1 A Discriminate Mutation Strategy in GA to Improving Prediction Accuracy

1.1 Multivariate Statistic Analysis

The first step is to analyse the conditional mean and variance for each attribute conditioning on 'target=1' and 'target=0', and calculate the T^2 metric for each attribute, where T^2 metric is defined as

$$T^2 = [\bar{X}_1 - \bar{X}_0]^2[(\frac{1}{n_1} + \frac{1}{n_0})S_p]^{-1}$$

in which

$$S_p = \frac{n_1 - 1}{n_1 + n_0 - 2}S_1 + \frac{n_0 - 1}{n_1 + n + 0 - 2}S_0$$

and n_1 and n_2 are the number of samples when $target = 1$ and $target = 0$.

Table 1: statistics of the attributes when target = 1

	mean	std
age	52.496970	9.550651
sex	0.563636	0.497444
cp	1.375758	0.952222
trestbps	129.303030	16.169613
chol	242.230303	53.552872
fbs	0.139394	0.347412
restecg	0.593939	0.504818
thalach	158.466667	19.174276
exang	0.139394	0.347412
oldpeak	0.583030	0.780683
slope	1.593939	0.593635
ca	0.363636	0.848894
thal	2.121212	0.465752

Table 2: statistics of the attributes when target = 0

	mean	std
age	56.601449	7.962082
sex	0.826087	0.380416
cp	0.478261	0.905920
trestbps	134.398551	18.729944
chol	251.086957	49.454614
fbs	0.159420	0.367401
restecg	0.449275	0.541321
thalach	139.101449	22.598782
exang	0.550725	0.499232
oldpeak	1.585507	1.300340
slope	1.166667	0.561324
ca	1.166667	1.043460
thal	2.543478	0.684762

Table 3: T^2 metric for each attribute

	Sp	X1-X0	T2
age	8.827614	-4.104480	143.414563
sex	0.444178	-0.262451	11.653548
cp	0.931148	0.897497	65.008119
trestbps	17.334947	-5.095520	112.557643
chol	51.687551	-8.856653	114.044344
fbs	0.356510	-0.020026	0.084538
restecg	0.521432	0.144664	3.016085
thalach	20.732938	19.365217	1359.265531
exang	0.416513	-0.411331	30.526314
oldpeak	1.017205	-1.002477	74.243914
slope	0.578929	0.427273	23.697672
ca	0.937450	-0.803030	51.693507
thal	0.565434	-0.422266	23.697943

According to the above table,

- The attributes, 'age', 'trestbps', 'chol', 'thalach', 'oldpeak', are having the largest variance. This is because these attributes are continues, and larger variance is expected for continuous attributes.
- The attributes, 'set', 'fbs', 'restecg', are non-continuous, and are having the smallest variances, so they provides the much less information according to the entropy theory.
- The remaining attributes 'cp', 'exang', 'slope', 'ca', 'thal', are non-continuous, and are having the largest variances, so they provide the most significant information for classification.

Further correlation analysis for 'cp', 'exang', 'slope', 'ca', 'thal', shows that, they are some correlation between 'exang' and other four attributes. After

excluding 'exang', the remaining four attributes, 'cp', 'slope', 'ca', 'tha' are the most significant attributes. During the GA evolution, these four attributes will enjoy less mutation probability (e.g. 10^{-3}).

	cp	slope	ca	thal	exang
cp	1.	0.11971659	-0.18105303	-0.16173557	-0.39428027
slope	0.11971659	1.	-0.08015521	-0.10476379	-0.25774837
ca	-0.18105303	-0.08015521	1.	0.15183213	0.11573938
thal	-0.16173557	-0.10476379	0.15183213	1.	0.20675379
exang	**-0.39428027**	**-0.25774837**	**0.11573938**	**0.20675379**	1.

1.2 Discriminate Mutation Strategy in GA

In the current GA algorithm, the individual is of a dimension of 13 (as there are total 13 attributes), so it requires population size of $2^{13} = 8192$ individuals to cover all the possible combinations. An improvement is to always select the attributes with the most significant information to be endowed with less mutation probability (e.g. 10^{-3}), while the remaining attributes will have higher mutation probability, in order to explore more individuals with higher fitness.

In the simulation, 'cp', 'slope', 'ca', 'tha' are the most significant attributes, so they will be enjoy much less mutation probability, i.e. 10^{-3}. The remaining 9 attributes will higher but equal mutation probabilities.

