

Computational Logic in GeoPKDD*

Fosca Giannotti¹, Alessandra Raffaetà², Chiara Renso¹

¹ KDD LAB - ISTI CNR - Pisa - {giannotti, renso}@isti.cnr.it

² Dipartimento di Informatica - Università Ca' Foscari Venezia - raffaeta@dsi.unive.it

Abstract. The rapidly growing collections of privacy-sensitive telecommunication data from mobile phones and other location-aware devices, are enabling new classes of applications. The space-time trajectories of the personal mobile devices and their human companions offer interesting practical opportunities to find behavioural patterns, to be exploited for several challenging applications. In this setting the MIUR project GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery) has been proposed with the aim of devising a novel geographic privacy-preserving knowledge discovery process. This paper describes the context and the objectives of the GeoPKDD project, focusing on the role played by computational logic inside such a new discovery process.

1 Introduction and Motivation

Spatio-temporal datasets are, and will be, growing rapidly, in particular, due to the collection of telecommunication data from mobile phones and other location-aware devices, as well as the daily collection of transaction data through database systems, network traffic controllers, web servers, sensors. The large availability of these forms of geo-referenced information is expected to enable novel classes of applications, where the discovery of consumable, concise, and applicable knowledge is the key step. As a distinguishing example, the presence of a large number of location-aware wirelessly connected mobile devices presents a growing possibility to access space-time trajectories of these personal devices and their human companions: trajectories are indeed the traces of moving objects and individuals. These trajectories contain detailed information about personal and vehicular mobile behaviour, and therefore offer interesting practical opportunities to find behavioural patterns, to be used for instance in traffic and sustainable mobility management, e.g., to study the accessibility to services. Clearly, in these applications privacy is a concern. As a prototypical application scenario, assume that source data are log transactions from mobile cellular phones, reporting user's movements among the cells in the network; these are streams of raw data (log entries) about users entering a cell – $(userID, time, cellID)$ – or, in the near future, even user's position within a cell – $(userID, time, cellID, X, Y)$ and, in the case of GPS/Galileo equipped devices, user's absolute position.

In this context, the MIUR project GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery) has been proposed with the aim of devising a novel geographic privacy-preserving knowledge discovery process. Such a process, illustrated in Fig. 1, consists of three main steps: trajectories reconstruction, knowledge extraction, and interpretation and delivery of the obtained information.

* This work has been partially supported by the MIUR Italian Project GeoPKDD.

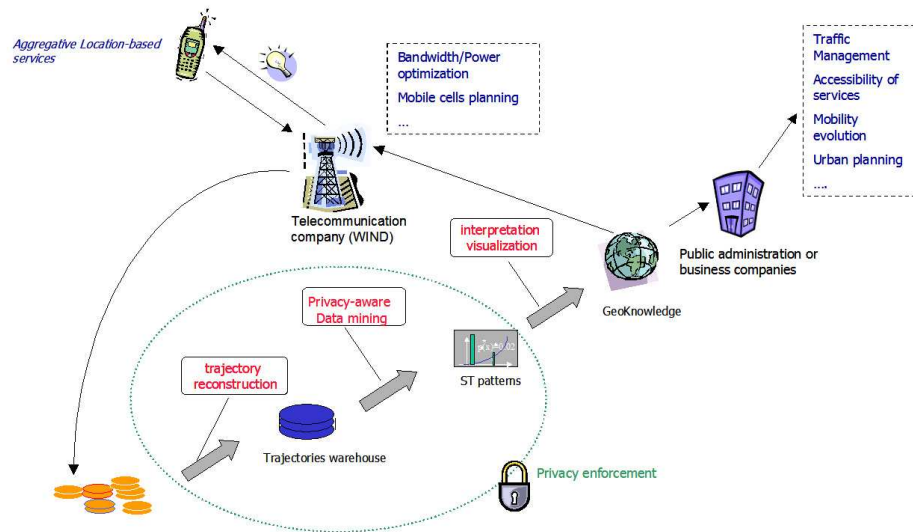


Fig. 1. The GeopKDD process

Trajectory reconstruction. In this basic phase, a stream of raw data about moving people has to be processed to obtain ready-to-use trajectories, building a privacy-aware trajectories warehouse. Reconstruction of trajectories is *per se* a challenging problem. The reconstruction accuracy of trajectories, as well as their level of spatio-temporal granularity, depend on the quality of the log entries, since the precision of the position may range from the granularity of a cell of varying size to the relative (approximated) position within a cell. Indeed, each moving object trajectory is typically represented as a set of localisation points of the tracked device, called *sampling*. This representation has intrinsic imperfection due to mainly two aspects. The first source of imperfection is the error measurement of the tracking device. The second one is related to the sampling rate and involves the trajectory reconstruction process that approximates the movement of the object between two localisation points. Although for some application, linear interpolation can be an acceptable approximation of the real trajectory, we believe that this could be a too coarse approximation and more sophisticated techniques are to be investigated to take into account the spatial, and possibly temporal, imperfection in the reconstruction process.

