Predicting Knowledge in an Ontology Stream*

Freddy Lécué
IBM Dublin Research Center, Ireland

Jeff Z. Pan
University of Aberdeen, UK

Abstract

Recently, ontology stream reasoning has been introduced as a multidisciplinary approach, merging synergies from Artificial Intelligence, Database, World-Wide-Web to reason on semantic augmented data streams. Although knowledge evolution and real-time reasoning have been largely addressed in ontology streams, the challenge of predicting its future (or missing) knowledge remains open and yet unexplored. We tackle predictive reasoning as a correlation and interpretation of past semantics-augmented data over exogenous ontology streams. Consistent predictions are constructed as Description Logics entailments by selecting and applying relevant cross-streams association rules. The experiments have shown accurate prediction with real and live stream data from Dublin City in Ireland.

1 Introduction

The semantic web [Berners-Lee *et al.*, 2001] is considered to be the future of the current web. The semantics of information is represented using rich description languages e.g., OWL the *Web Ontology Language*, which is underpinned by Description Logics (DL) [Baader *et al.*, 2003] to define web ontologies. Their dynamic extension i.e., ontology stream [Ren and Pan, 2011], is used for representing knowledge evolution [Huang and Stuckenschmidt, 2005]. Enriched with inference models e.g., stream-based (i) querying [Calbimonte *et al.*, 2010] for real-time filtering, (ii) reasoning [Valle *et al.*, 2009] for interpreting evolution or (iii) diagnosing [Lécué, 2012] for explaining anomalies, ontology stream is envisioned to span many real world applications. From semantics-empowered sensors [Sheth, 2010], to social semantic web [Auer *et al.*, 2006], all are examples of applications where multiple, large and expressive ontology streams have an important role.

Predictive inference, as a reasoning technique for estimating future (or missing) observations in a stream given some historical information, has been largely studied in Statistics, Database and Artificial Intelligence [Härdle, 1992] but not in the context of semantics-enhanced data as envisioned in an ontology stream. For instance, prediction in (raw) data stream mining applications is estimated by correlating current and past data patterns using different distance metrics [Gehrke *et al.*, 2001] between (syntactic) numeric/symbolic values. Even if [Papadimitriou *et al.*, 2005] augmented such models with cross-correlation over streams, they all fail in using and interpreting the underlying semantics of data, making prediction not necessarily consistent and accurate, specially if streams are characterized by many major changes over time (concept drift). From a different perspective, [Bloehdorn and Sure, 2007] combine machine learning and semantic web for predicting class-membership of data by mining its instances in ontologies. This approach cannot be applied since data to be ”class”-ed is not known beforehand in an ontology stream.

Facing these limitations, we address the problem of predicting knowledge in an ontology stream. Given some continuous knowledge, how do we capture time-evolving trends and patterns in the stream to make accurate predictions? Our approach determines consistent patterns by inferring and mining rules across exogenous streams, where their DL-based semantic representations are captured and interpreted through some static background knowledge. Predictive reasoning is addressed by analyzing stream evolution through correlation of its changes and their inconsistencies. Predictions, modeled as DL entailments, are derived consistent by selecting and applying relevant cross-streams association rules. The significance of rules is evaluated with respect to their (i) context of applicability, (ii) support, (iii) confidence over time, and (iv) DL consistency of their prediction. The experiments have shown accurate prediction with real, live, stream (semantics augmented) data from Dublin City in Ireland.

The remainder of this paper is as follows. Section 2 briefly reviews the logic we adopt together with ontology stream. In Section 3 we study knowledge correlation over time and rules mining in streams. Section 4 presents our consistent knowledge prediction approach. Section 5 reports experiment results on scalability and accuracy. Section 6 briefly comments on related work. Finally, Section 7 draws some conclusions.

