Advanced Algorithms
Final Course Project Specifications

Xiu-Jun GONG

College of Intelligence and Computing
gongxj@tju.edu.cn

1 Project topics

You are required to select one of topics below to finish your final project.

1.1 Advanced algorithms for TSP problem

The traveling salesman problem (also called the traveling salesperson problem or TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city? It is one of the most famous benchmarks, significant, historic, and very hard combinatorial optimization problem.

You are required to use or improve the following algorithms in literature for solving the TSP problem.

- Genetic algorithms: [1] , [2]
- Ant Colony Optimization algorithms: [3], [4]
- Deep learning: [5], [6]

1.1.1 Requirements

- Implementing/improving at lease two algorithms above (Using c/c++ or python)
- Evaluating on at least 5 datasets with more than 30 cities from TSPLIB
- Reporting your results including the shortest distances and running time.

1.1.2 Supplement materials

- Genetic algorithms: GitHub
- Ant Colony Optimization: pypi

1.2 Regression analysis for Beijing Multi-Site Air-Quality Data

The Beijing multi-site air-quality dataset [7] from (UCI) includes hourly air pollutants data from 12 nationally-controlled air-quality monitoring sites. The air-quality data are from the Beijing Municipal Environmental Monitoring Center. The meteorological data in each air-quality site are matched with the nearest weather station from the China Meteorological Administration. The time period is from March 1st, 2013 to February 28th, 2017. Missing data are denoted as NA. Its attribute information is shown in table 1.

Table 1: Attribute Information

No.	Name	description
1	No	row number
2	year	year of data in this row
3	month	month of data in this row
4	day	day of data in this row
5	hour	hour of data in this row
6	PM2.5	PM2.5 concentration (ug/m^3)
7	PM10	PM10 concentration (ug/m^3)
8	SO2	SO2 concentration (ug/m^3)
9	NO2	NO2 concentration (ug/m^3)
10	CO	CO concentration (ug/m^3)
11	O3	O3 concentration (ug/m^3)
12	TEMP	temperature (degree Celsius)
13	PRES	pressure (hPa)
14	DEWP	dew point temperature (degree Celsius)
15	RAIN	precipitation (mm)
16	wd	wind direction
17	WSPM	wind speed (m/s)
18	station	name of the air-quality monitoring site

Upon this dataset, we aim at investigating how to predict the value of PM2.5 concentration for each monitoring site using attributes from 12 to 17 plus 3 to 5.

To make use of attributes from 3 to 5 in your predictions, you can encode them as continuous variables to perform regular regression analysis (RRA) or just regard them as time series to perform auto-regression analysis (ARA).

1.2.1 Requirements

- Using three types of regressions including linear, non-linear and deep learning algorithms
- Training your models for each season in years from 2013 to 2017 (Leave 20% for testing) for at lease two monitoring sites
- Reporting testing results including MAE, RMSE and MAPE for at least $4*4*2 = 32$ experiments.

1.2.2 Evaluation Metrics

Supposed that there are n instances in your testing set, y_i is the real value, and $\hat{y}_i$ is the predicted value for instance i.

Mean Absolute Error (MAE) is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

Root Mean Squared Error (RMSE) is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{2}$$

Mean absolute percentage error(MAPE) is defined as:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3}$$

If the value of y is near to zero for some instances in your testing set, y_i in the denominator of above equation should be replaced by its mean value.

1.2.3 Supplement materials

- linear regression: sklearn
- non-linear regression: SVR, DecisionTreeRegressor, RandomForestRegressor
- deep learning regression: refer to the lecture of chapter 3.
- time series regression: prophet and its tutorial
- multiple input features time series regression: tutorial

1.3 Generation of promoter sequences

Given over 4500 natural promoter sequences, you are required design a GAN-based algorithm for generating a given number (at least 10000) of sequences with 61 bp length. These generated sequences must have similar distributions with provided ones.

1.3.1 Requirements

The algorithm is evaluated by the following aspects against the natural promoter sequences:

- Motifs by PSSM matrix
- Top k ($k \geq 10$) most frequently occurring k-mers (k=4,6)
- Motifs by DREME and FIMO from MEME suite

1.3.2 Supplement materials

- GAN algorithm: url
- Dataset (promter-seqs.fasta) can be download from here (Password: rpeN)
- The evaluation methods can refer to [8]

2 Preparing project reports

Your project report must contain the following items:

- Title: less than 60 words
- Abstract: 300 - 500 words
- Introduction: 1+ page(s)
- Methods: 2+ pages
- Results: 3+ pages
- Conclusion: 300 - 500 words
- References: 3+ citations

3 Submission

The following stuffs should be compressed into a file with name: 学号-姓名-题目.zip

- a src directory: source codes
- a data directory: not the raw data, but name list of datasets and outputs of experiments.
- a report file: in word or pdf format.
- a README file: explain the functions of all files in src and data directories

4 Scoring your project

Your score will be evaluated based on the following factors:

- Rationality of methods: 40%
- Significance of results: 40 %
- Clarity of source codes: 10 %
- Quality of report presentation: 10%

References

[1] Abid Hussain, Yousaf Shad Muhammad, M. Nauman Sajid, Ijaz Hussain, Alaa Mohamd Shoukry, and Showkat Gani. Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator. Computational Intelligence and Neuroscience, 2017:1–7, 2017.

[2] Jihene Kaabi and Youssef Harrath. Permutation rules and genetic algorithm to solve the traveling salesman problem. Arab Journal of Basic and Applied Sciences, 26(1):283–291, January 2019.

[3] Fadl Dahan, Khalil El Hindi, Hassan Mathkour, and Hussien AlSalman. Dynamic Flying Ant Colony Optimization (DFACO) for Solving the Traveling Salesman Problem. Sensors (Basel, Switzerland), 19(8), April 2019.

[4] Abdulqader M. Mohsen. Annealing Ant Colony Optimization with Mutation Operator for Solving TSP. Computational Intelligence and Neuroscience, 2016:1–13, 2016.

[5] Feidiao Yang, Tiancheng Jin, Tie-Yan Liu, Xiaoming Sun, and Jialin Zhang. Boosting Dynamic Programming with Neural Networks for Solving NP-hard Problems. Proceedings of Machine Learning Research, 95:726–739, 2018.

[6] Gorker Alp Malazgirt, Osman S. Unsal, and Adrian Cristal Kestelman. TauRieL: Targeting Traveling Salesman Problem with a deep reinforcement learning inspired architecture. arXiv:1905.05567 [cs], May 2019. arXiv: 1905.05567.

[7] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in Beijing. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2205):20170457, September 2017.

[8] Ye Wang, Haochen Wang, Lei Wei, Shuailin Li, Liyang Liu, and Xiaowo Wang. Synthetic promoter design in Escherichia coli based on a deep generative network. Nucleic Acids Research, 48(12):6403–6412, 2020.