Knowledge extraction. Spatio-temporal data mining methods must be developed to extract useful patterns out of trajectories. Spatio-temporal data mining is still in its infancy ([1, 3, 7, 13, 14]), and even the most basic questions in this field are still largely unanswered: what kinds of patterns can be extracted from trajectories? Which methods and algorithms should be applied to extract them? How can such patterns be effectively used to improve the comprehension of the application domain and to deliver better

services? How can privacy be guaranteed? The following basic examples give a glimpse of the wide variety of patterns and possible applications it is expected to manage:

- *Clustering*, the discovery of groups of “similar” trajectories, together with a summary of each group (see Fig. 2.(a)). Knowing which are the main routes (represented by clusters) followed by people during the day can represent a precious information for improving several different services to citizens. E.g., trajectory clusters may highlight the presence of important routes not adequately covered by the public transportation service.
- *Frequent patterns*, the discovery of frequently followed (sub)-paths (see Fig. 2.(b)). Such information can be useful in urban planning, e.g., by spotlighting frequently followed inefficient vehicle paths, which can be the result of a mistake in the road planning.
- *Classification*, the discovery of behaviour rules, aimed at explaining the behaviour of current users and predicting that of future ones (see Fig. 2.(c)). Urban traffic simulations are a straightforward example application for this kind of knowledge, since a classification model can represent a sophisticated alternative to the simple ad hoc behaviour rules, provided by domain experts, on which actual simulators are based.

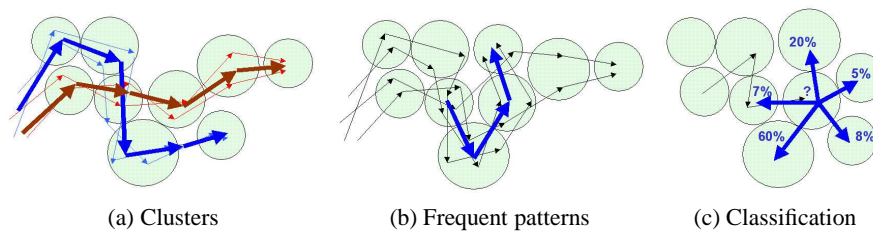


Fig. 2. Examples of extracted patterns.

Knowledge delivery. Extracted patterns are very seldom geographic knowledge ready-to-use: it is necessary to reason on patterns and on pertinent background knowledge, evaluate patterns interestingness, refer them to geographic information, find out appropriate presentations and visualisations.

We believe that a computational logic component could be profitably used as knowledge representation formalism and inference engine supporting reasoning.

In the next section we show some preliminary ideas on the role computational logic techniques play in the realisation of the knowledge interpretation and reasoning functionalities. Privacy issues will not show up in our discussion, since, according to the project plan, we think that they should be mainly addressed in the construction of the trajectory warehouse and in the definition of the spatio-temporal data mining algorithms.

2 Computational Logic for Knowledge Management

The availability of a suitable query language is clearly fundamental for a profitable use of data. Current (geographical) information systems, e.g., Oracle or ArcGIS, allow the user to exploit some consolidated mechanisms such as SQL-like queries, statistical functions and some spatial operations, thus offering an efficient and mature technology to query standard relational or spatial data. However they do not provide high level operations to handle the time and space related dimensions: nowadays the development of a spatio-temporal application with commercial systems often requires the programming of some ad-hoc components by means of procedural languages provided by the system itself for customisation purposes. The absence of specific spatio-temporal operations makes the customisation of applications extremely complex.

In order to provide a more efficient and flexible treatment of spatio-temporal data, several attempts to exploit the deductive capabilities of logics to reason on geographic data have been done in the literature [15, 11, 4, 2, 5]. Our research activity, briefly outlined below, follows this line.

The language. We defined the language STACLP [9], a constraint logic programming language with spatial and temporal annotations. The pieces of spatio-temporal information are given by pairs of annotations which specify the spatial extent of an object or of a property at a certain time period. The use of annotations makes time and space explicit while avoiding the proliferation of spatial and temporal variables and quantifiers. Annotated formulae come with inference rules that allow to combine annotations in order to derive new information. Moreover, STACLP supports both definite and indefinite spatial and temporal information. Definite information is modelled by $\text{atp}(X, Y)$, $\text{at } T$, $\text{thr}[(X_1, X_2), (Y_1, Y_2)]$ and $\text{th}[T_1, T_2]$ annotations specifying, respectively, the spatial and time point, the spatial region and the time interval in which a property holds. On the other hand, indefinite information is expressed by $\text{inr}[(X_1, X_2), (Y_1, Y_2)]$ and $\text{in}[T_1, T_2]$ annotations, expressing the fact that there are some point(s) in a given spatial region or temporal interval - which ones may not be known - in which a property holds. It is worth noticing that time and space can be discrete or dense, it depends on the application domain. Finally, STACLP allows to establish a dependency between space and time, thus permitting to model continuously moving points and regions.

