Data Science + Public Policy

Jeff Chen

2018-11-12

DIY: How much of the ground is covered in [vegetation/buildings/economic activity]?

Photographs contain data. Some are more structured than others. Satellite imagery, for example, can be directly used to infer patterns on the ground, especially relating to natural phenomena like agriculture. Free imagery is readily available from various satellite instruments such as Aqua/Terra MODIS (NASA), ASTER (NASA/Japan), VIIRS (NASA/NOAA), Landsat Operational Land Imager (NASA/USGS), among others. Private firms such as Digital Globe and Planet also operate their own satellites and provide commercial data services.

Suppose there is a need to know how much healthy vegetation is present in farm lands in the US heartland. Satellite imagery can easily be used to support this task. A U.S.-Japan team used the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument on the Terra satellite to capture images of crop fields in Kansas. The image below captures a 37.2-km x 38.8-km area where green areas indicate healthy vegetation.[1]

```r
#Library
  library(raster)
  library(digIt)

  img <- digIt("color_segment_kansas")
  plotRGB(img)
```

Suppose the following question were asked:

> How much of the crop field is covered in healthy vegetation?

In earth science, satellite imagery can be converted into vegetation indices, then cutoffs can be applied. The choice of a cutoff runs the risk of subjective biases. The alternative is to use k-means clustering to conduct *color quantization*, which is a process that reduces the number of colors in an image into fewer distinct colors.

Photographs are comprised of a three-dimensional array that essentially resembles three matrices sandwiched together. Each matrix is an $n \times m$ matrix for each red-green-blue (RGB). The goal is to cluster on the colors, which requires the each of the $n \times m$ matrices to be transformed into a two dimensional matrix with 3 columns (one for each color) and of length nm. K-means is applied to this matrix to obtain color groups.

Getting Started

In the wild, the `digIt()` function is not available. An image would normally need to be downloaded and loaded as a `brick()` using the `raster` package. For simplicity, we use the `digIt` library to download and load the ASTER data.

```r
#Library
  library(raster)
  library(digIt)

  img <- digIt("color_segment_kansas")
```

The image is converted into a matrix containing three vectors of equal length: one for each RGB value.

