Grouping galaxy based on photometric data

Ruiyang Gan, Steve Han, Mingwei Huang, Soyoung Lee, Yifan Leng

10/17/2018

Introduction

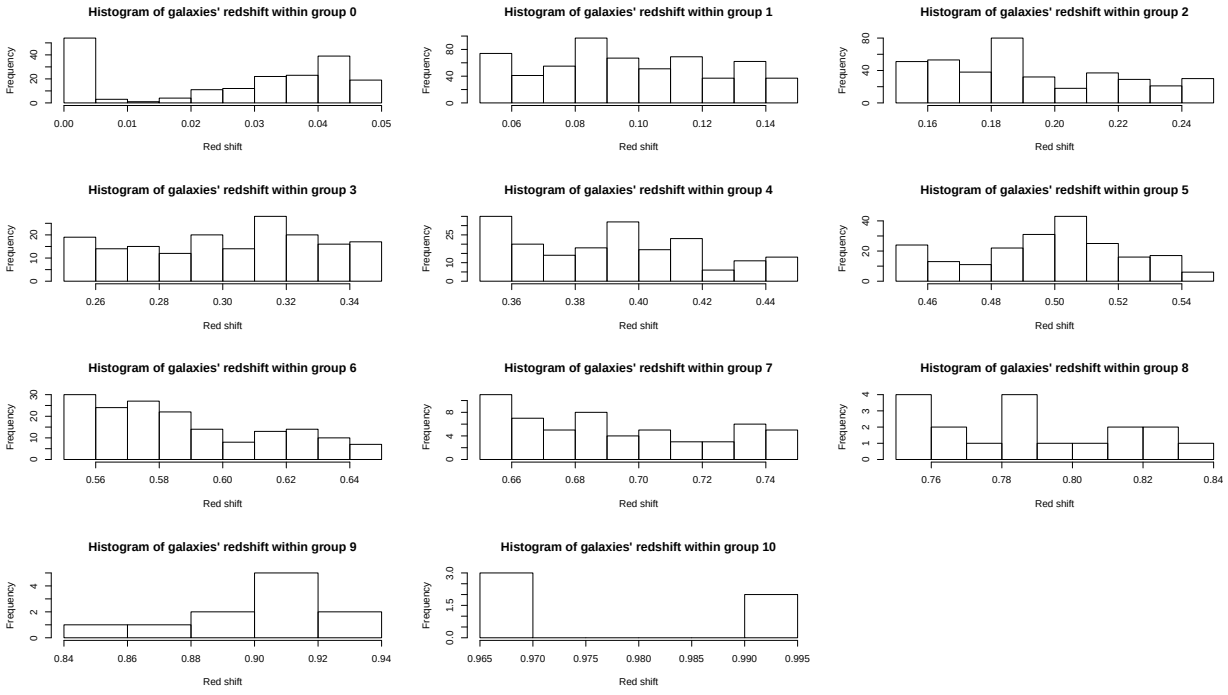
This report focuses on classifying galaxies with similar redshift into small groups with group size ≤ 10 but ≥ 2 (sosie pairs). To reach the goal, we will use a tree-like model: First we classfy galaxies into different groups based on their redshifts (as sosie pairs should not be too far away from each other). Next, we will use a density based, nearest-neighbor search based on normalized color magnitudes to search for sosie pairs within the groups that are classified to have similar redshifts. (*Note* : The notion of *density* is just describing how densely the points (galaxies) are clustered together, not to be confused with notion of *probability density* in probability and statistics).

One dimensional grouping of galaxies based on redshift (a crude measure of distance)

To group the galaxies by the redshift, we use the kernel function with bandwith `h` defined in the following:

$$k(z, z_i^*) = \{ \begin{array}{ll} 0 & |z - z^*| > h \\ i & |z - z^*| \leq h \end{array}$$

The kernel function allows us to label the galaxies by the center they are closest to. The centers are $z_i^* = [0.1, 0.2, ..., 1.3]$ since the redshift ranges from 0 to 0.995. The center are indexed with $i = 1, 2, ..., 5$.



Finding galaxy pairs with similar redshifts based on color magnitudes

After we obtain the preliminary grouping of galaxies based on their redshifts, we will find pairs within these groups. To accomplish the goal, we will use DBSCAN (a nearest-neghbor-searching approach) to find galaxy pairs based on magnitude of color that has been normalized with respect to its composite color magnitude of u.

