Homework 2: Unsupervised Learning and Clustering

Harvard CS 109B, Spring 2018

Jan 23, 2018

Contents

Homework 2 is due February 19, 2018

"Handy" Algorithms

In this assignment, you will be working with data collected from a motion capture camera system. The system was used to record 12 different users performing 5 distinct hand postures with markers attached to a left-handed glove. A set of markers on the back of the glove was used to establish a local coordinate system for the hand, and 11 additional markers were attached the the thumb and fingers of the glove. Three markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. Finally, 2 markers were attached to each finger with one above the fingernail and the other in the middle of the finger. A total of 36 features were collected resulting from the camera system. Two other variables in the dataset were the ID of the user and the posture that the user made.

The data were partially preprocessed. First, all markers were transformed to the local coordinate system of the record containing them. Second, each transformed marker with a norm greater than 200 millimeters was eliminated. Finally, any record that contained fewer than three markers was removed.

A few issues with the data are worth noting. Based on the manner in which data were captured, it is likely that, for a given record and user, there exists a near duplicate record originating from the same user. Additionally, There are many instances of missing data in the feature set. These instances are denoted with a ? in the dataset. Finally, there is the potential for imbalanced classes, as there is no guarantee that each user and/or posture is represented with equal frequency in the dataset.

The dataset, provided in CSV format, contains 78,095 rows and 38 columns. Each row corresponds to a single instant or frame as recorded by the camera system. The data are represented in the following manner:

`Class - Integer. The hand posture of the given obervation, with 1=Fist (with thumb out), 2=Stop (hand flat), 3=Point1 (point with index finger), 4=Point2 (point with index and middle fingers), 5=Grab (fingers curled as if to grab).`

`User - Integer. The ID of the user that contributed the record.`

`X0, Y0, Z0, X1, Y1, Z1, ..., X11, Y11, Z11 - Real. The x-coordinate, y-coordinate and z-coordinate of the twelve unlabeled marker positions.`

1 Missing Data Imputation

The data contain many missing values. Before attempting to perform any statistical procedures, we will need to address the missing data. One way to address missing data is to impute it.

Given the knowledge of how the data was collected, we can hypothesize that there are two ways in which the data might cluster together: by user and by posture. Perhaps the users have significantly different heights and/or hand sizes, resulting in the data generated by each user to be distinct from each other. Or, perhaps the hand postures are sufficiently unique such that the markers on the glove tend to be grouped together by the posture, regardless of who the user is. We will examine these hypotheses to see if either one provides a reasonable way to impute the data.

For this part of the assignment, you will want to use the following libraries:

```
# import libraries

library(dplyr)

library(ggplot2)

library(cluster)

library(mclust)

library(factoextra)

library(NbClust)

library(dbscan)
```

Write code to impute the missing data values with the mean of their respective feature (column), grouped by both users and postures. That is, you should create two new dataframes: one where the missing values are replaced by the mean of the user's feature, and one where the missing values are replaced by the mean of the posture's feature.

Hint: when loaded into R, the raw CSV might list the observations as factors. You will want to change that. One way to convert factors to numeric is to cast the columns of the dataframe like so:

```
for (i in 1:ncol(data)) {

  data[,i] <- as.numeric(as.character(data[,i]))

}
```

The "dplyr" package might also be useful for data cleaning.

Helper Function: Col means data imputation

Hint: function to impute means for one column, you will want to adapt this for all the necessary columns.

```
impute_column <- function(column) {

missing_indices <- which(is.na(column))

column_mean <- mean(as.numeric(column[-missing_indices]))

column[missing_indices] <- column_mean

return(column)

}
```

For data imputation, you can use this code after you make necessary adjustment above

First, create each imputed posture data frame

```
`for (posture in unique(mydata$Class)) {

  df <- mydata %>% filter(Class == posture)

  df_imputed_means <- impute_means(df)

  assign(paste0("imputed_posture_means", posture), df_imputed_means)

}`
```

Second, create each imputed user data frame

```
`for (user in unique(mydata$User)) {

  df <- mydata %>% filter(User == user)

  df_imputed_means <- impute_means(df)

  assign(paste0("imputed_user_means", user), df_imputed_means)
}`
```

Then, stack data frames back together (inserting appropriate variable names instead of ellipsis).

```
imputed_posture_all <- rbind(imputed_posture_means1, ..., imputed_posture_means5)

imputed_user_all <- rbind(imputed_user_means0, ... , imputed_user_means14)
```

Finally manage your stack and remove clutter (inserting appropriate variable names instead of ellipsis).

```
`rm(imputed_posture_means1, ... , imputed_posture_means5)

rm(imputed_user_means0, ... , imputed_user_means14)

rm(posture)
rm(user)
rm(df)
rm(df_imputed_means)`
```

2 Clustering with k-means

Now that we have imputed the missing values, we can investigate our hypotheses by examining how well the data clusters by user and by posture. In the following problem, we will explore a wider choice of options for the number of centroids.