[1] https://www.nasa.gov/topics/earth/earthmonth/earthmonth__2013__01.html

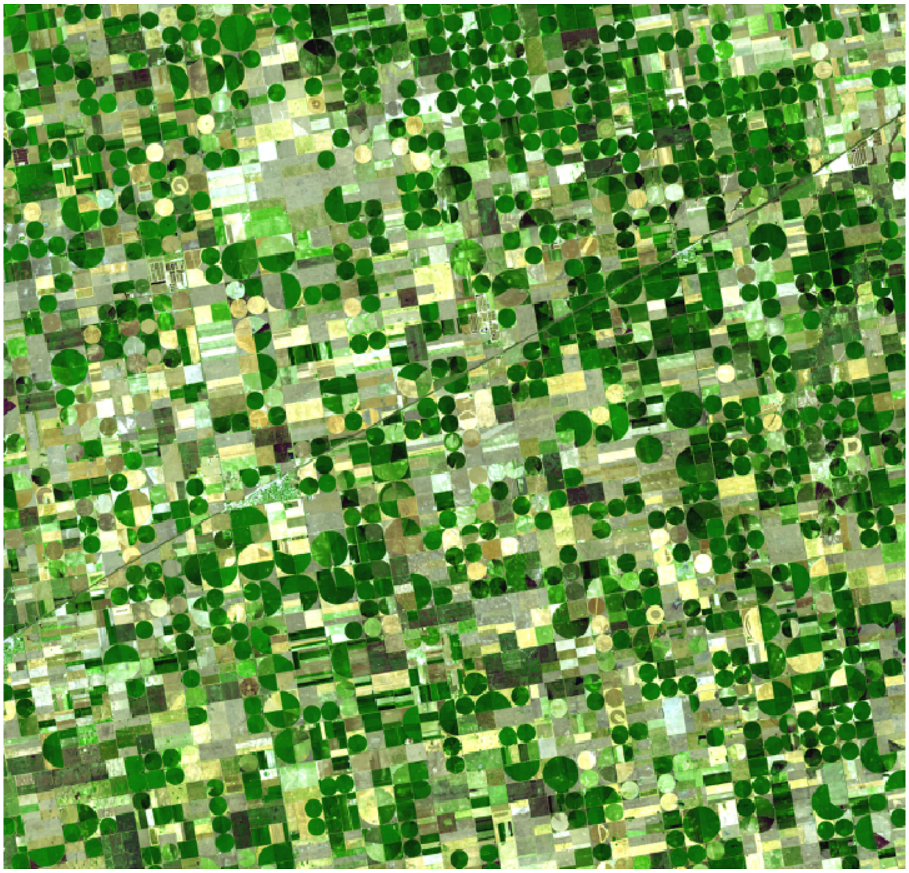

Figure 1: Crops in Finney County KS. Via NASA/GSFC/METI/Japan Space Systems, and U.S./Japan ASTER Science Team

```
#Dimensions
  dim(img)
```

```
## [1] 2481 2589    3
```

```
#Convert image into columns
  data <- cbind(as.vector(img[[1]]),
                as.vector(img[[2]]),
                as.vector(img[[3]]))
```

With the data in the right shape, k-means can be applied. In this example, we use the `kmeans()` function that is built into R:

`kmeans(x, k)`

where:

- `x` is a data frame or matrix of numerical values
- `k` is the number of clusters

The result of the `kmeans()` function contains a number of attributes such as the cluster assignment of each observation. To evaluate the fitness of the cluster, a silhouette statistic can be calculated using the `silhouette()` function in the `cluster` library:

`silhouette(cluster, distance)`

where:

- `cluster` is the cluster assignment.
- `distance` is a dissimilarity matrix of input features produced by `dist()`.

Given the size of the input matrix ($n = 2481 \times 2589 = 6423309$), the dissimilarity matrix is produced on a sample of $n = 20000$.

Below, k-means is tested for values of $k = 2$ to $k = 10$ using a random sample of $n = 20000$. Before the loop, the sample is taken, then the dissimilarity matrix is calculated using the `dist()` function. Within the loop, k-means results are assigned to the object `res` from which the cluster assignments are extracted. The silhouette is then calculated and assigned to the `sil` object, from which the mean silhouette is estimated from observation level silhouettes (third column).

```
#Load cluster library
  library(cluster)

#Calculate distance object using sample of n = 10000
  set.seed(10)
  subdata <- data[sample(data, 20000),]
  d <- dist(subdata)

#Set up placeholder for silhouette values
  sil.out <- data.frame()

#Loop through values of k
  for(k in 2:10){

    set.seed(20)

    #Run k-means, save to o
    res <- kmeans(subdata, k)

    #Get silhouette
```

