Project INF442-4

Constrained and k Shortest Paths

Contact: Martin (martin.krejca@polytechnique.edu)

Conceptual design by Juan-Antonio Cordero

1 Introduction

Finding a shortest path in a network is a fundamental problem in computer science and graph theory. The aim is to find a path from a given source to a stated destination at minimum cost. This problem shows up frequently when resources meed to be sent between two specific points as quickly, cheaply, or reliably as possible. In the domain of data science, these applications are commonly related to the reduction of dimensionality of high-dimensional data sets. In order to not lose important information in such a reduction, it is desirable that the mapping does not change the distances among its data points, that is, the *mapping* should be *isometric*. This approach is also known as *Isomap* [TSL00]. In this project, you are tasked to implement different algorithms for determining shortest paths and then use them ultimately for an Isomap setting.

In the following, we discuss the setting of this project as well as the different tasks. In total, there are *four* tasks (Tasks 1 to 4). For each task, various algorithms exist. You are required to implement *at least one* algorithm per task of the specified type. To this end, please also utilize the data sets mentioned in Section 5. If you implement more than one algorithm, this can only favor the grading positively. We encourage you to use the following information as pointers for looking up further information by yourself.

2 Preliminaries

We consider directed, weighted, connected graphs with nonnegative edge weights. For a vertex set V, an edge set $E \subseteq V^2$, and a function $w\colon E \to \mathbf{R}_{\geq 0}$, we denote such a graph by (V, E, w). A *path* P (of *size* $n \in \mathbf{N}$) *from* $s \in V$ *to* $t \in V$ is a sequence $(v_i)_{i=0}^{n}$ of vertices such that $v_0 = s$, that $v_n = t$, and that for all $i \in [n]$, it holds that $(v_{i-1}, v_i) \in E$. The *length* $w(P)$ of P is the cost of all edges forming P, that is, it holds that $w(P) = \sum_{i \in [n]} w(v_{i-1}, v_i)$. A *shortest* path from s to t is a path from s to t with *minimal* length among all possible paths from s to t. Last, we say that a graph is *connected* if and only if for each $(s, t) \in V^2$, there exists a path from s to t.

3 The Shortest-Path Problem and Its Variants

Given a directed, weighted, connected graph (V, E, w) and two vertices $s, t \in V$, the *shortest-path problem* asks for a shortest path from s to t. This problem has been well studied for decades. By now, there exists a plethora of efficient sequential as well as parallel algorithms. Famous sequential algorithms include Dijkstra's algorithm [Dij59] as well as the algorithm by Bellman [Bel58] (which computes shortest paths from a single vertex to *all* other vertices in the graph). An example for a parallel algorithm is a parallelization of Dijkstra's algorithm [CMM+98].

Task 1. Implement sequential and parallel algorithms for finding a shortest path in a given graph and with a given start vertex to a given end vertex.

3.1 Variants of Finding Shortest Paths

In many settings, computing a single path is not desirable or not feasible. We consider *two* such variants of the shortest-path problem: finding a *constrained* shortest path and finding k shortest paths.

Finding a Constrained Shortest Path

We are given a graph (V, E, w) (following all the properties mentioned in Section 2), two vertices $s, t \in V$, as well as a *delay* function $d\colon E \to \mathbf{R}_{\geq 0}$ and a bound $b \in \mathbf{R}_{\geq 0}$. The *delay of a path* P, denoted by $d(P)$, is defined analogously to the length of P by substituting w in the sum by d. The goal is now to find a shortest path P' from s to s with a delay of at most b, that is, it must hold that $d(P') \leq b$.

Task 2. Describe and implement algorithms for solving the problem of finding a constrained shortest path.

Finding k Shortest Paths

We are given a graph (V, E, w) (following all the properties mentioned in Section 2), a *single* vertex $s \in V$, as well as a positive integer k. The aim is to enumerate k (different, edge-wise nonempty) shortest paths from s that are shortest among *all* k-subsets of paths from s. In other words, the task is to return k paths from s to vertices that are closest to s, measured in w.

Watch out! The k-shortest-paths problem typically refers to a different problem than the one above.

Task 3. Describe and implement algorithms for solving the problem of finding k shortest paths as explained above.

4 Application to Isomap

We intend to apply our algorithms from before to Isomap [TSL00] on graph data. This algorithm relies on computing the following values:

1. The neighborhood graph of the input. To this end, one connects each vertex to a certain subset of vertices. Typically, one chooses either those within a certain vicinity $\varepsilon \in \mathbf{R}_{\geq 0}$ (for which we can use Task 2) or the k closest ones (for which we can use Task 3).
2. The square matrix of shortest distances among all data points. The shortest distances are measured in the neighborhood graph.
3. The eigendecomposition of the distance matrix from item 2. This decomposition represents the lower-dimensional embedding of the data. You are allowed to use external libraries for computing the eigendecomposition.

Task 4. Implement and evaluate the performance of Isomap for different values of ε and K.

5 Data Sets

For Tasks 1 to 3, please use specific problems from OR-Library:

- `http://people.brunel.ac.uk/%7Emastjjb/jeb/orlib/rcspinfo.html`

We ask you to use the files `rcsp1.txt`, `rcsp2.txt`, `rcsp3.txt`, and `rcsp4.txt`.

For Task 4, please use multiple data sets of your choice from the ALOI:

- `https://aloi.science.uva.nl/`

References

[Bel58] Richard Bellman. “On a routing problem.” In: *Quarterly of Applied Mathematics* 16.1 (1958), pp. 87–90. DOI: `10.1090/qam/102435` (see page 1).

[CMM+98] Andreas Crauser, Kurt Mehlhorn, Ulrich Meyer, and Peter Sanders. “A parallelization of Dijkstra’s shortest path algorithm.” In: *Proceedings of MFCS’98*. Vol. 1450. 1998, pp. 722–731. DOI: `10.1007/BFb0055823` (see page 1).

[Dij59] Edsger W. Dijkstra. “A note on two problems in connexion with graphs.” In: *Numerische Mathematik* 1 (1959), pp. 269–271. DOI: `10.1007/BF01386390` (see page 1).

[TSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. “A global geometric framework for nonlinear dimensionality reduction.” In: *Science* 290.5500 (2000), pp. 2319–2323. DOI: `10.1126/science.290.5500.2319` (see pages 1, 2).