Learning from Imbalanced Datasets

February 7, 2022

1 Description

Imbalanced datasets are those in which the classes are not represented equally, which results in classifiers disregarding the minority class. Typical ways of dealing with this issue are resampling the dataset (either upsampling the minority class/es or downsampling the majority class/es. In this project, you will test a new approach to dealing with imbalanced datasets, based on a mixture of supervised and unsupervised learning.

2 Tasks

1. Choose 3 datasets from UCI, Kaggle, etc. You will make them imbalanced, so make sure they are *not* imbalanced from the beginning. Small imbalances (e.g., 55% of one class) are fine. Load, inspect, and clean the datasets. For each of them, create three versions/surrogates (in addition to the original one) by subsampling one of the classes:
   (a) Low imbalance (65%)
   (b) Medium imbalance (75%)
   (c) High imbalance (90%)
2. To establish a baseline, perform stratified cross-validation on each of the datasets and their surrogates and train a random forest. Report baseline results using appropriate metrics.
3. Create 10 stratified folds (to ensure the imbalance ratio remains the same in each fold) for each of the datasets.
4. Using the data of 9 of these folds:
   - Using the Elbow method and the Silhouette method, identify the number of clusters in the dataset. There should be some level of agreement between these indices (or at least you should be able to identify lower and upper bounds).
   - Run k-means in the data set using the identified number of clusters. Select as final clustering that with the lowest output criteria.
   - For each cluster, identify its centroid and the number of samples of the minority class in that cluster (as per their labels). Save this information.
   - Train a random forest for each of the clusters that contains samples from more than one class (i.e., if a cluster only has samples for one of the classes, you don't need to train a classifier).
   - Given a sample x_i from the unseen fold (the one left out in (3))
     – Assign x_i to its closest cluster.
     – If this cluster has only instances of one class, assign to x_i that label. Otherwise, use the model trained with data from that cluster to assign a label to x_i.
5. Do the above for each permutation of 10 bins (like in cross-validation), and present the average and standard deviation of results for each of the datasets and their surrogates using *appropriate metric/s*.
6. Compare your results with the baseline results from (2). A boxplot of the cross-validation results for each method should help you decide which method is best under which conditions. Are the results significantly better with the new method (e.g., as determined by a permutation test)? How does the data imbalance affect the results?

3 Links

- UCI: https://archive.ics.uci.edu/ml/datasets.php
- Elbow method: https://blog.cambridgespark.com/how-to-determine-the-optimal-number-of-clusters-for-k-mean
- Silhouette method example: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html