Before we dive into the actual clustering algorithm, we will explain the features we use to cluster the galaxies. Usually, flux emiited by galaxies varies in both magnitude/intensity and the pattern of light they are emitting. Visually, magnitude/intensity represents how bright these lights are, while the pattern of light directly links to the color of the galaxy. Now, imagine that we have two "identical galaxy" (emitting same color with same level of brightness), but one is slightly further away from us. Therefore, the brightness/magnitude level in each color band will be different, but the magnitude ratio will be the same (as they are emitting same kind of light). Suppose we directly use the raw color magnitude to classify the galaxies, we will lose the pair. But if we normalize the magnitude in each color band with respect to magnitude of one color band, the two galaxies will have the same normalized color magnitude. For later experiement, we will use the composite color magnitude of g,r,i,z normalized with respect to the composite color magnitude of u.

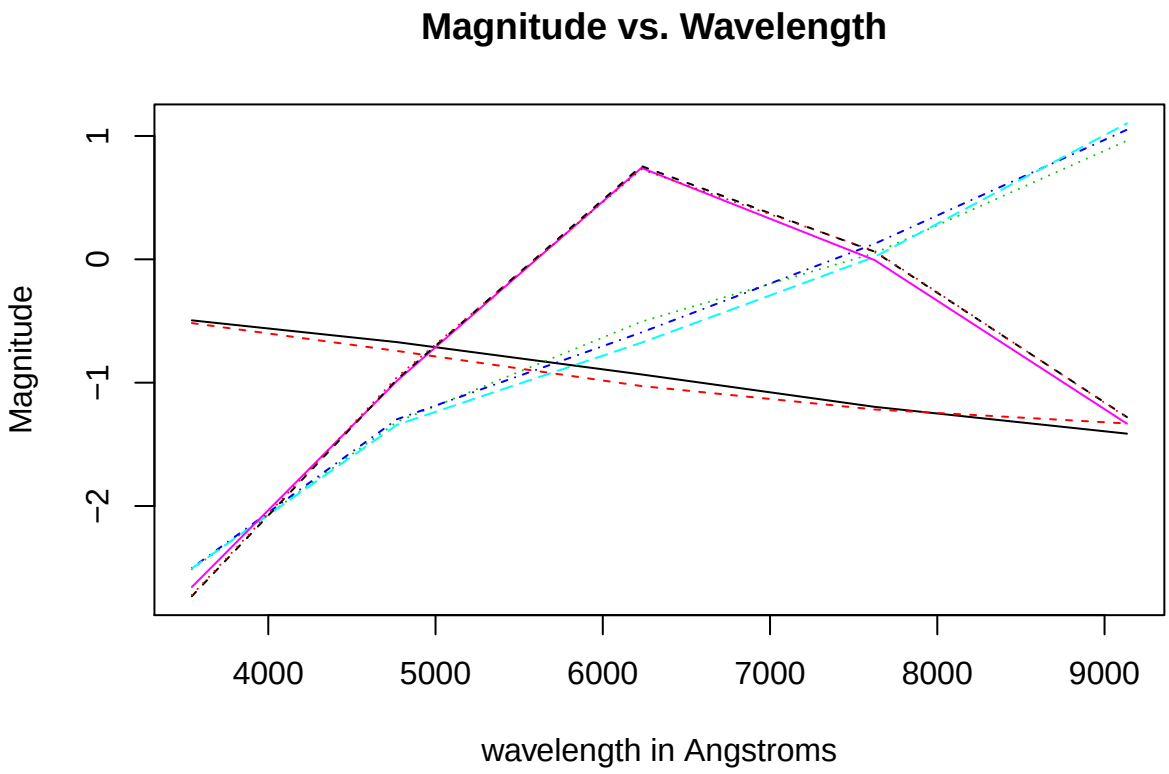
The principle of DBSCAN is a clustering method that is analagous to finding a "closest-pair" for a galaxy. THe concept builds around the notion of *core points* and *radius* ϵ. A point x is said to be a *core point* if it has at least *minPts* points (including x itself) within its neighborhood $B(x, \epsilon)$. Points x and y are said to be *connected* if $d(x, y) \leq 2\epsilon$, i.e. B(x,ϵ) $\cap$ B(y,ϵ) $\neq \emptyset$. If a point is not connected to any core point, then this point is regarded as a point, a point not in any "meaningful" cluster.

