined on disjoint sets of Laplacian eigenvalues.

□

5 Abstract

We investigate an M -channel critically sampled filter bank for graph signals where each of the M filters is supported on a different subband of the graph Laplacian spectrum. For analysis, on each subband, the graph signal is filtered using efficient polynomial approximation methods and then downsampled on a corresponding set of vertices chosen via non-uniform random sampling. We use an efficient decoder derived from the non-uniform sampling distribution to reconstruct the filtered signals on each band from their samples. We leverage an approximation of the spectral density function of the graph Laplacian to both approximate the number of eigenvalues in each band and design the filter bank to be more amenable to polynomial approximation, in order to reduce the resulting reconstruction error. We empirically explore the joint vertex-frequency localization of the dictionary atoms, the sparsity of the analysis coefficients, and the ability of the proposed transform to compress piecewise-smooth graph signals.

6 Filter Bank Design

In this section, we construct a scalable M -channel critically sampled filter bank where each of the M filters is supported on a different subband of the graph Laplacian spectrum. The ideal filter bank design requires full eigendecomposition the graph Laplacian, which takes $\Theta(N^3)$ operations with regard to the size of the graph. In order to circumvent these expensive computations, we estimate the spectral density function of the graph Laplacian and then adopt it to efficiently design the the filter bank. The filters are then stored and applied using the polynomial approximation methods discussed in [?, ?].

6.1 Estimate Spectral Density Function

Various methods have been proposed by past literature to estimate the spectral density functions, including the spectrum slicing method[?, ?], the Lanczos algorithm, the kernel polynomial method [?] and etc. We direct readers to [] for more information. Here, we give a brief overview of the KPM, which is the method we use in our implementation.

Given a filter $\hat{g}(\cdot)$, we notice that $\hat{g}(\mathcal{L})$ is a diagonal matrix, where $\hat{g}(\mathcal{L})_{i,i} = 1$ if the i th eigenvalue is covered by the filter and $\hat{g}(\mathcal{L})_{i,i} = 0$, otherwise. By computing the trace of $\hat{g}(\mathcal{L})$, which is the sum of the diagonal, we obtain the number of eigenvalues in the band specified by $\hat{g}(\cdot)$. This method scales linearly with regard to the size of the graph. In practice, the trace can be approximated using the following fact:

$$\text{tr}(\hat{g}(\mathcal{L})) = E[r^T \hat{g}(\mathcal{L}) r] \propto \frac{1}{I} \sum_{i=1}^I (r^i)^T \hat{g}(\mathcal{L}) r^i$$

We pick n linearly spaced points (x_1, \dots, x_n) over the entire spectrum. Using the trace estimator, we can easily compute the number of eigenvalues, y_i , in $[0, x_i]$ for $0 \leq i \leq n$. Finally, we use the monotonic cubic polynomial interpolation method with interpolation points $\{(x_i, y_i)\}_{0 \leq i \leq n}$ to form a smooth function that estimates the cumulative spectral density function.

6.2 Fast Filtering

Recall that $\hat{g}(\cdot)$ filters a signal f by $\hat{g}(\mathcal{L})f = U\hat{g}(\Lambda)U^T f$, where $\hat{g}(\Lambda) = (\hat{g}(\lambda_1), \dots, \hat{g}(\lambda_n))^T$. To avoid the computation of U and Λ in the filtering process, one common approach is to approximate the exact filter $\hat{g}(\cdot)$ by a polynomial function:

$$\hat{g} = \sum_{k=0}^{\infty} c_k T_k.$$

In our implementation, we allow two different polynoial approximation methods: Chebyshev and Jackson-Chebyshev approximations. For example, the truncated Chebyshev polynomial approximation can be represented as

$$\hat{g}(\mathcal{L})f = \frac{1}{2}c_0 f + \sum_{k=1}^{\infty} c_k \bar{T}_k(\mathcal{L})f \propto \frac{1}{2}c_0 f + \sum_{k=1}^K c_k \bar{T}_k(\mathcal{L})f$$

and each $\bar{T}_k(\mathcal{L})f$ can be efficiently computed recursively from $\bar{T}_{k-1}(\mathcal{L})f$ and $\bar{T}_{k-2}(\mathcal{L})f$.

