COMP 9102 Data Management and Information Retrieval
Assignment 2
Nearest Neighbor Similarity Queries
Due Date: Oct 27th, 2017 5:00pm

Question specification.

In this assignment, you are required to implement and benchmark nearest neighbor similarity queries using the two-step and multi-step similarity search algorithms. The algorithms should take as input a query vector q and should compute the nearest neighbor p of q.

Dimensionality Reduction：

1) Apply PCA (Principal Component Analysis) algorithm to reduce dimension of dataset to k. You should first centralize data points (by subtracting the mean value of data in each dimension) and apply a SVD (Singular Value Decomposition) on the data matrix A. See this link for reference.
2) You should first decompose A as $A=USV^T$. Then select first k column of U and S as U',S'. And B=U'*S' stores the mapped data points in the reduced dimension space.

Proof of correctness：

In your report, you need to prove that the distance in the reduced dimensional space is a lower bound to the distance in the original space.

R-tree index:

Construct an R-tree index for the reduced vectors.
Hint: You do not have to code the R-tree from scratch. Various libraries exist for the R-tree index. For example, you may refer to boost::geometry::index::rtree library for C++. You need to write test cases for R-tree insertion and queries, and make sure the library you used is correct.

Nearest neighbor similarity search：

Implement both the two-step nearest neighbor similarity algorithm and the multi-step nearest neighbor similarity algorithm. Please refer to lecture notes for details about these algorithms. Do remember to compare your result to a linear scan algorithm to assert correctness of your algorithms.

Dataset.

Navigate to https://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.html
1) Download ColorHistogram.asc.gz to your local folder.
2) We use Euclidean distance as the similarity measure.

Requirements.

1）Implement and test the requested indexes and algorithms.

2) Run a benchmark experiments where you test the performance of your algorithms. In an experiment you should run a huge number of similarity search queries and average the running time and the number of required distance computations in the original high-dimensional space. Your benchmark scheme should evaluate the performance of both algorithms and the linear scan algorithm on different choice of k (for example 5, 10 ,15) .

3) Write a short report about your implementation and the experiments and submit the report together with your code at the course website. Your code should be compilable without problems and you should include basic instructions on how to compile and use it. Your program can be written in your preferred programming language (e.g., C, C++, Java, Python, etc.). Your program must run within reasonable time.

Submission.

Please submit your assignment (one **ZIP** file) to moodle on or before 5:00pm, Oct 27th, 2017. Make sure all contents are readable.

Please feel free to post your questions on **Moodle forum**, or contact TA Daniel (dhding2@cs.hku.hk) if you encounter any difficulty in this assignment. We would be happy to help.