Image Segmentation Through Application of Pagerank Graph-Based Kmeans Clustering

n0 u3m!3M

12/15/22

Abstract

Implementation of graph-based k-means clustering method for image segmentation that utilizes Pagerank to select the centroids of node clusters, and utilizes alternative Pagerank algorithms to assign nodes to centroids, to segment images into k partitions. The approach achieved capability for the segmentation of a subject from a multi-colored background using a minimum number of clusters, a task that traditional color-based k-means clustering is unable to accomplish.

Introduction

This implementation of image segmentation uses an application of graph-based k-means clustering. The graph-based representation of the image is generated in multiple variations with consideration for multiple color spaces. The Pagerank algorithm is used for both choosing the cluster centroids and assigning nodes to those centroids. The implementation is coded in python to take advantage of C lang's paralleling processing computation speeds. This serves as a proof of concept and springboard into future variations of the concept.

The graph representations of the images are constructed under two methods. First, Color-Node uses each unique color channel value as a node with directed edges of 1 to other nodes which 4-neighbor the color in the image. Second, Color-Edge uses each pixel in the image as a node with directed edges to every other node weighted to similarity by color channel vector distance. The initial color spaces/channels tested are RGB, hue from HSV, and AB from Lab.

In addition to the Dijkstra Algorithm to assign each node to one of the respective centroids, the Personalized Pagerank (RWR) Algorithm with each centroid as the target uses the bidirectional importance of each node to the centroid to achieve each assignment.

A new Pagerank algorithm termed "Undone Edges Pagerank" is also tested in place of the Dijkstra Algorithm. For each previous cluster, the graph of nodes is partitioned into two cluster and non-cluster groups, and the outward directing edges of each non-cluster node and removed. Standard Pagerank is performed under the presumption that non-cluster nodes will rank higher if they are significant to the cluster node subgraph.

The initial goal is to segment a subject from a complex multi-colored pattern background with the minimum amount of partitions. Traditional k-means clustering often captures the most yet distinct colors in the background instead of the subject. The graph-based representation and Pagerank implementation should take advantage of its relational information to cluster distinct yet nearby colors together.

Background

One method to employ k means on a graph representation is to revert the graph representation of an image back into a vector value by embedding the graph into a vector in euclidean space. This can be done in multiple ways, starting with basic matrix factorization or dimension reduction techniques such as Principal Component Analysis. Graph kernels are also a common embedding method that can be implemented with tools such as GraKel library/tool by applying a Neighborhood Subgraph Pairwise Distance Kernel. Other common methods of embedding graphs are artificial neural networks, autoencoders, and convolutional neural networks which may derive deeper results implemented by tools such as the Spektral library/tool or DeepWalk library/tool (Longa, 2009). Future work can use these tools to form a comparison of graph-based k-means results with embedding k-means results.

Graph-based k means clustering has been well documented, for example in the application of clustering 3d mesh objects, with most cases using Dijkstra Algorithm in place of the distance algorithm, and standard Pagerank is also used in choosing centroids though not assigning to centroids (Hajij, 2020).

Image segmentation using a non-vector k means algorithm, though without a graph or Pagerank, has also been attempted in the past. This implementation used pixels as points instead of the traditional color value and a subtractive method with the euclidean distance of each pixel to assign to centroids (Dhanachandra, 2015).

Personalized Pagerank has also been implemented as image segmentation through Random Walk with Repeat, another name for Personalized Pagerank. This application does not use k means to segment but implements the segmentation component semi-manually, requiring the use of hand selecting the seed/initial pixels used for Personalized Pagerank. Pixels are used as nodes while edges are determined by a neighborhood system and assigned a Gaussian weight encoded with color changes between two nodes in Lab color. Personalized Pagerank was used to compute each pixel against every seeded/initial pixel set to generatively assign each pixel to a partition with good results (Kim, 2008). There exist multiple other implementations of image segmentation using variations of random walk ranging from superpixels to preserving canny edges.

In terms of clustering, researchers have implemented Personalized Pagerank as a graph-based clustering algorithm for use on social network graphs. One such example is an optimization that, although doesn't use k means but a variation of backward partitioning, does mention k means and suggests their algorithm that is linear in time complexity may perform faster than k-means which is superlinear in time complexity (Tabrizi, 2013).

Methods

This implementation consists of three major stages: the generation of an image's graph representation matrix, the k means node assignment to a centroid, and the k means selection of a new centroid.

