HighDimensionCluster

Jinchang Fan

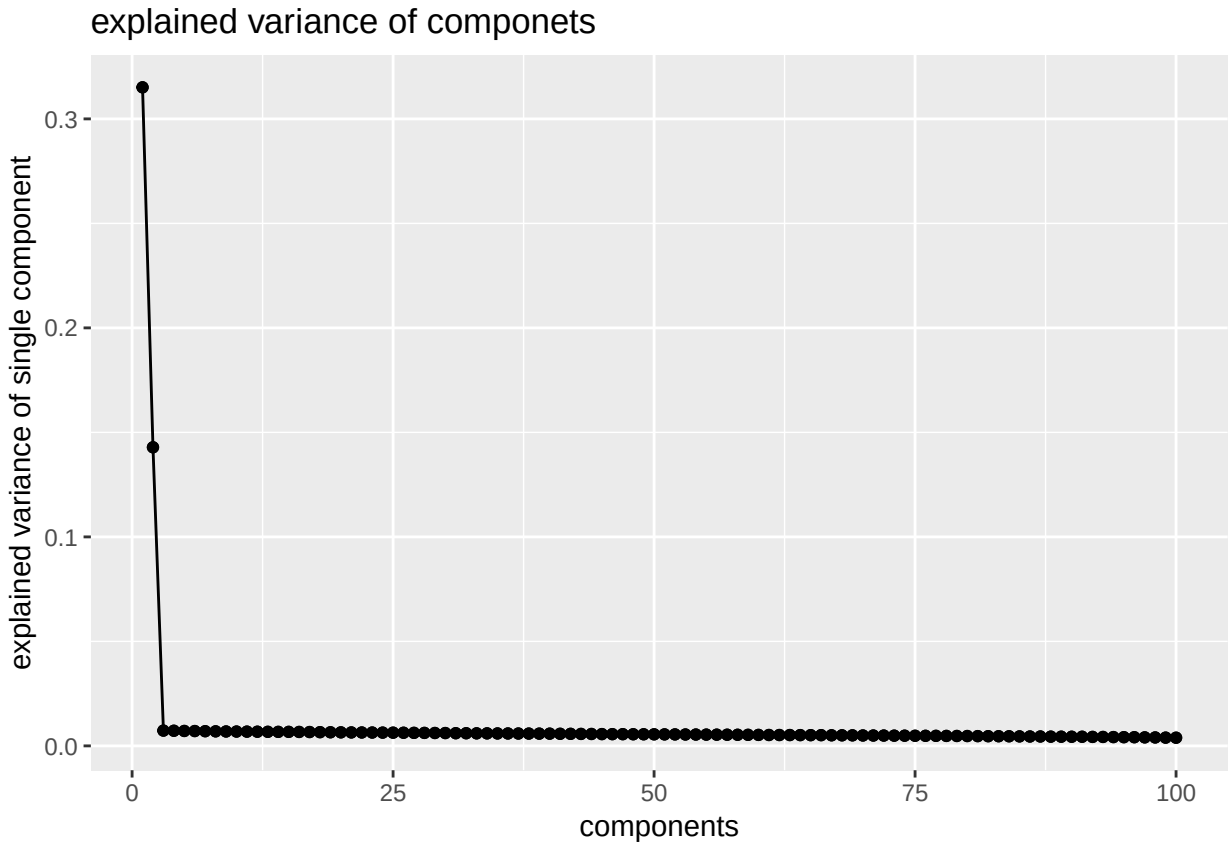
3/20/2019

1. Simulate high-dimensional data (p=1000) with three groups of observations where the number of observations is n=100 (R code to generate data is here).

