Interleaved Inductive-Abductive Reasoning for Learning Event-Based Activity Models

K. Dubba[1], M. Bhatt[2], F. Dylla[2], A.G. Cohn[1] and D.C. Hogg[1]

[1]School of Computing, University of Leeds, UK
[2]SFB/TR 8 Spatial Cognition, University of Bremen, Germany

Abstract. We propose an interleaved inductive-abductive model for reasoning about complex spatio-temporal *narratives*. Typed Inductive Logic Programming (Typed-ILP) is used as a basis for learning the domain theory by *generalising from observation data*, whereas abductive reasoning is used for noisy data correction by *scenario and narrative completion* thereby improving the inductive learning to get semantically meaningful event models. We apply the model to an *airport domain* consisting of video data for 10 turn-arounds.

Introduction

Behaviour interpretation and activity analysis from real-time video and other forms of sensory data has become the cornerstone of many application domains within the purview of areas such as Smart Environments and Cognitive Vision. A fundamental requirement within such application domains is the representation of dynamic knowledge pertaining to the spatial aspects of the environment within which an agent, system, or robot is functional. As the mutual interactions amongst the agents in these domains happen in *space* and also have associated *temporal extensions*, one may use spatio-temporal relations to model the processes represented by such interactions. Specifically, activities can be modelled as sets of spatio-temporal relations, obtained from sensory data, that hold during particular intervals.

General Inductive Logic Programming (ILP) [10] methods assume correct and noise-free data. In contrast, data from visual and other sensors tend to be noisy with high variability in the sample space. This leads to *over-fitted* models (i.e., more rules), as the model has to cover all the examples. A model with more rules can result in many false positives when used as a basis for event-recognition with test data.

We show how well-fitted, semantically meaningful event models can be learned from noisy data by interleaving induction and abduction. This acquires significance in cases where training data is scarce and noisy. The framework we present in this paper is generic and has been applied on events in a very challenging domain; a corpus of real video data that captures the events that occur on an *airport apron*, a pre-designated area at airports where logistical processes such as arrival, loading, unloading, or departure occur. Our evaluation and demonstration focusses on the synergy afforded by the inductive-abductive cycle, whereas our proposed model provides a blue-print for interfacing common-sense reasoning about space, events and dynamic spatio-temporal phenomena with quantitative techniques in activity recognition.

Related Work

Integrating induction-abduction has been attempted previously in different contexts. Abduction has been used in machine learning for *theory revision* where an initial theory and consistent sets of positive and negative examples are given and the theory is then revised to fit the instances [9]. In our framework, the examples themselves are noisy (i.e. incorrect) thereby requiring *observation data revision* in a manner that is consistent with the initially learned theory, and general common-sense knowledge about *space*, *spatial change*, and the dynamics of the domain.

Induction and abduction are also integrated in a logic programming context [7] where abductive derivability is used instead of deductive derivability. This approach requires the conjunction of examples to be covered instead of each example separately to ensure that the abductive explanations for different examples are consistent with each other, whereas in our approach each example is independent of each other (i.e. the *learning from interpretations* setting [2], where each example is a separate database) and can be dealt separately. Also the correctness of the learned model is relaxed as we do not require the learned model to not cover any negative examples. Note that many ILP approaches discard examples considering them as noisy by using a heuristic stopping criteria. This is not acceptable in cases where there is scarcity of training data, where learning from every example is important. In the present framework, we avoid learning from them by reasoning that they are corrupted and can easily be explained using the induced domain theory and commonsense spatio-temporal reasoning.

Commonsense, Space, and Change

Qualitative Spatial & Temporal Representation and Reasoning (QSTR) provides a commonsensical interface to abstract and reason about quantitative spatial information. *Qualitative spatial / temporal calculi* are relational-algebraic systems pertaining to one or more aspects of space such as *topology, orientation, direction, size* [4]. The basic tenets in QSTR consist of constraint based reasoning algorithms over an infinite (spatial) domain to solve *consistency* problems in the context of spatial calculi. The key idea here is to partition an infinite quantity space in finite disjoint categories, and utilise the special relational properties of such a partitioned space for reasoning purposes.