Graph Representation

The image's graph representation in matrix form is implemented in two methods, Color-Node Neighbors-Edge and Pixel-Node Color-Edge. The color spaces used are primarily RGB, Hue(SV), and (L)AB. For HSV, only the hue channel of the color space is used while in LAB only the a and b channels are used. The implemented code for RGB is also capable of being used for every other three-channel color space.

Color-Node Neighbors-Edge uses every unique color value in a color space as a node. For each Color-Node and the corresponding multiple pixels of the same color which exists on the image, the outgoing edge weight to the node of every 4-neighbors bordering the corresponding multiple pixels is set to 1 with non-neighboring nodes set to 0.

In this implementation, Hue has a maximum size of 180 nodes while RGB has a maximum size of 256^3 or 16,777,216 nodes. In order to decrease computation time and sparse matrices, a bin size parameter exists for RGB-Node and Lab-Node which divides each color channel value by the bin size value. For example, a bin size of 4 will result in RGB maximum size reduced to 64^3 or 262144 nodes. Bin size is only applicable for Color-Node Neighbors-Edge and not applicable for Pixel-Node Color-Edge.

Pixel-Node Color-Edge uses every pixel in the image as a node. For each Pixel-Node the edge weight between it and every other Pixel-Node is the similarity between the two nodes' color values. The similarity is an integer between 0 and 10, calculated linearly with 10 corresponding to the minimum distance between the two values and 0 corresponding to the maximum distance between the two values.

For example, in RGB-Edge, if one Pixel-Node has the color value (0,0,0) and another Pixel-Node has the (127,127,127), the resulting similarity is 5. In Hue-Edge, as hue values exist in a continuous cycle, 0 corresponds to a distance of 90, half of 180. A hue value of 179 and 5 would then also have a similarity of 9.

Graph Based K Means Clustering

Graph based k means clustering is achieved through modifying traditional k means clustering to operate on a graph. Centroid euclidean vectors are replaced by select nodes within the graph. The algorithms to assign nodes to centroids and select new centroids within an updated cluster are also based on the graph structure. The altered algorithm is as follows:

1. Set k random nodes as the initial cluster-centroids.
2. Assign all nodes to a centroid with a graph-based distance algorithm.
3. Reselect new centroids by using the highest Pagerank (argmax) of each cluster subgraph.
4. Repeat.

The central idea is in applying standard Pagerank on the subgraph, a matrix consisting only of the nodes assigned to the cluster, of each cluster to find the new centroid. As Pagerank finds the most significant node within a graph relative to every other node, the highest Pagerank serves as a practical replacement for mean, median, and other center calculations.

Distance Algorithm

In place of the Euclidean Distance Formula, three implementations of an alternative distance algorithm are compared.

The first is the commonly used Dijkstra Algorithm which existing implementations of graph-based k means clustering use. The Dijkstra Algorithm finds the shortest path between every node to each of the n centroids. The lowest path (argmin) from any node to the n centroids is the centroid that node is assigned to.

The second is the reliable Personalized Pagerank Algorithm. For each n centroids, n number of Personalized Pagerank is run over the whole matrix to find the bidirectional importance of every node to each of the n centroids. The highest rank (argmax) of the n ranks for any node is the centroid that node is assigned to.

The third, termed "Undone Edges Pagerank", is a modified Pagerank intended to relay the importance of any node to each centroid's cluster subgraph. The previous n clusters and their associated centroids are referenced to determine assignment instead of centroids alone. For each n run of Undone Edges Pagerank, the matrix is modified by having all outgoing edges of nodes outside the previous cluster removed, leaving only their incoming edges. Standard Pagerank is performed under the presumption that non-cluster nodes will rank higher if they are significant to the previous cluster's quasi-subgraph. The highest rank (argmax) of the n ranks for any node is the centroid that node is assigned to.

Limitations

Computing time is composed of two major components. First, setup time is required to translate the pixel image into the matrix graph representation of either Color-Node or Pixel-Node. Second, K means run time required for clustering which will use Pagerank and/or personalized Pagerank multiple times per iteration using the matrix graph.

The setup time for Color-Nodes is N, where N is the rows by cols of the image. The setup time of Pixel-Node graphs is N^2 computations. A 100x100 pixel image would cost 1000 relative computations to set up the Color-Node graph in comparison to 1,000,000 for Pixel-Nodes.

