

Report on Paper "Graph Spectral Image Processing"
by Gene Cheung et al.

Sourabh Antani

Introduction

This paper provides an overview of recent graph spectral techniques in GSP specifically for image/video processing. The topics covered include image compression, image restoration, image filtering, and image segmentation.

Definitions and Basics

Graph A graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ contains a set \mathcal{V} of N nodes and a set \mathcal{E} of M edges. Each existing edge $(i, j) \in \mathcal{E}$ is undirected and contains an edge weight $w_{i,j}$, which is typically positive; a large positive $w_{i,j}$ would mean that samples at nodes i and j are expected to be similar/correlated.

Graph Signal x on \mathcal{G} is a discrete signal of dimension N , one sample $x_i \in \mathbb{R}$ for each node i in \mathcal{V} . Assuming nodes are appropriately labeled from 1 to N , we can treat a graph signal simply as a vector $x \in \mathbb{R}^N$.

Edge weight Matrix \mathbf{W} $W_{i,j} = w_{i,j}$, has 0 on diagonal unless graph contains self loops, i.e. an edge from a node to itself.

Diagonal Degree Matrix \mathbf{D} $d_{i,i} = \sum_j W_{i,j}$.

Combinatorial Graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$. \mathbf{L} is symmetric, and PSD if all weights are positive, and hence, can be decomposed into $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ where $\mathbf{\Lambda}$ is a diagonal matrix containing real (and non-negative for PSD) eigenvalues λ_k . \mathbf{U} is the eigenmatrix composed of orthogonal eigenvectors \mathbf{u}_i as columns. the eigenvalues and eigenvectors here can be interpreted as **graph frequencies** and **graph frequency components** respectively. Together, they define **graph spectrum** for graph \mathcal{G} . \mathbf{U} can be used to decompose a graph signal x into its frequency components $\alpha = \mathbf{U}^T x$.

???WHY $D - W$? Why not $D + W$???

Smooth Signal A signal \mathbf{x} is considered smooth if the if the sample on a node x_i is similar to the samples x_j on neighboring nodes with large $w_{i,j}$. In frequency domain, this means that graph has mostly low frequency compnents and the coefficients $\alpha = \mathbf{U}^T x$ are zero for high frequency components. The smoothest signal is a constant vector and is the first eigen vecor corresponding to the eigen value 0. Expressing \mathbf{x} as $x_i - x_j$, smooth signal would mean that when the frequency (= eigen value λ) is high, the signal would be similar ($x_i - x_j$ is small). Hence the graph regularizer $\mathbf{x}^T \mathbf{L} \mathbf{x}$ is small and since \mathbf{L} is PSD, it is lower bounded by 0 which is achieved when \mathbf{x} is scalar multiple of the constant eigen vector.

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{(i,j) \in \mathcal{E}} w_{i,j} (x_i - x_j)^2 = \sum_k \lambda_k \alpha_k^2$$

If \mathbf{W} is interpreted as shift operator, the smoothness can be defined as the p -norm of the difference between the signal and its shifted version i.e. $\|\mathbf{x} - \frac{1}{|\lambda_{max}|} \mathbf{W} \mathbf{x}\|_p^p$

Image Compression

Image compression refers to the process of encoding an image \mathbf{x} onto a codeword $c()$, minimizing distortion in the reconstructed image for a given target bit-rate $R(t)$. Traditionally, this is done using lossy compression via 2-D transform that generates a vector of transform coefficient whose components are mostly small with only a few significantly different from 0.

There are many versions of a scheme where the coefficients α are quantized and loosely coded using compression algorithm. However the decoding requires the knowledge of transform used for encoding. Due to this, either a pre-determined transform must be used or the information about transform has to be supplied in addition to the compressed data, thus adding to the bit-rate.

Seeking optimal transform is an open research area. Discrete Cosine Transform (DCT) and wavelets use fixed set of basis vectors so these don't need to be communicated to decoder.

Karhunen-Loève transform (KLT) is based on eigendecomposition of covariance matrix (similar to Principle Component Analysis). It has been shown to be optimal for a Gaussian source under mean square error metric. KLT is signal-adaptive since the columns of \mathbf{U} (basis vectors) can be matched to a class of input signal. The drawbacks are: the transform matrix has to be communicated to decoder, the transform does not have a specific structure, hence a fast algorithm cannot be applied.

GFT is also based on eigen vector decomposition but is more flexible by encoding pixel similarities into weights of an undirected graph. This enables use of GFT on any types of image and also one can decide the degree of accuracy of encoded image. Graph Laplacian is also PSD which enables use of fast algorithms. Additionally commonly occurring transforms can be indexed which would require only the index information to be transmitted, thus reducing the overhead.

Graph Design is the process of choosing the graph topology and edge weights that make up the matrices used for GFT. Edge weight is conventionally computed as a function of photometric distance (difference in pixel values, in color components). However this can become very expensive. One scheme to reduce the overhead is to constrain the weights to 0,1 to simply encode if pixels are similar or not, or by using greedy optimization on edge detection or image segmentation algorithms. Yet another approach would be to segment the image into blocks and use quantizer optimized using probability density function and encode the blocks where the 'similarity' distribution is extreme (either strong or none) using weights of 0,1 and using $1, c$ where the similarity distribution is strong or weak.

