

Image Graph Spectrum

Chaitanya Patel

CVIT, IIIT-Hyderabad

June, 2017

Outline

Introduction to Image Graph

Image Graph

Definition of Image Graph

Function p

Laplacian

Incident Matrix

Laplacian Matrix

Obtaining optimum p

Laplacian's relation to function p

eigenvectors of L as p

Conclusion

Introduction to Image Graph

- ▶ For image segmentation, clustering, etc. Graph Spectral Analysis has been extensively used.
- ▶ Image is represented as a weighted complete graph. Each pixel represents a vertex.
- ▶ Edge weights are assigned as per affinity between pixels. Affinity can be defined on the basis of various properties :
 - ▶ Intensity difference between two vertices
 - ▶ Gradient difference between two vertices
 - ▶ Difference between some other feature calculated at each pixel
e.g. sift

Image Graph

- ▶ Image Graph is represented as $G(V, E, W)$
- ▶ V contains all image pixels as vertices. If there are total n pixels in the image then $|V| = n$
- ▶ E contains all pairwise relationship between every pair of vertices(pixels) thus making G a complete graph. $|E| = \binom{n}{2}$ for undirected graph
- ▶ The weight $w_{ij} \geq 0$ associated with an edge $(v_i, v_j) \in E$ encodes the affinity between the pixels represented by vertices v_i and v_j . We can collect these weights into an $n \times n$ affinity matrix $W = (w_{ij})_{i,j=1,\dots,n}$

Function p

- ▶ We want to define a function $p : V \rightarrow \mathbb{R}$ such that it is a continuous function i.e. difference between $p(v_i)$ and $p(v_j)$ inversely follows w_{ij}
- ▶ It is equivalent to say that we want to minimize

$$\lambda = \sum_{i=1}^n \sum_{j=1}^n w_{ij} (p(v_i) - p(v_j))^2$$

- ▶ Let Matrix P be defined as
$$\begin{bmatrix} p(v_1) \\ p(v_2) \\ \vdots \\ p(v_{|V|}) \end{bmatrix}$$

Incident Matrix

- ▶ For any directed graph $G(V, E)$, consider

$$V = \{v_1, v_2, \dots, v_{|V|}\}$$

$$E = \{e_1, e_2, \dots, e_{|E|}\}$$

- ▶ Incident Matrix ∇ is $|E| \times |V|$ matrix such that if k^{th} edge is from v_i to v_j with weight w_{ij} then
 - ▶ $\nabla_{ki} = +w_{ij}$
 - ▶ $\nabla_{kj} = -w_{ij}$
 - ▶ $\nabla_{km} = 0, \forall m \neq i, j$

Laplacian Matrix

- ▶ $L = \nabla^T \nabla$ is called laplacian of graph
- ▶ L is $|V| \times |V|$ matrix where

$$L_{ii} = \sum_{j=1}^{|V|} w_{ij}$$

$$L_{ij} = -w_{ij} \quad i \neq j$$

- ▶ $L = D - W$ where D is degree matrix and W is adjacency matrix

Laplacian's relation to function p

- We can show that $P^T LP = \frac{\lambda}{2}$

$$\begin{aligned} P^T LP &= P^T(D - W)P \\ &= P^T DP - P^T WP \end{aligned}$$

- Take $d_{ii} = (i^{th}$ diagonal entry in $D)$ and $p_i = p(v_i)$
- First term is

$$P^T DP = \sum_{i=1}^{|V|} d_{ii} p_i^2$$

- second term is

$$P^T WP = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} p_i p_j w_{ij}$$

Laplacian's relation to function p (cont.)

$$\begin{aligned} P^T L P &= \sum_{i=1}^{|V|} d_{ii} p_i^2 - \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} p_i p_j w_{ij} \\ &= \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij} p_i^2 - \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} p_i p_j w_{ij} \\ &= \frac{1}{2} \left(\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij} p_i^2 + \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij} p_j^2 - 2 \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} p_i p_j w_{ij} \right) \\ &= \frac{1}{2} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} w_{ij} (p_i - p_j)^2 \\ &= \frac{\lambda}{2} \end{aligned}$$

Courant-Fischer Formula

- ▶ Courant-Fischer Formula for any $n \times n$ symmetric matrix A

$$\lambda_1 = \min_{\|x\|=1} (x^T A x)$$

$$\lambda_2 = \min_{\substack{\|x\|=1 \\ x \perp v_1}} (x^T A x)$$

⋮

$$\lambda_{max} = \max_{\|x\|=1} (x^T A x)$$

Here v_1, v_2, v_3, \dots are eigenvectors corresponding to eigenvalues $\lambda_1, \lambda_2, \lambda_3, \dots$ where $\lambda_1 \leq \lambda_2 \leq \lambda_3 \dots$

- ▶ We know $P^T L P = \frac{\lambda}{2}$ and L is symmetric. Thus eigenvectors of L corresponding to smallest eigenvalues, represent such possible P 's that minimizes λ

Conclusion

- ▶ For any image, a weighted graph is constructed considering each pixel a vertex.
- ▶ Edge weights are assigned according to affinity of vertices.
- ▶ Laplacian is obtained from adjacency matrix using formula
$$L = D - W$$
- ▶ Normalized laplacian can be obtained by formula
$$L = I - D^{-\frac{1}{2}} WD^{-\frac{1}{2}}$$
- ▶ Eigen Decomposition of L gives v_1 as a trivial solution and v_2, v_3, \dots as desired solutions

Joint Spectral Correspondence for Disparate Image Matching

Matlab Implementation Details

Chaitanya Patel

CVIT, IIIT-Hyderabad

June, 2017

Outline

Dense SIFT features

Adjacency Matrix

Laplacian and its Eigen decomposition

Reconstruction using Eigenvectors

MSER features extraction and matching

Dense SIFT features

- ▶ Each image is resized to control total number of pixels
- ▶ One way is to use `vl_dsift` function. It will give 128D feature at each keypoint.
 - ▶ binsize 6 pixels
 - ▶ step(stride size) 4 pixels
- ▶ According to paper, sift feature is extracted for each key point for two scales i.e. two binsize : 10 and 6 pixels and concatenate them to create 256D feature at each key point
- ▶ For that, `vl_sift` is called twice for each binsize with keypoints specified in frames

Adjacency Matrix

- ▶ `pdist` is used to create adjacency matrix for intra image pixels
- ▶ `pdist2` is used to create adjacency matrix for inter image pixels
- ▶ cosine distance is used
- ▶ They are concatenated to create joint image graph adjacency matrix

Laplacian and its eigen decomposition

- ▶ Degree Matrix is obtained from Adjacency Matrix
- ▶ Normalized Laplacian is calculated using formula
$$L = I - D^{-1/2}WD^{-1/2}$$
- ▶ `eigs` is used with parameter `sm` to get 5-6 eigenvectors with smallest eigenvalues

Reconstruction using Eigenvectors

- ▶ Eigenvector is divided into two halves - one for each image
- ▶ values are put back to each key point and other values are linearly interpolated using interp2

MSER features extraction and matching

- ▶ `vl_mser` is used for feature extraction
- ▶ `knnsearch` is used for feature matching in both images











































