The Colors of Love: Topic/Image Color Mining via Clustering

Colin Shea-Blymyer

December 2019

Abstract

What are the colors of love? It is useful for an analyst or artist to understand associations between concepts and colors used to express them. In this paper, I introduce an algorithm that uses cluster analysis to determine the dominant colors of a set of images. To this end, I apply four different clustering algorithms, analyze one in depth, and compare their suitability to the task. I present the results of these color mining algorithms on a number of queries and discuss their results.

1 Introduction

The relationship between an object's color and the emotions it evokes has been the topic of psychological study for years [9]. Scientists and artists alike are interested in how certain colors might better suggest happiness, for instance, than others. Not to be outdone, marketing researchers have employed numerous schemes relying on human emotional association with colors [8], with some findings suggesting that our evaluation of a product often happens based on color alone [12].

Determining the dominant color of an image has applications in information retrieval, as well. Much as one may wish to search a database of text documents for those pertaining to "love", so too might one search a database of images for those containing colors associated to "red" [13]. Neither of these are a simple matter, however: "love" has many analogues in the English language, and "red" may range from maroon to pink in the mind of the user. Indeed, [13] points out that a user might ask, non-specifically, for "red" apples, but would be more specific with the colors of a car.

Traditionally, the clustering of colors in images has been used for image segmentation [2] and information retrieval [5]. More recently, this image processing technique has been used to detect anomalies [7], leveraging the unsupervised nature of clustering.

In this work, I approach the problem of associating a set of colors with a topic. Given a set of images representing a topic, I apply cluster analysis to determine what colors comprise the palette of the images. This approach shares some similarities with topic mining approaches. Where topic mining aims to find the words that best characterize a set of documents, this work aims to find the colors that best characterize a set of images. In short, both are a fashion of dimensionality reduction [4].

2 Methods

To obtain a set of images representing a topic I use a popular image search engine. Querying the search engine with the topic returns a set of images. Each image has an independent horizontal width w and vertical height h measured in pixels. Each pixel is a 3-tuple representing a point in RGB space (loosely: a color); i.e. a pixel $p = (r, g, b)$ where $r, g, b \in [0 : 255]$ and $r, g, b \in \mathbb{I}$ represents each pixel's intensity of red, green, or blue, respectively. Thus, each image can be represented as a $w \times h \times r \times g \times b$ array.

Each image array is then processed into a pixel-frequency set, where each entry denotes a pixel's frequency in that image. These pixel-frequency sets are then combined for all images in the set, resulting in a "bag-of-pixels" model of the image set. This "bag-of-pixels" can be used to produce a histogram of color presence in the image set - a common approach to color quantization in the past [13].

The bag-of-pixels can then be handed to a density-based clustering algorithm. For clustering algorithms that expect an array of points, the image arrays may be flattened, concatenated, and passed in. Such an array, however, can easily reach tens of millions of entries, while fewer than 17 million points exist in RGB space. Not only does this tend to produce over-saturated samples, it also causes analysis to scale poorly. To avoid these issues, a sample of each pixel p is taken at a scaled inverse of its frequency $f \in F$, where the scaling factor is $\left\lfloor \sqrt{(\min F)^{-1}} \right\rfloor$. The floor square root minimum frequency normalization term has the desirable property of dropping out rare pixel values and maintaining more common colors while shrinking the sample count to a more manageable size.

2.1 Pixel Clustering

The analysis in this work employed four well-known clustering algorithms: k-means [1], Gaussian mixture models (GMM) [11], mean shift [3], and DBSCAN [6]. Clustering algorithms are a class of unsupervised machine learning that exploit expected structure in a data set to bin data samples into like-structured clusters. K-means is an example of a centroid-based algorithm, GMM is a distribution-based algorithm, and DBSCAN and mean shift are density-based algorithms. These algorithms represent the major families of clustering algorithms that are suitable for larger number of samples.

2.1.1 K-means

The k-means clustering algorithm tries to find clusters based on the locality of points to a centroid. In this application, the k-means algorithm partitions pixels $p \in P$ into k clusters $S_1, \ldots, S_k \in \mathbf{S}$. The parameter k is set to 8 in this work to respect the seven basic colors and gray-scale. The problem can be easily stated as a minimization problem on the variance of the clusters:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{p \in S_i} \|p - \mu_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where μ_i is the centroid of S_i. Solving the problem results in each pixel being assigned to the cluster belonging to the closest centroid to it. This produces a Voronoi decomposition of the space of samples, where each cell defines a cluster. This solution is approached via an iterative update procedure:

Algorithm 1 k-means algorithm

Require: $P : p \in \mathbb{R}^3$
  initialize centroids randomly $\mu_1, \ldots, \mu_k \in \mathbb{R}^3$
  repeat
    $S_i \leftarrow \arg\min_j \|p_i - \mu_j\|^2$
    $\mu_k \leftarrow \frac{1}{|S_i|} \sum_{p_j \in S_i} p_j$
  until convergence

Improving the initialization of centroids makes this algorithm more effective at large scales, and mini-batch can be applied to k-means as well.