Graph Learning is process of defining a 'good graph' from data. Many methods are referenced but none described by the authors. *Need to review few references to understand.*

Reducing GFT Complexity some of the approaches are using index for commonly used transforms (this works for small block sizes), apply GFT to low-resolution images and use edge-adaptive filter to restore original resolution and exploiting graph symmetries where possible.

Directional Model is another popular way to reduce the complexity and overhead. The idea is to treat DCT as 2D square grid graph. In this case the eigen values are not distinct. In this case, the eigen bases for the Laplacian can be represented as rotation of the original vector. The angle of rotation must be chosen to match the directional characteristics of the image block. In this way, the angle encodes a large amount of detail in the image, thus reducing the overhead.

Applications of GFT range from video compression, inter- and intra-predicted video coding, multivideo

view etc.

Image Restoration

Image restoration is an inverse problem. Given a degraded image, one is tasked with restoring the original signal, e.g. denoising, interpolation, super-resolution, deblurring etc. A generic model can be viewed as $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$ where \mathbf{y} is degraded observation of the original \mathbf{x} , \mathbf{H} is the degradation matrix and \mathbf{z} is additive noise.

Image Denoising: $\mathbf{H}=\mathbf{I}$, \mathbf{z} is additive white Gaussian noise. Using Bayesian approach, a typical maximum *a posteriori* formulation as

$$\min_x ||y - x||_2^2 + \mu R(x)$$

where $R(x)$ is the negative log of signal prior or regularization term and μ is the weight parameter. The crux is to define a prior $R(x)$ that discriminates target signal against other candidates while keeping optimization computationally efficient.

Alternately, denoising can be viewed as filtering problem

$$x = D^{-1}W y$$

where $D^{-1}W$ is row stochastic and filter coefficients in \mathbf{W} are designated adaptively based on local/nonlocal statistics.

Sparsity of GFT Coefficients : One approach is to map an observed signal to a preselected transform domain assuming sparse signal representation and perform hard/soft thresholding on transform coefficients. Probabilistically, graph Laplacian can be interpreted as inverse covariance matrix of GMRF and hence de-correlates the input signal. Thus, appropriately chosen GFT can sparsify an signal resulting in smaller l_0 -norm. Assuming self-similarity, average patch \bar{y} is computed using $N - 1$ similar patches to a target patch y . For each pixel, the weight $w_{i,j}$ for each of the edges connecting to neighboring pixels is calculated using photometric distance. it can be shown that the computed eigenvectors for the image are robust to noise. The N similar patches are denoised jointly as:

$$\min_{\alpha} \sum_{i=1}^N ||y_i - U\alpha_i||_2^2 + \tau \sum_{i=1}^N ||\alpha_i||_0$$

where the weight parameter τ can be estimated using Stein unbiased risk estimator. The second term can be minimized using iterative soft-thresholding.

Graph Laplacian Regularizer : $R(x) = x^T L x$. This can also be interpreted as Tikhonov regularizer $||\Gamma x||_2^2$ where $\Gamma = U\Lambda^{1/2}U^T$ given $L = U\Lambda U^T$. Minimizing the regularizer means that the connected pairs by large edge weights will have similar sample values, i.e. energy of the signal resides mostly on low frequencies. The optimal solution x^* is calculated as

$$x^* = U \text{diag}\left(\frac{1}{(1 + \mu\lambda_1)}, \dots, \frac{1}{(1 + \mu\lambda_N)}\right) U^T y.$$

The resulting low-pass filter on y in GFT domain - smaller filter coefficient $(1 + \mu\lambda_i)^{-1}$ for larger λ_i - can be implemented efficiently using Chebychev polynomial approximation.

Alternatively, assuming that the Hermitian of the weight matrix \mathbf{W}^* is a polynomial of the matrix \mathbf{W} , the optimal MAP denoising filter with l_2 -norm is derived without matrix inversion as $g(\lambda_n) = \frac{1}{1 + \mu(1 - \lambda_n)^2}$

Also, the Laplacian regularizer can be interpreted from a continuous manifold perspective where weights are computed as gaussian of different in set of exemplar functions $f()$. The regularizer converges to a continuous functional $\int_{\Omega} \nabla x^T G^{-1} \nabla x (\sqrt{\det(G)})^{2\gamma-1} ds$ where $G = \sum_{n=1}^N \nabla f_n \nabla f_n^T$. Defining $D = G^{-1} (\sqrt{\det(G)})^{2\gamma-1}$, the optimization can be implemented as anisotropic diffusion $\delta_t x = \text{div}(D \nabla x^*)$. D is the diffusivity that determines how fast the image is diffused. For $\gamma < 1$, it can be shown that diffusion is two step process 1) forward diffusion smooths an image along edge and 2) backward diffusion that sharpens perpendicular to the edge.

