Assignment 15 - Homework - Clustering

Due Apr 19 by 11:59pm | **Points** 20 | **Submitting** a file upload | **File Types** pdf

1. Exploratory Data Analysis (5 pts)

1. Relatively how many terms appear in exactly one document?

```
   julia> irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
   julia> termfreq = Serialization.deserialize("./data/termfreq.jldata")
   julia> terms = Serialization.deserialize("./data/terms.jldata")
   julia> termfreq[1,1:end].nzind
84-element Vector{Int64}
```

> In the first document there are a total of 84 terms.

```
using DelimitedFiles # To use `writedlm` function
using Serialization
using Statistics
irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
termfreq = Serialization.deserialize("./data/termfreq.jldata")
terms = Serialization.deserialize("./data/terms.jldata")
number_of_terms_in_document = []
number_of_documents = length(irs990extract) # 260783
for row_of_documents in 1:number_of_documents
   row = termfreq[row_of_documents,1:end].nzind
   println(length(row))
   push!(number_of_terms_in_document,length(row))
end
println("***")
println("Minimum: ",Statistics.minimum(number_of_terms_in_document))
println("Maximum: ",Statistics.maximum(number_of_terms_in_document))
println("Average: ",Statistics.mean(number_of_terms_in_document))
println("***")
# takes about 5m20.298s to run
writedlm( "numberOfTerms.txt",  number_of_documents, ',')
# $ cat numberOfTerms.txt | sort -n | uniq -c > uniq.txt
```

> On average there are about 21.718225497827696 terms across all documents.
>
> In any one document relatively there can be as much as 18 terms and as little as 3 terms in a single document

2. Relatively how many terms appear at least 5 times?

```
# takes 0m6.408s to run
irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
termfreq = Serialization.deserialize("./data/termfreq.jldata")
terms = Serialization.deserialize("./data/terms.jldata")
number_of_terms_counter = 0
document = termfreq[1:end,:]
terms_appeared = termfreq = 0 .< document
count_of_words_appeared = sum(terms_appeared, dims = 1)
for word in 1:length(terms)
   if count_of_words_appeared[word] >= 5
      number_of_terms_counter += 1
   end
end
println(number_of_terms_counter)
```

> 14235 terms appear at least 5 times.

3. Show the 20 most frequent words. Words like "and", "to", "the" aren't especially meaningful. Which is the first word that you feel may be meaningful for characterizing the nonprofit? Why?

```
using Serialization
using Statistics
irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
termfreq = Serialization.deserialize("./data/termfreq.jldata")
terms = Serialization.deserialize("./data/terms.jldata")

top_twenty_array = Array{Int64}(undef, 20, 2)
for term in 1:length(terms)
   for index in 1:20
      if length(termfreq[1:end,term].nzind) > top_twenty_array[index,1]
            for row in 20:index
               top_twenty_array[row,1] = top_twenty_array[row-1,1]
               top_twenty_array[row,2] = top_twenty_array[row-1,2]
            end
            top_twenty_array[index,1] = length(termfreq[1:end,term].nzind)
            top_twenty_array[index,2] = term
            break
      end
   end
end
for row in 1:20
   println(terms[top_twenty_array[row,2]])
end
```

> `school` for it loosely argues that for a non-profit organization to exist with a focus to our education it may imply that our public education system lacks staff, resources, or basic assistance that our society/government fails to provide

4. How many documents contain "sacramento"?

```
julia>for term_column in 1:length(terms)# 79653
         if(cmp(terms[term_column], "sacramento")==0)
               println("index = ", term_column)
         end #end if
      end #end for
index = 63171
```

`sacramento` is stored in index = 63171

```
using Serialization
using Statistics
irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
termfreq = Serialization.deserialize("./data/termfreq.jldata")
terms = Serialization.deserialize("./data/terms.jldata")
sacramento_document_counter = 0
for document_row in 1:length(irs990extract) # 260783
   if(cmp(terms[63171], "sacramento")==0)
      sacramento_document_counter += 1
   end
end
println(sacramento_document_counter)
```

> 260783 documents contains "sacramento"

5. What's one element in `irs990extract` where the mission contains "sacramento"?

Come up with your own question similar to the questions above, and answer it.

2. Selecting a Subset (5 pts)

What do you do when your program doesn't run? Try using a subset of the data, the most important subset.

```
first10k = 1:10_000
termfreq10k = termfreq[first10k,:]
termAppeared = 0 .< termfreq10k # So if these terms are positive and the term did appear
wordAppearanceCount = sum(termAppeared, dims = 1)
```

1. Use one or more of the fields in `irs990extract` to define and pick the 10,000 largest nonprofits.

> PLACE_HOLDER

2. What's the largest nonprofit based on your definition? Does it seem reasonable?

> PLACE_HOLDER

3. Drop all words that don't appear at least twice in this subset.

> PLACE_HOLDER

We'll use this subset for the remainder of the assignment.

3. Principal Components Analysis (5 pts)

Fit the first 10 principal components, i.e. project the data down into a 10 dimensional subspace.

1. Interpret the principal ratio. What does it mean?

> PLACE_HOLDER

2. Plot the variances of the first 10 principal components as a function of the principal component number. What do you observe?

> PLACE_HOLDER

3. Which words have the relatively largest loadings in the first principal component? (These the absolute values of the entries of `projection()`.) Are these the kinds of words you expected? Explain.

> PLACE_HOLDER

Hints

1. Resources for interpreting principal components: Making sense of principal component analysis, eigenvectors & eigenvalues PCA and proportion of variance explained
2. Transpose the matrix to follow the structure described in the MultivariateStats documentation
3. If the program is too slow, try converting from a dense to a sparse matrix.

4. Clustering (5 pts)

Apply k means with k = 3 to the principal components of the subset of data. This means you should be fitting k means to a data matrix with 10,000 observations, and 10 features, which are the scores for each of the 10 principal components.

1. How many elements are in each group?

> PLACE_HOLDER

2. Which nonprofits are closest to the centroids? Feel free to use the function below.

> PLACE_HOLDER

3. k means should find a group of mission statements that are very similar. What happened?

> PLACE_HOLDER

Is it reasonable?

> PLACE_HOLDER

If we were to continue this analysis, what would you do next?

> PLACE_HOLDER

```
"""
    Find the indices of the data points that are closest to the centroids defined by the
"""
function close_centroids(knn_model)
    groups = knn_model.assignments
    k = length(unique(groups))
    n = length(groups)
    result = fill(0, k)
    for ki in 1:k
        cost_i = fill(Inf, n)
        group_i = ki .== groups
        cost_i[group_i] = knn_model.costs[group_i]
        result[ki] = argmin(cost_i)
    end
    result
end
```

5. Extra Credit (1 pt)

Fit k means with k = 3 to the entire original `termfreq` data. This takes around 18 hours to run. Did k means again find a group of mission statements that are very similar, following the same pattern as in the previous question?

> PLACE_HOLDER

Notes

- A sparse matrix and what sparsity means is that most of the entries are zeros
    - extract just one row I get a sparse vector out so one dimensional vector

```
    > irs990extract = Serialization.deserialize("./data/irs990extract.jldata")
    > termfreq = Serialization.deserialize("./data/termfreq.jldata")
    > terms = Serialization.deserialize("./data/terms.jldata")
    > length(irs990extract)
260783
    > size(termfreq)
(260783, 79653) # 260783 number of rows in our matrix and corresponds to

# row of termfreq that corresponds to the first element of irs990extract
```

```
> row1 = termfreq[1,1:end].nzind
```