Introduction to Machine Learning, Programming Assignment 2

Analyzing Feature Selection and Clustering Methods

Julie P. Garcia

Johns Hopkins University

Abstract

In this paper, we evaluate the Stepwise Forward Selection algorithm as a technique for performing Feature Selection. The SFS algorithm was used as wrapper for a Naïve Bayes Classifier to find the feature set that performs the best. The resulting reduced feature set was input into a K-means clustering algorithm that clustered the data based on the number of classes in the dataset. The labels produced by K-means clustering were then used to train the data with Naïve Bayes in order to evaluate K-means as a classifier. The Silhouette Coefficient was calculated to measure the performance of K-means a clustering algorithm. All algorithms were implemented in Python and run on three publicly available datasets that are commonly used for learning experiments.[3] In general, the results show that the K-means clustering algorithm works well for clustering data, however, it does work well as a classifier and should be used in conjunction with other classifying methods.

## Problem Statement

In this paper, we will analyze the performance of a feature selection algorithm Stepwise Forward Selection, wrapping a Naïve Bayes classifier. We will also test the K-means clustering algorithm as a method of clustering data and as a classifier.

### Algorithms Implemented

We first implemented Stepwise Forward Selection for feature selection and used it to wrap a Naïve Bayes classifier in order to test the performance of each proposed set of features. We then input the reduced feature set into the K-means clustering algorithm and clustered the data based on the number k classes. The K-means algorithm produced labels for each data set that were used as input to the Naïve Bayes algorithm again and performance was tested to see how well K-means classified the data. The Silhouette Coefficient was calculated to measure the performance of K-means as a clustering algorithm.

#### Stepwise Forward Selection

The Stepwise Forward Selection (SFS) algorithm is a feature selection technique in which you start with no features and keep adding them until thre is no performance improvement. If a feature causes performance to decline it is removed from the new feature set. Each feature is tested and performance is measured by a classifier algorithm. For this paper, we used Naïve Bayes as a classifier. The SFS algorithm acts as a wrapper to Naïve Bayes, and outputs an optimal set of features.[1] Steps to the algorithm are as follows:

1. Start with zero features in the new feature set.
2. Iterate through each feature, and perform the following:
    a. Add the feature to the test feature set

   b. Train the new feature set using a classifier (Naïve Bayes in this case)

   c. Test the performance of the classifier model on the test set.

   d. If this improves performance, add it to the new feature list

3. Return the new feature set

The pseudocode for the SFS Algorithm is shown here:

**Algorithm 10.1** Stepwise Forward Selection

```
function SFS(𝓕, 𝒟_train, 𝒟_valid, Learn())
    𝓕_0 ← ⟨⟩
    basePerf ← −∞
    repeat
        bestPerf ← −∞
        for all F ∈ 𝓕 do
            𝓕_0 ← 𝓕_0 + F
            h ← Learn(𝓕_0, 𝒟_train)
            currPerf ← Perf(h, 𝒟_valid)
            if currPerf > bestPerf then
                bestPerf ← currPerf
                bestF ← F
            end if
            𝓕_0 ← 𝓕_0 − F
        end for
        if bestPerf > basePerf then
            basePerf ← bestPerf
            𝓕 ← 𝓕 − bestF
            𝓕_0 ← 𝓕_0 + bestF
        else
            exit
        end if
    until 𝓕 ← ⟨⟩
    return 𝓕_0
```

**K-means Clustering**

The K-means clustering algorithm is an unsupervised learning technique for clustering data.[1] In this paper, we test it's performance as a clustering technique and also as a classifier. The algorithm was implemented as follows:

1. Randomly initialize the centers of each cluster by choosing them from the rows of samples

2. Loop the following steps until the centers converge
    a. Find the cluster center that is closest to current datapoint and label that row with the appropriate cluster
    b. Calculate the new centers by finding the mean of newly labeled clusters
    c. Break when the centers converge

The pseudocode for the K-means algorithm is shown here:

**Algorithm 10.3** *K*-Means Clustering

```
function KMEANS(D, k)
    initialize μ_1, ..., μ_k randomly
    repeat
        for all x_i ∈ D do
            c ← arg min_{μ_j} d(x_i, μ_j)          ▷ d() is the distance between x_i and μ_j
            assign x_i to the cluster c
        end for
        recalculate all μ_j based on new clusters
    until no change in μ_1, ..., μ_k
    return μ_1, ..., μ_k
end function
```

**Experimental Approach**

All algorithms were run on three datasets, Iris, Glass, and Spambase. For each of the three the datasets, data was first split into test and training set (1/3 and 2/3, respectively). Stepwise Forward Selection was performed on the training set, to reduce dimensionality. This algorithm was used as a wrapper for Naïve Bayes in order to test the performance of each combination of features. This new optimized feature set was input into the K-means clustering algorithm with k = the number of classes.

Once the dataset was clustered, these cluster labels were fed into the Naïve Bayes classifier and tested on the test data in order to see how K-means performed as a classifier. The Silhouette Coefficient of the clusters was used to test the performance of K-means as a clustering algorithm. It is calculated as follows:

1. For each data point in the set, calculate the average distance to all other objects in the cluster, call this *a*.
2. For each data point in the set, calculate the average distance to each point in the other clusters and take the minimum, call this *b*.
3. Calculate the silhouette coefficient:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

4. Evaluate how the k-means clustering algorithm did overall:

$$sil(C) = \frac{1}{|D|} \sum_{x_i \in D} s_i$$

**Results**

After running the algorithms on each dataset multiple times, the average best performance of the Stepwise Forward Selection method, was 70% for the Iris dataset, 63% for the Glass dataset and the Spambase dataset was 82%. These results show that with 56 features as opposed to the Iris datasets 4 and the Glass datasets 7, the Spambase dataset was better suited for Feature Selection by SFS.

The K-means algorithm performed well as a clustering algorithm, as the silhouette scores averaged .79 for the Iris dataset, .70 for Glass and .78 for the Spambase dataset. However, as a classifier K-means performed poorly, often at 100% error rate.

**Summary**

In summary, the Stepwise Forward Selection method for Feature Selection seems to work better on datasets with a large number of features. the K-means clustering algorithm works well for clustering data, however, it does not work well as a classifier and should be used in conjunction with other classifying methods.

References

1. Alpaydin, E. (2014). *Introduction to machine learning*. Cambridge, MA: The MIT Press.
2. Bursac, Z., Gauss, C. H., Williams, D. K., & Hosmer, D. W. (2008). Purposeful selection of variables in logistic regression. *Source Code for Biology and Medicine, 3*(1). doi:10.1186/1751-0473-3-17
3. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.