Example 1. A moving point can be modelled easily by using a clause of the form:

$$\text{moving_point atp}(X, Y) \text{ at } T \leftarrow \text{constraint}(X, Y, T)$$

For instance, consider a car running on a straight road with speed v and assume that its initial position at time t_0 is (x_0, y_0) . The moving car can be represented by the clause

$$\text{car_position atp}(X, Y) \text{ at } T \leftarrow X = x_0 + v(T - t_0), Y = y_0 + v(T - t_0)$$

whose body “computes” in the obvious way, the position (X, Y) of the car at time T .

Such a language offers facilities for handling spatio-temporal information and has been used successfully as a high level interface on the top of GIS [10, 5]. Rules can be used to represent general knowledge about the collected data (background knowledge),

and deductive capabilities can provide answers to queries that require some inference besides the crude manipulation of the data. In addition, as shown e.g. in [8] the language can reconcile both deductive and inductive inference mechanisms. Induction can help extracting implicit knowledge from data and, according to the impressive success in the knowledge discovery in the database field, it can provide a powerful support to decision making.

Trajectories. In STACLP, starting from a set of spatio-temporal locations and by using linear interpolation, an approximation of trajectories can be easily modelled: the location points are represented by means of atp/at annotations and then the straight line between the two end points is expressed as a constraint.

Specifically, a set of localisations of each object o , say (x_i, y_i, t_i) for $i = 1, \dots, N$, can be represented by the following N STACLP facts:

```
fix(o) atp (x1, y1) at t1
fix(o) atp (x2, y2) at t2
:
fix(o) atp (xN, yN) at tN
```

Such localisations define the *core* of the trajectory of object o , which is then *completed* by defining all the intermediate points through linear interpolation using the following STACLP rules:

```
traj(o) atp (X, X) at T :- fix(o) atp (X, Y) at T.
traj(o) atp (X, Y) at T :- fix(o) atp (X1, Y1) at T1,
                           fix(o) atp (X2, Y2) at T2,
                           succ(T1, T2), T1 < T < T2,
                           X=(X1(T2-T)+X2(T-T1))/(T2-T1),
                           Y=(Y1(T2-T)+Y2(T-T1))/(T2-T1).
```

In the body of the second rule, approximate points (x, y) are computed by using the equation for the line passing through two given points. The presence of the (standard) *successor* predicate *succ*, defined as true for all and only the couples of (strictly) consecutive location points, ensures that no other observation exists between times t_1 and t_2 , i.e., the interpolation is performed only between consecutive location points.

Uncertainty. In applications of geographical nature, data rarely come truly free of errors: imperfection in some form or another is an endemic feature of geographic information. In particular, trajectories of moving objects - our data of interest - may not be completely specified or identified with sufficient accuracy (e.g. we may only know that a given object crossed an area, without knowing the exact path followed while crossing such an area). A representative case is given by the *log* transactions: each entry just represents the presence of the device in the cell, thus identifying each position of the moving object with a degree of imprecision. This feature can be immediately represented in STACLP exploiting the *in* annotation for indefinite spatial localisations.

A further challenging issue to investigate is the possibility of extending our framework to cope with different aspects of uncertainty in spatio-temporal data (i.e. degrees of certainty, fuzzy and rough sets). Following a typical idea in the related literature (for

a survey, see [12]), a possible way to deal with uncertain data in our formalism consists of considering a lattice of annotations that model different degrees of certainty (e.g., the interval $[0,1]$ with the usual ordering is the lattice commonly used for fuzzy logic).

Reasoning Architecture for GeoPKDD. The knowledge interpretation phase can be intended as a reasoning step that takes place only after the extraction of patterns from raw data (see Fig. 1). The rules apply to patterns, possibly combining them with background knowledge, to perform some kind of high level reasoning. However, it can be convenient to see, more generally, knowledge interpretation as an activity that applies at various steps of the discovery process, from raw data and trajectories representation, to extracted patterns and background knowledge, possibly to final end-user application.