Jackson-Chebyshev method adds in one more damping multiplier, g_j^K , to alleviate the Gibbs oscillations near the boundaries:

$$\hat{g}(\mathcal{L})f \propto \frac{1}{2}g_0^K c_0 f + \sum_{k=1}^K g_k^K c_k \bar{T}_k(\mathcal{L})f.$$

For the scalable filter bank, instead of storing the exact filters, we store the coefficients of Chebyshev or Jackson-Chebyshev polynomial approximations and the filters will be applied to any given signal accordingly. See [?], [?] for more information on polynomial approximation and [?] for more fast filtering techniques. Two examples of polynomial approximated filters using Jackson-Chebyshev coefficients are shown in the figure below.

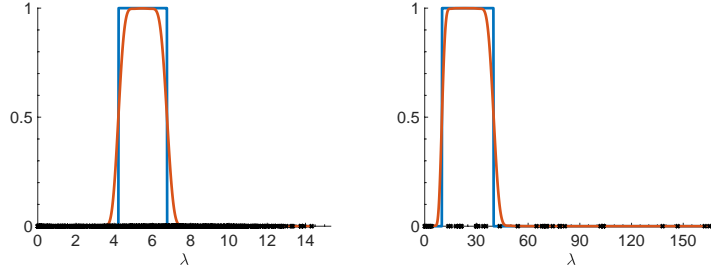


Figure 1: The first two examples are ideal and approximated filter banks for a more evenly distributed spectrum. The other two examples are for a graph with noticeable spectral gaps.

6.2.1 Adjust Band Ends to Minimize Approximation Error

When selecting the band ends for each of the M ideal filters, we consider two factors: spectrum-adaptation and band spacing method. In our implementation, the filter bank can be either spectrum-adapted or not, and it can be either evenly spaced or logarithmic spaced.

Without a full knowledge of the spectrum of the graph Laplacian, we cannot pick the band ends the way as we do for ideal filter banks. We first pick the band ends based on the approximated spectral functions. While satisfying the spectrum-adaptation and spacing requirements, we shift our initial choices of band ends in a small interval in order to reduce the resulting reconstruction error.

Consider the exact filter \hat{g} and the polynomial approximated filter $\hat{h}(\mathcal{L})$. If K is the order of the polynomial approximation, the filtering error is

$$\|\hat{g}(\mathcal{L}) - \hat{h}(\mathcal{L})\| = \max_{l=0, \dots, n} |g(\lambda_l) - h(\lambda_l)| \leq \sup_{\lambda \in [0, \lambda_{max}]} |g(\lambda) - h(\lambda)| = \Omega(\rho^{-K})$$

Notice that for Chebyshev and Jackson-Chebyshev approximations, the filtering error is always clustered around band boundaries (figure 1). If the boundaries are set in spectral gaps, i.e. there are no or very few eigenvalues around it, the approximated filters will not affect the filtering accuracy too much, thus leading to a small reconstruction error.

Our approach to locate the spectral gaps is simply to find the minimum of the probability distribution of Laplacian eigenvalues over a specified interval. Let the w be the width of the adjustment interval. For an initial choice of the band end $m \in [\tau, \tau + w]$, we intend to find τ such that the cumulative density function $F(\tau + w) - F(w)$ is minimized. That is, we want to minimize the probability of eigenvalues existing in the interval $[\tau, \tau + w]$. When w is small, it is equivalent to find the τ such that the probability distribution of the Laplacian eigenvalues $p(\tau)$ is minimized. This process is illustrated in the figure below.

7 Non-Uniform Sampling and Reconstruction

In this section, we select the downsampling sets using the non-uniform random sampling method and reconstruction the original signal from samples based on the the decoder derived from the

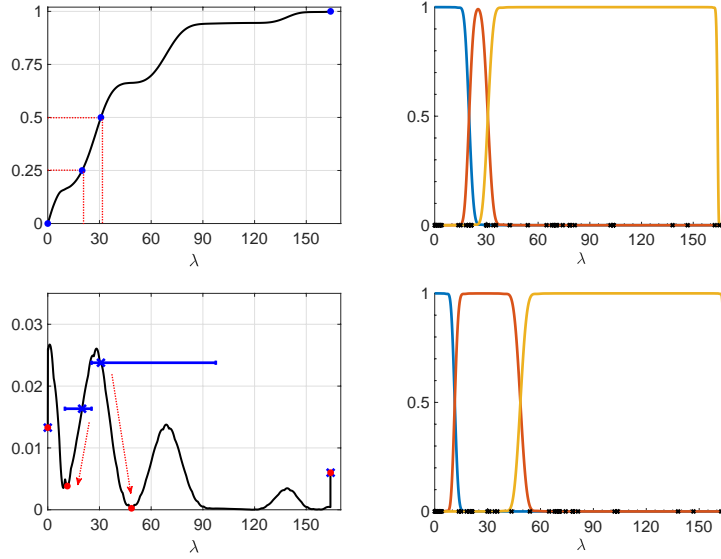


Figure 2: (a) shows the initial band end choices for a spectrum-adapted and logarithmic spaced filter bank of 3 filters. (b) is the approximated filter bank based on the initial choices. In (c), the green dots indicate the shifted band ends and (d) is the update filter bank.