We will first use the k-means algorithm to carry out the clustering.

(a) Using the "kmeans" function in R, run kmeans on the features using 14 centroids (representing the 14 users). Do not run the algorithm on the entire dataset, as the eventual visualization can become unwieldy. Instead, obtain a random sample of 2,000 observations without replacement, and run the algorithm on the sampled values. Set a seed at '42' and set the 'nstart' parameter in the kmeans function to '46' to ensure that we can check your results. Hint: You can take a random sample of the dataframe's indices using R's "sample" function:

```
# take random sample
```

```
set.seed(42)

samp <- sample(x=1:length(imputed_user_target), size=2000, replace=F)

cluster_target <- imputed_user_target[samp]

cluster_features <- imputed_user_features[samp,]
```

(b) Use the “fviz-cluster” function to visualize the results of your clustering algorithm (you will probably want to press the “Zoom” button in the plots section of R Studio so that you can see the results on a larger plot). How much of the variance in the data is explained by the first two principal components? Does it look like the data separate into 14 distinct clusters?

(c) Compare the results from your clustering algorithm to the actual users. Specifically, make a bar plot showing the assigned cluster from kmeans against the actual user of the observation. Have the area of each bar correspond to the the percentage of observations that belong to a given user. Based on this graph, does it look like the data clusters well by user?

Hint: You will probably want to make use the “geom-bar” function in ggplot2 to do this.

(d) Repeat all of the above steps, but group by posture rather than by user. That is, run the kmeans algorithm with 5 centroids instead of 14. Construct the same plots and answer the same questions.

(e) What do the results of the bar plot clustered by posture suggest about the data? Why does this make sense in the context of what we know about the problem?

(f) Using all of the information gleaned from this problem, how do you recommend the missing data be imputed? Why?

3 Clustering Evaluation

In the previous problem, we used k-means with 5 and 14 centroids to decide how we should impute missing data. In this problem, we will investigate various ways of evaluating the quality of a clustering assignment.

(a) Use the elbow method to evaluate the best choice of the number of clusters, plotting the total within-cluster variation against the number of clusters for k-means clustering with k $\in$ (1, 2, ... 15).

(b) Use the average silhouette to evaluate the choice of the number of clusters for k-means clustering with k $\in$ (1, 2, ... 15). Plot the results.

(c) Use the gap statistic to evaluate the choice of the number of clusters for k-means clustering with k $\in$ (1, 2, ... 15). Plot the results. Be patient - this might take a few minutes.

(d) After analyzing the plots produced by all three of these measures, discuss the number of clusters that you feel is the best fit for this dataset. Defend your answer with evidence from the previous parts of this assignment, the three graphs produced here, and what you surmise about this dataset.

4 Other Clustering Algorithms

Up until now, we have used the k-means algorithm to cluster the data. In this problem, we will explore other methods used to create clusters.

(a) Hierarchical clustering: Implement agglomerative clustering (using Ward’s method) and divisive clustering. Plot the results of these algorithms using a dendrogram and interpret the results. Hint: Use the “agnes” and “diana” functions, respectively.

(b) Soft clustering: Run fuzzy clustering and a Gaussian mixture model on the scaled features. For the fuzzy clustering, run the algorithm with 5 and 14 clusters and plot the results using the “fviz-silhouette” function. For the Gaussian mixture model, the "Mclust" algorithm chooses the optimal number of clusters internally; report the number of clusters it selects. Also display the membership probabilities for the first 10 observations in your sample.

Hint: Use the "fanny" and "Mclust" functions, respectively. You might need to adjust the "memb.exp" parameter to something between 1 and 2 to get the function to run correctly. Make sure to include analysis for trial-and-error of fanny parameters. Justify your results.

(c) (AC 209b students only) Density-based clustering: Apply DBSCAN to the data. Determine a reasonable combination of ϵ, the radius of the neighborhood around an observation, and the number of nearest neighbors within the ϵ-neighborhood to be considered a core point. You should construct a knee plot to determine the choice of ϵ. Summarize the results using a principal components plot, and comment on the clusters and outliers identified. How does the clustering produced by DBSCAN compare to the previous methods? Read Section 2.2 of the R vignette on DBSCAN

https://cran.r-project.org/web/packages/dbscan/vignettes/dbscan.pdf to learn about the OPTICS algorithm.

1. Describe the difference in goal between the DBSCAN and OPTICS algorithm. You may need to refer to the references cited within.

2. Run the OPTICS algorithm on the data within the dbscan package. Choose (and justify) an appropriate value of ϵ and the minimum number of points in the ϵ-neighborhood. Interpret the results of the clustering.

Hint: Make sure to also use and plot `extractXi()` with parameter `xi=0.5` to properly visualize your results. See documentation suggested above.