In this sense, an architecture which looks promising combines the efficiency of data management systems (i.e., DBMS, trajectories warehouse or Geographical Information System), with the flexibility and the expressive power of computational logic (e.g., STACLPL) in a two-layered structure, as illustrated in Fig. 3. The data of interest (log transactions, trajectories, patterns) are exploited at two levels: they can be kept in the data management systems or exported at the logical level. The reasoning, logic-based layer can ask the data management systems to access data and to perform some demanding operations on data, e.g. visualisation or data mining operations, which are offered as primitive (and thus implemented efficiently) by such systems. The data stored in the lower layer can be exported to the logic-based layer in order to perform complex, deductive and/or inductive reasoning functionalities, which are not supported by standard data management systems. The strategy will be to exploit the efficiency of the data management systems as much as possible, exporting data into the logical representation only if strictly necessary to perform analysis not provided by the underlying systems.

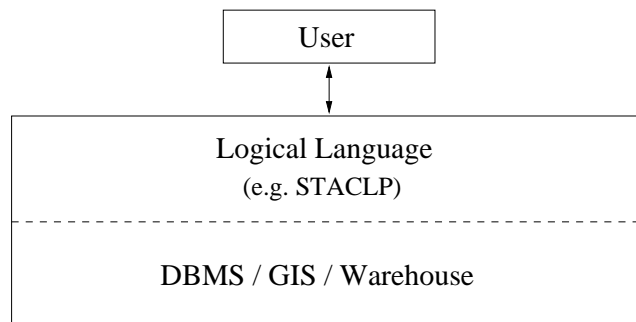


Fig. 3. Architecture of the system.

As an example, in the case of our project dealing with trajectories, the data management system could include the *trajectory warehouse* and offer some (spatio-temporal) data mining primitives. The logical layer could make use of STACLPL, enriched with a collection of primitive predicates, e.g., *trajectory(Tr_id, prop)* which calls the trajectory warehouse in order to obtain suitably identified trajectories, or *cluster(dataset, patterns)*

which calls an external data mining algorithm, getting back the extracted patterns. Trajectories could be exported and represented at the logic layer, using the language STA-CLP as already discussed.

This two-layered architecture has been already experimented in [5] where the logical language is on top of a GIS. A similar approach is followed also in [6], where the deductive database language *LDL++* is extended to deal with data mining primitives implemented as external calls. We believe that these approaches can represent a starting point for the definition of the reasoning architecture for GeoPKDD.

References

1. T. Abraham. *Knowledge Discovery in Spatio-Temporal Databases*. PhD thesis, School of Computer and Information Science, Faculty of Information Technology, University of South Australia, 1999.
2. J. Chomicki and P.Z. Revesz. Constraint-Based Interoperability of Spatiotemporal Databases. *GeoInformatica*, 3(3):211–243, 1999.
3. S. Gaffney and P. Smyth. Trajectory clustering with mixture of regression models. In *KDD Conf.*, pages 63–72. ACM, 1999.
4. S. Grumbach, P. Rigaux, and L. Segoufin. Spatio-Temporal Data Handling with Constraints. *GeoInformatica*, 5(1):95–115, 2001.
5. P. Mancarella, A. Raffaetà, C. Renso, and F. Turini. Integrating Knowledge Representation and Reasoning in Geographical Information Systems. *International Journal of GIS*, 18(4):417–446, June 2004.
6. G. Manco. *Foundations of a Logic-Based Framework for Intelligent Data Analysis*. PhD thesis, Dipartimento di Informatica, University of Pisa, 2001.
7. M. Nanni. *Clustering Methods for Spatio-Temporal Data*. PhD thesis, Dipartimento di Informatica, Università di Pisa, 2002.
8. M. Nanni and F. Turini A. Raffaetà, C. Renso. A Declarative Framework for Reasoning on Spatio-Temporal Data. In *Spatio-Temporal Databases. Flexible Querying and Reasoning*, pages 75–104. Springer, 2004.
9. A. Raffaetà and T. Frühwirth. Spatio-Temporal Annotated Constraint Logic Programming. In *PADL'01*, volume 1990 of *LNCS*, pages 259–273. Springer, 2001.
10. A. Raffaetà, C. Renso, and F. Turini. Enhancing GISs for Spatio-Temporal Reasoning. In *ACM GIS'02*, pages 35–41. ACM Press, 2002.
11. S. Spaccapietra, editor. *Spatio-Temporal Data Models & Languages (DEXA Workshop)*. IEEE Computer Society Press, 1999.
12. V. S. Subrahmanian. Uncertainty in Databases and Knowledge Bases. In *Advanced database systems*. Morgan Kaufmann Publishers Inc., 1997.
13. N. Sumpter and A. Bulpitt. Learning spatio-temporal patterns for predicting object behaviour. *Image and Vision Computing*, 18(9):697–704, 2000.
14. I. Tsoukatos and D. Gunopulos. Efficient mining of spatiotemporal patterns. In *Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD 2001*, volume 2121 of *LNCS*, pages 425–442, 2001.
15. M. F. Worboys. *GIS - A Computing Perspective*. Taylor & Francis, 1995.