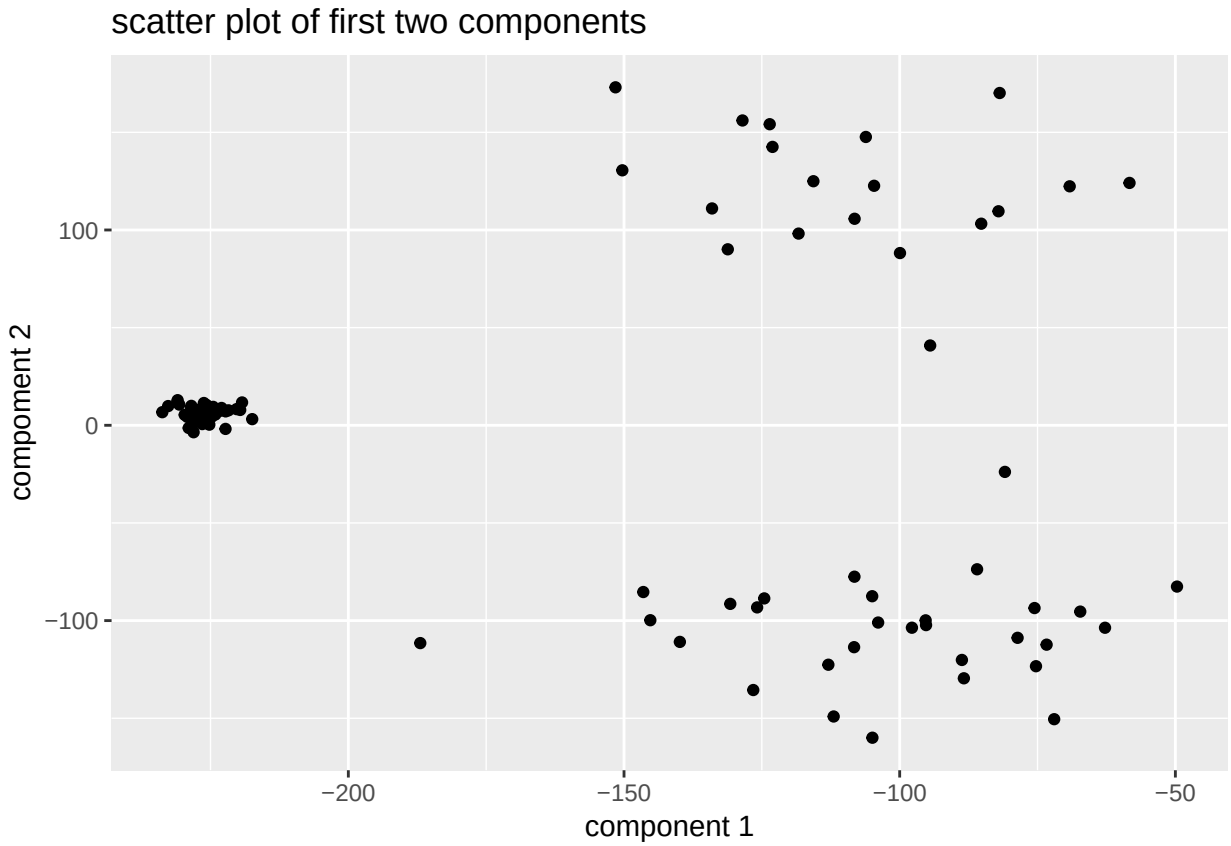
```
set.seed(2019)
n_rows = 1000
n_cols = 100
n_genes = 1000
n_cells = 100

k=3
x_mus = c(0,5,5)
x_sds = c(1,0.1,1)
y_mus = c(5,5,0)
y_sds = c(1,0.1,1)
prop1 = c(0.3,0.5,0.2)

comp1 <- sample(seq_len(k), prob=prop1, size=n_cols, replace=TRUE)
samples1 <- cbind(rnorm(n=n_cols, mean=x_mus[comp1],sd=x_sds[comp1]),
                  rnorm(n=n_cols, mean=y_mus[comp1],sd=y_sds[comp1]))

proj <- matrix(rnorm(n_rows* n_cols), nrow=n_rows, ncol=2)
A1 <- samples1 %*% t(proj)
A1 <- A1 + rnorm(n_genes*n_cells)
```

