Spectral Graph Partitioning and its application in image segmentation

Prince Zizhuang Wang

College of Creative Studies, UC Santa Barbara

Abstract. Graph partitioning can be stated as a problem of partitioning a graph G into subgraphs such that the number of edges or the weight sum of the edges crossing the subgraphs is minimized. It has a lot of applications in computer vision, machine learning, parallel computing, and circuits design. In machine learning specifically, a lot of recent works on data mining make use of various graph partitioning algorithms to learn unsupervisely the clusters of large scale data such as social networks and knowledge graphs. Unfortunately, partitioning large scale graph falls into the category of NP-hard problems. Hence, various heuristics algorithms have to be used in order to get an approximated solution. In this paper, we mainly discuss spectral graph partitioning, one of the most important heuristics algorithms aiming at partitioning large scale graphs, which make use of eigenvectors of the Laplacian matrix associated with the graph. We will then discuss some applications of spectral graph partitioning algorithms in image segmentation, a very important task in computer vision field. We will see that partitioning the graph representation of a given image input based on eigenvectors of Laplacian matrix can give us meaningful segments which in many cases divide the salient object in the image from the background. We then tested one of these algorithms over a large scale image dataset designed for image classification. As the spectral graph partitioning algorithm successfully partitions the images into individual objects and backgrounds, we see a great potential of using this as a backbone for more complicated tasks such as unsupervised classification and object detection frameworks.

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introduction

Spectral algorithms has become of the most important heuristics methods for graph partitioning algorithms. It was developed based on Fiedler's work [1] on the algebraic connectivity of a graph. Fiedler suggests that the second smallest eigenvalue of the Laplacian matrix representation of a given graph has a strong relationship with its algebraic connectivity and hence we can partition the graph into two sub parts based on the component of the eigenvector associated with this eigenvalue. We therefore call this eigenvector as Fiedler's vector.

Even though the idea is easy to understand, the intuitive behind spectral method is not very clear, as there is no easy explanation about why partitioning a graph based on Fiedler's vector can divide the original graph into two sub components such that the cut edges across two components are minimized. To fully understand spectral methods, we need analyze some important properties of Fiedler's vector and the cuts based on that. One important theorem of spectral partitioning, as Fiedler suggested, is that two subgraphs resulting from spectral method are always connected. Furthermore, [2] noticed the relation between cheeger constant and Fiedler eigenvalue and suggests that spectral partitioning with smaller Fiedler value produce better partitions. In the next section, we will briefly introduce the problem set up of graph partitioning algorithms, and more details of spectral method.

2 Graph Partitioning

Given a connected graph $G(V, E)$, a partition divides all the vertices into two disjoint sub-graphs, A and B. If we let $|E(A, B)|$ be the number of edges crossing the two sub-graphs, then an optimal cut can then be defined as a bi-partition that minimize the following ratio,

$$\phi(G) = \min \frac{|E(A,B)|}{|A|} \tag{1}$$

where $|A| < n/2$, n is the total number of vertices in graph G. Let the adjacency matrix, $A(G)$ be defined as the a matrix whose entry at (i, j) is 1 if (i, j) is an edge in the graph, and is 0 otherwise. Let D be a matrix where its i^{th} diagonal element is equal to the degree of i^{th} vertex in graph G. Then the Laplacian matrix can be defined as

$$L(G) = D - A$$

The Laplacian matrix of an undirected graph is then symmetric positive semidefinite, meaning that all eigenvalues are non-negative. If the graph is connected, then the second smallest eigenvalue, which we refer to as Fiedler value, is strictly great than 0. In [1], Fiedler related this value with the algebraic connectivity of the graph G. Intuitively, if a given graph has more edges, then its algebraic connectivity is higher since it is 'more connected'. It is observed that the more connected a graph is, meaning that there are edges, then the higher its Fiedler value can be. Generally, Fiedler value gives good partition that minimizes the $\phi(G)$ ratio if the algebraic connectivity of the given graph is small. Later we will see that this is because the ratio in (1) can be bounded according to the eigenvectors of Laplacian matrix $L(G)$.

