MATH089 Project 6 - Graph Laplacian and Spectral Clustering

Posted: 11/15/21

Due: 11/22/21, 11:55PM

Science usually arises from observation of regularity in data. A first step in all scientific analysis is the organization of such observations into groups based upon some similarity criterion. Grouping through quantitative mathematical procedures requires the definition of a distance between observations and a procedure to form observation clusters. This project is an introduction to this basic task in data science using classification of letter images as an example.

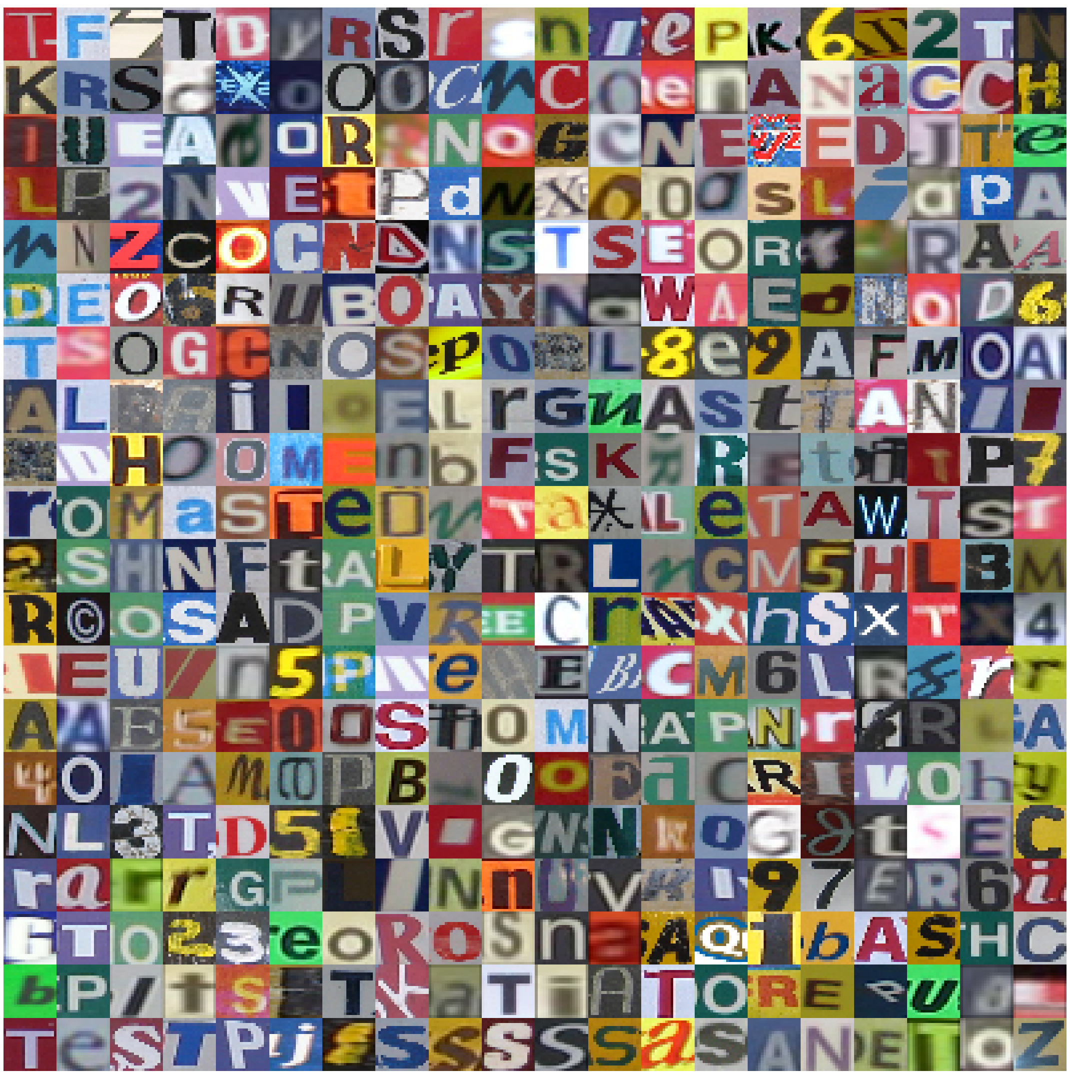

Figure 1. First 400 characters in Kaggle data base (First Steps With Julia | Kaggle)

Tasks:

- Establish a distance function suitable for character identification
- Establish number of clusters
- Test clustering procedure for new character images

1 Introduction

(Read First Steps With Julia | Kaggle and research k-Means algorithm, e.g., Wikipedia)

2 Methods

2.1 Import data

- The Kaggle database on character images contains 6220 files in .Bmp format, each of size 20 x 20 pixels. The ImageMagick package allows the images to be imported into Julia.