2.1.2 Gaussian mixture models

A Gaussian mixture model is a probabilistic model that attempts to estimate the underlying distributions to which each observed sample belongs. In other words, the model tries to find a number of Gaussian distributions that generate the observed data. GMMs use an iterative process much like algorithm 1. Indeed, this approach can be seen as a generalization of k-means that takes into account further information about the distribution (rather than just the centroid). To solve this problem, GMMs employ the expectation-maximization algorithm to fit the parameters θ of k Gaussian distributions (algorithm 2).

In this work, the Bayesian information criterion is used to determine the number of components in the mixture model, however the algorithm does not scale well to larger data sets.

2.2 Mean shift

Mean shift clustering clusters points based on the nearest, densest area, as determined by kernel density estimation. As such, it is considered a density

Algorithm 2 expectation-maximization algorithm

Require: $P : p \in \mathbb{R}^3$; latent data $\mathbf{Z}$; likelihood function L
  initialize parameter estimate θ_t
  repeat
    $Q(\theta|\theta_t) \leftarrow \mathbb{E}_{\mathbf{Z}|P,\theta_t}(\log L(\theta; P, \mathbf{Z}))$
    $\theta_{t+1} \leftarrow \arg\max_{\theta} Q(\theta|\theta_t)$
  until convergence

based clustering technique, but shares characteristics with k-means and GMMs in its iterative updates (algorithms 1 and 2). This algorithm maximizes the density of each cluster, shifting the mean of each kernel until density gains are no longer made. While the algorithm does not take the number of clusters as a parameter, it does require a bandwidth for the kernel density estimation step. The bandwidth itself can be estimated, however this leads to increased complexity on an already complex update step. Thus, mean shift does not perform well for larger data sets.

2.2.1 DBSCAN

The DBSCAN algorithm is a density based clustering approach that differentiates points as being members of different high-density areas separated by low-density areas. This allows clusters to take many more shapes than the generally convex hulls of k-means and GMMs. DBSCAN takes parameters to define whether or not another point should be considered a neighbor, and to define how many neighbors are required to consider a point as part of a cluster. DBSCAN performs well with larger data sets, but somewhat worse as the data becomes more fragmented.

3 Results

Associating the dominant colors of a set of images with a specified topic requires the set of images itself to be associated with the topic. To this end, I employ an HTML parser (Beautiful Soup) to extract images from a query to an image search algorithm (Google Images). This process extracts 100 images; some of which are discarded due to formatting issues (e.g. animated ".gif"). This generally supplies me with more than 70 images to analyze, as detailed in the Methodology section. Analysis is performed in python, using the Scikit-learn [10] implementation for clustering algorithms.

3.1 The Colors of Love

To demonstrate the insights gained by this style of analysis, I present the results of this process on the query "love". A visual inspection of the web results of this search (presented in figure 1) primes our expectation for this result. We see the

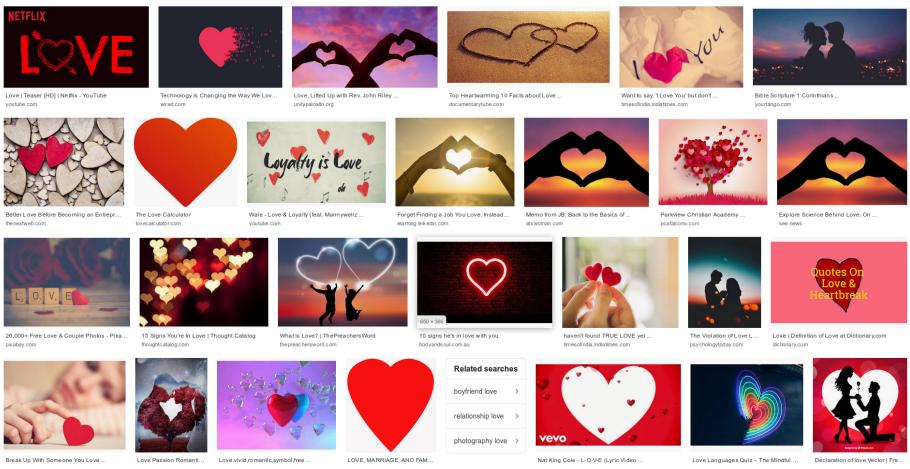


Figure 1: Google Image search results for query "love".

saturated reds of Valentine's day, but much more color is present than reds and pinks. Indeed, the histogram of colors (figure 2) shows that blues and yellows punctuate the pink and red of hearts and the black and white of backgrounds.

