Image Segmentation Using Maximum Spanning Tree on Affinity Matrix

Qiang He[1] and Chee-Hung Henry Chu[2]
[1]*Department of Mathematics, Computer and Information Sciences*
Mississippi Valley State University
Itta Bena, MS 38941
[2]*The Center for Advanced Computer Studies*
University of Louisiana at Lafayette
Lafayette, LA 70504-4330
USA

1. Introduction

The objective of image segmentation and clustering is to extract meaningful regions out of an image. A graph theoretic approach, as an alternative to the gradient-based methods, is usually based on the eigenvectors of an affinity matrix [3,4,5,6,7]. The theoretical foundation of this development is the Spectral Graph Theory [1], through which the combinatorial features of a graph can be revealed by its spectra. This characteristic can be applied into graph partitioning and preconditioning. The typical eigendecompostion based segmentation work is called the normalized cuts [6]. The normalized cut measure incorporates the local similarity within cluster and total dissimilarity between clusters. The minimization of this measure is to solve the Rayleigh quotient, a generalized eigenvalue solving problem. However, solving a standard eigenvalue problem for all eigenvectors has exponential complexity and is very time consuming. Shi [6] made use of the sparsity of the affinity matrix and introduced the Lanczos method to simplify the computation of eigenvalues.

In this paper, we give a new image segmentation algorithm using the maximum spanning tree [2]. Our method works on affinity matrix and addresses the physical meanings of an affinity matrix. Instead of computations of eigenvalues and eigenvectors, we proved that the image segmentation could be transformed into an optimization problem: finding the maximum spanning tree of the graph with image pixels as vertices and pairwise similarities as weights. Section 2 describes the related theory and Section 3 gives the experimental results on synthetic and real data to illustrate the performance of this algorithm. Finally, we draw a conclusion and discuss future work in Section 4.

2. Method descriptions

In this section, we first discuss the characteristics of affinity matrix and then define an optimization measure based on the weighted graph associated with an image. The solution to the optimization problem satisfies the clustering standard with maximal within-class similarity and minimum between-class similarity.

2.1 Affinity matrix

The affinity matrix is a symmetric matrix and describes the pairwise pixel similarity. Every element $W_{i,j}$ of an affinity matrix $\mathbf{W}$ represents the similarity between pixels i and j. There are various definitions for the similarity measures. In general, $W_{i,j}$ can be defined as

$$W_{i,j} = e^{-\| x_i - x_j \|^2 / 2\sigma^2} \tag{1}$$

where $\|.\|$ is Euclidean distance and σ is a free parameter. This is somewhat similar to the definition to Gaussian distribution.

The characteristics of an affinity matrix (or similarity measures) are listed as follows.

- Symmetric property
  The affinity matrix is symmetric, that is, $W_{i,j} = W_{j,i}$. So it can be diagonalized.
- Unit normalization
  That is, $0 \le W_{i,j} \le 1$. The similarity $W_{i,j}$ between pixels i and j increases as $W_{i,j}$ goes from 0 to 1 while the dissimilarity decreases.
- Transitive property
  If pixels i and j are similar and pixels j and k are similar, then pixels i and k are similar.
- Coherence property
  That is, $W_{i,j} \ge W_{l,k}$ holds for $\forall i,j,l,k$ if pixels i and j are in the same cluster while pixels l and k are in different clusters.
- If similarity $W_{i,j}$ is greater than some threshold, then we say that pixels i and j are similar.

2.2 Similarity measure for cluster and whole image

Now, we define a similarity measure for one cluster and the whole image. If we consider the affinity matrix represents a weight matrix of a complete graph with all pixels as vertices. Then, there is a maximum spanning tree for this complete graph. Obviously, there is a subaffinity matrix for every cluster. We define the cluster similarity measure as the product of those weights in its maximum spanning tree. That is,

$$S_h = \prod_{i=1}^{N_h - 1} P_i^h \tag{2}$$

where h represents cluster number, N_h is the number of entities (pixels) in cluster h, and P_i^h are weights in the maximum spanning tree of cluster h. Because of the symmetric, transitive, and coherence properties of affinity matrix, we can understand this as follows. Given a pixel p in cluster h, in order to find all pixels in cluster h, we first find the pixel q with maximum similarity to pixel p. Then we find another pixel not in set $\{p,q\}$, but with maximum similarity either with p or with q. Repeatedly, until all pixels of cluster h are found. We can see that this measure is reasonable to represent the maximum within-cluster similarity for cluster h.

