JMLR: Workshop and Conference Proceedings 1:1–4, 2016 Submitted 9/25/2016; Published 9/25/2016

Introduction to Machine Learning - EN 605.449.81 - Lab 2

Max Henry MHENRY22@JHU.EDU

Abstract

In this lab, I implemented stepwise forward selection and genetic algorithm feature selection and tested their performance using a LDA-like objective function. I found that feature subsets returned from genetic algorithms provided much better results but their run-time and overhead was much higher than SFS.

Keywords: stepwise forward selection, genetic algorithm, k-means, HAC

1. Problem Statement

The purpose of the assignment was to implement stepwise forward feature selection and genetic algorithm feature selection. The feature subsets of each algorithm were tested using both k-means and HAC clustering and a LDA-like objective function. The number of classes for each data set was used as k in k-means and was used to cut the HAC tree at the correct level.

I expect feature selection using the genetic algorithm will produce a feature subset that outperforms the feature subsets returned by SFS for both k-means and HAC clustering.

2. Algorithms

2.1. Stepwise Forward Selection

Stepwise forward feature selection is method of dimensionality reduction that starts with an empty subset of features and incrementally adds a feature to the tested feature subset, grades its performance on the model and keeps it if is the best performer for that round of testing. This incremental approach is repeated until all the features are added or the performance of the new feature subset would be lower than the current best performance seen so far.

2.2. Genetic Algorithm Feature Selection

Applying a genetic algorithm to feature selection follows the same outline as a typical genetic algorithm process, namely, selection, crossover, mutation, model evaluation and replacement. The form of the population for this particular lab is a bit string with length equal to the number of total features of the non-reduced data instance where 1 means the feature should be included and used for training and 0 means the feature should be left out. During selection, two parents are selected proportion to the parents fitness. In the case of this experiment, fitness is the score the model achieved using equation 1. In crossover, two parents are combined to create two children feature subsets using a random position in

© 2016 M. Henry.

the parent strings. In mutation, each bit in the offspring is randomly flipped with a small probability. Using these offspring feature subsets, a new model is trained and evaluated using equation 1. Replacement happens when one adds the offspring into the population if their performance is better than the bottom-performing feature subsets.

2.3. K-means

K-means uses centroids, the mean value of each attribute over all the instances clustered together, to cluster unseen instances. In the case of this experiment, the number of centroids is equal to the number of classes one expects to see in the input data. The process begins with random centroids. In this experiment, centroids are initially created by randomly choosing a value between the minimum and maximum value for each attribute based on the input data. From there, the algorithm calculates the distance between each point in the inputted data and each centroid. A data instance is then assigned to the centroid it is closest to and the centroids are recalculated. In this experiment, the algorithm proceeds until the centroids stop updating by more than 0.001. At that point the centroids are returned.

2.4. HAC

Hierarchical agglomerative clustering is a simple way of clustering data instances together. One starts by putting each data instance in their own cluster. Then one calculates the Euclidean distance from each cluster to every other cluster and merges the two clusters that are closest together. This process repeats until a single cluster has been formed. This process produces a dendrogram that can be cut horizontally to retrieve the "active" clusters at a given point in the algorithm.

3. Experimental Approach

3.1. Data Cleaning

Missing data was filled in using the conditional probability of the values occurring given the underlying class example. In addition, class labels were removed during clustering.

3.2. Experiments

Three data sets (glass, iris, and spam) were run in 4 configurations: SFS based on k-means, SFS based on HAC, genetic algorithm based on k-means, and genetic algorithm based on HAC. The feature sets they returned and their score based on equation 1 is reported here.

4. Results

	K-means	HAC
SFS	[1,3], 11.44	[0], 5.68
GA	[0,1,2,3,4,5,6], 22.52	OBE

Table 1: Features returned from Glass dataset

	K-means	HAC
SFS	[1], 0.74	[0], 0.77
GA	[0,1], 0.31	[1,2], 0.69

Table 2: Features returned from Iris dataset

	K-means	HAC
SFS	[0], 0.012	OBE
GA	[2,6,8,9,16,18,19,23,24,25,26,29,30,34,39,40,41,43,44,47,49,50], 0.0009	OBE

Table 3: Features returned from Spam dataset

5. Conclusions

Seen above, the rigor of the genetic algorithm feature selection outperforms SFS only on the glass dataset. Any insight is muddied by the fact that full feature selection runs did not finish in time to appear in this report. These include only the HAC processes and is caused by its poor run-time. Although concluding too much from these results is dangerous, it is clear with the results I obtained that the overhead of GE feature selection is not always warranted and SFS can prove quality dimensionality reduction in less time. This is the exact opposite of what I expected to happened as I thought the thoroughness of GE would allow it to exploit the feature subset combinations SFS skip over.

6. Summary

The poor run-time of HAC variants hampered the reporting strength of this experiment but it appears SFS methods provide similar dimensionality reductions for less overhead.

7. References

Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.

Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). "14.3.12 Hierarchical clustering". The Elements of Statistical Learning (PDF) (2nd ed.). New York: Springer. pp. 520528. ISBN 0-387-84857-6. Retrieved 2009-10-20.

"Clustering - Hierarchical." Clustering - Hierarchical. N.p., n.d. Web. 25 Sept. 2016.

UCI machine learning repository: Glass identification data set. (1987, September 1). Retrieved September 25, 2016, from https://archive.ics.uci.edu/ml/datasets/Glass+Identification

UCI machine learning repository: Iris Data set. (1988, July 1). Retrieved September 25, 2016, from https://archive.ics.uci.edu/ml/datasets/Iris

UCI machine learning repository: Spambase Data set. (1990, July 1). Retrieved September 25, 2016, from https://archive.ics.uci.edu/ml/datasets/Spambase

Goldberg, David E., and John H. Holland. "Genetic algorithms and machine learning." Machine learning 3.2 (1988): 95-99.

John, George H., Ron Kohavi, and Karl Pfleger. "Irrelevant features and the subset selection problem." Machine learning: proceedings of the eleventh international conference. 1994.