The method of spectral graph partitioning is very simple. Given the graph G and its Laplacian matrix $L(G)$, we compute and label the eigenvalues of G as $\lambda_1 < \lambda_2 < \lambda_3 ... < \lambda_n$. Let x_2 be the Fiedler vector, that is, the eigenvector corresponds to λ_2. Since there are n vertices and $x_2 \in \mathcal{R}^n$, each vertex can be assigned to a component of x_2. Then, the graph can be bi-partitioned into two subsets G_1 and G_2 if we put vertex i into G_1 if $x_2(i) < s$ and put it into G_2 if otherwise. The choice of the splitting value s can be various according to different tasks. For bisection, s can be simply chosen as the median of all the components of Fiedler vector. And, as suggested by the following theorem given by [3], there exists a s such that the ratio $\phi(G)$ is minimized.

Theorem 1. *Let $G(V, E)$ be a graph on n nodes of maximum degree d, let Q be its laplacian matrix, and let ϕ be the cut ratio. For any vector $x \in \mathcal{R}^n$ such that $\sum x_i = 0$,*

$$\frac{x^T Q x}{x^T x} \geq \frac{\phi^2}{2d}$$

Moreover, there is an s so that the cut had ratio upper bounded by $\sqrt{2d\frac{x^T Q x}{x^T x}}$

It will soon be clear why Fiedler vector can be used to achieve the best ratio. Since the Laplacian matrix is symmetric, the Rayleigh quatient of a vector with respect to Laplacian matrix can be defined as,

$$\frac{x^T L x}{x^T x}$$

. Furthermore, Rayleigh achieves its minimum when x is chosen as the Fiedler vector, that is

$$\lambda_2 = \min \frac{x^T L(G) x}{x^T x}$$

where x is the Fiedler vector. Therefore, by the above theorem, we have

$$\lambda_2 \geq \frac{\phi^2}{2d}$$

Hence, as this was mentioned previously, if the Fiedler value is small, or the graph is less connected, then spectral method can find a partition with good cut ratio.

The question now becomes when we have small Fiedler value to get good partition. [3] proves that for planer graph, the Fiedler value can be upper bounded by $O(1/n)$. More specifically, given a planar graph G with n nodes of degree a most δ, the Fiedler value of G is upper bounded by

$$\frac{8\delta}{n}$$

Then, by Theoreom 1, G has Fiedler cut of ratio bounded by $O(1/\sqrt{n})$.

For graphs in higher dimensions, [3] also proved that the Fiedler value of a subgraph G of an intersection of k-nearest neighbor graph of n points in d dimension is bounded by $O(k^{1+2/d}/n^{2/d})$, and therefore the cut ratio is bounded by $O(k^{1+1/d}/n^{1/d})$. And the Fiedler value of a k-ply neighborhood system is bounded by $\gamma_d \delta^2 (k/n)^{1/d}$, where $\gamma_d = 2(\pi + 1 + \pi/)^2 (A_{d+1}/V_d)^{2/d})$, which gives a cut ratio of $O(\delta(k/n)^{1/d})$. Therefore, for graphs embedded in higher dimensions, the Fiedler value can still be small if the graph is less connected and therefore the bipartition given by spectral method is guaranteed to be good.

Fig. 1. Mask RCNN
Source: https://github.com/matterport/MaskRCNN

3 Graph partitioning and Image segmentation

Like graph partitioning, image segmentation can be viewed as a problem of partitioning an image input into multiple segments. It has a lot of applications in object detection and classification, as many segmentation algorithms can successfully divide objects and the background into two different subsets given the original images. Recently, a lot of works following the machine learning trends are proposed to do the task of segmenting images. In the Mask RCNN paper, [4], the authors make use of a combination of linear classifier and convolutional neural network, where the linear classifier is built with the help of ground truth segments to tell which local parts of the original image contains the objects, and then a convolutional neural network is trained to tell what the object is contained in that local part. While have some decent results, one problem of these CNN-based approach is that it has to rely on a set of pre-defined feature maps to extract local information. Since the number of feature maps is limited and their shapes are restricted, in the end it can only get segments of restricted shapes that only have rectangles. Furthermore, since these models use a linear classifier over parts of the images that only contains local information, it fails to segment the image based on more global information of the input, such as the spatial relations of objects in different parts of the images, the whole background color, the brightness of the entire image, and etc. To segment an image without any help from machine learning model and make use of only the prior knowledge such as the eigenvalues of the matrix representation of an image, we therefore consider the other approach for image segmentation using spectral graph partitioning.

