VU Machine Learning
WS 2020

Exercise 3.3

Automated Machine Learning

Nysret Musliu

This is one of possible topics for exercise 3. See other possible topics in tuwel. You have to select only one topic for exercise 3

Exercise 3.3

- Automated Machine Learning
  - Implementation of a metaheuristic algorithm for automated selection/configuration of machine learning algorithms
  - Comparison with other state of the art approaches
  - Group work (like in the first two assignments)
  - Presentations: after the submission

Implementation of algorithm

- Implement a metaheuristic algorithm (local search, genetic algorithms…) that searches for the best machine learning technique (and best hyperparameters) for a particular classification/regression data set
- *Simulated annealing and tabu search can be used, but they should be combined with other algorithms (e.g., iterated local search)*

Implementation of algorithm

- Search space:
  - At least five available machine learning algorithms
  - Most important hyperparameters that should be tuned for each of these algorithms. You can specify for each hyperparameter a reasonable range of possible values
  - The aim is to find a solution (the best algorithm/hyperparameters) in the search space that optimizes an evaluation score (e.g., classification accuracy or RMSE)
- Please write me an email if you have any questions

Comparison with other approaches

- Compare you approach with two state of the art AutoML systems (e.g. auto-sklearn, TPOT…)
- Use for comparison four classification or regression data sets (you can also use the data sets from the previous assignments)
- Time limit: you should use at least 1h per data set

Submission

- Your implementation
- More than 20 slides with this structure
  - Main information for your implementation: representation of solution, neighborhoods (or crossover operators), evaluation function, parameters used for implemented technique…
  - Selected state of the art AutoML systems for comparison
  - Discussion of results
- No report needed for this assignment

Presentations/Discussion of assignment

- Discussion of code
- Implementation issues
- Discussion of results and your findings

Appendix: Metahauristic Techniques

Iterated Local Search (ILS)

http://www.econ.upf.edu/~ramalhin/PDFfiles/2001_MIC_FILS.pdf

As a local search algorithms technique within ILS you should either use **simulated annealing or tabu search (see appendix)**

Genetic algorithms

See appendix or chapters 3 in http://www.cs.vu.nl/~gusz/ecbook/ecbook-course.html

or Memetic algorithms (combination of genetic algorithms and local search)

Implementation of other algorithms is also possible

– Please write me an email if you are interested to implement some other algorithm: nysret.musliu@tuwien.ac.at

Appendix A

A short introduction to local search techniques/genetic algorithms

Definition of search problem

- Given e search space S together with its feasible part

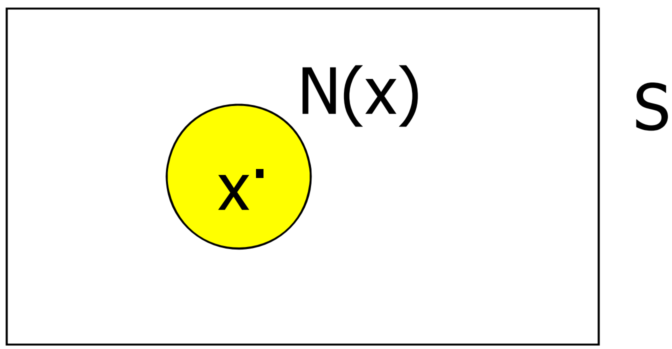
  $F \subseteq S$, find $x \in F$ such that

  $eval(x) \leq eval(y)$ for all $y \in F$

- x that satisfies the above condition is called global optimum (for minimization problem)

Neighbourhood and local optima

- Region of the search space that is near particular point in the space



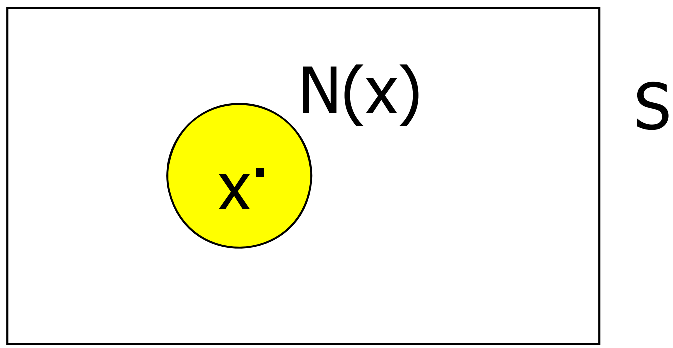
- A potential solution $x \in F$ ia a local optimum with respect to the neighborhood N, if and only if

  $eval(x) \leq eval(y)$,

  for all $y \in N(x)$

Local Search Techniques

- Are based on the neighbourhood of the current solution



- The solution is changed iteratively with so called neighbourhood relations (moves) until an acceptable or optimal solution is reached

Hill Climbing Algorithm

1. Pick a random point in the search space
2. Consider all the neighbours of the current state
3. Choose the neighbour with the best quality and move to that state
4. Repeat 2 through 4 until all the neighbouring states are of lower quality
5. Return the current state as the solution state

Simulated Annealing

- Is based on the analogy from the thermodynamics
- To grow a crystal, the row material is heated to a molten state
- The temperature of the crystal melt is reduced until the crystal structure is frozen in
- Cooling should not be done two quickly, otherwise some irregularities are locked in the crystal structure

Simulated Annealing

```
Prozedure simulated annealing
 begin
    t=0
    Intialize T
    select a current solution v_c at random
    evaluate v_c
    repeat
      repeat
        select a new solution v_n in the neighborhood of v_c
         if eval(v_c) < eval(v_n) then  v_c =v_n
         else if random[0,1) < e^((eval(v_n)-eval(v_c))/T) then v_c=v_n
      until (termination-condition)
      T=g(T,t)
      t=t+1
    until (halting-criterion)
 end
```

SA – problem specific questions

- What is a solution?
- What are the neighbors of a solution?
- What is a cost of a solution
- How do we determine the initial solution

SA – specific questions

- How do we determine the intial “temperature” T”
- How do we determine the cooling ration g(T,t)?
- How do we determine the termination condition?
- How do we determine the halting criterion?

- STEP 1: $T = T_{max}$

  select v_c at random

- STEP 2: pick a point v_n from the neighborhood of v_c

  if *eval*(v_n) *is better than the val*(v_c)

  then select it $(v_c = v_n)$

  else select it with probability $e^{\frac{-\Delta eval}{T}}$

  repeat this step k_T times

- STEP 3: set $T = rT$

  if $T \geq T_{\min}$

  then goto STEP 2

  else goto STEP 1

Basic Tabu Search

```
Procedure Tabu-Suche
begin
 Initialize tabu list
 Generate randomly Initial Solution s_c
 Evaluate s_c
 repeat
   Generate all neighborhood solutions of the solution s_c
   Find best solution s_x in the neighborhood
   if s_x is not tabu solution then s_c= s_x
   else if 'aspiration criteria' is fulfilled then
        s_c= s_x
   else
      find best not tabu solution in the neighborhood s_nt
      s_c = s_nt
   Update tabu list
 until (terminate-condition)
end
```

See
https://dbai.tuwien.ac.at/staff/musliu/ProblemSolvingAI/Class7TabuSearch.pdf

Simple Genetic Algorithm

```
{
    initialize population;
    evaluate population;
    while TerminationCriteriaNotSatisfied
    {
        select parents for reproduction;
        perform recombination and mutation;
        evaluate population;
    }
}
```

Genetic Algorithms: A Tutorial, Wendy Williams:

https://www.csd.uwo.ca/~mmorenom/cs2101a_moreno/Class9GATutorial.pdf