Integrating Data by Discovering Topological and Proximity Relations among Spatiotemporal Entities

Georgios M. Santipantakis and Christos Doulkeridis and Akrivi Vlachou and George A. Vouros

Link discovery (LD) is the process of identifying relations (links) between entities that originate from different data sources, thereby facilitating several tasks, such as data deduplication, record linkage and data integration. Existing LD frameworks facilitate data integration tasks, over multidimensional data. However, limited work has focused on spatial or spatiotemporal LD, which is typically much more processing-intensive due to the complexity of spatial relations. This chapter targets spatiotemporal link discovery, focusing on topological and proximity relations, proposing a framework with several salient features: support both for streaming and archival data, support of spatial relations in 2D and 3D, flexibility in terms of input consumption, improved filtering techniques, use of blocking techniques, proximity-based LD instead of merely topological LD, and a data-parallel design and implementation. The efficiency of the proposed spatiotemporal LD framework is demonstrated by means of experiments on real-life data from the maritime and aviation domains.

Data integration is a critical task for applications managing data that originates from different and often heterogeneous data sources. In the current era data of massive volume that are often generated at unprecedented rates in the form of data streams, and in different representation models, formats and modalities, big data integration raises additional challenges.

Georgios M. Santipantakis
Dept. of Digital Systems, University of Piraeus, Karaoli & Dimitriou 80, 18534 Piraeus, e-mail: gsant@unipi.gr

Christos Doulkeridis
Dept. of Digital Systems, University of Piraeus, Karaoli & Dimitriou 80, 18534 Piraeus, e-mail: cdoulk@unipi.gr

Akrivi Vlachou
Dept. of Digital Systems, University of Piraeus, Karaoli & Dimitriou 80, 18534 Piraeus, e-mail: avlachou@unipi.gr

George A. Vouros
Dept. of Digital Systems, University of Piraeus, Karaoli & Dimitriou 80, 18534 Piraeus, e-mail: georgev@unipi.gr

One significant step to facilitate data integration is link discovery, which is defined as the process of identifying relations (links) between entities/objects that originate from different data sources. Different applications of link discovery tasks include:

- (Record linkage between data sources): Consider a merge taking place between two companies, which requires that their customer databases need to be reconciled. Customers are described by their names and identifying common customers requires joining records based on alphanumeric values (names). However, in practice, using exact matching does not work sufficiently well, due to various reasons, such as spelling errors (e.g., "Jon Smith" vs. "John Smith"), use of middle name (e.g., "George Vouros" vs. "George A. Vouros"), etc. Therefore, approximate matching techniques (a.k.a. approximate or fuzzy joins) are required, which constitute a special case of link discovery between entities.
- (Entity resolution): Consider two different data representations corresponding to two entities, where the problem is to determine if the two entities are identical. For instance, a Wikipedia page describing an athlete and a record in a table of a relational database of athletes. The problem is to discover that two entities refer to the same real-world object.
- (Deduplication): Consider the case of two persons that perform data entry using different naming conventions. In this case, we may encounter multiple records that are not identical, yet they refer to the same object. As a representative example one can think of "Los Angeles" vs. "LA", "fifth avenue" vs "5th avenue", etc.

In order to support application scenarios such as the above, efficient and effective link discovery techniques are sought. This book chapter focuses on a special case of link discovery, where the underlying data sets are of spatiotemporal nature and the relations to be discovered are also spatial or spatiotemporal. Specifically, we present the design and implementation of a generic link discovery framework tailored for spatiotemporal data. We present its extensible design and flexibility in terms of supported spatiotemporal data types and relations and we outline a state of the art method for link discovery of topological relations. We further present improvements that apply to certain cases of spatiotemporal data and may lead to performance gains, as well as methods for proximity-based link discovery. The presented methods also consider the case of more complex geometries, such as polylines.

The generic architecture of the stLD framework is illustrated in Figure 1. Its input is two data sources (not necessarily disjoint) represented in RDF format. The output of stLD is also provided in RDF and can be either: (a) only the linked entities discovered, or (b) the linked entities concatenated with the input RDF fragment. The first option allows to decouple the processing of input data sets from the processing of discovered links, and reduces the overall link discovery time. The second option provides synchronized and sequential RDF fragments to the output, which is beneficial for certain applications that need to process enriched input entities (i.e., with additional spatiotemporal links and properties).