After we define cluster similarity measure, we further define a similarity $\mathbf{S}$ for the whole image, as follows

$$\mathbf{S} = \prod_{h=1}^{c} S_h = \prod_{h=1}^{c} \prod_{i=1}^{N_h - 1} P_i^h \tag{3}$$

where c is the number of clusters of image or number of segmentation components.
For convenience, sometimes, we use log on $\mathbf{S}$. We have

$$\log \mathbf{S} = \sum_{h=1}^{c} \log S_h = \sum_{h=1}^{c} \sum_{i=1}^{N_h - 1} \log P_i^h \tag{4}$$

Next, we will show that to maximize the similarity measure $\mathbf{S}$ is to maximize the within-cluster similarity and minimize the between-cluster similarity, which is preferred by the clustering and image segmentation.
Proposition 1. The following optimization problem

$$\underset{h,i}{\operatorname{argmax}} \mathbf{S} = \underset{h,i}{\operatorname{argmax}} \prod_{h=1}^{c} \prod_{i=1}^{N_h - 1} P_i^h \tag{5}$$

guarantees that the within-cluster similarity is maximum and the between-cluster similarity is minimal.
Proof. By contradiction.
Assume that there is a pixel p in cluster m is misclassified into cluster n. In the maximum spanning tree of cluster m, pixel p either connects two edges in the middle of the tree or connects one edge as a leaf node. Consider that the pixel p is removed from the cluster m. If p is a leaf node, then one its associated edge (also the weight) is removed from the maximum spanning tree. If p is in the middle of the tree, then two its associated edges (also the weights) are removed from the maximum spanning tree. But a new edge must be added to connect the two separate parts into a new maximum spanning tree. When pixel p is added into cluster n, it is either in the middle of the tree or exists as a leaf node. However, because of the coherence property of affinity matrix, the pairwise similarity between p and any pixel in cluster n is the smallest in the maximum spanning tree of cluster n, then p can not be added in the middle of the maximum spanning tree of cluster n. So p is added as a leaf node and one more edge (also the weight) is added onto the new maximum spanning tree of cluster n.
If p is removed as a leaf node from the cluster m, its cluster similarity measure S_m becomes

$$S'_m = \frac{S_m}{w_r} \tag{6}$$

where w_r is the removed weight.

If p is removed as a node in the middle of the maximum spanning tree of cluster m, its cluster similarity measure S_m becomes

$$S'_m = \frac{w_a S_m}{w_b w_c} \tag{7}$$

where w_b and w_c are the removed weights from the maximum spanning tree and w_a is the added new weight. From Prim's algorithm [2] (for minimal spanning tree, but inverse weights of maximum spanning tree, we can use it), $w_a \le w_b$ or $w_a \le w_c$. Or else, $w_a > w_b$ and $w_a > w_c$, w_a will be in the maximum spanning tree.

After *p* is added as a leaf node into the cluster *n*, its cluster similarity measure S_n becomes

$$S'_n = w_d S_n \tag{8}$$

where w_d is the added weight. Because of coherence property, $w_d < w_r, w_a, w_b, w_c$.

Then either,

$$S'_m \times S'_n = \frac{S_m}{w_r} \times w_d S_n = \frac{w_d}{w_r} S_m S_n < S_m S_n \tag{9}$$

or

$$S'_m \times S'_n = \frac{w_a S_m}{w_b w_c} \times w_d S_n = \frac{w_a w_d}{w_b w_c} S_m S_n < S_m S_n \tag{10}$$

Therefore, the maximum within-cluster similarity and the minimal between-cluster similarity are guaranteed under the above similarity measure.

2.3 Maximum spanning tree

After we define the above optimization problem, we want to solve it. We show that the above optimization problem can be solved by finding a maximum spanning tree for the complete weighted graph of an image. First, we give a brief introduction to Prim's algorithm, which, in graph theory, is used to find a minimum spanning tree for a connected weighted graph. If we inverse all weights of affinity matrix, to find a maximum spanning tree of the original graph is equavalent to finding the minimum spanning tree of the graph with new weights. So Prim's algorithm can be used to find a maximum spanning tree.

Prim's algorithm is an algorithm that finds a minimum spanning tree for a connected weighted graph, where the sum of weights of all the edges in the tree is minimized. If the graph is not connected, then it only finds a minimum spanning tree for one of the connected components.

The time complexity of the Prim's algorithm is $O(|E|\log|V|)$, where $|E|$ is the number of edges in the graph and $|V|$ is the number of nodes. For a complete graph, the number of edges is $|E| = \binom{|V|}{2} = \frac{|V|(|V|-1)}{2}$. This is also the maximum number of edges that a graph can have. So the time complexity of the Prim's algorithm is also $O(|V|^2 \log|V|)$. If we only use the transitive property of affinity matrix and compute the local similarity for a pixel, then there are only eight similarities for a pixel between it and its eight neighboring pixels. The number of edges becomes $|E| = 8|V|$ and the complexity becomes $O(|V|\log|V|)$. This will reduce the complexity considerably. On the other hand, the complexity to a standard eigenfunction problems takes $O(|V|^3)$, where $|V|$ is the number of nodes in the graph.