Under this discriminate mutation strategy, the initialization of the population at the start of the GA should also be adapted. In the initial population, the individual should always have the four most significant attributes, 'cp', 'slope', 'ca', 'tha', activated.

The implementation of the discriminate mutation strategy is listed in the following,

```
def my_init_individual(container, func, n):
    is_less_mutable = zoo.get_is_less_mutable()
    return container((1 if is_less_mutable[i] else func()) for i in range(n))

# create the individual operator to fill up an Individual instance:
toolbox.register("individualCreator", my_init_individual, creator.Individual, toolbox.ze

def my_mutFlipBit(individual, indpb):
    for i in range(len(individual)):
        if random.random() < indpb[i]:
            individual[i] = type(individual[i])(not individual[i])
    return individual,

# Flip-bit mutation:
# indpb: Independent probability for each attribute to be flipped
toolbox.register("mutate", my_mutFlipBit, indpb=zoo.get_mutate_indpb())
```

This discriminate algorithm is named, 'discriminate RF (v2)'. Under this discriminate mutation strategy, it is found that GA algorithm with the RF algorithm can predict higher accuracy than the traditional 'equal-mutation' strategy.

The accuracy for the five best solutions in 3 random epics are listed in the following table. The average highest accuracy is 0.854839.

best-5-solutions	1st-epic	2nd-epic	3rd-epic	average
0	0.854623656	0.854946237	0.854946237	0.85483871
1	0.851505376	0.854946237	0.854731183	0.853727599
2	0.854731183	0.854623656	0.851397849	0.853584229
3	0.851612903	0.85483871	0.851505376	0.85265233
4	0.851612903	0.851505376	0.848172043	0.850430108

The accuracy for the five best solutions in 3 random epics are listed in the following table. The average highest accuracy is 0.862151, which is 0.007312 higher than the traditional RF.

best-5-solutions	1st-epic	2nd-epic	3rd-epic	average
0	0.862150538	0.862150538	0.862150538	0.862150538
1	0.858924731	0.858817204	0.855591398	0.857777778
2	0.858817204	0.855591398	0.852150538	0.855519713
3	0.855591398	0.855591398	0.849139785	0.85344086
4	0.852150538	0.852150538	0.849139785	0.851146953

1.3 Support Vector Machine (SVM) vs. Random Forest (RF) Algorithm

The advantage of RF is its versatility and it is a very handy algorithm. But its biggest limitation is that a large number of trees constructions can make the algorithm too slow and ineffective for real-time application. Other approaches with similar or even superior performance but will less computation cost are preferred. In the simulation, it is proposed to use the Support Vector Machine (SVM) classifier. It is much faster than the RF, and simulations have demonstrated that SVM can have on average better performance than the RF algorithm, and SVM can converge more quickly than the RF classifier. This algorithm is named, 'discriminate SVM'.

The accuracy for the five best solutions in 3 random epics are listed in the following table. The average highest accuracy is 0.868065, which is 0.013226 higher than the traditional RF.

best-5-solutions	1st-epic	2nd-epic	3rd-epic	average
0	0.868064516	0.868064516	0.868064516	0.868064516
1	0.868064516	0.868064516	0.868064516	0.868064516
2	0.861290323	0.861290323	0.861290323	0.861290323
3	0.858172043	0.858172043	0.858172043	0.858172043
4	0.858387097	0.858172043	0.858387097	0.858315412

1.4 Discrimination Optimization

In the above discrimination based on T^2 and correlation, 'cp', 'slope', 'ca', 'tha' are the selected as the most significant attributes. Another selection is to include 'exang'. Though 'exang' has larger correlation coefficients with the other attributes, the maximum absolute correlation is 0.39428027 which is a weak correlation. In an optimized version of the discrimination, five attributes, 'cp', 'slope', 'ca', 'tha', plus, 'exang' are selected as the most significant ones, and experiments show that this optimization provides on average better accuracy. This algorithm is named, 'discriminate RF (v3)'.

