Genetic Algorithms - Scikit Learn (Overview)

Genetic algorithms mimic the natural selection process to search for best solutions. Genetic selection process under module scikit-learn as follows,

Installation

The simplest way to install sklearn-genetic is using pip

```
pip install sklearn-genetic
```

or on anaconda prompt

```
conda install -c conda-forge sklearn-genetic
```

System Requirements

1. Python >= 2.8
2. scikit-learn >= 0.20.4
3. DEAP >= 1.0.3

Example Best features selection from given dataset using GA:

```
from sklearn import datasets, naive_bayes
import pandas as pd
from genetic_selection import GeneticSelectionCV
digit = datasets.load_digits(n_class=10)
x=digit.data
y=digit.target
estimator_gnb=naive_bayes.GaussianNB()
model=GeneticSelectionCV(
    estimator_gnb,
    cv=None,
    scoring='accuracy',
    fit_params=None,
    max_features=5,
    verbose=0,
    n_jobs=1,
    n_population=500,
```

```
    crossover_proba=0.5,
    mutation_proba=0.1,
    n_generations=50,
    crossover_independent_proba=0.1,
    mutation_independent_proba=0.05,
    tournament_size=3,
    n_gen_no_change=None,
    caching=True,
)

selector = model.fit(x, y)
print(selector.support_)
```

Parameter Explanation:

1. *estimator* :fitting algorithm(GaussianNB)
2. *cv=None* :Cross Validation
3. *scoring='accuracy'*: Evaluation methods(like accuracy, r2 etc)
4. *fit_params=None*:Model fitting parameter
5. *max_features=5*: Maximum number of features to be selected from dataset
6. *verbose=0*: Data log
7. *n_jobs=1:* Number of Jobs
8. *n_population=500:* Initial Population
9. *crossover_proba=0.5:* Crossover probability for genetic algorithm
10. *mutation_proba=0.1:* Mutation probability for genetic algorithm
11. *n_generations=50;* Number of generation
12. *crossover_independent_proba=0.1:* Crossover independent probability for each feature
13. *mutation_independent_proba=0.05:*Mutation independent probability for each feature
14. *tournament_size=3:* Parent selection process size
15. *n_gen_no_change=None:*it automatically terminate after getting best result if set to integer
16. *caching=True:*if you want cached the result set to True otherwise False

Output:

Selector.support_ gives True for best selected features among all and false for rest.