Next, we show that the optimization problem can be solved by finding the maximum spanning tree and then removing the $c-1$ minimal weights.

Proposition 2. The following optimization problem

$$\operatorname*{argmax}_{h,i} \mathbf{S} = \operatorname*{argmax}_{h,i} \prod_{h=1}^{c} \prod_{i=1}^{N_h - 1} P_i^h$$

can be solved by finding the maximum spanning tree of the graph associated with the image and then removing the $c-1$ minimal weights.

Proof. Prim's algorithm is used to find the maximum spanning tree.

Searching the maximum spanning tree of the whole image starts from a pixel x in some cluster m. According to coherence property of affinity matrix, the maximum spanning tree of cluster m must be put into the maximum spanning tree of the image first. Then the maximum spanning tree of cluster m will connect some pixel y in another cluster n through a between-cluster edge. From pixel y, the maximum spanning tree of cluster n is put into the maximum spanning tree of the image next. Repeatedly, all maximum spanning trees of clusters will be put in the maximum spanning tree of the image. The $c-1$ minimal weights connect those maximum spanning trees of c clusters into the maximum spanning tree of the whole image. Therefore, we obtain the solution

$$\max \mathbf{S} = \frac{T}{\prod_{l=1}^{c-1} w_l} = \frac{\prod_{k=1}^{N-1} T_k}{\prod_{l=1}^{c-1} w_l} \tag{11}$$

where

$T = \prod_{k=1}^{N-1} T_k$ is the product of all weights in the maximum spanning tree of the whole image.

w_l are the $c-1$ minimal weights.

3. Experimental results

We test our algorithm using synthetic data and real data. The synthetic data are binary data, generated by drawing white squares on the black background. The real data is a kid picture. The number of clusters (components) is give as prior here. The selection of number of clusters is a model selection problem that depends on the application, and is beyond the scope of our discussion here.

Test on synthetic data

The synthetic data are a picture with two white squares on the black background. The similarity is computed using Equation 1 based on intensity values. We chose $\sigma = 0.1.$ Then the within-class similarity is

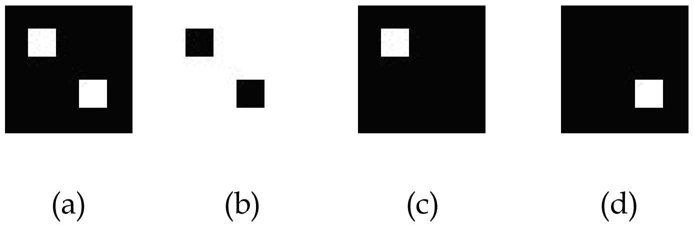
$$e^{-\| x_i - x_j \|^2 / 2\sigma^2} = e^{-\| 0-0 \|^2 / 2 \cdot 0.1^2} = e^{-\| 1-1 \|^2 / 2 \cdot 0.1^2} = e^{0} = 1 \tag{12}$$

and the between-class similarity is

$$e^{-\| x_i - x_j \|^2 / 2\sigma^2} = e^{-\| 1-0 \|^2 / 2 \cdot 0.1^2} = e^{-50} \approx 0 \tag{13}$$

For the data with two white squares, the maximum spanning tree of the image has two edges with zero weights. All other edges have weight one. The zero weight edges are two minimal weights and separate the maximum spanning tree of the image into three spanning trees. One of the new spanning trees represents the background and the other two represent two foreground squares.

The segmentation results are shown in Figure 1. We can see that we obtain perfect results.



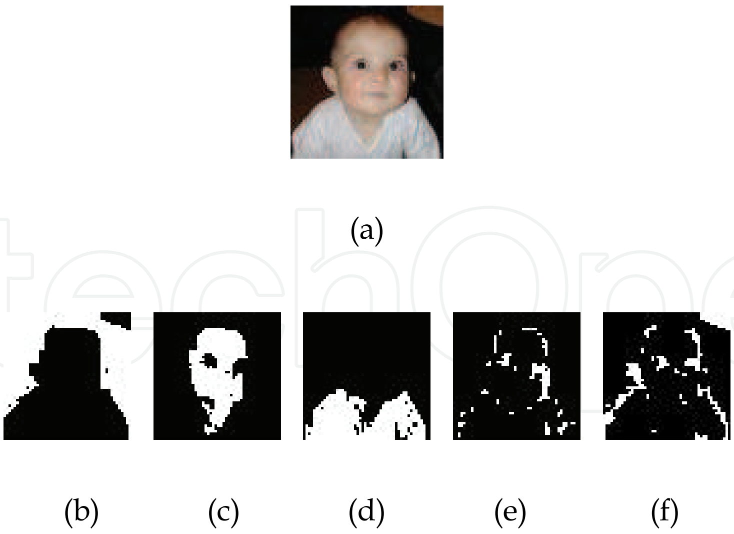
(a) (b) (c) (d)