2. Perform k-means to identify the number of clusters in the data.

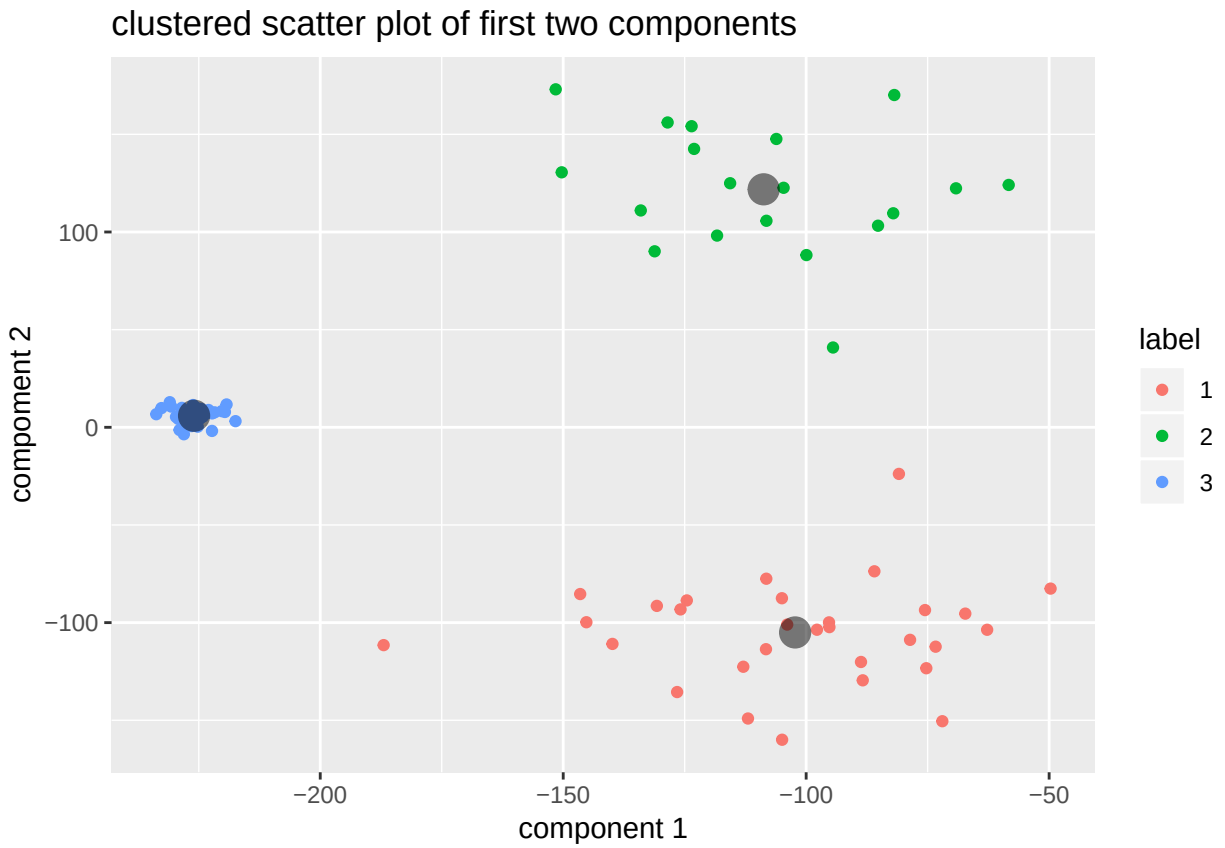
```
time0 <- Sys.time()
mysvd <- svd(A1)
A1_pc <- A1 %*% mysvd$v[,1:2]
km_pca <- kmeans(A1_pc,3)
time_pca <- Sys.time() - time0

center <- data.frame(pc1 = km_pca$centers[,1], pc2 = km_pca$centers[,2], label = as.factor(1:3))

qplot(x = c(1:100), y= mysvd$d/sum(mysvd$d) )+geom_line()+geom_point()+
  labs(title='explained variance of componets', x='components',y='explained variance of single componen
```




```
qplot(A1_pc[,1],A1_pc[,2])+labs(title ='scatter plot of first two components',x='component 1', y='compo
```




```
data.frame(pc1 = A1_pc[,1], pc2 = A1_pc[,2], label = as.factor(km_pca$cluster)) %>%
  ggplot( aes(pc1,pc2, color = label))+geom_point()+
  annotate('point', x = km_pca$centers[,1], y = km_pca$centers[,2], size = 5, alpha = 0.5)+
  labs(title ='clustered scatter plot of first two components',x='component 1', y='compoment 2')
```




```
time0 <- Sys.time()
km <- kmeans(A1, 3)
time_km <- Sys.time() - time0
```

3. To assess the accuracy, calculate the adjusted rand index and then calculate the within clusters sum of squares.

```
#given data and their cluster label, calculate within clusters sum of squares(ss), between cluster ss a
#input: X matrix n*p; label seq n
#output:
SSE_cluster <- function(X,label){
  X_c <- apply(X, 2, function(x) x-mean(x))
  SST <- sum(diag(t(X_c) %*% X_c))
  SSW <- 0
  for(l in unique(label)){
    X_temp <- X[which(label==l),]
    X_tc <- apply(X_temp, 2, function(x) x-mean(x))
    SSW <- SSW + sum(diag(t(X_tc) %*% X_tc))
    cat(l, ' ', sum(label==l), ' ', sum(diag(t(X_tc) %*% X_tc)),'\n')
  }
  return(list(SST = SST, SSW = SSW, ratio = SSW/SST))
}

SSE_cluster(A1, km_pca$cluster)
```

```
## 1   30   74670.85
## 3   51   51517.96
## 2   19   48537.4
```

```
## $SST
## [1] 1155191
##
## $SSW
## [1] 174726.2
##
## $ratio
## [1] 0.1512531
```

```
SSE_cluster(A1, km$cluster)
```

```
## 3   22   45563.42
## 2   70   476963.9
## 1   8   11874.74
```

```
## $SST
## [1] 1155191
##
## $SSW
## [1] 534402.1
##
## $ratio
## [1] 0.4626094
```

```
adj.rand.index(comp1, km_pca$cluster)
```

```
## [1] 1
```

```
adj.rand.index(comp1, km$cluster)
```

```
## [1] 0.546132
```

4. Record both metrics.

Then, create a data visualization summarizing the results.