2 Background

Both static background knowledge and semantics of stream data are represented using an ontology. Dynamic knowledge

[*] The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement IDs 318201 (SIMPLI-CITY) and 286348 (K-Drive).

is then captured by reasoning on stream data descriptions in this ontology. We focus on DL as a formal knowledge representation language to define ontologies since this logic offers good reasoning support for most of its expressive families and compatibility to W3C standards e.g., OWL 2. We review (i) DL basics of $\mathcal{EL}^{++}$, (ii) ontology stream and its behavior.

2.1 $\mathcal{EL}^{++}$ Description Logics

We illustrate our work with DL $\mathcal{EL}^{++}$ [Baader *et al.*, 2005] where satisfiability and subsumption are decidable. The selection of this DL fragment, which is the logic underpinning OWL 2 EL and the basis of many more expressive DL, has been guided by (i) the expressivity which was required to model semantics of data in our transportation domain (Section 5), and (ii) the scalability of the underlying basic reasoning mechanisms required in our stream context.

A signature Σ, noted $(\mathcal{CN}, \mathcal{RN}, \mathcal{IN})$ consists of 3 disjoint sets of (i) atomic concepts $\mathcal{CN}$, (ii) atomic roles $\mathcal{RN}$, and (iii) individuals $\mathcal{IN}$. Given a signature, the top concept $\top$, the bottom concept $\bot$, an atomic concept A, an individual a, an atomic role r, $\mathcal{EL}^{++}$ concept expressions C and D can be composed with constructs:

$$\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$$

We slightly abuse the notion of atomic concepts to include $\top$, $\bot$ and nominals [Horrocks and Sattler, 2001] i.e., individuals appearing in concept definitions of form $\{a\}$.

The DL ontology $\mathcal{O} \doteq < \mathcal{T}, \mathcal{A} >$ is composed of a TBox $\mathcal{T}$ and ABox $\mathcal{A}$. A TBox is a set of concept and role axioms.

Example 1. ***(TBox Fragment)***
Figure 1 presents a TBox where e.g., $CongestedRoad$ (2) denotes the concept of "road with at least a bus in a heavy traffic". The $busStatus$ role (4) informs about the dynamics of buses e.g., stopped, or $inMove$.

$$\exists from.Area \sqcap to.Area \sqcap \exists travel.Bus \sqsubseteq Road \sqcap \exists with.Bus \tag{1}$$
$$Road \sqcap \exists with.(Bus \sqcap \exists congested.Heavy) \sqsubseteq CongestedRoad \tag{2}$$
$$Road \sqcap \exists with.(Bus \sqcap \exists congested.Light) \sqsubseteq FreeRoad \tag{3}$$
$$busStatus \circ stopped \sqsubseteq congested \text{ \% Roles composition} \tag{4}$$
$$Road \sqcap \exists travelTime.Normal \sqsubseteq Road \sqcap \exists with.OnTimeBus \tag{5}$$
$$CongestedRoad \sqcap FreeRoad \sqsubseteq \bot \text{ \% Incompatibility} \tag{6}$$
$$\{r_1\} \sqsubseteq Road \quad (7) \qquad \{r_2\} \sqsubseteq Road \quad (8) \qquad \{r_3\} \sqsubseteq Road \quad (9)$$

Figure 1: $\mathcal{EL}^{++}$ TBox Fragment $\mathcal{T}$ (with internalized ABox).

$\mathcal{EL}^{++}$ supports General Concept Inclusion axioms (GCIs, e.g. $C \sqsubseteq D$ with C is subsumee and D subsumer) and role inclusion axioms (RIs, e.g., $r \sqsubseteq s$, $r_1 \circ \cdots \circ r_n \sqsubseteq s$). An ABox is a set of concept assertion axioms e.g., $a : C$, role assertion axioms e.g., $(a; b) : R$, and individual in/equality axioms e.g., $a \neq b$ or $a = b$.