The stLD framework follows the *filter-and-refine* methodology for performing the link discovery task. In the filtering step, a blocking method is employed to drastically reduce the number of candidate pairs of entities, whereas in the refinement step

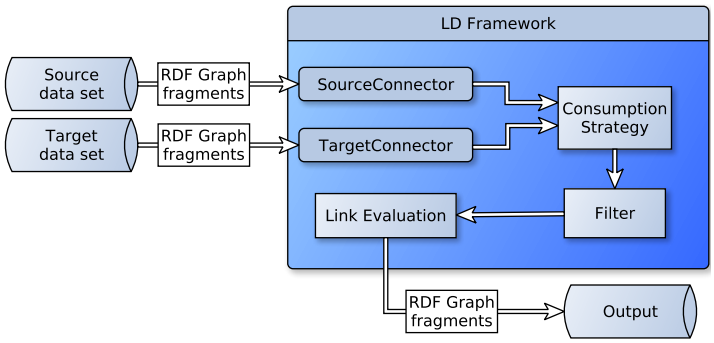

Fig. 1 High-level architecture of the spatiotemporal link discovery framework stLD.

the candidate pairs need to be examined to check if they satisfy the relation of interest. The methods presented in this chapter assuming an equi-grid, i.e., a grid constructed by cells of equal size. The blocking method is responsible to detect the cells that enclose the spatial representation of a given entity. Doing so, links are to be investigated only between entities blocked in the same cell.

Consider a spatial representation σ in a data source S of an entity that overlaps with one or more cells and k spatial representations $\{\tau_1, \ldots, \tau_k\}$ in the data source T that also overlap with the given cell. Our observation is that if σ is disjoint to all the spatial representations in $\{\tau_1, \ldots, \tau_\kappa\}$ in this cell, then we can safely infer that there are no topological relations (with the exception of "disjoint") to be discovered in this cell between σ and any spatial representation in $\{\tau_1, \ldots, \tau_\kappa\}$. Motivated by this observation, we propose in this chapter the *MaskLink technique* to explicitly represent the empty space within cells as yet another spatial representation. Thus, for each grid cell c, we construct an artificial polygon called *mask* of c, which is defined as the difference between the cell c and the union of spatial representations $\{\tau_1, \ldots, \tau_\kappa\}$ overlapping with the cell. We also show that by having the mask of a cell as yet another spatial representation, we can devise an efficient algorithm for link discovery that eagerly avoids comparisons to geometries for spatial representations enclosed in the mask of a cell. For the typical case where a cell contains several spatial representations, this technique prunes several candidate pairs of entities, saving computational time in the refinement step of the LD process. The MaskLink technique is applicable also for link discovery of proximity relations between spatial representations of entities in S and T.

We also discuss an improvement on the blocking method, namely the refined blocking, where the minimum necessary set of cells is computed that are needed to cover any given polyline, w.r.t. the granularity of the grid. The general idea of the method is to compute the cells for each point of the polyline, and for the case of closed geometries, the cells for all the interior points. This approach will provide the minimum necessary cells to cover the polyline (the proof is trivial). A naive implementation would be to iterate all the points of the given polyline (including the interpolated points with respect to the grid granularity) and compute

the corresponding cells. This method however would result to linear complexity to the number of points in a polyline, which will be costly for polylines consisting of many points.

The proposed "refined blocking" method recursively segments the polyline and terminates when both ends of the segment are within the same cell. This method has linear complexity to the number of cells that will be used for the geometry with respect to the grid granularity. The cost of refined blocking is not higher than the cost of using simply the MBR of the geometry, since the computation of MBR iterates through all the points of the geometry to decide the minimum and maximum latitude/longitude values of the MBR. On the other hand, the proposed method has a worst-case scenario of iterating all the points (i.e., the case where each point of the geometry should be placed in a separate cell with respect to the grid granularity).

This chapter reports experimental results on topological and proximity relations using the methods presented. We compare the MaskLink technique for all topological relations to RADON using the data sets CLC and NUTS. RADON requires that the entire data sets are loaded in memory, which may not always be possible for all use cases of LD. We overcome this memory limitation, by loading NUTS (as the smaller data set) in memory, and accessing CLC in batches of lines. For a fair comparison of techniques, we repeat the same procedure for MaskLink, although it can be directly applied on the given data sources.

We observe that the MaskLink is using less memory compared to RADON and it consistently outperforms RADON. When comparing MaskLink to a typical grid based implementation (namely the baseline), we observe that the gain achieved by MaskLink increases with the size of the data source, indicating that large instances of the problem cannot be efficiently solved by the baseline algorithm.

Regarding proximity relations, MaskLink outperforms the baseline consistently, but this time by a much larger margin, since the problem of proximity link discovery is harder in general. It is very significant to notice that the baseline is not scalable for this problem, as it does not terminate in reasonable time when the input size is larger than 1,500K. In contrast, MaskLink scales gracefully with the size of input data.