This set of images includes more than 100 million pixels, and more than 2 million unique color values of a possible 16.8 million colors. Before clustering the pixels, I plot the pixels in *RGB* space (3), providing an understanding of how different clustering techniques will assign labels.

In figure 3, we can see streaks of magenta, patches of blue, clusters of red, and a bifurcated, gray-scale band of color across the space. With this in mind, the results of the four clustering algorithms can be interpreted.

As seen in figure 4, the mean shift algorithm estimates the lowest number of clusters; seemingly sticking to four corners of *RGB* space. The eight clusters required of k-means fairly closely resemble the eight estimated to exist by GMM. This suggests the GMM failed to incorporate much information beyond the centroids of its Gaussian distributions. DBSCAN offers the largest number of clusters (a common trend in these experiments), following the particular lines of color gradient used in the pictures. Curiously, the mean colors of the clusters identified by DBSCAN lend themselves to be, themselves, clustered. Many of these clusters appear as shades of others nearby. Subjectively, the shades of red and pink identified by each of the clustering algorithms match the expectation of Valentine's day colors. The presence of black reflects its common use as a background color. Blue, however, comes as a surprise. Looking back to figure 1, we can see the presence of blue in the hues of the sky, or as a highlight to more central reds. Taken artistically, the presence of softer blues contrasts the passionate reds and the gentle pinks, presenting a more balanced picture.

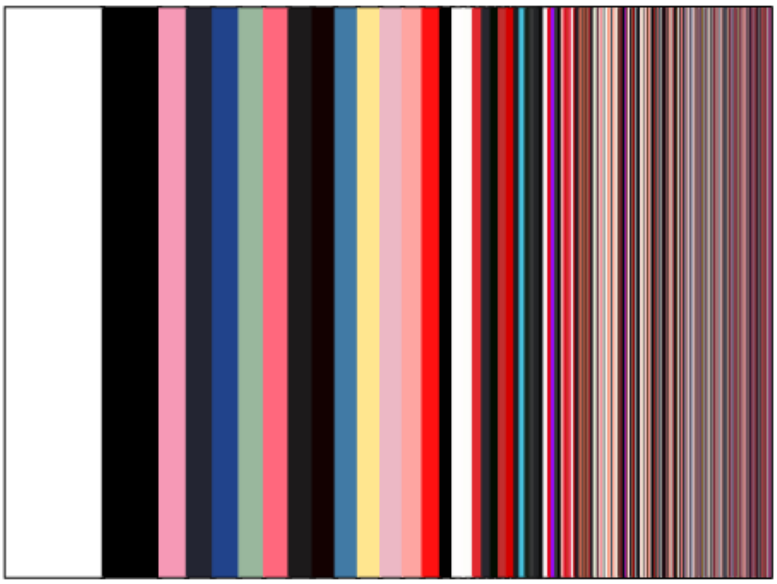


Figure 2: Histogram of the 100 most common colors present in image set gathered for query "love". Width of each bar represents the relative log-frequency of the color depicted in the bar. Bars are ordered, descending, from left to right by their frequency in the dataset.

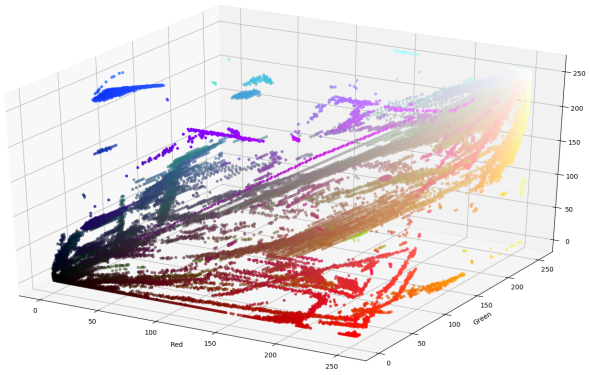


Figure 3: Plot of the pixels from the "love" image set in *RGB* space. Each point represents a pixel value, and its color reflects that value.