Table 1 sketches completion rules [Baader *et al.*, 2005] that are used to classify $\mathcal{EL}^{++}$ TBox $\mathcal{T}$ and entail subsumption for any concept in $\mathcal{CN}_{\mathcal{T}}$. Reasoning with such rules is PTime-Complete [Baader *et al.*, 2008a]. We internalize ABox into TBox axioms (along $\rightsquigarrow$) so (i) rules in Table 1 can be applied on both axioms, (ii) TBox reasoning (e.g., subsumption, satifiablility) can be performed on internalized ABox axioms.

$$a : C \rightsquigarrow \{a\} \sqsubseteq C \qquad (a, b) : r \rightsquigarrow \{a\} \sqsubseteq \exists r.\{b\}$$
$$a \doteq b \rightsquigarrow \{a\} \equiv \{b\} \qquad a \neq b \rightsquigarrow \{a\} \sqcap \{b\} \sqsubseteq \bot$$

Besides considering an internalized ABox, we assume that the $\mathcal{EL}^{++}$ TBox is normalized, and all subsumption closures are pre-computed [Baader *et al.*, 2005]. Due to limited space, we do not formally introduce the notions of interpretation, and entailment ($\models$) here.

R_1	If $X \sqsubseteq A,\ A \sqsubseteq B$ then $X \sqsubseteq B$
R_2	If $X \sqsubseteq A_1, \cdots A_n,\ A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$ then $X \sqsubseteq B$
R_3	If $X \sqsubseteq A,\ A \sqsubseteq \exists r.B$ then $X \sqsubseteq \exists r.B$
R_4	If $X \sqsubseteq \exists r.A,\ A \sqsubseteq A',\ \exists r.A' \sqsubseteq B$ then $X \sqsubseteq B$
R_5	If $X \sqsubseteq \exists r.A,\ A \sqsubseteq \bot$ then $X \sqsubseteq \bot$
R_6	If $X \sqsubseteq \exists r.A,\ r \sqsubseteq s$ then $X \sqsubseteq \exists s.A$
R_7	If $X \sqsubseteq \exists r_1.A,\ A \sqsubseteq \exists r_2.B,\ r_1 \circ r_2 \sqsubseteq r_3$ then $X \sqsubseteq \exists r_3.B$

Table 1: $\mathcal{EL}^{++}$ TBox Completion Rules (no datatypes).

2.2 Ontology Stream and its Evolution

We represent knowledge evolution by a dynamic, evolutive versions of ontologies in Definition 1 [Ren and Pan, 2011].

Definition 1. ***(Ontology Stream)***
An ontology stream $\mathcal{O}_m^n$ from point of time m to point of time n is a sequence of ontologies $(\mathcal{O}_m^n(m), \mathcal{O}_m^n(m+1), \cdots, \mathcal{O}_m^n(n))$ where $m, n \in \mathbb{N}$ and $m < n$.

$\mathcal{O}_m^n(i)$ is a snapshot of an ontology stream $\mathcal{O}_m^n$ (stream for short) at point of time i, referring to a set of axioms in a DL $\mathcal{L}$. A transition from $\mathcal{O}_m^n(i)$ to $\mathcal{O}_m^n(i+1)$ is an update. We will consider streams $\mathcal{O}_0^n$ for the sake of clarity and will assume all its snapshots to be consistent w.r.t. $\mathcal{T}$. Inconsistent snapshots generate special cases during the stream auto-correlation and mining processes in Section 3, easy to be handled but not described due to space restriction.

Example 2. ***(Ontology Stream)***
Figure 2 illustrates three partial ontology streams $\mathcal{O}_0^9$, $\mathcal{P}_0^9$ and $\mathcal{Q}_0^9$ through some snapshots at point of time $i \in \{6, 7, 8\}$.

