Project 3: Identifying leaf clusters

Addison James, Bin Zhuo
Feifei Lei, Nandhita Narendra Babu

Overview

When talking about big data, one topic that can never be ignored is machine learning. Machine learning is generally defined as a set of methods that can detect patterns in data, predict future data, or give decision supports under uncertainty. These properties of machine learning can also be categorized as: classification, clustering and prediction.

In this particular project, we were assigned a question to identify the leaf clusters. The leaf data set contains a column identifying the species that can be ignored during clustering and then used to verify the results. There were 40 species numbers, but only 30 of these species were present in the leaf data set. We used two clustering methods. k-Means clustering analysis was used to make 30 clusters to see how well this approach recovers the real data; model-based clustering was used to group the leaves at higher level. In the following report, we will talk about the machine learning methods we applied, summary the findings of our project and discuss the assumptions and limitations of the methods we used, as well as the scalability.

Method Descriptions

In clustering analysis, "all methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality", stated by Jasonb in the article "A Tour of Machine Learning Algorithms".

k-Means clustering requires the analyst to specify the number of clusters to extract, then aims to partition *n* observations into *k* clusters. The principle is to make the observation belongs to the cluster with the nearest mean. In our project, the function `kmeans` was employed to conduct the k-Means clustering analysis to separate leaves into species. First the `kmeans` function is used to cluster the data into $k = 30$ clusters, so that it could potentially cluster all the observations into 30 separate species based on the 14 covariates available. The covariates used were: eccentricity, aspect ratio, elongation, solidity, stochastic convexity, isoperimetric factor, maximal indentation depth, lobedness, average intensity, average contrast, smoothness, third_moment, uniformity, and entropy.

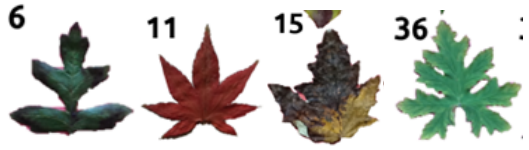
Another clustering methods we applied in project is model-based clustering. We used mclust software to implement higher level clustering. It is a model-based clustering, whose covariance parameterization and number of clusters are selected via BIC. The clustering is done under normal mixture modeling via EM algorithm. We used the `Mclust` function, which selects the optimal model according to BIC for EM initialized by Guassian Mixture Models, to perform the clustering analysis. `Mclust` function chooses the model and number of clusters with the largest BIC. The input to `Mclust` includes the components and the covariance structures. The object produced by `Mclust` is a list with components describing the estimated model.

Findings

For k-Means method, verifying that the observations are clustered into the species may sound simple, but it is actually challenging, even when the species is already known. Two possible solutions are presented and compared below. The first examines each cluster and assigns a species number to each cluster based on the most frequent species in the cluster. This can fail to create a one-to-one relationship between the species and cluster because there may be a tie or two different clusters may be assigned to the same species. The other solution is to examine each species and assign the species to the clusters, the same way as previously described, but reversing the role of clusters and species.

The results from the first method (most frequent species in each cluster) show that 138 of the 340 observations (41%) ended up in the "correct" cluster. This method of evaluations successfully assigned 19 of the 30 clusters (63%) to a species. The second method (most frequent cluster in each species) gave a less optimistic evaluation showing that 126 observations were "correctly" clustered (37%) and 17 of the 30 species were assigned a cluster (57%). These results shows this clustering algorithm does a poor job of grouping leaves of the same species based on the known covariates.

For model-based clustering, after scaling the original data, we reached an optimal cluster number of 6. An example of the outcome is as follows:



Cluster	Species (percentage of observation falling in that cluster)
1	3 (60%), 9(64%), 13(85%), 30(50%), 33(64%)
2	1(92%), 2(90%), 4(62%), 7(60%), 23(64%), 24 (69%), 27(73%), 29(100%), 32(55%), 26(33%)
3	10(92%), 25(100%), 28(50%)
4	5(92%), 12(92%), 14(67%), 22(58%), 35(55%)
5	6(100%), 11(100%), 15(100%), 36(100%)
6	8(100%), 31(55%), 34(100%)

Discussions

When considering the capability of large datasets, both `kmeans` and `Mclust` can well handle the problem. There's no specific statement in R indicates that `kmeans` is suitable to the large datasets. But the algorithm used in this function is the algorithm of Hartigan and Wong (1979), which performs better than other k-Means algorithms. And we know that k-Means is faster than hierarchical clustering. For function `Mclust`, there's no doubt that this function is able to cope with large datasets very well. In the article "mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation" (2012), the authors pointed out that `Mclust` includes a provision for using a sampled data in the hierarchical phase before applying EM to the whole dataset, which allows to extend the method to larger datasets.

Though these methods are easy handling and capable to larger datasets, there are still some limitations. k-Means clustering tends to cluster the observations into comparable spatial extent, while some practical problems may require different cluster shape. In this case, k-Means clustering will lose points on accuracy. `Mclust` provides only 10 of the 14 possible variance-covariance structures based on the eigenvalue decomposition. Additionally, function package mclust has no direct provision to handle missing values.