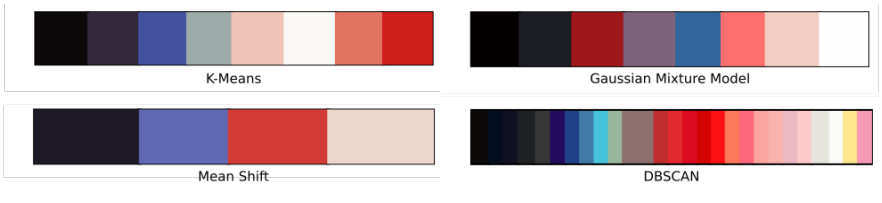
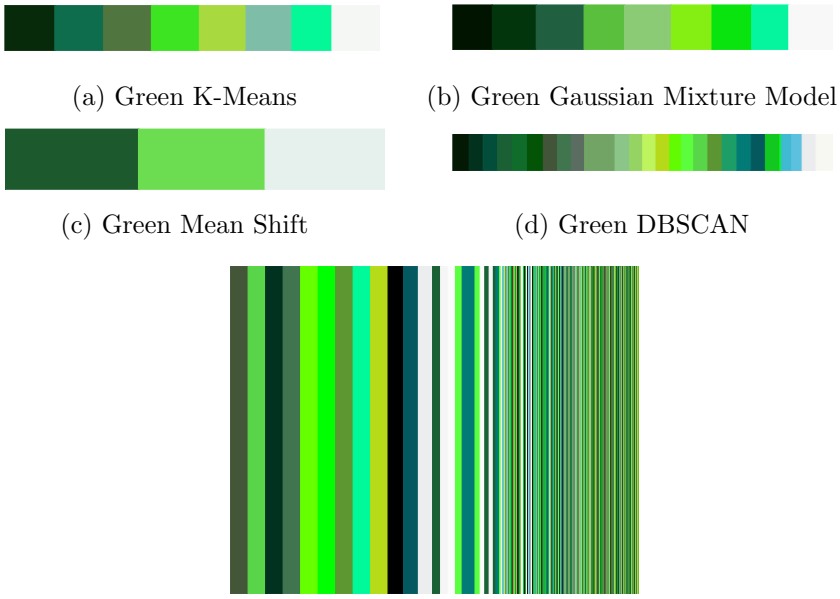


Figure 4: The results of four clustering techniques on the pixels of the "love" image set. The width of each bar represents the log of the relative size of a cluster, and the bar's color corresponds to the color of the centroid of that cluster. Bars are ordered, ascending, from left to right by their associated centroid's distance from black: the *RGB* point (0, 0, 0).



(a) Green K-Means

(b) Green Gaussian Mixture Model

(c) Green Mean Shift

(d) Green DBSCAN

(e) Histogram

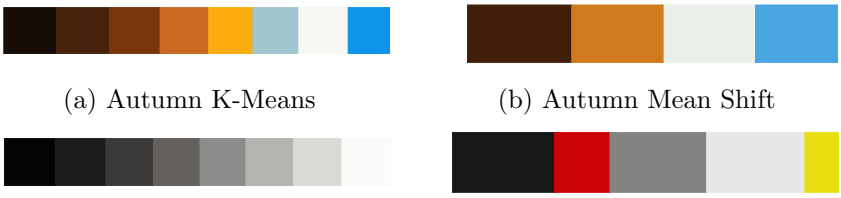
Figure 5: Clustering results for the query "green".

3.2 Further Experiments

As a sort of sanity check, I have performed this analysis on queries specifying a color, e.g. "green". Figure 5 shows that the analysis does discover an abundance of green pixels. In particular, the mean shift algorithm reduces the color space to three clusters - a beacon of expected performance when the query itself is a single color. Of note, however, is the distribution of colors in the histogram. While the image set for the query "green" maintains the general logarithmic distribution apparent for "love" in figure 2, the more common colors better share the spotlight, so to speak. That is, the more common colors in the "green" set appear in similar proportions, while in the "love" set, even colors that are slightly less common appear in much smaller quantities.

This variety of analysis can be easily performed on any topic or search query that comes to mind. Some queries present colors that the clustering algorithms largely agree on, while others seem to defy consensus (figure 6). This seems to be related to the paradigm, or lack thereof, for presenting images on the given topic.