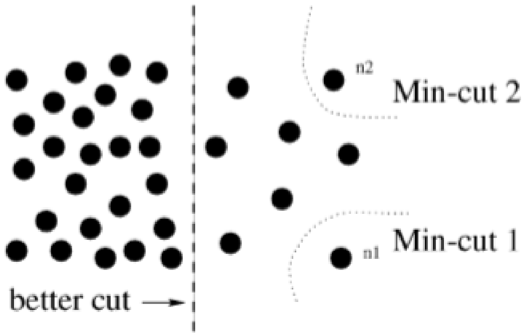

Fig. 2. bad partition with minimum cut
source: https://people.eecs.berkeley.edu/ malik/papers/SM-ncut.pdf

3.1 Spectral graph partitioning and Normalized cuts

[5] proposed an image segmentation method that make use of graph partitioning. The idea simply to convert an image into its graph representation and then partition the graph into subgraphs based on the eigenvectors of its Laplacian matrix. The segmented image can be created by converting the partitioned graphs back to image representation.

To convert an image into its graph representation, each pixel point is interpreted as a node and each pair of node is connected by a weighted edge. The edge between pair of nodes is made by measuring the pixel value and feature similarities between two different pixel nodes. With the brightness and spatial information, the following objective is proposed to define the weight of each edge (i, j),

$$w_{ij} = e^{\frac{-||F_{(i)} - F_j||_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-||X_{(i)} - X_j||_2^2}{\sigma_X^2}} & \text{if} ||X(i) - X(j)||_2 < r \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $X(i)$ stands for spatial location of pixel point i. The spatial similarity is large for pair of pixels points that are close to each other, and will drop down to 0 if two pixels are at distance greater than a threshold value. $F(i)$ stands for other feature information of a given pixel point i, such as brightness, intensity, color, or texture information. Like spatial similarity, pair of pixels with similar feature vectors have larger feature similarity, and therefore larger edge weight.

Given a partition of a graph G with two subgraphs A and B, define the cut value to be the weight sum of edges crossing these two disjoint subsets, that is,

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (3)$$

As explained in the first section, the problem of graph partitioning can then seen as an optimization problem which tries to find a partition that minimizes the cut value between subgraphs. However, it is observed by [6] that cutting graphs based on the above objective often results in cutting small sets of isolated nodes in the graph. Intuitively, if a pixel node in a graph is isolated and far away from other nodes, then by our formulation (2), the edge weight of all the edges connecting this node with all the other nodes will be small. Therefore, the program will favors cutting only this isolated node out, as the its edge weight only contributes a smaller amount to the cut value, which results in smaller $cut(A, B)$.

To address this problem, [5] proposed to used a normalized cut value, which is,

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{4}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is defined as the sum of weight of the edges connecting all the nodes in A with all other nodes in graph G. In this way, isolated pixels will no longer have small cut value, since the cut value of be a large portion of the $assoc$ value from that small set to all the other nodes.

Let D be a diagonal matrix whose diagonal element is the degree of each node. [] demonstrates that minimizing the normalized cut is equivalent to minimizing the Rayleigh quotient with respect to the generalized Laplacian matrix $D - W$,

$$min_x Ncut(x) = min_y \frac{y^T (D - W) y}{y^T D y} \tag{5}$$

where y is setting as $y = (1 + x) - b(1 - x)$, with the condition that $y^T D1 = 0$. Since the Rayleigh quotient with the Laplacian matrix reaches to its minimum when it is the second smallest eigenvalue, the above optimization problem can then be converted a problem of solving a generalized eigenvalue system,

$$(D - W)y = \lambda D y \tag{6}$$

And, since Rayleigh quotient with respect to Laplacian matrix is minimized at Fiedler vector, we have the solution,

$$y_1 = argmin \frac{y^T (D - W) y}{y^T D y} \tag{7}$$

where y_1 is the Fiedler vector of the generalized Laplacian matrix.

