Genetic and Forward Selection Algorithms

Dec 2019

I. Part 1

A. Imbalance dataset

By checking the dataset, we can understand that the dataset is highly imbalance. To deal with this issue, I duplicate data points in minority class around 22 times to have a more balanced dataset. I do it on both train and test set. It also worth mentioning that this process is done in a random manner.

B. Classifier Selection

We design two algorithms for a cost sensitive classification problem .To do so, I have implemented both genetic and forward selection algorithms.

For selecting the classifier, I tried a couple of classifiers both on genetic and forward selection algorithm. But I finally chose decision tree. For choosing this classifier I have listed two reasons as follows:

- Because genetic algorithm takes time to find the best feature set, it is better to choose a simple classifier to have a more efficient algorithm.
- I tested other classifiers but I found out that they are not performing significantly different from decision tree.

.

C. Fitness function

I defined a joint fitness function which I use it in both forward selection and genetic algorithm.

$$fitness = acc - \frac{\frac{cost_i}{max(cost)}}{(acc+1)} + max(cost)$$

where $max(cost)$ is the maximum cost among feature costs which is (76.11) and $cost_i$ is the cost of feature i (while using forward selection algorithm).$cost_i$ is the cost of all features in our subset of features in genetic algorithm. acc is the accuracy of the classifier containing k of total $n$features($k <= n$).

D. Stopping condition

For stopping the model I used a couple of iterations and by plotting the fitness function in each iteration, we can find out that the fitness function doesn't change after a couple of iterations, and it reaches a saturation point so we stop in that point. I will discuss this part more in implementation details of each model.

II. Part 2

A. Forward selection algorithm

For this part, I implemented the forward selection algorithm. Forward selection is a greedy algorithm which tries to find the best feature at each step and add it to the empty set. I use the mentioned fitness function for feature evaluation. I also use the similar stopping criteria for this algorithm meaning that, when the fitness function does not change after adding a feature to our set, I stop at that point and return the set of features.

The main characteristic of this algorithm is that it is fast. So it is better for choosing features from a large set of features.

I have written the algorithm as follows :

Algorithm 1: Forward Selection

```
Result: Final feature set and their costs
finalSet = empty set; tempSet = empty set; initialization;
while fitness function does not changes more than a
 threshold do
    for features in feature set do
        if feature not in finalSet then
            tempSet = finalSet
            append new feature to tempSet;
            fit the model with the features in tempSet;
            calculate fitness;
            sort features in tempSet based on their
              fitness in decreasing order;
        else
    end
    add the first item in tempset to finalSet/;
end
```

As we see in the algorithm, at the end when the differences in fitness function is not more than a threshold(for example 0.1), I stop adding new features. However because it is a greedy algorithm, it may not give the best result.

B. Selected features

In this part I will write the selected features based on the algorithm. In the first iteration, 21th feature is added to the set because in the absence of 19th and 20th features, 21th feature is our best choice. Then one by one, features are added in the following order from left to right in the set.

$$finalFeatureSet = \{21, 17, 3, 10, 13\}$$

TABLE I

EXPERIMENT RESULTS FOR FORWARD SELECTION ALGORITHM ON TRAINING SET.

	c=1	c=2	c=3	total acc	cost
selected features	1.0	1.0	1.0	1.0	51.7
all features	1.0	1.0	1.0	1.0	

TABLE II

EXPERIMENT RESULTS FOR FORWARD SELECTION ALGORITHM ON TEST SET.

	c=1	c=2	c=3	total acc	cost
selected features	0.98	0.92	0.96	0.96	51.7
all features	0.90	0.99	0.99	0.98	

C. Experiment results

As shown in Table 1. the results show that with the selected features , we can get a high accuracy. However, because of its greedy way of solving the problem, it does not does not decrease the cost as much as possible and it does not give the best result possible. The fitness function at the end is equal to 72.79.

III. PART 3

A. Genetic Algorithm

As mentioned in the previous part, just like the forward selection algorithm I use the Decision Tree. Using a simple classifier is much more important for genetic algorithm because in terms of using complicated classifiers, it may take longer time to find the best subset.

After knowing the result of the decision tree on the test set, I implemented the genetic algorithm. For this task I used a population size equal to 10. I tried others but this one was enough to produce a good result.I use bitstring representation as mentioned in the question. Each bit represents a feature. I also reached my best result in 80 iterations. Number of iterations is selected based on the saturation of the fitness function at a point around 80 iterations.

As I mentioned before I am using population size equal to10. So after each iteration I save 60 percent of the population with higher fitness and for the other40 percent, I produce new off-springs with the help of mutation and crossover.

