Assignment: “Scalable Processing of Top-k Dominating Queries”
M.Sc. Data and Web Science 2019-2020 - Mining of Massive Datasets

Introduction
In this project, you will work with multi-dimensional data. Given a potentially large set of *d*-dimensional points, where each point is represented as a *d*-dimensional vector, we need to detect the *k* points that have the highest **domination score**. We say that a point p dominates another point q, when p is as good as q in all dimensions and it is strictly better in at least one dimension. We will assume that small values are preferable. For example, the point p(1, 2) dominates q(3, 4) since 1 < 3 and 2 < 4. Also, p(1, 2) dominates q(1, 3) since although the have the same x coordinate, the y coordinate of p is smaller than that of q.

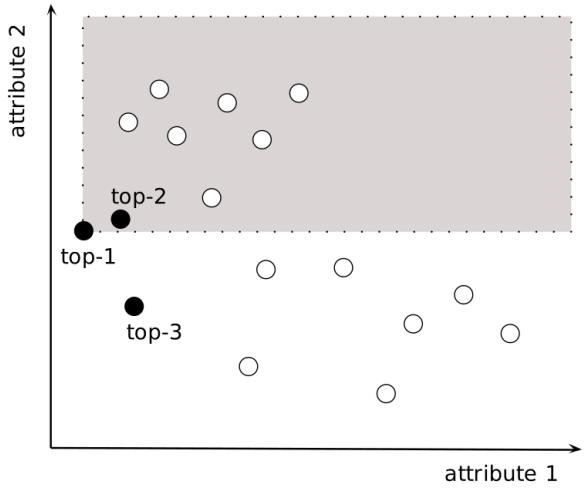

Figure 1. Example of a top-3 dominating query.

An example for top-3 is shown in Figure 1. The dominating region of the top-1 point is shown gray. Any point that falls inside this region is dominated by top-1. Based on the Figure the top-1 point has a domination score of 8, since it dominates 8 other points.

Requirements
Given a dataset of *d*-dimensional points you should implement a scalable and efficient algorithm to detect the *k* most important points, i.e., the top-*k* points that have the highest domination scores. Your algorithm must implemented in Scala and you should use Apache Spark. Note that the parameter *d* (number of dimensions) depends on the dataset whereas the parameter *k* is user-defined and must be given as an input to the algorithm. The point coordinates in general may be double numbers, so you should treat coordinate values as doubles. You should provide results for different values of *k*, different dimensionalities, different data distributions and different data cardinalities. (**Tip**: *start with a small 2-dimensional dataset in order to be able to check if your algorithm provides the correct results*.). Examples of data distributions in the 2-d space are given in Figure 2.

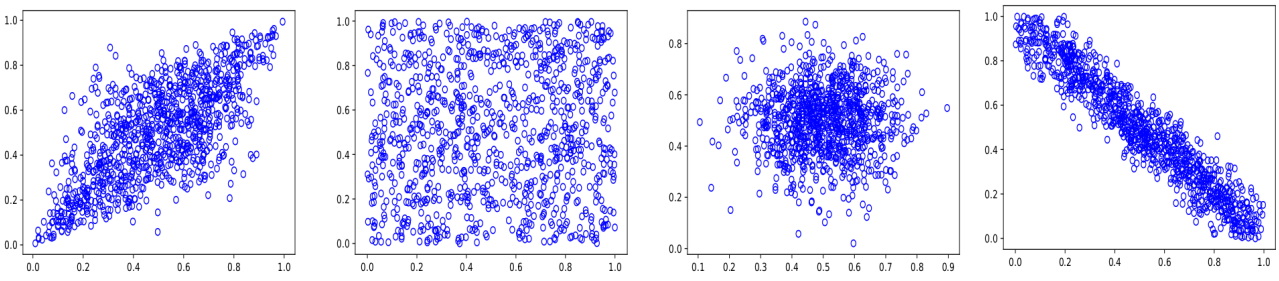

Figure 2. Different distributions for 2-d (from left to right): correlated, uniform, normal, anticorrelated.

Deliverables

You should deliver the source code of your solution, the code to generate the data distributions and a report describing what you did, in detail. Also, you need to prepare 9-10 slides for the presentation session that will take place at the end of the semester. Note that, the code for data generation can be in another programming language, e.g., Java, Python or anything you prefer. However, the code for your solution must be in Scala.

Bibliography

http://www4.comp.polyu.edu.hk/~csmlyiu/journal/vldbj_kdom.pdf

http://delab.csd.auth.gr/~apostol/pubs/isps2015_tpm.pdf

https://link.springer.com/article/10.1007/s11280-015-0340-6