Image segmentation using Bayesian stochastic blockmodels

Pavlin Gregor Poličar
University of Ljubljana
Faculty of Computer and Information Science
Email: pavlin.g.p@gmail.com

***Abstract*—Image segmentation is a very broad field that combines methods from various areas of expertise. Graph-based techniques achieved some early successes in this field and are still in use today. However, community detection techniques have progressed significantly over the past few years, although none have been applied to this problem until quite recently, where the Louvain modularity optimization method was applied and appeared to achieve reasonable results. However, modularity optimization is just one of numerous community detection algorithms developed in recent years. This raises the question: how well would other more recent community detection algorithms perform for image segmentation? In this work we examine the most recent developments in Bayesian stochastic blockmodeling and attempt to apply them to images. We examine the difficulties that arise and possible solutions. Our results indicate that stochastic blockmodels and its variants are not a viable method for image segmentation and discuss the reasons why most community detection algorithms would perform poorly.**

I. Introduction

The problems of image segmentation and perceptual grouping remain one of the most popular challenges in computer vision today. Image segmentation is the process of partitioning an image into multiple segments, usually corresponding to different objects or regions present in the image. Intuitively, a good segment is a subset of pixels in the image, which are spatially close to each other and are fairly uniform with regards to color and brightness.

However, one can never be sure what a ”good” segmentation truly is, since image segmentation is an inherently hierarchical task. Therefore no true segmentation exists, since it can be defined on multiple levels of granularity. Instead of finding the *correct* answer, we should strive to find a *useful* answer in the context of our application. Figure 1 shows the disagreement between human annotators to highlight these differences. If human experts cannot agree on a *correct* segmentation, what hope does an algorithm have? Ideally, we could design a method that would produce a hierarchy of segments in an image at different granularities, but we quickly run into computation challenges. Therefore any low-level segmentation technique cannot and should not strive to produce a correct segmentation, but should produce a useful segmentation in the context of our application.

Graph-based techniques have proved to be quite effective for this task, most notably – the normalized cut algorithm and its variants proposed in 2000 [1] are still some of the most widely used methods today due to their simplicity and computation efficiency. These methods and other graph-flow based methods have proven to be quite useful in this domain.

Figure 1. Human annotators produce different segmentations for the same image. The true image is shown in the top left, and eight possible annotations are shown. Clearly, no one single segmentation is the ”correct” answer – different segmentations are possible at different granularities. The image and segmentations can be found in the Berkley segmentation benchmark dataset under the id 254054.

In recent years, much attention has been given to community detection and the implementation of efficient algorithms in the field of network science. Some of the most notable such algorithms are greedy modularity optimization [2], random walk based algorithms e.g. Infomap [3] and stochastic blockmodels [4]. All of these algorithms have their drawbacks, in some way or another. Both greedy optimization and infomap are discriminative methods, which partition graph nodes into communities. Stochastic blockmodels are a generative model and are more closely described in Section III. For a long time, the main drawback of stochastic blockmodels is that the number of blocks or communities had to be known in advance. However, recent work by Peixoto [5] do away with this limitation by posing the problem in a Bayesian framework.

All of the three methods could be useful in the context of image segmentation due to the fact that they can produce hierarchical partitions. However, since the focus of this paper is the application of stochastic blockmodels, we limit ourselves these models and their nested variants [6].

This paper is organized as follows. In Section II, we describe how image segmentation can be posed as a graph-based problem. We then shift our focus onto a recent paper by Browet et al. [7] who propose an image segmentation scheme based on a modified greedy modularity optimization method. In Section III we provide an overview of the stochastic blockmodel and recent developments. Finally, in Section IV, we show some experimental results on real images. Finally, Section V concludes the paper and discusses suggestions for future work.