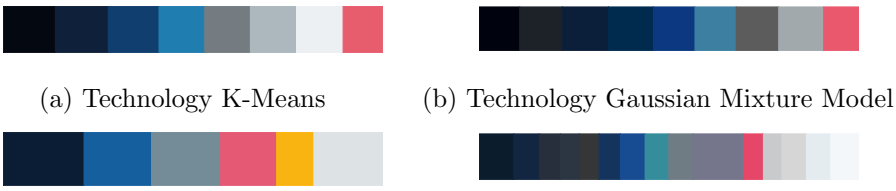
Finally, I present the results of this analysis on the image set gathered from the query "technology" in figure 7. These results show that technology is often depicted as blue (perhaps IBM set the trend?), and often accented with a splash of red.



(a) Autumn K-Means (b) Autumn Mean Shift

(c) Punk K-Means (d) Punk Mean Shift

Figure 6: Clustering results for the "coherent" image set "autumn", and the "incoherent" image set "punk".



(a) Technology K-Means (b) Technology Gaussian Mixture Model

(c) Technology Mean Shift (d) Technology DBSCAN

Figure 7: Clustering results for the query "technology".

4 Conclusion

While the clustering of colors in an image has been performed in the past, its use in concert with an image search engine to extract color features for topics is novel. In this work I present a color analysis process to link topics to colors that commonly represent them. This process has the benefit of requiring relatively little data to extract useful features, and leverages existing information retrieval tools. Furthermore, with careful selection of clustering algorithm, this process will scale well to larger data sets.

While this technique does perform admirably with only a few dozen images, the results shown in figures 3 and 4 suggest that more data may better resolve the distributions of colors used in a set of images. In figure 3, one can identify clear lines of gradient and patches of hues present in possibly only one or two images of the set. As data size grows, these irregularities should regularize, or become drowned out by the presence of other colors. However, in this analysis memory-saving techniques were already employed to analyze the data on a relatively light computer. With increased data size, analysis may become restricted to higher performance environments, or more strict memory-saving techniques will have to be implemented.

One particular challenge to analysis of color space is its density. As mentioned earlier, the unique colors of the "love" image set occupied almost one 16th of its space. With increasing data set sizes, and less cohesive query results,

an even greater proportion of the space may be filled. This lack of empty space can make it difficult for density based clustering algorithms to draw boundaries. These problems may be addressed, however, with techniques from topic mining. This is supported by the analogy between this work's "bag-of-pixels" and the "bag-of-words" model in natural language processing. The use of more complex, topic mining models should be conscious of the difference in the temporal nature of language and the convolutional nature of images - a difference that is avoided in the "bag-of" approaches.

Another challenge in color space analysis is the discrepancy between color space and human perception. This is especially apparent in *RGB* space, where pinks are closer to light blues than to reds. Future work in this space should consider translating images into a more human-friendly color space such as *HSV*. Further, the shapes of clusters found by the likes of DBSCAN in particular can flaunt human expectation. Consider a gradient from red to blue; if a consistently dense chain of colors stretches from one corner to another, DBSCAN would likely consider it a unique cluster. Meanwhile a human observer would be hesitant to put blue and red in the same bin, even in the presence of a magenta gradient. This problem is less obvious on a gradient from black to white, as humans have a more difficult time distinguishing shades of gray from shades of magenta, despite the red-blue corners being closer together in *RGB* space.

While this approach to topic/image analysis is relatively simple, it serves as a tool that is easy to implement and requires little data to return interesting results.

References

[1] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[2] M. Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics, and image processing*, 52(2):145–170, 1990.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):603–619, 2002.

[4] S. P. Crain, K. Zhou, S.-H. Yang, and H. Zha. *Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond*, pages 129–161. Springer US, Boston, MA, 2012.

[5] Y. Deng, B. Manjunath, C. Kenney, M. S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on image processing*, 10(1):140–147, 2001.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[7] S. D. Khirade and A. Patil. Plant disease detection using image processing. In *2015 International conference on computing communication control and automation*, pages 768–771. IEEE, 2015.

[8] L. I. Labrecque and G. R. Milne. Exciting red and competent blue: the importance of color in marketing. *Journal of the Academy of Marketing Science*, 40(5):711–727, 2012.

[9] K. NAz and H. Epps. Relationship between color and emotion: A study of college students. *College Student J*, 38(3):396, 2004.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. Gaussian mixture models and k-means clustering. *Numerical recipes. The art of scientific computing, 3rd ed.: Cambridge University Press*, 843:846, 2007.

[12] S. Singh. Impact of color on marketing. *Management decision*, 44(6):783–789, 2006.

[13] J. R. Smith and S.-F. Chang. Single color extraction and image query. In *Proceedings., International Conference on Image Processing*, volume 3, pages 528–531. IEEE, 1995.