

Lab #8: Unsupervised Learning

Biostat 216: Machine Learning in R for the Biomedical Sciences

Principal Components Analysis

1. Read in the breast cancer imaging data “ispy1doctored2.csv” into a data frame called `dat`
2. Perform principal components on just the variables "pixelVolT0", "pixelVolT1", "pixelVolT2", "pixelVolTfinal", "pixelVolPctChgT0_T1", "ftvPeT0", "ftvPeT1", "ftvPeT2", "ftvPeTfinal", "ftvPePctChgT0_T1", "age", "MRI_LD_T0", "MRI_LD_T1", "MRI_LD_T2", "MRI_LD_Tfinal".
Use the command `pr.out <- prcomp(~ pixelVolT0 + pixelVolT1 + pixelVolT2 + pixelVolTfinal + pixelVolPctChgT0_T1 + ftvPeT0 + ftvPeT1 + ftvPeT2 + ftvPeTfinal + ftvPePctChgT0_T1 + age + MRI_LD_T0 + MRI_LD_T1 + MRI_LD_T2 + MRI_LD_Tfinal, data = dat, scale=TRUE)`.
The `scale=TRUE` argument makes all the variables be normalized before performing PCA.
3. Look at the `center` and `scale` components of `pr.out`. These correspond to the means and standard deviations of the variables that were used for scaling prior to implementing PCA. E.g. `pr.out$center`
4. What is the dimension of the part of the data frame that you performed PCA on? Hint: you can find it as an object in `pr.out` – check `?pr.out`
5. Now look at the `rotation` matrix component of `pr.out` which gives the principal component loading vector. How many PCs are there? Does that agree with what you would expect?
6. Plot the first two principal components with `biplot(pr.out, scale=0)`. The `scale=0` argument ensures that arrows are scaled to represent the loadings.
7. The red arrows represent the loadings and their contributions to each PC. Why do you think there are many arrows that are almost overlapping?
8. Look at the `sdev` component of `pr.out`. This gives the standard deviation of each PC.
9. Generate a vector `pr.var` that is the variance explained by each PC by squaring the standard deviations.
10. Generate a new vector `pve` that is the proportion of variance explained by each vector. Generate the sum of `pve` as a confirmation that you are on the right lines.
11. Generate plots of proportion of variance explained and cumulative variance explained against PC number. You will need the function `cumsum` which computes the cumulative sum of elements in a numeric vector, e.g. for `a1 <- c(2,1,3,4)`, `cumsum(a1)` gives `[1] 2 3 6 10`
12. How many PCs do you need to explain at least 80% of the variance?

K-means clustering

13. Subset `dat` to only the columns "pixelVolTfinal", "ftvPeTfinal", "age", "MRI_LD_Tfinal". Name the new object `datsub`
14. Perform K -means clustering on the subset of variables for $K = 3$. Use 30 random starting configurations.
15. Output the cluster assignments of the 3 clusters
16. Generate a table of agreement between the cluster labels and the outcome pathologic complete response `pCR`. Do you think the cluster allocations are illuminating with respect to `pCR`?

Hierarchical clustering

17. The `hclust()` function implements hierarchical clustering in R. We will plot the hierarchical clustering dendrogram using complete, single, and average linkage clustering, with Euclidean distance as the similarity measure. Start by creating the distance matrix between all pairs of observations using the `dist()` function: `distx <- dist(datsub)`.
18. Generate hierarchical clustering with `method="complete"`. Use the command `hc.complete <- hclust(distx, method="complete")`.
19. Generate similar objects for `average` and `single` linkage.
20. Plot each of the dendrograms using the `plot` function.
21. Determine the cluster labels for each observation associated with cutting the dendrogram with 4 clusters using the `cutree()` function.