II. Related work

A. Image segmentation as a graph problem

Before we can apply graph-based methods for image segmentation, the image itself must be represented as a graph. This is typically done by building an undirected weighted graph G, where each node represents a pixel in the input image. Each weighted edge of the graph G represents the similarity between pairs of pixels i and j and is stored in the weighted adjacency matrix W. Typically, we define W_{ij} as

$$W_{ij} = \begin{cases} e^{\frac{d(i,j)^2}{\sigma_d^2}} e^{\frac{|F(i)-F(j)|^2}{\sigma_F^2}} & \text{if } d(i,j) < d_{\text{max}}, \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

where the term $d(i,j)$ represents spatial proximity using some distance metric between pixels i and j (most commonly the Euclidean distance) and $F(i)$ is some feature vector based on the properties of the pixel. This feature vector is typically the scalar intensity of the pixel for gray-scale images or the HSV transform for colored images [7]. d_{max} specifies an upper bound on pixels that we will consider based on their spatial proximity to the pixel in question. This parameter is necessary due to practical considerations, since the total number of connections between pixels is n^2, which is, in practical regards, computationally unfeasible, therefore it is necessary to restrict ourselves to some region around individual pixels. To illustrate this, we can consider a small image by today's standards, a 480×320 image. This image contains 153,600 pixels in total and were we to connect every single pixel to one another, this would result in 23,592,960,000 edges. This not only has computer memory implications but will also impact the running time of our algorithms. The last two parameters σ_d and σ_F are user-defined parameters that control the impact of pixel distances and pixel similarities in the edge weight. A lower value of σ_d induces faster decay i.e. distances will drop off exponentially, whereas larger values will induce a slower decay. Likewise, a lower value of σ_F induces faster decay i.e. the higher the difference in pixel features, the lower the similarity. An illustration of an image being represented as a graph by following this simple conversion method is shown in Figure 2. The original image of the koala bear hugging a tree branch will be shown later multiple times.

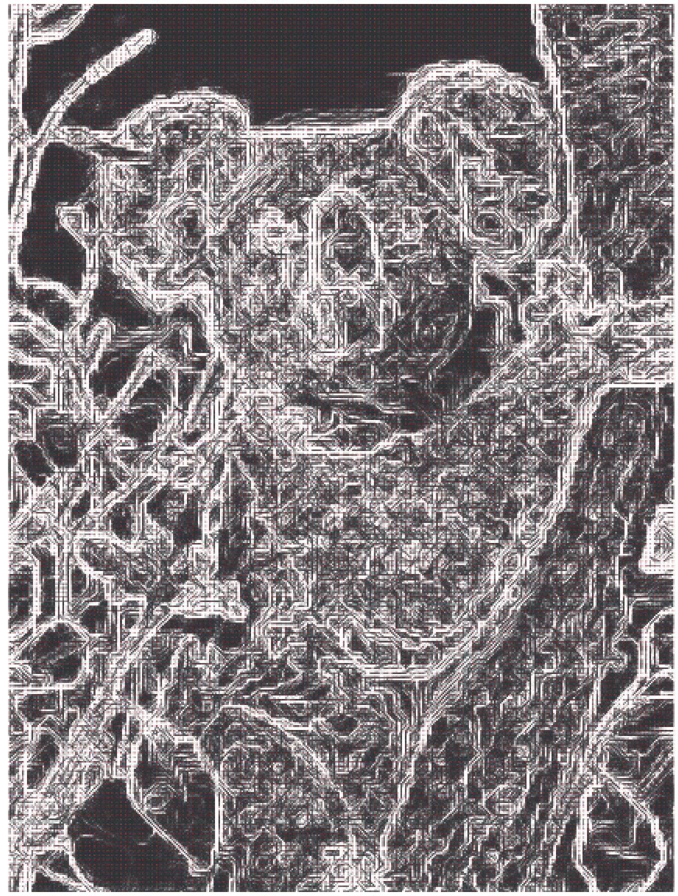

Figure 2. We convert an image of a koala bear hugging a tree branch into a graph using the method described above. The image consists of graph edges whose thickness is determined be the edge weight. We can clearly see tightly connected regions of the image by dark areas, whereas the image edges are clearly visible due to the absence of thick edges.