sampling distribution. This method is first studied in [?] only for bandlimited signals. We extend this idea to all \mathcal{R} -concentrated signal models.

First, we need to determine the number of samples needed for each band in order to ensure critically sampling and perfect reconstruction. It has been shown in [?] that any signal whose graph Fourier transform has exactly k non-zero coefficients can be perfectly recovered from samples of that signal on k appropriately selected vertices. That is, the number of samples needed equals the number of eigenvalues covered in the band. We use the spectral density function estimated in the previous section to approximate the number of eigenvalues in each band. TODO: force the number of samples in the last band. Or use the overcomplete transform

The goal of the non-uniform random sampling method is to minimize the graph weighted coherence by imposing a sampling distribution on graph vertices. Graph weighted coherence is a quantity that measures of the localization of some Fourier modes on the nodes of the graph. The probability of node i being selected is $p_i \propto \|U_{\mathcal{R}}^T \delta_i\|_2$, which characterizes how much energy of δ_i is concentrated on the Fourier modes covered in \mathcal{R} . When this quantity is large, we would want high probability of selecting this node since there exists \mathcal{R} -concentrated signals whose energy is solely concentrated at the i th node. Including this node in our sample will increase our chance of perfect reconstruction later.

Theorem ([?], Theorem 3.2). Consider any graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, W\}$ with its Laplacian $\mathcal{L} = U\Lambda U^*$. Notice that \mathcal{L} is real, symmetric and positive semi-definite and thus the columns of U are orthonormal and its eigenvalues are non-negative. We propose the following sampling procedure:

1. Sampling Distribution. Let $\mathcal{R} \subseteq \{0, 1, \dots, n-1\}$. Define $\mathbf{p} \in \mathbb{R}^n$ as the sampling distribution on vertices $\{1, 2, \dots, n\}$ such that

$$\mathbf{p}_i = \frac{\|U_{\mathcal{R}}^T \delta_i\|_2^2}{|\mathcal{R}|}, \text{ for } i = 1, 2, \dots, n. \quad (1)$$

The sampling distribution \mathbf{p} minimizes the graph weighted coherence. We associate the matrix $P = \text{diag}(\mathbf{p}) \in \mathbb{R}^{n \times n}$.

2. Subsampling Matrix. Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_m\}$ be the subset of nodes drew independently from the set $\{1, 2, \dots, n\}$ according to the sampling distribution \mathbf{p} . Define M as the random subsampling matrix with the sampling distribution \mathbf{p} such that

$$M_{ij} = \begin{cases} 1 & \text{if } j = \omega_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

for all $i \in \{1, 2, \dots, m\}$ and $j \in \{1, 2, \dots, n\}$.

From the m samples obtained using the sampling method above, we can reconstruction all \mathcal{R} -concentrated signals accurately by solving the optimization problem

$$\min_{\mathbf{f} \in \text{span}(U_{\mathcal{R}})} \|P_{\Omega}^{-1/2}(M\mathbf{f} - \mathbf{y})\|_2, \quad (3)$$

which estimates signal $\mathbf{f} \in \mathbb{R}^n$ from $\mathbf{y} \in \mathbb{R}^m$.

Let $\epsilon, \delta \in (0, 1)$. Let \mathbf{f}^* be the solution of the problem (3). With probability at lease $1 - \epsilon$, the following holds for all $\mathbf{f} \in \text{span}(U_{\mathcal{R}})$ and all $\mathbf{n} \in \mathbb{R}^m$:

$$\|x^* - x\|_2 \leq \frac{2}{\sqrt{m(1-\delta)}\|P_{\Omega}^{-1/2}\mathbf{n}\|_2} \quad (4)$$

provided that

$$m \geq \frac{3}{\delta^2} v_{\mathbf{p}}^k \log\left(\frac{2k}{\epsilon}\right). \quad (5)$$

8 Illustrative Examples II: Approximate Calculations

- Make sure to have some very large examples and show computation times