To find pair of points (or small group of points), we will need to let minPts, minimal number of points in core point's neighborhood, be 2. (*Note*: minPts is not equivalent to the number of points found in a cluster, but it sets lower bound for the number of points in a cluster). At the same time, we will let the radius of the neighborhood, ϵ, also be sufficiently small, as we need sosies to be nearly identical to each other.

Now, we will need to specify the specific value of the two parameters eps and minPts. For minPts, we can set it to 2, as the lowest number of points in a sosie pair has to be 2. Since we have normalized and scaled our composite magnitude of color band g,r,i,z, we can choose a fixed radius for our eps here. Due to the low complextiy of DBSCAN, an empirical approach here would be running the clustering algorithm multiple times and finding the appropriate eps. (A better approach here might be using hdbscan, which only requires setting minPts)

Another important setting for DBSCAN is to determine what metric $d(\cdot)$ we use to define ϵ and its induced neighborhood. The justification of using a euclidean metric is very tricky. In the original paper of DBSCAN, they use euclidean metric as their measure of distance. Since they only discuss clustering in a 2-dimensional space, using euclidean metric is a natural choice. However, we are clustering small groups based on more than 3 features (4 normalized color magnitude and AB ratio of each color bands in full model), using Euclidean metric may not be a very good choice here. However, using a euclidean metric in our small scale experiment (where we only use composite color magnitude normalized with respect to composite manitude of color band u).

Apply dbscan clustering on our simulated data (prelabled because we created them)



We simulated three Sosie groups, and each sosie group has three galaxies. We plot the galaxies' simulated magnitude against wavelength, and the Sosie galaxies should have the same shape but different shift.

```
##
##     1 2 3
##   1 2 0 0
##   2 0 3 0
##   3 0 0 3
```

We then run DBSCAN clustering algarithms on our simulated data, and the galaxies are perfectly clustered. The result in shown in the table above.

A small scale Experiment

Sosie query used in the data set

```
SELECT TOP 2000
   p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z,
   p.cModelMag_u as c_u, p.cModelMag_g as c_g, cModelMag_r as c_r,
   p.cModelMag_i as c_i, p.cModelMag_z as c_z,
   p.deVAB_u, p.deVAB_r, p.deVAB_i, p.deVAB_z,
```

```
   p.deVAB_g,
   p.expAB_u, p.expAB_g, p.expAB_r, p.expAB_i,
   p.expAB_z,
   s.specobjid, s.class, s.z as redshift
FROM PhotoObj AS p
   JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE
   s.z BETWEEN 0 AND 10
   and s.class = "GALAXY"
```

We use the following features from the `PhotoObj` and `SpecObj` in SDSS databse to cluster galaxies with similar colors and distance. The features we used for cluster on distance and color magnitudes are defined bwlow:

`z`: Best redshift when excluding QSO fit in BOSS spectra (right redshift to use for galaxy targets). It is our crude estimate of a galaxy's distance from earth.

The features we used to clutser galaxies based on its emitted color bands are defined in the following:

`cModelMag_u`: Composite Model Magnitude of ultravolet defined as

$$F_{composite} = \text{fracDeV} * F_{deV} + (1 - \text{fracDev}) * F_{exp}$$

where The coefficient (clipped between zero and one) of the de Vaucouleurs term is stored in the quantity fracDeV. FdeV and Fexp are the de Vaucouleurs and exponential fluxes (not magnitudes) of the object in question.

`cModelMag_g`: Composite Model Magnitude of green band.

`cModelMag_r`: Composite Model Magnitude of red band.

`cModelMag_i`: Composite Model Magnitude of near infrared band.