$$\mathcal{O}_0^9(6) : \{bus7\} \sqsubseteq Bus \sqcap \exists busStatus.(\exists stopped.Light) \tag{10}$$
$$: \exists from.\{RockRd\} \sqcap \exists to.\{YorkRd\} \sqcap \exists travel.\{bus7\}$$
$$\sqsubseteq \{r_1\} \sqcap \exists with.\{bus7\} \tag{11}$$
$$\mathcal{P}_0^9(6) : \{r_1\} \sqsubseteq \exists adj.(\{r_2\} \sqcap \exists withWeather.HighHumidity) \tag{12}$$
$$\mathcal{Q}_0^9(6) : \{r_1\} \sqsubseteq \exists adj.(\{r_3\}) \sqcap \exists travelTime.Normal) \tag{13}$$
$$\mathcal{O}_0^9(7) : \{bus7\} \sqsubseteq Bus \sqcap \exists busStatus.(\exists stopped.Heavy) \tag{14}$$
$$: \exists from.\{RockRd\} \sqcap \exists to.\{YorkRd\} \sqcap \exists travel.\{bus7\}$$
$$\sqsubseteq \{r_1\} \sqcap \exists with.\{bus7\} \tag{15}$$
$$\mathcal{P}_0^9(7) : \{r_1\} \sqsubseteq \exists adj.(\{r_2\} \sqcap \exists withWeather.LowHumidity) \tag{16}$$
$$\mathcal{Q}_0^9(7) : \{r_1\} \sqsubseteq \exists adj.(\{r_3\} \sqcap \exists travelTime.Long) \tag{17}$$
$$\mathcal{O}_0^9(8) : \{bus7\} \sqsubseteq Bus \sqcap \exists busStatus.(\exists stopped.Heavy) \tag{18}$$
$$: \exists from.\{RockRd\} \sqcap \exists to.\{YorkRd\} \sqcap \exists travel.\{bus7\}$$
$$\sqsubseteq \{r_1\} \sqcap \exists with.\{bus7\} \tag{19}$$
$$\mathcal{P}_0^9(8) : \{r_1\} \sqsubseteq \exists adj.(\{r_2\} \sqcap \exists withWeather.HighHumidity) \tag{20}$$
$$\mathcal{Q}_0^9(8) : \{r_1\} \sqsubseteq \exists adj.(\{r_3\} \sqcap \exists travelTime.Long) \tag{21}$$

Figure 2: Ontology Streams $\mathcal{O}_0^9$, $\mathcal{P}_0^9$, $\mathcal{Q}_0^9$ at time $i \in \{6, 7, 8\}$.

By applying completion rules in Table 1 on both background knowledge $\mathcal{T}$ and some streams $\mathcal{O}_0^n$, we are able to infer axioms which are specific to some snapshots.

Example 3. ***(Subsumption Reasoning in Ontology Stream)***
(22), (23)*, as dynamic knowledge are entailed from $\mathcal{T}$ (Figure*

1), stream $\mathcal{O}_0^9$ (Figure 2), and rules in Table 1. For instance $\{r_1\}$ is entailed to be free in $\mathcal{O}_0^9(6)$; and congested in $\mathcal{O}_0^9(8)$.

$$\mathcal{T} \cup \mathcal{O}_0^9(6) \models^{(1),(3),(4),(10),(11)}_{R_1,R_2,R_3,R_7} \{r_1\} \sqsubseteq FreeRoad \qquad (22)$$

$$\mathcal{T} \cup \mathcal{O}_0^9(8) \models^{(1),(2),(4),(18),(19)}_{R_1,R_2,R_3,R_7} \{r_1\} \sqsubseteq CongestedRoad \qquad (23)$$

The evolution of a stream is captured along its changes i.e., *new*, *obsolete* and *invariant* ABox entailments from one snapshot to another one in Definition 2.

