CS 594 Geometric Algorithms for Data Analysis

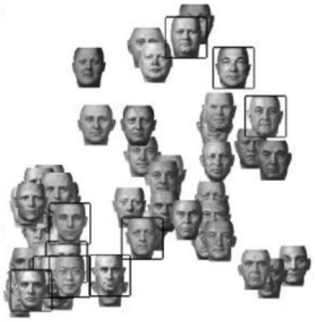
Instructor: Anastasios Sidiropoulos

Method of instruction: The instruction will be based on the following main components:

- During the first half of the course, the instructor will present various fundamental methods and ideas from computational geometry and discuss their applications in data analysis. Any necessary prerequisites will also be discussed during this time.
- During the second half of the course, the students will read and present research papers.
- The students will work on a project of their interest that incorporates ideas discussed in the class. The students will have the option to either conduct original research or experimentally evaluate prior work. The project will be performed in teams of 1–3 students. The students will be encouraged to start thinking about possible research topics early in the semester. The instructor will hold frequent meetings with each team to guide their progress.

Narrative description: Complex data sets arise in a plethora of application domains from measurements of various physical processes, history of financial transactions, logs of user activity in a network, and so on. The analysis of such data sets is therefore a task of increasing importance for science and engineering. Even though in many applications there is an abundance of such raw inputs, extracting meaningful information can often be a major computational challenge. In most cases, this difficulty is due to the lack of a useful representation of the data.

Over the recent years, geometric methods have become an indispensable tool for overcoming this difficulty. The reason behind this development is the fact that a data set endowed with pairwise similarities can be naturally interpreted as a geometric space. Such data sets include DNA sequences, statistical distributions, collections of news articles, and so on. Under this interpretation, several important data analytic questions can be understood as geometric computational problems. For example, the problem of classification can be expressed as geometric partitioning. Similarly, the problem of fitting a model to a set of measurements can be thought of as interpolation in some appropriate geometric space.



In these contexts, the main algorithmic challenges occur in high-dimensional, or more generally, complex metric spaces. Contemporary Computational Geometry aims at addressing the above challenges via the design of efficiently algorithmic methods for the analysis of these spaces.

Goal: In this course, the students will be exposed to algorithmic methods from computational geometry for the analysis of high-dimensional and complex data sets. Emphasis will be given on understanding the state of the art of these methods, and on developing intuition about which methods are appropriate in various application contexts.

Student deliverables: The students will have to read all the papers, and they will be expected to actively participate in all the lectures. Furthermore, each student will present at least one research paper to the class. For the final project, the students will have to submit a proposal of their selected topic within the first half of the course, a final report at the end of the class, and they will be asked to give a brief presentation on their findings.

Class meetings: There will be two 75' meetings per week.

Prerequisites: The course will be accessible to students with a wide range of backgrounds, including both theoretical and applied areas of computer science, as well as mathematics. Some familiarity with algorithms will be assumed, equivalent to a CS 401-level course.

Exams: There will be no exams.

Readings: Selected research papers from the following tentative list of topics:

- *Compression:* How to compress a data set, by introducing only a small error (for example, distance oracles, graph spanners, dimensionality reduction, and so on).
- *Sketching:* How to summarize a large data set, so that relevant information can be recovered from a small sketch (for example, via coresets).
- *Similarity search:* How to find similar elements in a large data set (for example, approximate nearest-neighbor search, locality sensitive hashing, and so on).
- *Metrical simplification:* How to approximate a "complicated" metric space by a simpler one (for example, embedding arbitrary metric spaces into random trees).
- *Clustering:* How to partition a large data set into a few classes of closely-related elements (for example, k-means, k-median, k-center, spectral clustering, and so on).
- *Outlier detection:* How to remove irrelevant/erroneous elements from a data set.
- *Metric learning:* How to transform a geometric data set, so that it agrees with the opinions of experts, and how to use this transformation for prediction.

The discussion of the above topics will include motivational examples from various application domains, such as visualization, machine learning, bioinformatics, statistics, networking, and streaming/massively parallel computing.

Overlap with other courses: To the best of my knowledge, this course does not have a significant overlap with any other CS 594 courses from recent years.

The problem of clustering is discussed in CS 412 and in CS 583 by prof. Liu. However, in the proposed course, clustering will be covered from the point of view of algorithm design. Several different clustering objectives will be compared in terms of their computational difficulty, and the emphasis will be on presenting the currently best-known approximation algorithms. This treatment of the subject aims to complement the discussions in more applied courses.

The problem of dimensionality reduction is discussed in CS 412. In the proposed course, this topic will be covered at a greater depth, and as a specific instance of a more general "compression" theme, which includes methods such as graph spanners, distance oracles, and so on.

The topic of prediction is covered in great depth in the course CS 594 by prof. Ziebart (Fall 2013). In the proposed course, prediction will only be mention in passing, as a motivational example for the methods of metric learning.