For crossover, I chose 20 percent of the surviving pack. In this part I choose the crossover rate equal to 0.8. For mutation I tried mutation rate equal to 0.3.I have also tried choosing random rate and it also gave a good result.The crossover rate and mutation rate are written in pseudocode too. For choosing the number of iterations, I kept track of the fitness function to check where it reaches a saturation. I chose the number of iterations around the saturation point. As we see in Figure 1 fitness function does not change that much after a certain number of iterations. In Figure 1 and Figure 2 the average fitness function in a population and the maximum fitness function is illustrated. .The algorithms are written in pseudocode format.

Algorithm 2: Genetic

Result: Final feature set and their costs
initialize the population with size 10;
Input: $features, cost$
Output: $accuracy, cost, populatioinstance$
`train(`F`)`
for *i from 1 to max iteration* **do**
    fitness();
    crossoverAndMutate();
choose the population instance with higher fitness;
return $accuracy, cost, populationinstance$
End Function

Algorithm 3: Calculate fitness

Result: Fitness
Input: $populationinstances$
Output: $fitnessvalue$
`fitness(`F`)`
for *i from 1 to populationSize* **do**
    fit the classifier;
    calculate fitness;
return $fitnessvalue$ **End Function**

Algorithm 4: Crossover and mutate

Result: new population
Input: $populationinstances$
Output: $newpopulation$
function `crossoverAndMutate()`**:**
    survived = chose 0.6 of population based on fitness
    $crossoverRate = 0.84$
    $mutationRate = 0.3$
    index = $crossoverRate * numberOfFeatures$
    offspring$_1$ = $survived[1][: index] + survived[2][index :]$
    offspring$_2$ = $survived[1][index :] + survived[2][: index]$
    index = $mutationRate * numberOfFeatures$;
    for *i from 1 to index* **do**
        do the mutation on selected third and fourth survived set.
    offspring$_3$ = $survived[3]$
    offspring$_4$ = $survived[4]$
    add offsprings to population list.
End Function

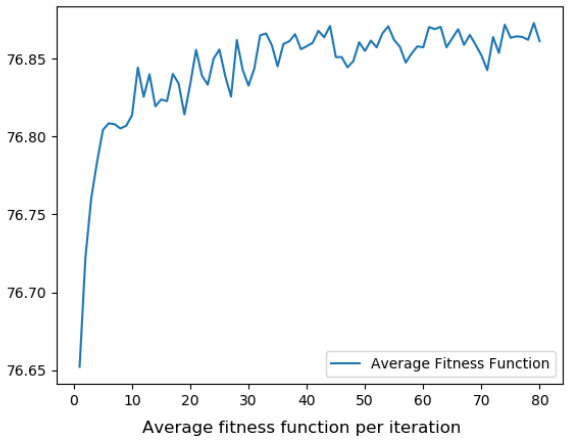

Fig. 1. Average Fitness function

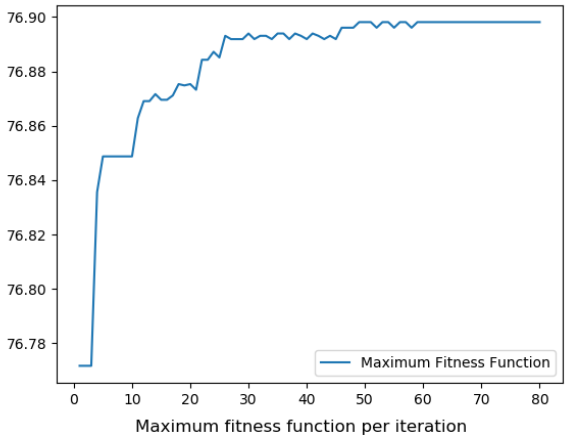

Fig. 2. Average Fitness function

B. Selected features

The selected feature set for this algorithm are as follows.

$$finalFeatureSet = \{3, 8, 17\}$$

Unlike forward selection, these features are examined together and there is no order in adding them.

C. Experimental Results

In the following tables I have written the experimental results for genetic algorithm.

It worth mentioning that the fitness value for genetic algorithm is 76.88 .

TABLE III

EXPERIMENT RESULTS FOR FORWARD SELECTION ALGORITHM ON TRAINING SET.

	c=1	c=2	c=3	total acc	cost
selected features	0.99	0.93	0.97	0.97	24.78
all features	1.0	1.0	1.0	1.0	

TABLE IV

EXPERIMENT RESULTS FOR FORWARD SELECTION ALGORITHM ON TEST SET.

	c=1	c=2	c=3	total acc	cost
selected features	0.99	0.91	0.96	0.95	24.78
all features	0.90	0.99	0.99	0.98	

IV. PART 4

In this part I will to compare these two approaches. The results show that the genetic algorithm performs better. The reason behind this is its ability to considered combined features for selecting the best feature set. However, in forward selection we add features one by one. In terms of running time however, forward selection works much better if our feature set is small.