Definition 2. ***(ABox Entailment-based Stream Changes)*** *Let (i) $\mathcal{L}$ be a DL, (ii) $\mathcal{O}_0^n(i)$, $\mathcal{O}_0^n(j)$ be snapshots in $\mathcal{O}_0^n$, (iii) $\mathcal{T}$ be $\mathcal{L}$-axioms, $\mathcal{G}$ its ABox entailments. Stream changes occurring from $\mathcal{O}_0^n(i)$ to $\mathcal{O}_0^n(j)$ (where $i < j$), denoted by $\mathcal{O}_0^n(j)\nabla\mathcal{O}_0^n(i)$, are ABox entailments (entailments for short) in $\mathcal{G}$ being new ($\mathcal{G}_{new}^{i,j}$), obsolete ($\mathcal{G}_{obs}^{i,j}$), invariant ($\mathcal{G}_{inv}^{i,j}$).*

$$\mathcal{G}_{new}^{i,j} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup \mathcal{O}_0^n(j) \models g \ \wedge \mathcal{T} \cup \mathcal{O}_0^n(i) \not\models g\} \qquad (24)$$

$$\mathcal{G}_{obs}^{i,j} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup \mathcal{O}_0^n(j) \not\models g \ \wedge \mathcal{T} \cup \mathcal{O}_0^n(i) \models g\} \qquad (25)$$

$$\mathcal{G}_{inv}^{i,j} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup \mathcal{O}_0^n(j) \models g \ \wedge \mathcal{T} \cup \mathcal{O}_0^n(i) \models g\} \qquad (26)$$

$\mathcal{G}_{new}^{i,j}$ reflects knowledge we obtain by moving from $\mathcal{O}_0^n(i)$ to $\mathcal{O}_0^n(j)$ while $\mathcal{G}_{obs}^{i,j}$ denotes knowledge we lose. $\mathcal{G}_{inv}^{i,j}$ captures stability of knowledge. Definition 2, extending the *definition of change* in [Huang and Stuckenschmidt, 2005] to support ABox entailment-based changes, provides basics for understanding how knowledge is connected among snapshots.

Example 4. ***(ABox Entailment-based Stream Changes)*** *Table 2 illustrates changes (new, inv-ariant, obs-olete) occurring from $\mathcal{O}_0^9(i)_{6\leq i\leq 7}$ to $\mathcal{O}_0^9(8)$ through entailements $\{r_1\} \sqsubseteq FreeRoad$, $\{r_1\} \sqsubseteq CongestedRoad$. E.g., $\{r_1\}$ is a new $CongestedRoad$ in $\mathcal{O}_0^9(8)$ with respect to $\mathcal{O}_0^9(6)$.*

Snapshot	$\mathcal{O}_0^9(8)\nabla\mathcal{O}_0^9(7)$			$\mathcal{O}_0^9(8)\nabla\mathcal{O}_0^9(6)$		
Changes	*new*	*inv.*	*obs.*	*new*	*inv.*	*obs.*
$\{r_1\} \sqsubseteq FreeRoad$						✓
$\{r_1\} \sqsubseteq CongestedRoad$		✓		✓		

Table 2: ABox Entailment-based Stream Changes.

3 Ontology Stream Auto-Correlation & Mining

We present *auto-correlation* and *association rules*, as a basis for predicting knowledge in ontology streams.

3.1 Auto-Correlation in An Ontology Stream

Understanding correlations between a stream $\mathcal{O}_0^n$ and its own past is important for detecting knowledge similarity, repeats, periodicity, even divergence. Definition 3 revisits the concept of auto-correlation [Bracewell, 1999] in Signal Processing for capturing knowledge *similarity* at various point of time in $\mathcal{O}_0^n$.

Definition 3. ***(Auto-Correlation in An Ontology Stream)*** *Let (i) $\mathcal{L}$ be a DL, (ii) $\mathcal{T}$ be $\mathcal{L}$-axioms, (iii) $\mathcal{O}_0^n$ be a $\mathcal{L}$-ontology stream. The (symmetric) auto-correlation between time i and j in $[0,n]$ of $\mathcal{O}_0^n$, denoted by $A(\mathcal{O}_0^n(i), \mathcal{O}_0^n(j))$, is:*