Color-Node graphs have another distinct advantage. Most images do not use the full range of unique color channel values or combinations, usually anywhere between 50-90%. For hue, the maximum unique values are 180. The matrix in each Pagerank is at max 180 by 180. No matter the size of the image, the matrix used will never exceed the maximum size. For AB it is 256*265 or 67840. For RGB and other standard three-channel color values, it is 256*265*256 or 17367040. For these cases, the bin size can reduce by lowering the total levels of intensity in each color channel. For example, a bin size of 8 will reduce RGB to 32*32*32 or 32768 which is more manageable as a 32768 by 32768 matrix.

Bin size is not just to reduce computation time but also to handle cases of sparse matrices. Inadequate segmentations under particular images against particular graph and distance algorithms can be improved by increasing the bin size. This method of grouping is similar to a technique in handling sparse utility matrices for recommender systems.

For Pixel-Node graphs, are not limited by the overall numbers of color values, and do not bin sizes are not applied. Each matrix is determined by the rows by cols of the image square. A 640 × 480 pixels image is a 307200 by 307200 matrix.

Results

The results of the testing are satisfactory. Differing combinations of graph type, color, and distance algorithms were tested in Table 1. Without the use of bin size on a small image, 12 out of 18 results were able to capture the relative shape of the island. Personalized Pagerank Distance performed the best. Between Undone Edges PageRank and Dijkstra Distance one often succeeds when the other fails. Each graph type has varying results, with some preferred over others depending on the application.

Table 1

Comparison of Graph and Distance Combinations on small_island.jpeg (70x52)

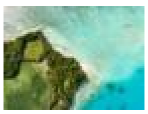
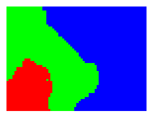
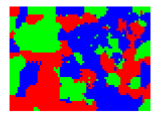
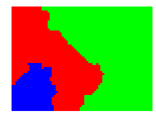
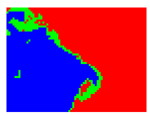
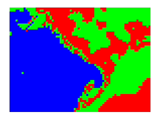
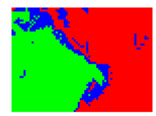
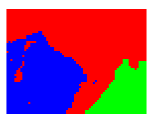
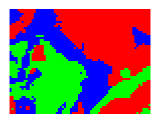
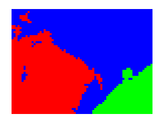
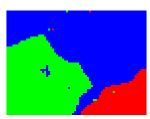
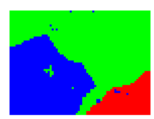
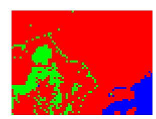
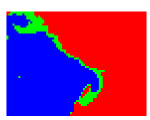
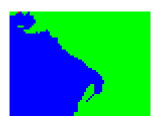
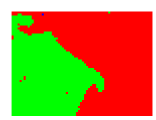
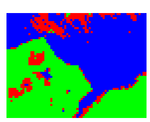
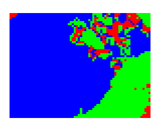
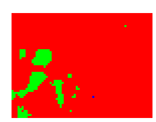
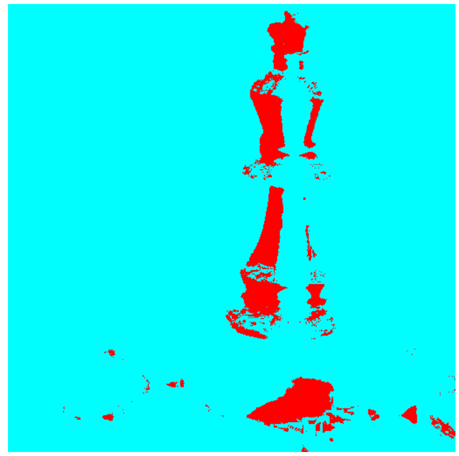
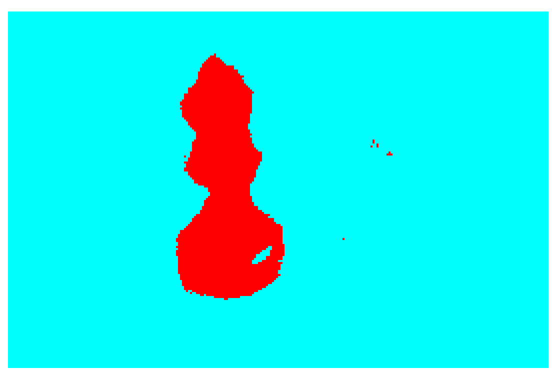
	Personalized PageRank	Undone Edges PageRank	Dijkstra Distance
RGB Node Neighbors Edge			
Hue(SV) Node Neighbors Edge			
(L)AB Node Neighbors Edge			
Pixel Node RGB Edge			
Pixel Node Hue(SV) Edge			
Pixel Node (L)AB Edge			

Table 2 demonstrates at least one combination is capable of segmenting a subject over a multi-colored background with minimum k clusters. In an image consisting of a checkered background like black and white, traditional k means clustering using pixel and their vector color values is likely to capture a cluster of white pixels and a cluster of black pixels with the subject split in between. Rgb-Node allows this implementation to associate neighboring black and white color values and capture this association with Personalized Pagerank.