This is not a new approach. In fact, it was first proposed (to the best of our knowledge) in 2000 by Shi & Malik alongside normalized cuts. Normalized cuts attempt to find such a set of edges that creates disconnected components when removed from the graph, such that the sum of the weighted edges is minimal. These methods are typically referred to as *cuts* [1]. These methods produce good results and are still in use today.

B. Community detection for image segmentation

Normalized cuts and most previous graph-based methods utilize flow optimization techniques. More recently, community detection algorithms have been tested for the image segmentation task. Browet et al. propose a variant of the Louvain method for greedy modularity optimization. The tests were performed on the Berkley segmentation benchmark dataset (BSDS) [8] and appears to work reasonably well.

They show that the standard greedy modularity optimization approach typically used for community detection produces coherent regions, but most objects in the images are over-segmented. They argue that this oversegmentation is due to the fact that modularity optimization algorithms cannot yield communities with long chains of nodes. They propose two solutions to tackle this problem. Firstly, increasing d_{max} is the theoretically sound solution, but is often, as previously discussed, unfeasible for all but the smallest images. The second remedy, which they then implement is to modify the null model used in modularity. We next describe their approach.

Modularity is a well-studied measure of community goodness [9]. It provides a nice trade-off between the edges actually observed in the graph and the number of edges we would expect to appear at random given a random null model. The basic expression for the modularity measure for weighted networks is given in Equation 2.

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} \left(w_{ij} - N_{ij} \right) \delta_{C_i C_j} \tag{2}$$

where n is the number of nodes, w_{ij} is the edge weight, $\delta_{C_i C_j}$ is the Kronecker delta function and N_{ij} is the expected number of edges we would expect to appear at random in a configuration model between nodes i and j and is defined as $N_{ij} = \frac{k_i k_j}{2m}$, where k_i is the sum of the weights of all edges of node i. In this case, m is the sum of weights of all the edges in the graph and is a scaling parameter ensuring that $Q \in [-1, 1]$.

They then propose a modified version of modularity, which enables regions to cover large portions of the graph. More precisely, they introduce a weight matrix Λ, with which they modify the null model $N_{ij} = \Lambda_{ij} \frac{k_i k_j}{2m}$, where

$$\Lambda_{ij} = \begin{cases} 1 & \text{if } d(i,j) \leq d_{\Lambda}, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Figure 3. Image segmentation using the Louvain greedy modularity optimization method (center) produces an oversegmented image. Browet et al. [7] propose a modified modularity measure which empirically allows communities to cover larger regions of the network (right).

This modified version of modularity appears to produce reasonable results, shown in Figure 3. However, the method is never actually benchmarked to other image segmentation techniques and no score is provided [7], therefore we cannot know how well it compares to other state of the art image segmentation techniques.

III. Stochastic Blockmodels

The stochastic blockmodel (SBM) [4] is a generative model for community detection, which makes use of the statistical properties of the network to infer the high level modular structure. In its most basic form, the model assigns each node i to one of K modules denoted by z_i. Each edge exists with probability $\Theta_{z_i z_j}$ where Θ is a $K \times K$ matrix which values are the probabilities of links appearing between blocks z_i and z_j.

SBMs are interesting from another standpoint. Most community detection algorithms rely on the assumption that communities are subsets of nodes in the network that tend to form edges with nodes in the same community. This is known as assortative mixing. A nice property of SBMs is that they are not limited to this kind of connectivity because they only observe the statistical properties of the network. Therefore they can easily detect other types of structure in the network such as disassortative mixing and core-periphery structure under the same inference framework.

A fundamental limitation of the classical SBM is that they are defined strictly for simple graphs or multigraphs. This means that they cannot make use of various additional data such as edge weights, which are usually a very rich source of information. Moreover, in the case of image segmentation, the number of blocks K must be known in advance. This clearly does not fit into a general framework for detecting regions of interest in images.

The first aforementioned issue was addressed by Aicher et al. [10], who generalized the SBM to include edge weights drawn from any exponential family distribution as additional covariates. The inference model also learns from the presence and absence of edges. This allows the model to make use of information that would otherwise be discarded when using previous techniques such as simple thresholding.