$$\begin{cases} \dfrac{|\mathcal{G}_{inv}^{i,j}|}{|\mathcal{G}_{new}^{i,j}|+|\mathcal{G}_{inv}^{i,j}|+|\mathcal{G}_{obs}^{i,j}|} & \textit{if}\ \mathcal{T} \cup \mathcal{O}_0^n(i) \cup \mathcal{O}_0^n(j) \not\models \bot \\ \dfrac{|\mathcal{G}_{inv}^{i,j}|}{|\mathcal{G}_{new}^{i,j}|+|\mathcal{G}_{inv}^{i,j}|+|\mathcal{G}_{obs}^{i,j}|} - 1 & \textit{otherwise} \end{cases} \qquad (27)$$

where the expressions in between $|$ refer to its cardinality i.e., the number of new, obsolete and invariant ABox entailments obtained from $\mathcal{O}_0^n(i)$ to $\mathcal{O}_0^n(j)$ using rules in Table 1.

With values in $[-1, 1]$, (27) captures negative and positive correlations. The snapshots $\mathcal{O}_0^n(i)$ and $\mathcal{O}_0^n(j)$ are negatively correlated if they are inconsistent i.e., at least one concept has an empty extension in $\mathcal{O}_0^n(i) \cup \mathcal{O}_0^n(j)$ with respect to $\mathcal{T}$. The contradictions and their logical consequences make the correlation negative. Alternatively, they have a positive correlation i.e., $\mathcal{O}_0^n(i)$ and $\mathcal{O}_0^n(j)$ share some *"elements"* (i.e., ABox entailments) of knowledge. In both cases, the number of *invariant* entailments has a strong and positive influence on the auto-correlation. On contrary, the number of *new* and *obsolete* ABox entailments, capturing some differentiators in knowledge evolution, has a negative impact. When an inconsistency occurs, the value 1 is subtracted to (27) instead of considering its additive inverse. This ensures that the invariant factor has always a positive impact on autocorrelation.

Evaluating (27) is in worst case polynomial time with respect to $\mathcal{T}$ and $\mathcal{O}_0^n$ in $\mathcal{EL}^{++}$. Indeed its evaluation requires (i) ABox materialisation, and (ii) basic set theory operations from Definition 2, both in polynomial time [Baader *et al.*, 2005; 2008b].

Example 5. ***(Auto-Correlation in An Ontology Stream)*** *All entailments in $\mathcal{O}_0^9(8)$, $\mathcal{O}_0^9(7)$ and $\mathcal{O}_0^9(6)$, derived using Table 1, are required for evaluating (27) with $i, j \in \{6, 7, 8\}$. In the simple case of $\mathcal{O}_0^9(8)\nabla\mathcal{O}_0^9(6)$, we reach the number of 28 invariants, 16 new, 13 obsolete entailments, hence an auto-correlation of -0.508. The negative correlation, due to inconsistency of $\mathcal{T}\cup\mathcal{O}_0^9(6)\cup\mathcal{O}_0^9(8)$, is caused by (6), (22-23). The evaluation of $A(\mathcal{O}_0^9(7), \mathcal{O}_0^9(8))$, scoring 1, is simpler due to the similar knowledge at points of time 7 and 8 of $\mathcal{O}_0^9$.*

Remark 1. ***(Auto-Correlation Extensions)*** *It is straightforward to extend Definition 3 for capturing finer-grained levels of auto-correlation. For instance the degree of inconsistency, among others, could have been discriminated further by reporting the proportion of concepts with an empty extension. It consists in considering and weighting (27) with the number of inconsistent axioms over two snapshots. Such a modification may fit better some applications e.g., in case our capture of inconsistency is too loose.*

Given a point of time $j \leq n$, its set of correlated snapshots $\mathcal{O}_0^n(i)_{,i\in[0,n]\setminus\{j\}}$ can be compared and ordered by applying $\leq$ on their $A(\mathcal{O}_0^n(i), \mathcal{O}_0^n(j))$ (Figure 3). Thus the best correlated snapshots of $\mathcal{O}_0^n(j)$ can be retrieved with respect to (27).