Table 2

Capturing Subject From Checkered Background With Minimum of 2 Clusters (Rgb-Node and Personalized Pagerank With Bin Size of 4)

Original	Segment
	
	
	

In Table 1, one of the failed results was RGB Node Neighbors Edge Undone Edges PageRank. Table 3 demonstrates how increasing bin size improves segmentation by addressing the sparse matrix.

Table 3

RGB Node Neighbors Edge, Undirected PageRank, With Varying Bin Sizes.

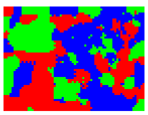
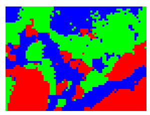
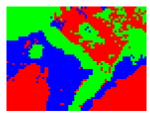
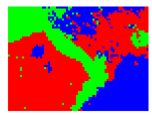
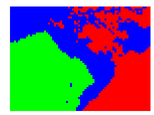
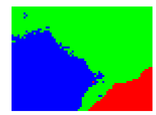
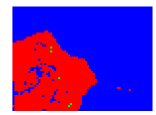
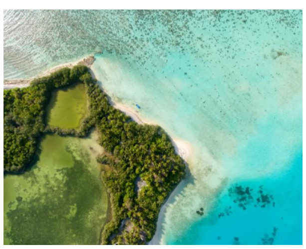
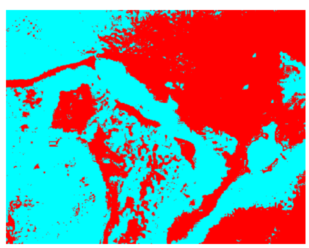
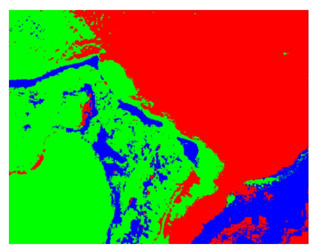
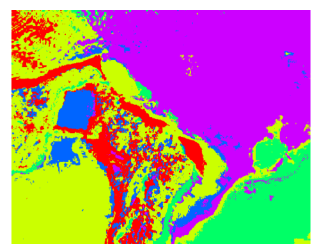
None	4	8	16	32	64	128
						

Table 4 demonstrates the difference in k number of clusters used to segment.

Table 4

Lab Node Neighbor Edge, varying clusters k on island.jpeg (700 x 524)

	2	3	5
			

Conclusion

Image segmentation using Pagerank k means clustering has been demonstrated to be feasible. In addition to variations in color spaces/channels as a parameter, which standard k means clustering also takes advantage of, the implementation includes variations in graph representation and distance algorithm as parameters. Each unique combination of parameters is suitable for a different application. One demonstrated niche application in the implementation is the capability of segmenting a subject over a multi-colored background using a minimum number of clusters.

References

Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using K -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54, 764–771. https://doi.org/10.1016/j.procs.2015.06.090

Hajij, M., Said, E., & Todd, R. (2020). PageRank and The K-Means Clustering Algorithm. *arXiv e-prints*. https://doi.org/10.48550/arXiv.2005.04774

Kim, T. H., Lee, K. M., & Lee, S. U. (2008). Generative image segmentation using random walks with restart. *Lecture Notes in Computer Science*, 5304, 264–275. https://doi.org/10.1007/978-3-540-88690-7_20

Longa, A. (2019). *Graph Embedding in 2D* (Master's dissertation, University of Trento, Trento, Italy). Retrieved from https://antoniolonga.github.io/tesi/Graph_Embedding_in_2D.pdf

Tabrizi, S. A., Shakery, A., Asadpour, M., Abbasi, M., & Tavallaie, M. A. (2013). Personalized pagerank clustering: A graph clustering algorithm based on random walks. *Physica A: Statistical Mechanics and Its Applications*, 392(22), 5772–5785. https://doi.org/10.1016/j.physa.2013.07.021