Machine Learning 1, Course Project 2019

Very important – Read before starting

- The deadline for completing and submitting your assignment is strictly Friday 10th January 2020 at 18:00.
- VLE will be set up to not accept late submissions meaning that you will get zero marks if late.
    - Please plan ahead (it is recommended that you try and upload and verify your work a day before).
    - Technical problems, internet connectivity issues, lost backups, cats eating laptops, etc… are not valid excuses.
- You must complete a plagiarism declaration form and include it in your report. Submissions without the form will not be considered.
- Projects must be submitted using VLE only. Physical copies or projects (including parts of) sent by email will not be considered.
- For your convenience, a draft and final submission area will be set up in VLE. Only projects submitted in the final submission area will be graded. Projects submitted to the draft area are not considered.
- It is suggested that after submitting your project, you re-download it and check it just in case. It is your responsibility to ensure that your upload is complete, valid, and not corrupted. You can re-upload the assignment as many times as you wish within the deadline.
- Your project must be submitted in ZIP format without passwords or encryption. Project submitted in any other archiving format (e.g. RAR, 7Z, etc…) will not be considered.
- The total size of your ZIP file should not exceed 38 megabytes.
- Your submission should include your report in PDF format, your source code, and executable file(s).
- It is expected that you submit a quality report with a proper introduction, discussion, evaluation of your work, and conclusions. Also, make sure you properly cite other people's work that you include in yours (e.g. diagrams, algorithms, etc…).
- In general, I am not concerned with which programming language you use to implement this project. However, unless you develop your artifact in BASIC, C, C++, Objective C, Swift, Go, Pascal, Java, C#, Matlab, or Python, please consult with me to make sure that I can correct it properly.
- This is not a group project.
- Plagiarism will not be tolerated.

Project

This assignment is about learning about Ant Colony Optimisation (ACO) and use it to solve instances of the Travelling Salesman Problem (TSP).

TSPLIB is a library of TSP instances that you should use as datasets for this assignment. See the dataset FAQs for how to interpret the data. Link: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

Requirements:

- Design and implement an ACO to solve **symmetric** TSP instances taken from TSPLIB.
- Evaluate the performance of both the ACO-based methods using at least four instance sizes (number of cities) of your choice for each method – choose your sizes wisely so that your evaluation makes sense. Structure your evaluation as follows:

  Instance name: burma14.tsp
  ACO method: *<setup>*
  Results: *<your evaluation, interpretation>*
  …
  …

  Instance name: bays29.tsp
  ACO method: *<setup>*
  Results: *<your evaluation, interpretation>*
  …
  …
  and so on…

Report:

- In your report, dedicate a section about how ACO works, and how it is applicable to TSP. Basically, this is your literature review section.
- Describe how you would deal with the TSP problem using Genetic Algorithms (encoding, etc…) and compare the method to ACO. Note that the methods for GA/TSP we did in class might not necessarily be the best for general cases.
- Describe how you used ACO to tackle the problem (parameters, etc…).
- Must contain an evaluation discussing results, strengths, weaknesses, and limitations. I expect proper experimental procedure discussing your setup, expected outcomes, results, and a good discussion.