The accuracy for the five best solutions in 3 random epics are listed in the following table. The average highest accuracy is 0.875161, which is 0.020323 higher than the traditional RF.

best-5-solutions	1st-epic	2nd-epic	3rd-epic	average
0	0.87516129	0.87516129	0.87516129	0.87516129
1	0.871827957	0.871827957	0.87172043	0.871792115
2	0.87172043	0.87172043	0.868387097	0.870609319
3	0.871827957	0.871827957	0.868387097	0.870681004
4	0.868387097	0.868387097	0.868602151	0.868458781

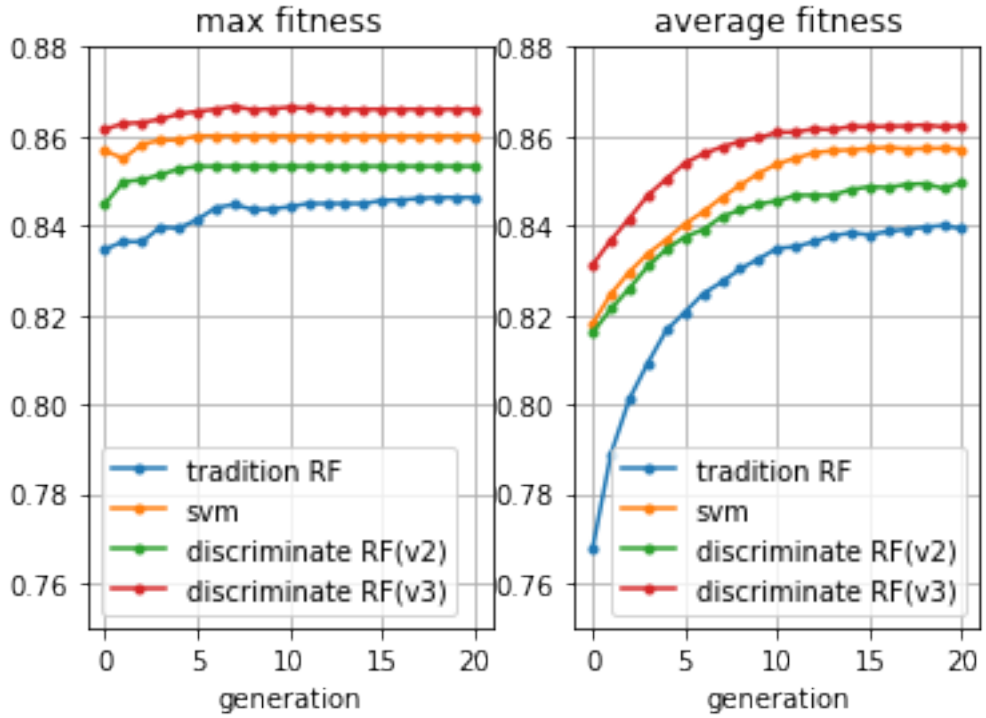
1.5 Simulation Performance

After running 10 random epics separately for the 'traditional RF', 'discriminate SVM', 'discriminate RF (v2)', and 'discriminate RF (v3)', The average accuracy for the 5 best solutions are summarized in the following table.

best solution accuracy	traditional RF	discriminate RF(v2)	discriminate SVM	discriminate RF(v3)
0	0.854946	0.862108	0.868065	0.875161
1	0.854516	0.858871	0.868065	0.871817
2	0.85414	0.857516	0.86129	0.871387
3	0.851925	0.854581	0.858172	0.87114
4	0.851161	0.851903	0.858366	0.868409

From the above table, it is observed that, the 'discriminate RF (v3)' has the highest accuracy on average, up to 0.875161. The 'discriminate SVM' has the second highest accuracy, and the 'discriminate RF (v2)' has the third highest accuracy. All these algorithms are better than the 'traditional RF'.

The following figure shows the convergence speed for the four algorithms. As observed, the 'discriminate RF (v3)' has the highest max fitness as well as the fastest convergence rate. The 'discriminate SVM' is the second best and 'discriminate RF (v2)' is the third best. All are better than the 'traditional RF' with equal mutation strategy.



1.6 Complexity Analysis

Suppose:

- n: number of samples in training. In this case, n = 303
- m: number of attributes. In this case, m = 13
- k: number of less significant attributes in the discriminate strategy. In this case $k = 9$ or $k = 8$
- N: number of trees in the forest in Random Forest Algorithm. In this case, N = 150

In one evaluation of the Random Forest Algorithm, the complexity is

$$O(N * m * n * log(n))$$

In one evaluation of the SVM, the complexity is

$$O(m^2 * n)$$

In the tradition RF algorithm, the search space is 2^m, so its complexity is

$$O(2^m * N * m * n * log(n))$$

In the discriminate RF algorithm, the search space is reduced to 2^k, so its complexity is

$$O(2^k * N * m * n * log(n))$$

In the discriminate SVM algorithm, the search space is reduced to 2^k, and its complexity is

$$O(2^k * m^2 * n)$$