In order to pursue our goal, an **Axiomatic Characterisation of the Spatial Theory** is necessary. Many spatial calculi exist, each corresponding to a different aspect of space. Here, it suffices to think of one spatial domain, e.g., topology, with a corresponding mereotopological axiomatisation by way of the binary relationships of the RCC-8 fragment $\mathcal{R}_{rcc8}$. From an axiomatic viewpoint, a spatial calculus defined on $\mathcal{R}$ has some general properties (P1–P5), which can be assumed to be known apriori. To realize a domain-independent spatial theory that can be used for reasoning (e.g., spatio-temporal abduction) across dynamic domains, it is necessary to formalize a domain-independent spatial theory (Σ_{space}) which preserves the high-level axiomatic semantics of these generic properties. For reasons of space, we only sketch the properties P1–P5 and neglect the formal axiomatization.

(P1–P2) The **Basic Calculus Properties** (Σ_{cp}) describe the *jointly exhaustive & pairwise disjoint* (JEPD) property, i.e., for any two entities in $\mathcal{O}$, one and only one spatial relationship from $\mathcal{R}$ holds in a given situation. The JE property of $n = |\mathcal{R}|$ base relations can be axiomatized by n ordinary state constraints and, similarly, the PD property can be axiomatized by $[n(n - 1)/2)]$ constraints. Other miscellaneous properties such as symmetry & asymmetry can be expressed in the same manner.

(P3) The primitive relationships in $\mathcal{R}$ have a *continuity structure*, referred to as its **Conceptual Neighbourhood** (Σ_{cn}) (CND), which determines the direct, continuous changes in the quality space (e.g., by deformation and / or translational motion).

(P4) From an axiomatic viewpoint, a spatial calculus defined on $\mathcal{R}$ is (primarily) based on the derivation of a set of **Composition Theorems** (Σ_{ct}) between the JEPD set $\mathcal{R}$. In general, for a calculus consisting of n JEPD relationships, $[n \times n]$ compositions are precomputed. Each of these composition theorems is equivalent to an ordinary state constraint, which every n-clique spatial situation description should satisfy.

(P5) Additionally, **Axioms of Interaction** (**Σ_{ai}**) are necessary when more than one spatial calculus is modelled in a non-integrated manner (i.e., with independent composition theorems). These axioms explicitly characterize the relative entailments between inter-dependent aspects of space, e.g., topology and size.
Now, let $\Sigma_{space} \equiv_{def} [\Sigma_{cp} \cup \Sigma_{cn} \cup \Sigma_{ct} \cup \Sigma_{ai}]$ denote a domain-independent spatial theory that is based on the axiomatisations encompassing (P1–P5).

Physically Plausible Scenarios. Corresponding to each spatial situation (e.g., within a hypothetical situation space), there exists a situation description that characterizes the spatial state of the system. It is necessary that the spatial component of such a state be a 'complete specification', possibly with disjunctive information. For k spatial calculi being modelled, the initial situation description involving m domain objects requires a complete n-clique specification with $[m(m-1)/2]$ spatial relationships for each calculus. In the following, we need to define a scene description to be **$\mathcal{C}$-Consistent**, i.e., compositionally consistent, if the n-clique state or spatial situation description corresponding to the situation satisfies all the composition constraints of every spatial domain (e.g., topology, orientation, size) being modelled. If more than one calculus is modelled the inter-dependent constraints (P5) must hold as well.
From the viewpoint of model elimination of narrative descriptions during an (abductive) explanation process, *$\mathcal{C}$-Consistency* of scenario descriptions is a key (contributing) factor determining the commonsensical notion of the *physically realizability* of the (abduced) scenario completions. Bhatt and Loke [1] show that a standard completion semantics with *causal minimization* in the presence of frame assumptions and ramification constraints preserves this notion of $\mathcal{C}$-Consistency for Σ_{space} within a logic programming framework, as well as with arbitrary basic actions theories.

The Inductive-Abductive Framework

We interleave inductive and abductive commonsense reasoning about space, events and change within a logic programming framework. Induction is used as a means to learn event models by generalizing from sensory data, whereas abductive reasoning is used for noisy data correction by *scenario and narrative completion* thereby improving the learning.