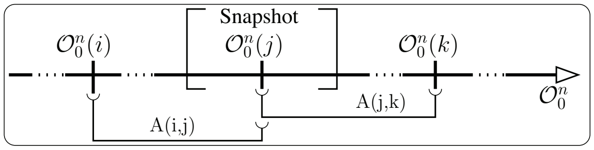


Figure 3: Illustration of Auto-Correlation in Stream.

Example 6. ***(Auto-Correlation Comparison and Ordering)*** *Following Example 5, $A(\mathcal{O}_0^9(7), \mathcal{O}_0^9(8)) > A(\mathcal{O}_0^9(6), \mathcal{O}_0^9(8))$ i.e., $\mathcal{O}_0^9(8)$ has a stronger auto-correlation with $\mathcal{O}_0^9(7)$.*

A similarity of auto-correlation scores does not necessarily mean similar knowledge. Indeed (27) does not reflect which invariant, new and obsolete entailments are shared.

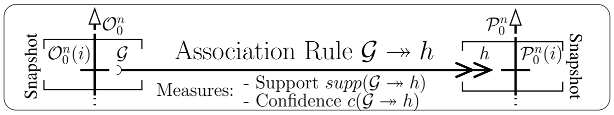
3.2 Association Rules Mining in Ontology Streams

Discovering interesting relations and rules between elements of knowledge in multiple streams is crucial for identifying how knowledge could be associated at a given point of time. Definition 4 revisits the concept of association rules [Agrawal *et al.*, 1993] in the context of ontology streams.

Definition 4. ***(Association Rules in Ontology Streams)*** *Let (i) $\mathcal{L}$ be a DL, (ii) $\mathcal{T}$ be $\mathcal{L}$-axioms, (iii) $\mathcal{O}_0^n$, $\mathcal{P}_0^n$ be $\mathcal{L}$-streams, (iv) $i \in [0, n]$ be a point of time, (v) $\mathcal{G}$ be axioms such that $\mathcal{T} \cup \mathcal{O}_0^n(i) \models \mathcal{G}$, (vi) h be an axiom such that $\mathcal{T} \cup \mathcal{P}_0^n(i) \models h$. $\mathcal{G} \twoheadrightarrow h$ is an association rule in $\mathcal{O}_0^n \times \mathcal{P}_0^n$ iff:*

$$\mathcal{T} \cup \mathcal{P}_0^n(i) \not\models \mathcal{G} \qquad (28) \qquad \mathcal{T} \cup \mathcal{O}_0^n(i) \not\models h \qquad (29)$$

Definition 4 identifies an association rule (Figure 4) as an *implication* relation between axioms $\mathcal{G}$ and h. (28-29) ensure that consequent h is "associated" with antecedent $\mathcal{G}$ in neither $\mathcal{O}_0^n$ nor $\mathcal{P}_0^n$ in at least one point of time $i \in [0, n]$. These constraints are important for scalability as they discard trivial rules i.e., associations which can be directly entailed by a snapshot of a unique stream: $\mathcal{T} \cup \mathcal{P}_0^n(i)$ or $\mathcal{T} \cup \mathcal{O}_0^n(i)$. These rules could be useful if snapshots have missing information.



$\mathcal{O}_0^n$
Snapshot
$\mathcal{O}_0^n(i)$
$\mathcal{G}$
Association Rule $\mathcal{G} \twoheadrightarrow h$
Measures: - Support $supp(\mathcal{G} \twoheadrightarrow h)$
- Confidence $c(\mathcal{G} \twoheadrightarrow h)$
$\mathcal{P}_0^n$
h
$\mathcal{P}_0^n(i)$
Snapshot

Figure 4: Illustration of Association Rules in Stream.

