




COSC2636/2632 Big Data Management

Assignment 2

	Assessment Type: Individual assignment; no group work. Submit online via Canvas→Assignments→Assignment 2. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements. This assessment supports CLO 1-5.
	Due date: 23:59pm, Sunday of Week 8; Deadlines will not be advanced but they may be extended. Please check Canvas→Syllabus or via Canvas→Assignments→Assignment 2 for the most up to date information. As this is a major assignment in which you demonstrate your understanding, a university standard late penalty of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
	Weighting: 18 marks

1. Overview

Problem Description:

Given a set of existing routes in social network, a passenger wants to go from source A to destination B, your need to implement a method to return the k nearest direct routes (user can follow to travel from A to B) efficiently. The basic requirements are below:

1. k, A, B are arbitrary and can be input by users.
2. Using the searching idea from paper to **achieve early termination, without checking all candidates in the dataset**. Therefore, a **brute-force method that scans all candidates is not acceptable**.

Dataset:

We provide a dataset of real people's travel routes in Los Angeles. You can find it in the directory "index/la_trips.txt" of the IKNN code package.

For example,

558483,33.595985412597656,-117.71573638916016

558483,34.05152130126953,-118.29872131347656

558483,33.70815658569336,-117.7821273803711

558483,33.74482345581055,-118.41141510009766

Stores a route of 4 points and the route id is 558483. Each line in the file represents a particular stop's latitude and longitude of this route.

Another file called "index/la_points.txt" stores all the unique points covered by those routes after our pre-processing.

Step-by-Step Guidance to complete this assignment:

a. Read paper

- i. Section 1 to 4.1 of the paper "Searching Trajectories by Locations – An Efficiency Study", which is introduced in both the lecture and tutlab.
- ii. Understand the input, output and search paradigm (Algorithm 1).
- iii. Familiar with the operations of R-tree (which you have learnt in lecture 2-3 and tutorial & lab in the first 5 weeks).

b. Have a look at the LA trajectory dataset (available at /index/la_trips.txt in the IKNN code package).

c. Understand how to store and index trajectories.

- i. Store trajectory as points.
- ii. Create R-tree index based on all the points (see /index/la_points.txt in the IKNN code package).
- iii. Building a mapping table to show the relationships between point and trajectory. (can be maintained in the main memory)

d. How to search k nearest routes by a pair of points?

- i. How to compute the distance from query to the route?
- ii. Build spatial index.
- iii. **Filter impossible routes in advance based on bound comparison.**
- iv. **Refine the remaining candidate routes.**

Requirement:

Note that in this assignment, we provide most part of the code, and what you need to implement is actually d(iii) and d(iv) highlighted above (details are in Sec 4.1 of the paper).

The code skeleton is located in the path (i.e., src/rmit/IKNN.java). What you need to do: (1) understand the code skeleton and the function of each API; (2) implement the core functions in "IKNN.java"; (3) after you finish all the codes, run the "src/rmit/Test.java" with correct input formats to check the program.

Before implementing the code:

You need to create a database to insert all trajectories to a database, for later use at "refinement" step (step d(iv)) the points of dataset. Here, we give the following guidance on how to do the insertion of data.

1. Install MySQL
2. Run SQL file in the db folder "src/db/tb_la_dataset.sql".
3. Change the setting in Settings.java.

Sample Test Case:

The sampled test case "Sample - TestCase.txt" is contained in the assignment folder.

What you will learn:

First, you need to understand the distance model between query and trajectory, and know how to compute the bound, and observe whether some trajectories can be skipped without checking, and see how different parameters may affect the performance, like the number of returned results. Further, you need to see how many routes can be discarded compared with the brute-force, and the effect on the result of various k.

If there are questions, you can email the tutors.

2. Assessment criteria

We will evaluate your program in terms of its correctness and efficiency on our test cases. And we use 36 test cases with different k values to evaluate your program, and each test case is worth 0.5 mark.

Note: The marking is based on the correctness of your implemented algorithm, as well as the efficiency upon the correctness of the results obtained.

1. For the correctness, if the top-k results are not totally right, then the mark will be **zero**.
2. For the efficiency, all your programs will be evaluated under the uniform computer setting with 8GB memory and Intel(R) Core (TM) i5-7200 CPU @ 2.50GHz processor. Generally, the results can be obtained in a few seconds. However, if the program adopts brute-force or strategy without early termination, then it will take way longer time than expected. Thus the mark of the test case with more than **30 seconds** running time will be **zero** without hesitation, no matter whether the results are right.
3. No marks will be given for this assignment if you use brute force (a.k.a. scanning the whole dataset to get the top k items) to solve this problem, or do not use the code skeleton we provided.
4. No marks will be given if you change the function "main()" (e.g., output format and default parameter setting.).

3. Learning Outcomes

Please refer to the link ("<http://www1.rmit.edu.au/courses/050436>") for the learning outcome.

4. Referencing guidelines

What: This is an individual assignment and all submitted contents must be your own (except for the main body of the code provided if any). If you have used sources of information other than that, you must give acknowledge the sources and give references using IEEE referencing style.

Where: Add a code comment near the work to be referenced and include the reference in the IEEE style.

How: To generate a valid IEEE style reference, please use the [citethisforme tool](#) if unfamiliar with this style. Add the detailed reference before any relevant code (within code comments).

5. Submission format

Submit **one .java file** showing the final output of your program via [Canvas→Assignments→Assignment 2](#). It is the responsibility of the student to correctly submit their files. Please verify that your submission is correctly submitted by downloading what you have submitted to see if the files include the correct contents.

Note: One single java file (i.e., IKNN.java): it contains your implementation of the IKNN algorithm.

Naming convention: StudentNumber_A2.java (e.g., s1234567_A2.java)

6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [University website](#).

7. Assessment declaration

When you submit work electronically, you agree to the [assessment declaration](#).