1BM22CS241

Sanjeet Prajwal Pandit

```
#N-Queens using Simulated Annealing

#!pip install mlrose-hiive joblib
#!pip install --upgrade joblib
#!pip install joblib==1.1.0
import mlrose_hiive as mlrose
import numpy as np

def queens_max(position):
    no_attack_on_j = 0
    queen_not_attacking = 0
    for i in range(len(position) - 1):
        no_attack_on_j = 0
        for j in range(i + 1, len(position)):
            if (position[j] != position[i]) and (position[j] !=
position[i] + (j - i)) and (position[j] != position[i] - (j - i)):
                no_attack_on_j += 1
        if (no_attack_on_j == len(position) - 1 - i):
            queen_not_attacking += 1
    if (queen_not_attacking == 7):
        queen_not_attacking += 1
    return queen_not_attacking

objective = mlrose.CustomFitness(queens_max)

problem = mlrose.DiscreteOpt(length=8, fitness_fn=objective,
maximize=True, max_val=8)
T = mlrose.ExpDecay()

initial_position = np.array([4, 6, 1, 5, 2, 0, 3, 7])

#The simulated_annealing function returns 3 values, we need to capture
all 3
best_position, best_objective, fitness_curve =
mlrose.simulated_annealing(problem=problem, schedule=T,
max_attempts=500,
                                                          init_state=
initial_position)

print('The best position found is:', best_position)
print('The number of queens that are not attacking each other is:',
best_objective)
```

OUTPUT:

```
The best position found is: [4 0 7 5 2 6 1 3]
The number of queens that are not attacking each other is: 8.0
```

```
#Travelling Salesman Problem
import mlrose_hiive as mlrose
import numpy as np
from scipy.spatial.distance import euclidean

# Define the coordinates of the cities
coords = [(0, 0), (1, 5), (2, 3), (5, 1), (6, 4), (7, 2)]

# Calculate the distances between each pair of cities
distances = []
for i in range(len(coords)):
    for j in range(i + 1, len(coords)):
        dist = euclidean(coords[i], coords[j])
        distances.append((i, j, dist))

# Create a fitness function for the TSP using the distance matrix
fitness_dists = mlrose.TravellingSales(distances=distances)

# Define the optimization problem
problem = mlrose.TSPOpt(length=len(coords), fitness_fn=fitness_dists,
maximize=False)

# Define the simulated annealing schedule
schedule = mlrose.ExpDecay(init_temp=10, exp_const=0.005, min_temp=1)

# Solve the problem using simulated annealing and print the result
structure
result = mlrose.simulated_annealing(problem, schedule=schedule,
max_attempts=100, max_iters=1000, random_state=2)
print("Result structure:", result)

# If the result is a tuple, unpack it accordingly
if isinstance(result, tuple) and len(result) == 2:
    best_state, best_fitness = result
else:
    best_state, best_fitness = result[0], result[1]

# Display the results
print("Best route found:", best_state)
print("Total distance of best route:", best_fitness)
```

```
!pip install mlrose-hiive joblib
!pip install --upgrade joblib
!pip install joblib==1.1.0
```

```
Requirement already satisfied: mlrose-hiive in
/usr/local/lib/python3.10/dist-packages (2.2.4)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (1.1.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-
packages (from mlrose-hiive) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-
packages (from mlrose-hiive) (1.13.1)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.10/dist-packages (from mlrose-hiive) (1.1.3)
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (from mlrose-hiive) (2.2.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from mlrose-hiive) (3.4.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->mlrose-hiive)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas->mlrose-hiive)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas->mlrose-hiive)
(2024.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->mlrose-
hiive) (3.5.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas->mlrose-hiive) (1.16.0)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (1.1.0)
Collecting joblib
  Using cached joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Using cached joblib-1.4.2-py3-none-any.whl (301 kB)
Installing collected packages: joblib
  Attempting uninstall: joblib
    Found existing installation: joblib 1.1.0
    Uninstalling joblib-1.1.0:
      Successfully uninstalled joblib-1.1.0
Successfully installed joblib-1.4.2
Collecting joblib==1.1.0
  Using cached joblib-1.1.0-py2.py3-none-any.whl.metadata (5.2 kB)
Using cached joblib-1.1.0-py2.py3-none-any.whl (306 kB)
Installing collected packages: joblib
  Attempting uninstall: joblib
    Found existing installation: joblib 1.4.2
    Uninstalling joblib-1.4.2:
      Successfully uninstalled joblib-1.4.2
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
```

```
imbalanced-learn 0.12.4 requires joblib>=1.1.1, but you have joblib
1.1.0 which is incompatible.
Successfully installed joblib-1.1.0
```

OUTPUT:

```
Result structure: (array([1, 0, 3, 5, 4, 2]), 21.0293485853026, None)
Best route found: [1 0 3 5 4 2]
Total distance of best route: 21.0293485853026
```