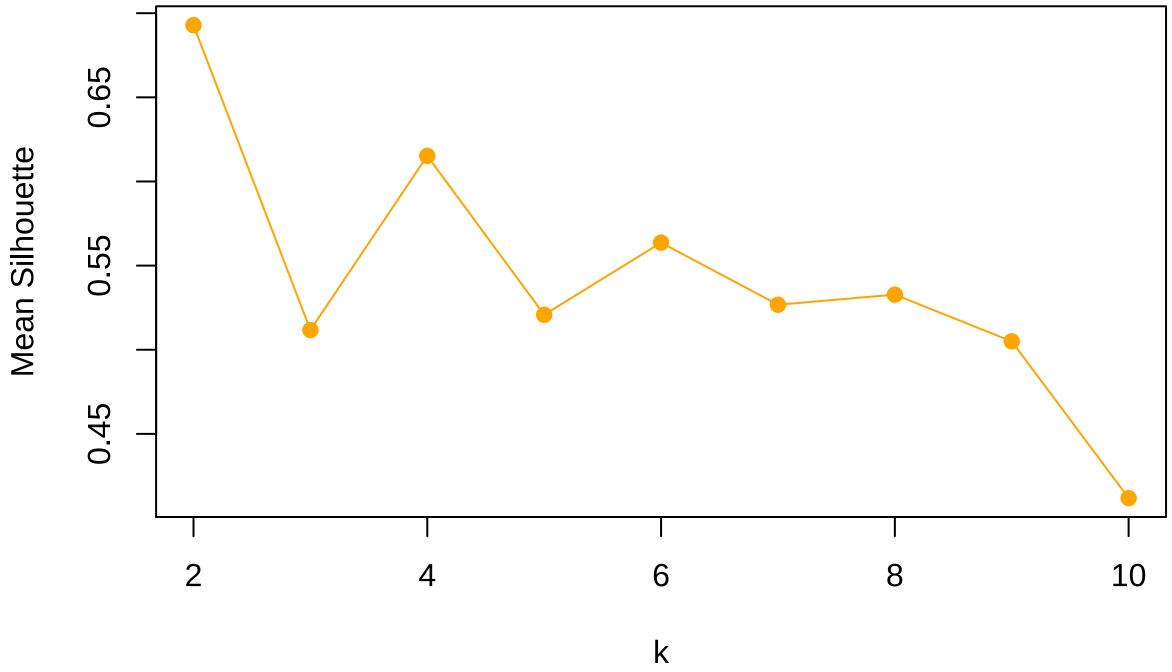

Figure 2: Mean silhouette by k

```
    sil <- silhouette(res$cluster, d)

    #Get summary values of silhouette
    temp <- data.frame(k.level = k,
                       avg = mean(sil[,3]))
    sil.out <- rbind(sil.out, temp)
  }
```

In color quantization exercises, lower values of k should be used. In the case below, the grid search suggests that $k = 2$ provides the most favorable cluster results.

```
#Plot result
  plot(sil.out[, c("k.level", "avg")], type = "l", col = "orange",
       ylab = "Mean Silhouette", xlab = "k")
  points(sil.out[, c("k.level", "avg")], pch = 19, col = "orange")
```

The k-means model is then estimated on the entire data set for $k = 2$. To visually check our results, we need to convert the vector of cluster assignments to a matrix with the same dimensions as the original image `img`. The matrix contains all the same information as an image, but is not in the right data class. Using `raster()`, the matrix can be converted into an raster image format containing cluster assignments.

```
#K values
  set.seed(123)
  res <- kmeans(data, 2)

#Convert cluster labels into matrix
  mat <- matrix(res$cluster,
                ncol = ncol(img),
                nrow = nrow(img),
                byrow = TRUE)
  img2 <- raster(mat)
```

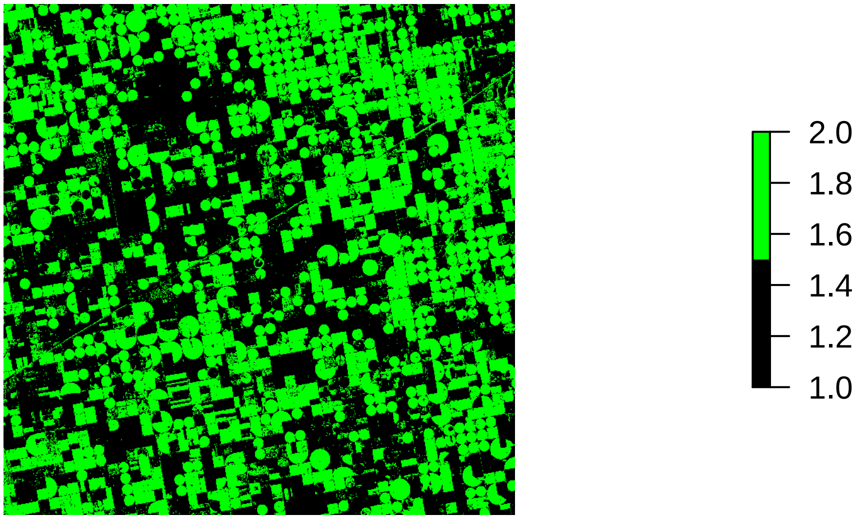

Figure 3: Color quantization for k = 3

With the data in the right form, the cluster assignments are rendered as an image. Notice that the healthy green areas are coded in green, which corresponds with cluster #2.

```r
plot(img2, box=FALSE, yaxt = "n",  xaxt = "n",
     frame.plot = FALSE, col = c("black", "green"))
```

To calculate the proportion of the land that is covered in healthy vegetation as well as approximate land area, we can use the following calculation:

```r
prop <- mean(mat == 2)
print(paste0("%Area = ", prop))
```

```
## [1] "%Area = 0.482713349147612"
```

```r
print(paste0("km2 = ", 37.2 * 38.8 * prop))
```

```
## [1] "km2 = 696.729139625697"
```