Theory Formation by Induction When the data is relational in nature, it is natural to use ILP for learning models. From among the different learning settings available in ILP, we use the *learning from interpretations* setting when each example is a separate database in itself; this is compatible with most event learning scenarios, where each example is independent of other examples. This learning from interpretations approach has been found to be very efficient for large scale problems [2].

Explanation by Abduction Diametrically opposite to projection and planning is the task of post-dictum or explanation, where given a set of time-stamped observations or snap-shots, the objective is to explain which events and/or actions may have caused the observed state-of-affairs. Explanation problems demand the inclusion of a narrative description, which is essentially a distinguished course of actual events about which we may have incomplete information [8]. Narrative descriptions are typically available as sensory *observations* from the real execution of a system or process. Given narratives, the objective is often to assimilate / explain them with respect to an underlying process model and an approach to derive explanations.
The *abductive explanation* problem can be stated as follows [6]: ***Given***: Theory T, observations G, find an explanation $\triangle$ such that: $T \cup \triangle \models G$ and $T \cup \triangle$ is consistent, i.e. the observation follows logically from the Theory extended with the explanation.

Given incomplete narrative descriptions, e.g., corresponding to only some ordered time-points in terms of high-level spatial (e.g., topological, orientation) and occurrence

Data: E^+, E^-: training sets
B: Background Knowledge
Result: H: learned theory
begin
  $H \longleftarrow \emptyset$
  $\triangle \longleftarrow \emptyset$
  while $E^+ \neq \emptyset$ **do**
    $Rule \longleftarrow Induce(B, E^+, E^-)$
    $H \longleftarrow H \bigcup Rule$
    $E^+ \longleftarrow E^+ - E^+_{Rule}$
    $\triangle \longleftarrow Abduce(B, H, E^+)$
    $E^+ \longleftarrow E^+ - E^+_{\triangle}$

Alg. 1: Interleaved Induction and Abduction ($\mathcal{IIA}$)

```
Ex:1
   dis(arr_zone,obj(aircraft(obj45)),6661,7137).
   tch(arr_zone,obj(aircraft(obj45)),7138,29114).
   tch(arr_zone,obj(veh(light_veh(gpu(obj54)))),7154,8161).
   dis(arr_zone,obj(veh(heavy_veh(loader(obj2)))),749,30380).
Ex:2
   dis(arr_zone,obj(aircraft(obj68)),2342,2663).
   tch(arr_zone,obj(aircraft(obj68)),2664,29524).
Ex:3
   dis(arr_zone,obj(veh(light_veh(trolly(obj0)))),285,21494).
   tch(arr_zone,obj(aircraft(obj41)),4458,32404).
   tch(arr_zone,obj(veh(light_veh(trolly(obj24)))),1712,32405).
Ex:4
   dis(arr_zone,obj(aircraft(obj33)),2435,6987).
   tch(arr_zone,obj(veh(heavy_veh(loader(obj27)))),2197,2310).
   dis(arr_zone,obj(veh(heavy_veh(loader(obj27)))),2311,2645).
Ex:5
   dis(arr_zone,obj(veh(light_veh(trolly(obj61)))),5450,5621).
   tch(arr_zone,obj(veh(light_veh(trolly(obj61)))),5622,6007).
   tch(arr_zone,obj(veh(light_veh(trolly(obj0)))),4951,7133).
Ex:6
   dis(arr_zone,obj(aircraft(obj45)),6661,7137).
   tch(gpu_zone,obj(aircraft(obj45)),7138,8139)
   tch(arr_zone,obj(aircraft(obj45)),8140,29114).
```

Fig. 1: Examples

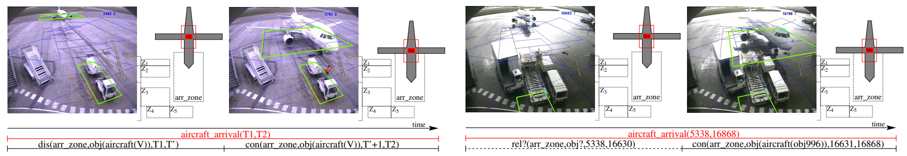
information, the objective of explanation is to derive one or more paths from the branching situation space, that could best-fit the available narrative information.
The abductive derivation of facts that explain how the scene changed from initial situation to the final situation, primarily involves non-monotonic reasoning in the form of minimising change, in addition to making the default assumptions about inertia, and an appropriate treatment of ramification constraints [1]. Abductive explanations are usually restricted to ground literals with predicates that are undefined in theory, namely the *abducibles*.