The second issue was just recently addressed by Peixoto [5], who built on the ideas of Aicher et al. and produce a nonparametric Bayesian approach, which is capable of probabilistically inferring the number of blocks in the model. The model relies on a Bayesian hierarchy of uninformative hyperpriors that remain agnostic of the number of groups, the sizes of the groups and the partition of the nodes. A very detailed derivation of this model is given in [11]. Peixoto shows that this nonparametric model is capable of correctly recovering community structure from various empirical networks. The inference step used is very efficient, making use of Markov Chain Monte Carlo (MCMC) sampling, that requires only $O(E)$ operations per sweep, where E is the number of edges in the network.

SBMs can recover the flat community structure of networks i.e. they provide the most probable partition of the nodes. At the same time, we would often be equally or more interested in the higher level overview of the network structure – how these observed communities interact with each other. In other words, we would be interested in the hierarchical decomposition of the network, observing different partitions of nodes at each level of the hierarchy.

The nested stochastic blockmodel is an extension to the basic SBM that enables precisely this. The basic idea of the nested SBM is illustrated in Figure 4 [6]. The idea is simple, recursively apply the SBM to the network, and then merge nodes in the same blocks until no more statistically significant block structure can be recovered.

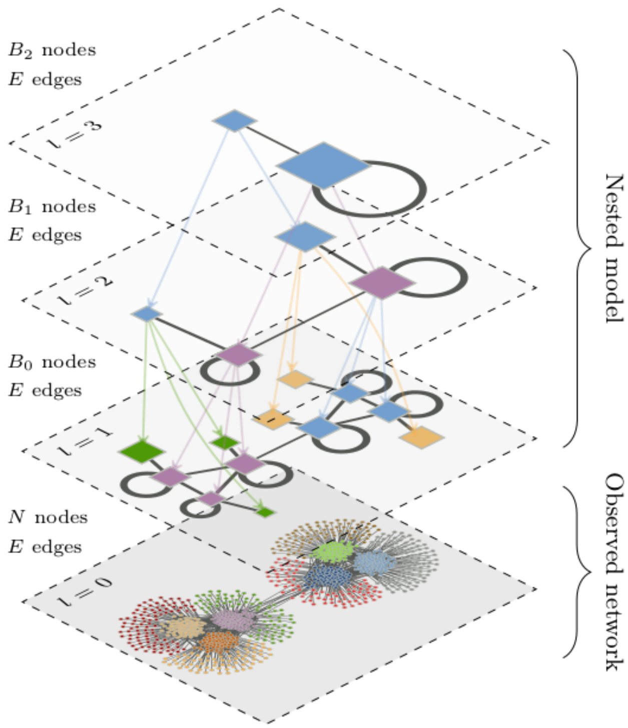

Figure 4. Example of a nested stochastic block model with three levels, and a generated network at the bottom. The top-level structure describes a core-periphery network, which is further subdivided in the lower levels.

IV. Results

We first apply the SBM to graphs obtained from images as described in Section II-A. The segmentations are shown in Figures 5 and 6.

Figure 5. Image segmentation using a nonparametric SBM produces an overly segmented image. The image can be found in BSDS under the id 24004.

Clearly, SBMs suffer from a similar drawback as the one discussed in the Louvain modularity optimization problem. A good community is defined as partitions of the graph where edges among nodes are more likely than edges outside of the community. This fundamental definition inherently contradicts

Figure 6. Another example using the nonparametric SBM. Again, the results show an overly segmented image. The image can be found in BSDS under the id 69015.

our graph-building process, with which we are forced to limit the number of edges we add to the graph. As such, most community optimization algorithms will struggle to find good communities and will produce similar results to what we observe with the SBM. While the regions it finds are reasonable, they are far too oversegmented to be useful in any context. The results of the SBM remind of another useful computer vision technique called superpixel – a preprocessing step which groups together similar regions in an image to make computation simpler. However, even though the SBM produces similar results, it is computationally far more expensive, and should not be considered as a substitute.