The above generalized eigenvalue problem can also be shown to be equivalent to a standard eigenvalue problem,

$$D^{-1/2}(D - W)D^{-1/2}x = \lambda x$$

whose solution can be approximated by Lanczos method.

3.2 Recursive cut

Bisection partitioning is useful and good enough when there is only object in the image and the background is not so various, which, unfortunately, is rarely the case. Ideally, when there are multiple objects in the image, we want our model to be able to partition the images into multiple parts such as each object falls into different subsets.

To do this, a recursive call is used to further partition into each subgraph. For each recursion, check whether the current partition should be further divided by looking at the stability of the cut and the $Ncut$ value. If decide to continue partitioning, then feed the subgraph as an input to an eigenvalue problem to sovle for the Fiedler vector, which is then used to partition this subgraph using spectral method.

4 Experiments on CelebA

We tested Normalized cut over CelebFaces Attributes Dataset (CelebA). CelebA is a large-scale face dataset consists of 200k faces from celebrities. The dataset contains images covering large pose variations and background change. We first create a Region Adjacency Graph representation for each individual face image, and then we call recursively call over normalize cut spectral partitioning method, where we manually set the number of segments for each image to be 400.

Fig. 3. Spectral partitioning over CelebA with 400 cuts

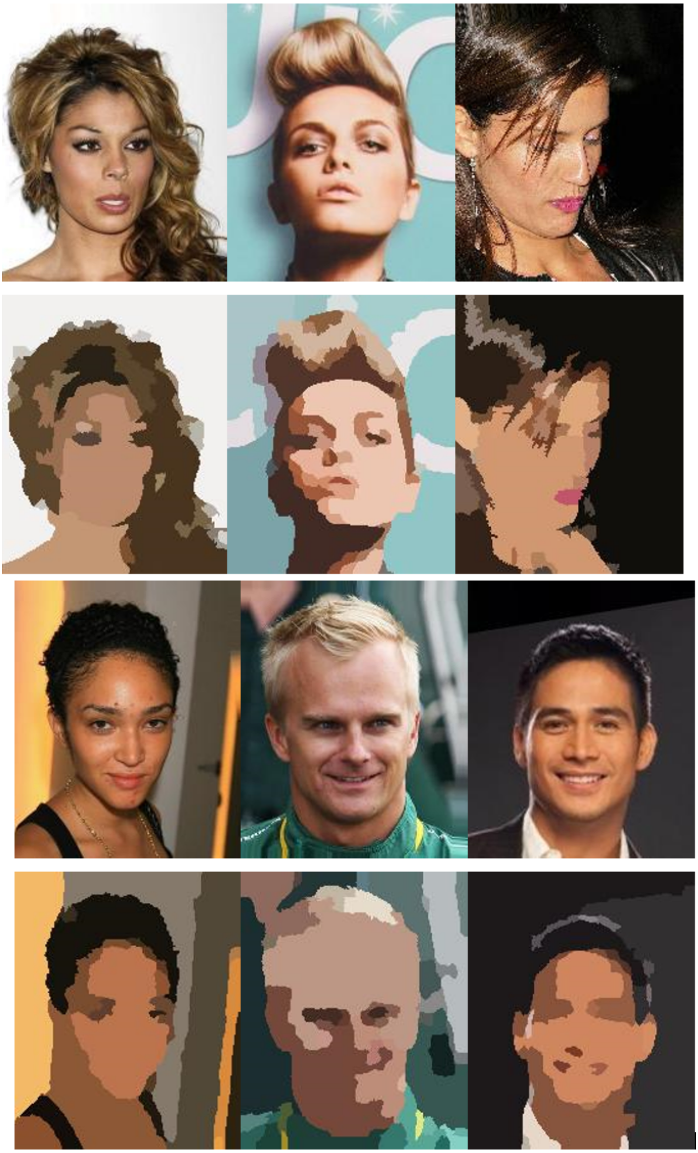

Fig. 4. Spectral partitioning over CelebA with 400 cuts

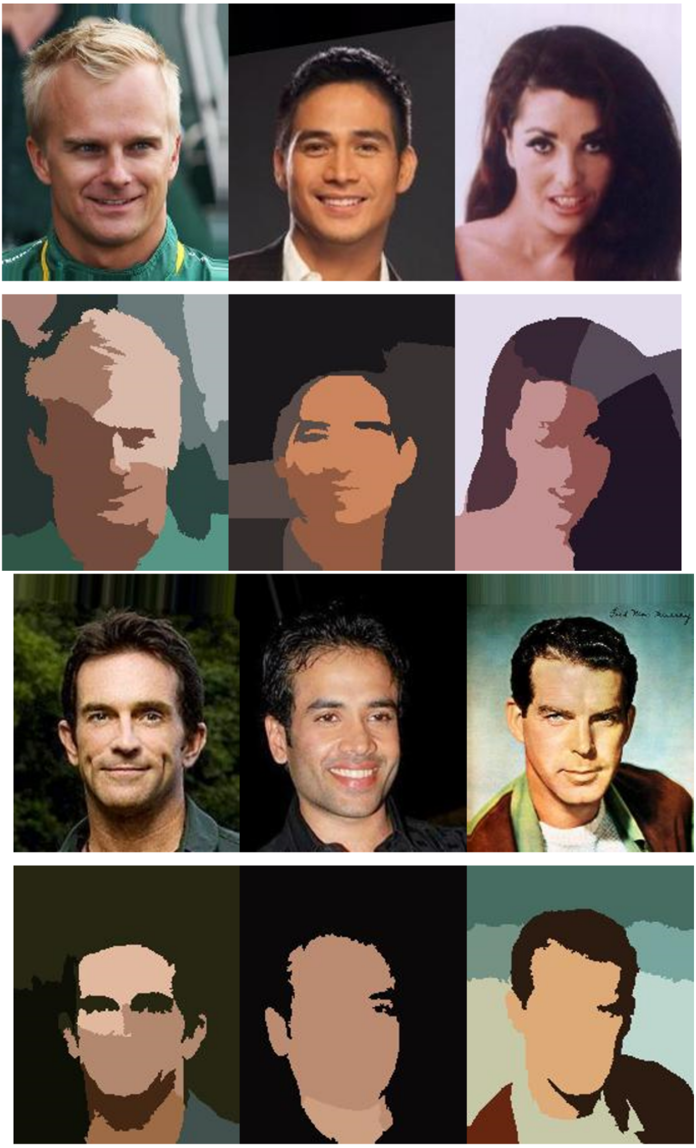

Fig. 5. Spectral partitioning over CelebA with 20 cuts

5 Conclusion and Future Work

We briefly reviewed spectral graph partitioning algorithms and discussed its application in image segmentation tasks. We tested the model over CelebA dataset, and we observed that the model can find meaningful partitions over each face image. Now the problem becomes, with a partitioned image whose subsets contain individual objects, how can we use it to do object detection and classification. This is not a trivial task, as each partition of an image has various shapes, while the input of a classification model such as convolutional neural network is required to have a fixed shape. We plan to explore some recent works on graph convolutional neural network, which does classification over graph inputs. We want to see whether we can use a graph CNN to do classification directly over the partitioned subgraphs, so that we don't need to convert each partitioned subgraph back to image representation for classification task.

References

1. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak mathematical journal **23**(2) (1973) 298–305
2. Spielmat, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. In: Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on, IEEE (1996) 96–105
3. Mihail, M.: Conductance and convergence of markov chains-a combinatorial treatment of expanders. In: Foundations of computer science, 1989., 30th annual symposium on, IEEE (1989) 526–531
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on, IEEE (2017) 2980–2988
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8) (2000) 888–905
6. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE transactions on pattern analysis and machine intelligence **15**(11) (1993) 1101–1113