Once the Julia environment has been updated to contain the ImageMagick and Images packages, they can be used to analyze the character image data.

```
∴ using Images, ImageMagick
```

```
∴ cd(homedir()*"/courses/MATH089/clustering/testResized")
```

```
∴ files=readdir(); c=load(files[1]); nFiles=size(files); mx,my=size(c);
  [nFiles mx my]
```

$$\left[\ [\ 6220\ ]\ \ 20\ \ 20\ \right] \tag{1}$$

```
∴ c[1,1]
```

```
RGB{N0f8}(0.475,0.424,0.514)
```

```
∴ Gray(c[1,1])
```

```
Gray{N0f8}(0.451)
```

```
∴ Float64(Gray(c[1,1]))
```

0.45098039215686275

```
∴ C=Float64.(Gray.(c)); imshow(C,cmap="gray");
```

2.2 k-means algorithm

- Clustering is a common data analysis procedure, and a Julia package is available.

```
∴ import Pkg; Pkg.add("Clustering")
```

```
   Resolving package versions...
   Installed NearestNeighbors — v0.4.9
   Installed Clustering ——————— v0.14.2
    Updating `~/.julia/environments/v1.6/Project.toml`
  [aaaa29a8] + Clustering v0.14.2
    Updating `~/.julia/environments/v1.6/Manifest.toml`
  [aaaa29a8] + Clustering v0.14.2
  [b8a86587] + NearestNeighbors v0.4.9
Precompiling project...
[32m  ✓ [39m[90mNearestNeighbors[39m
[32m  ✓ [39mClustering
  2 dependencies successfully precompiled in 3 seconds (288 already
precompiled)
```

```
∴
```

The k-means algorithm seeks to group n observations, each of dimension d into k clusters. For this application an observation is one of the character images and $d = 3m_x m_y$ using all color channels.

```
∴ using Clustering
```

```
∴
```

2.2.1 Single vector representation of color image

Each pixel in the image contains three color channels

```
∴ c[1,1]
```

RGB{N0f8}(0.714,0.153,0.188)

```
∴ red(c[1,1])
```

0.714 N 0 f 8

```
∴ Float64(red(c[1,1]))
```

0.7137254901960784

Define a function to form a single vector of size $3m_x m_y$ from an image of size $m_x \times m_y$

```
∴ function FormObs(c)
    mx,my = size(c); d = 3*mx*my;
    r=Float64.(red.(c)); g=Float64.(green.(c)); b=Float64.(blue.(c));
    reshape([r g b],(d,1))
  end
```

FormObs

```
∴ x=FormObs(c);
```

```
∴ size(x)
```

$$\begin{bmatrix} 1200 \\ 1 \end{bmatrix} \tag{2}$$

```
∴ size(c)
```

$$\begin{bmatrix} 20 \\ 20 \end{bmatrix} \tag{3}$$

2.2.2 Apply *k*-means on data subset

The data set contains vectors $\boldsymbol{x}_i$ for $i = 1, \ldots, n$, where n is the number of images, and each $\boldsymbol{x}_i$ contains $d = 1200$ numbers. The goal of the k-means algorithm is to define sets S_j for $j = 1, \ldots, k$ that contain similar images, presumably of the same letter. Let $S = \{S_1, \ldots, S_k\}$. The k-means algorithm minimizes the variance in each cluster

$$\min \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in S_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|^2$$

```
∴ function FormDataSet(n0,n,f)
    c = load.(f[n0:n0+n-1])
    X = FormObs(c[1])
    for i=2:n
      X = [X FormObs(c[i])]
    end
    return(X)
  end
```

FormDataSet

```
∴ n0=200; n=100; X=FormDataSet(n0,n,files); size(X)
```

$$\begin{bmatrix} 1200 \\ 100 \end{bmatrix} \tag{4}$$

```
∴ kGroups=2; K=kmeans(X,kGroups);
```

```
∴ S=assignments(K); nS=counts(K); C=K.centers;
```

```
∴ S[1:10]'
```

$$\begin{bmatrix} 2 & 1 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 1 \end{bmatrix} \tag{5}$$

```
∴ size(C)
```

$$\begin{bmatrix} 1200 \\ 2 \end{bmatrix} \tag{6}$$

```
∴
```

2.3 Display *k*-means results

2.3.1 Display centers

Each center represents an image. Display it.

3 Results

4 Discussion

(Present your conclusions on the use of random numbers to simulate real-life models)