Fig. 1. Image segmentation on synthetic data. (a). the original image (b). the segmented background (white part) (c). the first segmented square (white part) (d). the second segmented square (white part)

Test on real data

The real data is a kid picture. The similarity is computed using Equation 1 based on average values of three channel values. We still chose $\sigma = 0.1.$ The segmentation results are shown in Figure 2. The results are reasonable. Since boundary contours of kid face and shirt are not consistent with the background and foreground (kid face or shirt), they can be clearly separated as we chose 5 components.

In real images, because of noise and outliers in cluster, some within-cluster similarities are very small. Correspondingly, some very small weights do not represent the between-cluster separation weights. In practice, we follow the order of edges from the Prim's algorithm and pick those edge weights with considerable differences from (viz. smaller than) its previous and afterwards edge weights as the between-cluster separation edges. This method works robustly for real images. In practice, the χ^2 -statistic [4] for histograms may give a better similarity measure for color and texture.



(a)

(b) (c) (d) (e) (f)

Fig. 2. Image segmentation on kid picture. (a). original image (b). the segmented background (white part) (c). the segmented kid face (white part) (d). the segmented shirt (white part) (e,f). the segmented contours (white part)

4. Conclusions

In this paper, we presented a new graph-based image segmentation algorithm. This algorithm finds the maximum spanning tree of the graph associated with the image affinity matrix. Instead of solving eigenvalues and eigenvector, we proved that the image segmentation could be transformed into an optimization problem: finding the maximum spanning tree of a graph with image pixels as vertices and pairwise similarities as weights. In our future work, we will explore different similarity measures and test the segmentation algorithm on more data.

5. References

[1] Chung, F.R.K.: *Spectral Graph Theory*. American Mathematical Society, 1997.

[2] Cormen, T.H., Leiserson, C.E., Rivest, R. L., and Stein, C.: *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001.

[3] Perona, P. and Freeman, W. T.: A factorization approach to grouping. In *Proc ECCV*, H. Burkardt and B. Neumann, editors, 1998, 655--670.

[4] Puzicha, J., Rubner, Y., Tomasi, C., and Buhmann, J. M.: Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings of the IEEE International Conference on Computer Vision*, 1999, 1165--1173.

[5] Scott, G.L. and Longuet-Higgins, H. C.: Feature grouping by relocalisation of eigenvectors of the proximity matrix. In *Proc. British Machine Vision Conference*, 1990, 103-108.

[6] Shi, J. and Malik, J.: Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, 731-737.

[7] Weiss, Y.: Segmentation using eigenvectors: a unifying view. In *Proceedings IEEE International Conference on Computer Vision*, 1999, 975-982.

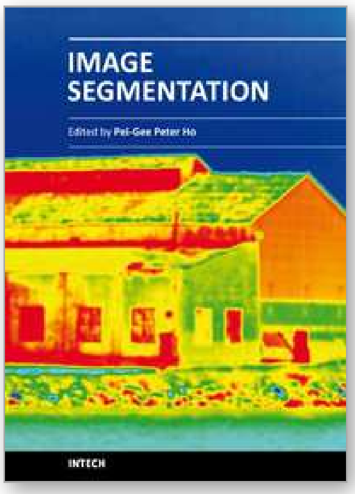


Image Segmentation
Edited by Dr. Pei-Gee Ho

ISBN 978-953-307-228-9
Hard cover, 538 pages
Publisher InTech
Published online 19, April, 2011
Published in print edition April, 2011

It was estimated that 80% of the information received by human is visual. Image processing is evolving fast and continually. During the past 10 years, there has been a significant research increase in image segmentation. To study a specific object in an image, its boundary can be highlighted by an image segmentation procedure. The objective of the image segmentation is to simplify the representation of pictures into meaningful information by partitioning into image regions. Image segmentation is a technique to locate certain objects or boundaries within an image. There are many algorithms and techniques have been developed to solve image segmentation problems, the research topics in this book such as level set, active contour, AR time series image modeling, Support Vector Machines, Pixon based image segmentations, region similarity metric based technique, statistical ANN and JSEG algorithm were written in details. This book brings together many different aspects of the current research on several fields associated to digital image segmentation. Four parts allowed gathering the 27 chapters around the following topics: Survey of Image Segmentation Algorithms, Image Segmentation methods, Image Segmentation Applications and Hardware Implementation. The readers will find the contents in this book enjoyable and get many helpful ideas and overviews on their own study.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821