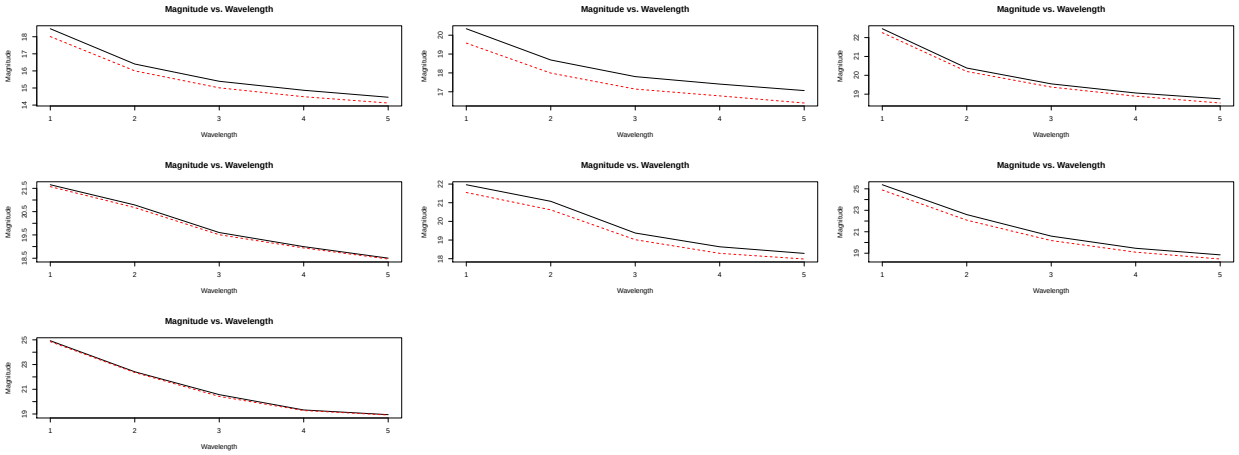
`cModelMag_z`: Composite Model Magnitude of infrared band.

The feature we will use in the next step to futher clutser similar-distance-and-color galaxies with similiar shapes is defined below:

`deVAB_u`, `deVAB_g`, `deVAB_r`, `deVAB_i`, `deVAB_z`: elliptical version of de Vaucouleurs defined on the AB axis ratio in five color band.

`expAB_u`, `expAB_g`, `expAB_r`, `expAB_i`, `expAB_z`: the exponential model defined on the AB axis ratio in five color band.

Now that we have found the clustering based on the redshift and normalized (w.r.t magnitude of u) and magnitude, we will see if these groups have similar pattern in color magnitudes according to barplots of color magnitude.



From the above plot showing the magnitude versus the wavelength for each sosie pairs, we notice that the clustering algorithm has found pairs of galaxies that has similar magnitude function, except they have differnet intensity of the light (in the plot, the two curve has similar shape but different intercept). Notice that in some plot, we have very many lines overlaid in the same plot but with significantly different magnitude pattern. It doesn't mean that they are sosie pairs, they are just classified as noise by DBSCAN.

Question and idea for next step

For now, we only perform pairs finding on a small data set using only redshift and normalized composite color magnitude. To proceed from here, we can add in the AB ratio of each color band in our feature space and perform clustering. AB ratio provides us with information about shape of the galaxy, as it captures shape of an elliptical galaxy. We can further filter by color band and calculate AB ratio for each color band. However, adding AB ratio will also increase the dimension in our feature space. Then the use of euclidean metric will demise the clustering algorithm due to the infamous curse of dimensionality. Thus, we can try different metric in our clustering algorithm, such as using infinite-norm as our metric or use a weighted metric. Another solution will be using a hierarchial classification model.

Appendix: Computer code :

Simulations:

```
# Read our simulated data
sim_data = read.csv("sim_data.csv")


cluster_dbscan <- function(data, eps_val){

  data.cMag <- subset(data, select=c("c_u","c_g","c_r","c_i","c_z"))
    # Normalize magnitude with respect to u's magnitude
```

```
  data.cMag <- data.cMag[,2:5]/data.cMag[,1]

    # Scale the normalized magnitude to have unit variance
    data.cMag <- scale(data.cMag)

    # Run DBSCAN to find the sosie pairs within groups classified by the
    # magnitude of the four color bands
    # The minPts (number of points required for a point to become a core point)
    # is set to 2 since we are looking for pairs
    rslt <- dbscan(data.cMag, minPts = 2, eps = eps_val)

    return(rslt$cluster)
```