We also consider several different values for the user parameters which regulate the strength of distance and pixel similarities as edge weights. We found that no setting of these parameters drastically improves the results.

Next, we consider the nested SBM. The results are shown in Figures 7 and 8. Again, we see that this produces overly segmented images even at higher levels of the hierarchy. We make two surprising observations.

Firstly, at higher levels of the hierarchy the nested SBM begins to merge together blocks which would not be considered good segments. This can be seen in a number of segments which cross apparent edges in the image e.g. in Figure 8 at the fifth level of the hierarchy, the koalas foot and branch are contained within the same segment. At higher levels of the hierarchy these differences become even more apparent, and are not included here for the sake of space. This problem is most likely caused by the fact that we include only a small number of edges for each pixel, and the model cannot detect meaningful blocks given this substantial lack of information. The other possible explanation for this could be that we did not stumble upon a good setting of parameters when converting the image to a graph (despite testing several parameter configurations).

Our second observation is that the segmentation obtained from the observed network, i.e. the first level of the hierarchy is far more segmented than the one obtained from the basic SBM.

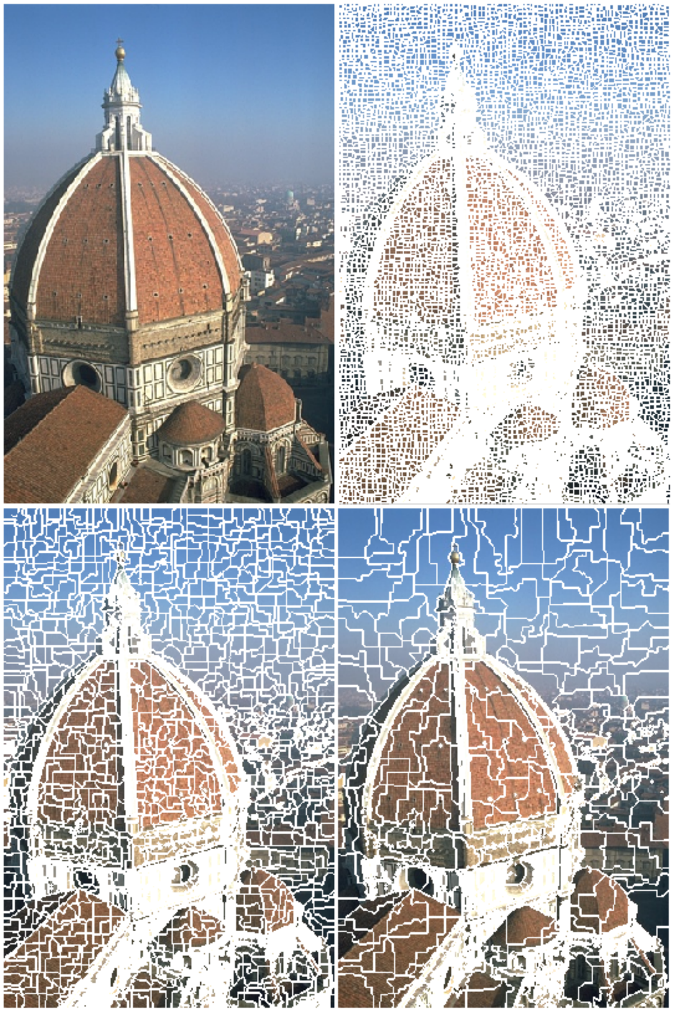

Figure 7. The segmentations obtained from the nested SBM. We display the original image, the first, third and fifth level of the hierarchy, respectively.