Interleaved Induction and Abduction ($\mathcal{IIA}$)

In general, ILP systems use a covering algorithm to learn models from examples. The search ranges over a hypothesis lattice and candidate hypotheses are evaluated based on the number of positive and negative examples it covers. Examples can be corrupted by noise resulting in missing and wrong facts. In such cases, more rules are learned than necessary in order to cover all the examples. As the number of rules for a concept increases, they may give many false positives when used for classification/recognition in test examples. In order to avoid learning from corrupt examples, we try to identify examples as being corrupted by explaining them through abduction using the already induced model as the background theory. The main assumption we make here is that the noise in the examples is not consistent.
The pseudo algorithm is given in Alg 1. Initially we start the induction algorithm that induces an initial hypothesis based on a score function that depends on the number of positive and negative examples covered and the length of the hypothesis. The positive examples covered by this hypothesis are removed from the list of positive examples yet to be covered. The induced theory along with background knowledge is used to try to explain the uncovered examples treating each example as a narrative. Abduction gives several possible explanations of different lengths (based on the number of abduced facts). The explanations are rejected if they have a length more than a specified threshold. Furthermore, given the formulation of the spatial theory Σ_{space}, $\mathcal{C}$-Consistency of abduced explanations is ensured. The examples with an explanation whose length is less than the specified threshold are removed from the positive examples list that are yet to be covered, as they are now considered to be covered by the already induced model. This process of induction and abduction is repeated until all the positive examples are covered.

Application, Implementation and Evaluation

The *airport apron* under consideration has been equipped with six cameras that record logistical activities from different angles. Altogether, 10 data sets each reflecting a turnaround have been used. Each video is on average 40 mins long ($50,000$ frames, 15 frames per sec). Tracking has been done on videos from six cameras separately and then fused to provide a ground plane interpretation. In our experiments, we apply a subset of the RCC-5 relations: disconnected (dis), touches (tch), and surrounds (sur). We specify



(a) Spatial primitive based event modelling

(b) Narrative completion (of data video) and previously learned model

Fig. 2: $\mathcal{IIA}$ Scenario and Narrative Completion; E.g., *aircraft_arrival*

zones on the apron area, that designate certain areas where events happen, in accordance with the International Air Transport Association (IATA) regulations. Prolog rules that implement a temporal interface to Allen interval relations are given as background information in the ILP system. The data for each video has on average 350 spatio-temporal facts. To provide positive and negative examples for an event for learning their models, we used deictic spatial and temporal terms to provide the boundaries of the event in the video [5]. These roughly specify when and where the event occurs in the video. These deictic terms are used to obtain the spatial relations among objects involved in the scene for that particular event, and this set of spatio-temporal facts forms the positive example; and the rest of the data is considered as a negative example for that event. The $\mathcal{IIA}$ algorithm is applied on the spatio-temporal data computed from the tracking data to learn event models and these models are used for recognizing the events in the test turnaround.

Induction and Abduction. In order to exploit the tree structured type hierarchy of agents involved in events, we exploit the Typed-ILP system [5] which uses a type refinement operator to traverse the hypothesis lattice while searching for a hypothesis. We use *Hyprolog*, a logic programming framework capable of abductive inference [3]. We have encoded the spatial theory Σ_{space} using this framework that contains the conceptual neighbourhood graph and the JEPD relationships of the spatial relations used. To explain our approach consider the following fragments of actually occurring data sets (`Ex:1` - `Ex:6` in Fig. 1) for the event *Aircraft Arrival*:
We obtain the following model for *Aircraft Arrival* event learned by the ILP approach from the first two examples of the given examples with *arr_zone* denoting a specific zone on the apron and any $\mathtt{T}_i$ denotes a time point.

```
aircraft_arrival(T1,T2,T3,T4) :-
  dis(arr_zone,obj(aircraft(V)),T1,T2),tch(arr_zone,obj(aircraft(V)),T3,T4),
  meets(T1,T2,T3,T4)).
```

This rule states that an *aircraft arrival* takes place if there is some interval an aircraft is disconnected to *arr_zone* directly followed by an interval, i.e *meets*, where the same aircraft is connected to *arr_zone*. This model does not cover any other examples apart from Ex:1 and Ex:2. Ex:3 has a missing *dis* relation related to the aircraft whereas Ex:4 has a missing *tch* relation (Fig. 2(b)). Ex:5 has a wrong type associated with the vehicle whereas in Ex:6 the zone is different in the *tch* relation. These represent the typical data corruption at a higher level because of tracking error at lower level video processing.