Rolling window:

```
kernel = function(x, h) {
    centers = seq(from=0.1, to=1.3, by=0.1)
    for (i in 1:length(centers)) {
        if (abs(x-centers[i]) < h){
            return(i)
        }
    }
    return(0)
}

redshiftGroup = unlist(lapply(dat_small$redshift, kernel, h=0.05))

# dat_small.redShift <- subset(dat_small, select=c("redshift"))
# # Fit Gaussian Mixture model on the redshift using BIC
# # to select the optimal number of groups for gaussian mixture
# BIC <- mclustBIC(dat_small.redShift)
# plot(BIC)
# mod1 <- Mclust(dat_small.redShift, x = BIC)
# redshiftGroup <- mod1$classification
total.redShiftGroups <- length(table(redshiftGroup))
```

DBSCAN:

```
cluster_dbscan <- function(data, eps_val){

  data.cMag <- subset(data, select=c("c_u","c_g","c_r","c_i","c_z"))
    # Normalize magnitude with respect to u's magnitude

  data.cMag <- data.cMag[,2:5]/data.cMag[,1]

    # Scale the normalized magnitude to have unit variance
    data.cMag <- scale(data.cMag)

    # Run DBSCAN to find the sosie pairs within groups classified by the
```

```
    # magnitude of the four color bands
    # The minPts (number of points required for a point to become a core point)
    # is set to 2 since we are looking for pairs
    rslt <- dbscan(data.cMag, minPts = 2, eps = eps_val)

    return(rslt$cluster)
}


# Extract the magnitudes of color bands u,g,r,i,z
# we will use to classify galaxies into pairs
dat_small.cMag <- subset(dat_small, select=c("c_u","c_g","c_r","c_i","c_z"))

cMagGroup <- sapply(sort(unique(dat_small$redShiftLabel)), function(d) {
    # Select the group with same redshift labelling
    dat_small.cMag.d <- dat_small.cMag[redshiftGroup == d,]
    # Normalize magnitude with respect to u's magnitude
    dat_small.cMag.d <- dat_small.cMag.d[,2:5]/dat_small.cMag.d[,1]

    # Scale the normalized magnitude to have unit variance
    dat_small.cMag.d <- scale(dat_small.cMag.d)

    # Run DBSCAN to find the sosie pairs within groups classified by the
    # magnitude of the four color bands
    # The minPts (number of points required for a point to become a core point)
    # is set to 2 since we are looking for pairs
    # Since the color magnitude has been scaled to unit variance, therefore I will
    # use .05 as the radius.
    # A better approach here would be using a hierachial model without using a fixed
    # radius. Scaled magnitude to have unit variance allow us to choose the
    # However, for large datasets, such hierarchial model will take up large
    # memory space and become computationally difficult.
    rslt.d <- dbscan(dat_small.cMag.d, minPts = 2, eps = .05)
    return(rslt.d$cluster)
    })

# Attach the magnitude grouping to the original data.frame
dat_small <- data.frame(dat_small, "cMagLabel" = numeric(nrow(dat_small)))
i = 1
for(d in sort(unique(dat_small$redShiftLabel))) {
    dat_small$cMagLabel[dat_small$redShiftLabel == d] <- cMagGroup[[i]]
    i = i+1
}

# Within groups with similar redshift, we can randomly select groups with the
# same color magnitude labeling and plot their color magnitude of u,g,r,i,z
par(mfrow = c(ceiling(total.redShiftGroups/3), 3))
i = 0
for(d in sort(unique(dat_small$redShiftLabel))){
    # Randomly select a pair with the same cMag label within the group
    # with same redshift label, ignore noise group
    i = i+1
    # not displaying the reshift group with only noise labels
```

```
    if (length(unique(cMagGroup[[i]])) > 1) {
        # level 0 means noise.
        # We only display galaxies that are clustered with others (sosie)
        cMagLabel = unique(cMagGroup[[i]])
        sosies = cMagLabel[which(cMagLabel != 0)]
        cMag.level.random <- sample(sosies,1)
        pairs.row <- which(
            (dat_small$redShiftLabel==d) & (dat_small$cMagLabel == cMag.level.random))
        sosie <- subset(dat_small[pairs.row,], select=c("c_u","c_g","c_r","c_i","c_z"))
        sosie <- as.matrix(sosie)
        matplot(t(sosie), type="l",
                        main = "Magnitude vs. Wavelength",
                        xlab = "Wavelength", ylab = "Magnitude")
    }
}
```