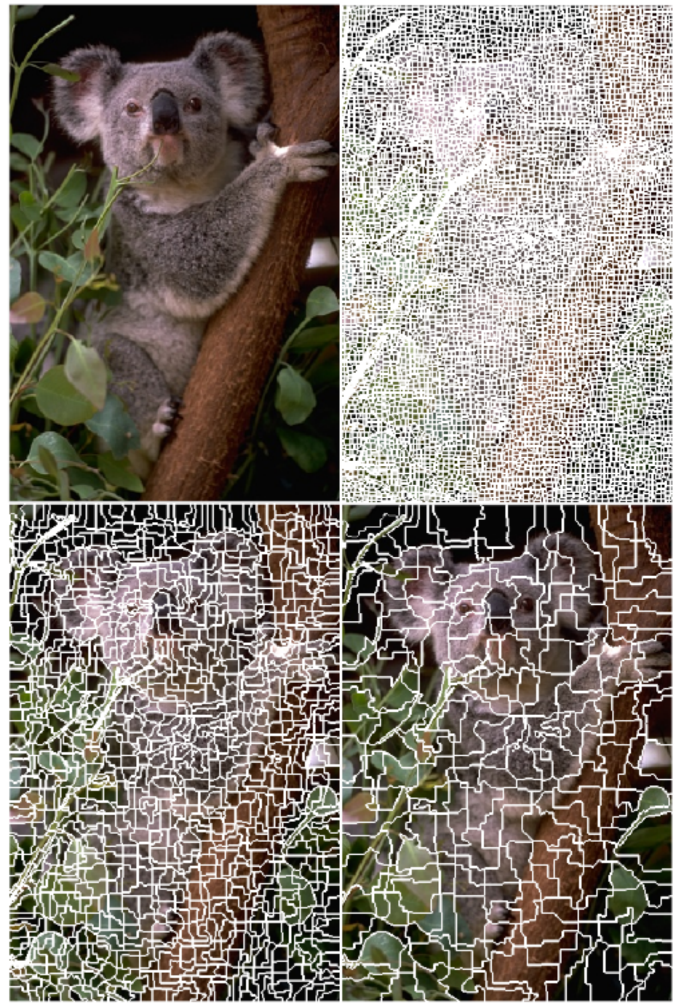

Figure 8. Another segmentations obtained from the nested SBM. We display the original image, the first, third and fifth level of the hierarchy, respectively. We observe that some of the segments are reasonable, others contain edges and sections that we would not group together.

A comparable segmentation is only obtained at the fourth or fifth level of the hierarchy. We speculate that this could be due to the nested SBM being more forgiving in the model selection process, since the nested SBM has more flexibility given multiple levels. The basic SBM does not have this luxury and must provide a good partition straight away.

V. Conclusion

In the presented work, we explore stochastic blockmodels and the nested variant for the task of image segmentation. We show that these are not suitable approaches for image segmentation and discuss some of the issues that arise when attempting to apply community detection algorithms to this particular task. A possible application in computer vision for SBMs and other community detection methods would be *superpixel*, a preprocessing technique that finds groups of similar pixels in close spatial proximity and groups them together in order to reduce the dimensionality of the image. However, the field of computer vision has already produced far more efficient methods for extracting these superpixels that run in linear time. Our SBM approach, does not and should therefore not be considered for this task since the produced results are not any better than those of established methods.

Another issue with this approach is parameter selection to induce the initial graph from the image. We have found no straightforward way to select the most appropriate parameters and they often differ from image to image.

Community detection algorithms are not well suited to this particular domain due to various issues discussed in this paper. A possible extension to this work would be to test community detection methods on preprocessed images with superpixel techniques mentioned above. However, this is likely to produce results comparable to the nested SBM.

References

[1] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[2] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[3] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[4] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[5] T. P. Peixoto, "Nonparametric weighted stochastic block models," *arXiv preprint arXiv:1708.01432*, 2017.

[6] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Physical Review X*, vol. 4, no. 1, p. 011047, 2014.

[7] A. Browet, P.-A. Absil, and P. Van Dooren, "Community detection for hierarchical image segmentation.," in *IWCIA*, vol. 11, pp. 358–371, Springer, 2011.

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001.

[9] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[10] C. Aicher, A. Z. Jacobs, and A. Clauset, "Learning latent block structure in weighted networks," *Journal of Complex Networks*, vol. 3, no. 2, pp. 221–248, 2014.

[11] T. P. Peixoto, "Bayesian stochastic blockmodeling," *arXiv preprint arXiv:1705.10225*, 2017.