
David C. Anastasiu∗
San José State University
San José, CA, USA

Huzefa Rangwala
George Mason University
Fairfax, VA, USA

Andrea Tagarelli
University of Calabria
Rende (CS), Italy

ABSTRACT
Finding nearest neighbors is an important topic that has attracted much attention over the years and has applications in many fields, such as market basket analysis, plagiarism and anomaly detection, community detection, ligand-based virtual screening, etc. As data are easier and easier to collect, finding neighbors has become a potential bottleneck in analysis pipelines. Performing pairwise comparisons given the massive datasets of today is no longer feasible. The high computational complexity of the task has led researchers to develop approximate methods, which find many but not all of the nearest neighbors. Yet, for some types of data, efficient exact solutions have been found by carefully partitioning or filtering the search space in a way that avoids most unnecessary comparisons.

In recent years, there have been several fundamental advances in our ability to efficiently identify appropriate neighbors, especially in non-traditional data, such as graphs or document collections. In this tutorial, we provide an in-depth overview of recent methods for finding (nearest) neighbors, focusing on the intuition behind choices made in the design of those algorithms and on the utility of the methods in real-world applications. Our tutorial aims to provide a unifying view of “neighbor computing” problems, spanning from numerical data to graph data, from categorical data to sequential data, and related application scenarios. For each type of data, we will review the current state-of-the-art approaches used to identify neighbors and discuss how neighbor search methods are used to solve important problems.

CCS CONCEPTS
• Information systems → Nearest-neighbor search; • Theory of computation → Nearest neighbor algorithms.

KEYWORDS
nearest neighbor search, graph construction, sparse data.

1 TUTORIAL OUTLINE
The tutorial (https://bit.ly/2VF5GFg) will first provide a formal definition of nearest neighbor search (NNS) and related problems and a summary of classical space partitioning-based approaches and their limitations. The importance of the NNS problem will be motivated by a series of application domains and methods that use NNS as a black-box kernel. The notion of neighbor is central for a plethora of problems and tasks in clustering, multi-label classification [11, 15, 17], anomaly detection, network/graph mining, recommender systems [1, 14, 19], bioinformatics and computational genomics [6, 9, 16], to name just a few. Depending on the domain, the notions of neighbor and neighborhood are key to enabling the modeling of a variety of phenomena; in the social network context, for example, such phenomena range from homophily effects to behavioral or state-transition dynamics, from information spread to information inference. Modeling such phenomena is essential for a number of classic problems in graph mining, such as community search, detection and evolution, link prediction, influence propagation, trust inference [12]. We will showcase the use of NNS methods as a key component in solving many of these problems.

The last few years have brought considerable advances in NNS methods. Some recent methods have focused on the discovery of new hash functions that, in the expectation, more closely relate objects [8, 13, 21]. Zhang and Zhang [21], for example, developed a hashing technique based on metric embeddings for edit distance that significantly outperforms all previous methods for DNA and other long string searches. Several works have shown that data-dependent hashing can outperform distribution-agnostic techniques. For metric spaces, a series of efficient approximate search methods rely on a navigable small world graph with nodes corresponding to the searched objects. Moreover, for objects best represented as sparse vectors, a variety of index traversal strategies achieve near-optimal search performance by leveraging the sparsity in the data and ignoring the majority of the object comparisons that are not similar enough [2–5, 13]. Anastasiu and Karypis, for example, developed several effective filtering techniques that leverage the Cauchy-Schwarz inequality applied to vector subspaces which enabled exact nearest neighbor graph construction methods to outperform even approximate solutions that were tuned to achieve at least 95% recall. Finally, some recent methods have expanded the scope of the search problem. Yu et al. [20] studied the case where the search has a limited computational budget, while Morales et al. [10] define a time-dependent similarity function for computing streaming similarity self-joins. In this tutorial, we will describe these and other recent state-of-the-art methods and provide guidance for choosing an appropriate method for finding neighbors in different domains.
REFERENCES