Narrative Completion If there are any missing facts in the narrative, the abduction system derives ground facts that are consistent with the domain and commonsense knowledge. If there are any wrong facts in the narrative, to come up with explanations, the abduction system would normally have to first derive a negated fact to say that the given fact is wrong and then has to derive the correct fact. With narrative completion, it is possible to cover all the examples given above, with one single Aircraft Arrival model learned. This avoids learning spurious rules to cover these corrupted examples thus giving us more semantically meaningful models.

To evaluate our approach, we compare the rules learned using only induction and rules learned using the $\mathcal{IIA}$ algorithm. The first column in Table 1 shows the events that we

Events	pos_ex	●	□	No. of e.g.s covered by abd.
Rear Loading	10	2	1	1
Rear Unloading	8	2	1	1
Aircraft Arrival	10	5	2	4
Aircraft Departure	10	4	2	3
GPU Arrival	6	3	1	2

● Num of rules with only Induction □ Num of rules with $\mathcal{IIA}$

Table 1: Integrated Induction-Abduction ($\mathcal{IIA}$) Results

considered for the experiments, the second column shows the number of instances of that particular event in the 10 turnarounds. The third column shows the number of rules learned by using only ILP while the fourth column shows the results using the $\mathcal{IIA}$ algorithm while fifth column shows the number of examples that were not covered by the induced rules but were explained using abduction and hence no rules learned from them. By interleaving induction and abduction we are able to avoid learning spurious rules as shown by the results. In most of the cases the number of rules are reduced by 50%. We also observed that the rules that could be learned from examples explainable by abduction do not semantically correspond to the events. Note that many ILP approaches discard examples considering them as noisy by using a heuristic stopping criteria while in our case we carefully avoid learning from them by reasoning that they are corrupted and can easily be explained using the induced domain theory and commonsense and spatio-temporal reasoning.

Conclusion and Outlook

With a formalization of a domain-independent spatial theory, and an inductive-abductive learning and reasoning cycle, we show how semantically meaningful event models can be learnt. By using abduction, we learn tight models in the presence of noisy data. This is very important in cases where there is scarcity of training data.
Remaining as future work, a way is needed to exploit the learned rules so that the noisy test data can be explained as being covered by the rules in a computationally efficient way. The abductive reasoning phase will benefit from probabilistic model ranking / filtering extensions, e.g., in the context of decision-theoretic approaches such as DTProbLog.

References

1. Mehul Bhatt and Seng Loke. Modelling dynamic spatial systems in the situation calculus. *Spatial Cognition and Computation*, 8(1):86–130, 2008.
2. H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3(1):59–93, 1999.
3. H. Christiansen and V. Dahl. HYPROLOG: A new logic programming language with assumptions and abduction. *Logic Programming*, pages 159–173, 2005.
4. A G Cohn and J Renz. Qualitative spatial reasoning. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.
5. Krishna S. R. Dubba, Anthony G. Cohn, and David C. Hogg. Event model learning from complex videos using ILP. In *ECAI*, pages 93–98, 2010.
6. A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of logic and computation*, 2(6):719, 1992.
7. E. Lamma, P. Mello, M. Milano, and F. Riguzzi. Integrating induction and abduction in logic programming. *Information Sciences*, 116(1):25–54, 1999.
8. Rob Miller and Murray Shanahan. Narratives in the situation calculus. *J. Log. Comput.*, 4(5):513–530, 1994.
9. Raymond J. Mooney. Integrating abduction and induction in machine learning. In P. A. Flach and A. C. Kakas, editors, *Abduction and Induction*, pages 181–191. Kluwer Academic Publishers, 2000.
10. S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13(3&4):245–286, 1995.