In Low-dimension manifold model, the graph is interpreted as patches of d-pixels lying on a d-dimension space lying on a low dimension manifold. This can be iteratively solved using weighted graph Laplacian method .

Graph Total Variation can be optimized using local gradients $\nabla_i x \in \mathbb{R}^N$ as $(\nabla_i x)_j = (x_j - x_i) W_{i,j}$. Here, the isotropic total variation is defined as $\sum_{i \in \mathbb{E}} \|\nabla_i x\|_2$. The TV-norm (defined above) is convex but non-smooth. Hence there are specialized algorithms to minimize it with fidelity term such as proximal gradient algorithms.

Weiner Filter approaches the denoising problem from statistical point of view and minimizes the mean square error. This method also works on non-gaussian noise.

Image Deblurring : H is the blurring operator, which may or may not be known.

One approach is to generate a stochastic matrix by row and column pre and post multiplying W by a diagonal matrix $K = C^{-1/2} W K C^{-1/2}$ so that $I - K$ is symmetric positive semi-definite. This is then used in a minimization problem to deblur the image.

Soft decoding of JPEG images JPEG image encoding is done by transforming 8×8 block via DCT followed by quantizing the coefficients ($q_i = \text{round}(Y_i/Q_i)$) before entropy encoding them. Hence at decoding time, one must find the quantization parameter. This can be done using a random walk graph Laplacian. The iterative graph laplacian promotes PWS and restores the high frequencies of the image recreating the image. For patch restoration, it is desirable to apply the same filtering strength for all the patches. $x^T L_r^T L_r x$ as regularized random walk graph laplacian.

Graph Based Image Filtering

f

Smoothing and Diffusion A simple approach to smoothing is to determine edge weights as euclidian distances and use low-pass filtering to give binary weights to the edges. This method works well but becomes difficult as number of vertices grows. Another approach is to interpret the weights as a heat map and then the solution is given by diffusion equation.

Edge-Preserving Smoothing The bilateral filter can be considered as a graph filter as follows:

$$\hat{x} = (D - 1)Wx = D^{-1/2}U_n(I - \Lambda_n U_n^T D^{1/2}x)$$

With eigenvalues between 0 and 2, the bilateral filter implicitly designs the GFT basis and Graph spectral filter. For example the filter

$$\tilde{h}(\lambda) = \frac{1}{1 + \rho \tilde{h}_r(\lambda)}$$

is a low pass filter if $\tilde{h}_r(\lambda)$ is a high pass filter and is an optimum solution of $\argmin_x \|y - x\|_2^2 + \rho \|H_r x\|_2^2$

Domain transform is performed in 4 steps. 1) compute the pixels as geometric, photometric or combination in each color channel. 2) place the value of the node on the node positions. 3) Perform a low pass filter 4) replace the filtered signal back on the pixels

The deformed pixel position after the domain transform can be viewed as solution to a linear problem. Then the original image recovery amounts to finding the inverse of the linear operator (matrix)

Computation Modern image processing requires very large graph Laplacians. E.g. 4K image contains 8×10^6 pixels and the laplacian has to be square of that number. Hence for speed, the initial eigen values are partially calculated or approximated.

Alternately, the spectral response can be represented as a polynomial in λ . These polynomial can be approximated using chebyshev or other methods.

Image Segmentation

Image segmentation is used for computer vision and also for some of the methods discussed above. The segmentation methods basically are energy-based methods that optimize the energy upon segmentation.

Graph Cut is partitioning of vertices into disjoint and connected sets. The cost of the cut is the sum of weights of the edges that are severed to make the cut.

Max-Flow and Min-Cut : Min-cut is a cut that has the lowest cost. The max-flow/min-cut solution can be found using max-flow algorithm but faster iterative methods are also available.

Normalized Cuts : Min-cut tends to give regions with very few vertices. In order to avoid this, Normalized cut is used. Here the cost is considered to be the fraction of edge weights that are severed to the total edge weight of all edge going from a segment to all vertices in the graph. This becomes an eigen-decomposition problem on Normalized Laplacian matrix. The eigen vector corresponding to second smallest eigen value provides a normalized cut. The one for third smallest eigen value gives the partition of the first two regions.

Mumford-Shah model is a segmentation model to minimize the functional:

$$J(u, \Gamma)_{i=1}^n = \int_{\Omega} (u - f)^2 dx + \beta \int_{\Omega} \Gamma_1 |\nabla u|^2 dx + v \sum_{i=1}^n \int_{\Gamma_1} ds$$

Γ are the interphases.

The graph cut can be achieved in a minimum sense or Discrete sense and algorithms for each type exist.

This image segmentation technique can also be applied to classification and multi-labeling problem.

Graph Bi-Laplacian is the second order graph-laplacian. For example, when a pixel is connected to four neighbors with equal weight, it represents finite element methods.

Bibliography

- [1] Cheung, Gene, Magli, Enrico, Tanaka, Yuichi, Ng, Michael K. "Graph Spectral Image Processing".
Proceedings of the IEEE, vol 106, No. 5, pp. 907-930 May 2018