Example 7. ***(Association Rules in Ontology Streams)*** *At point of time* 8 *in streams of Figure 2, we obtain rules:* (30) *from $\mathcal{O}_0^9 \cup \mathcal{P}_0^9$ to $\mathcal{Q}_0^9$;* (31) *from $\mathcal{O}_0^9$ to $\mathcal{Q}_0^9$ among a large set.*

$$(23), (20) \twoheadrightarrow (21) \qquad (30) \qquad\qquad (11) \twoheadrightarrow (13) \qquad (31)$$

For instance, (30) *denotes the rule: "if $\{r_1\}$ is congested and humidity is high on $\{r_2\}$ then travel time is long on $\{r_3\}$".*

As the number of rules grows exponentially with the number of entailments in $\mathcal{O}_0^n$ and $\mathcal{P}_0^n$, it is neither practical nor desirable to mine all potential rules in $\mathcal{O}_0^n \times \mathcal{P}_0^n$. To measure their significance and select interesting rules, we adapt the well-known concepts of *support* (Definition 5) and *confidence* (Definition 6), introduced in the database community.

Definition 5. ***(Axioms Support in Ontology Stream)*** *Let (i) $\mathcal{L}$ be a DL, (ii) $\mathcal{T}$ be $\mathcal{L}$-axioms, (iii) $\mathcal{O}_0^n$ be a $\mathcal{L}$-stream, (iv) $\mathcal{G}$ be axioms. The support $supp(\mathcal{G}) \in [0, 1]$ is the proportion of snapshots in $\mathcal{O}_0^n$ where $\mathcal{T} \cup \mathcal{O}_0^n(i) \models \mathcal{G}$, $i \in [0, n]$.*

Example 8. ***(Axioms Support in Ontology Stream)*** *Let $\mathcal{O}_5^8$, $\mathcal{P}_5^8$, $\mathcal{Q}_5^8$ be respectively streams $\mathcal{O}_0^9$, $\mathcal{P}_0^9$, $\mathcal{Q}_0^9$ restricted to time interval* $[5, 8]$ *where knowledge at point of time* 5 *extends* $[6, 8]$ *in Figure 2. Table 3 illustrates the support of some of their axioms, grouped by syntactic similarity, stream-dereferenced e.g.,* (17) $\doteq$ (21)*.* (23) *is entailed at point of time* 5, 7, *and* 8 *of $\mathcal{O}_5^8$ but not at* 6*, thus a support of $\frac{3}{4}$.*

Ontology Streams		$\mathcal{O}_5^8$					$\mathcal{P}_5^8$		$\mathcal{Q}_5^8$	
Axioms $g \in \mathcal{G}$		(22)	(23)	(10)	(11)	(18)	(16)	(20)	(13)	(21)
Point of Time	5		✓		✓	✓		✓		
	6	✓		✓	✓			✓	✓	
	7		✓		✓	✓	✓			✓
	8		✓		✓	✓		✓		✓
$supp(g)$		$^1/_4$	$^3/_4$	$^1/_4$	1	$^3/_4$	$^1/_4$	$^3/_4$	$^1/_4$	$^1/_2$

Table 3: Support of Axioms $g \in \mathcal{G}$ in $\mathcal{O}_5^8$, $\mathcal{P}_5^8$, $\mathcal{Q}_5^8$.

Definition 6. ***(Confidence of an Association Rule)*** *Let (i) $\mathcal{L}$ be a DL, (ii) $\mathcal{O}_0^n$, $\mathcal{P}_0^n$ be $\mathcal{L}$-streams, (iii) $\mathcal{G}$ be sets of axioms and h be an axiom, and (iv) $\mathcal{G} \twoheadrightarrow h$ be an association rule in $\mathcal{O}_0^n \times \mathcal{P}_0^n$. The confidence c of $\mathcal{G} \twoheadrightarrow h$ in* $[0, 1]$ *is:*

$$c(\mathcal{G} \twoheadrightarrow h) \doteq \frac{supp(\mathcal{G} \cup h)}{supp(\mathcal{G})} \qquad (32)$$

with $supp(\mathcal{G} \cup h)$, as the support of $\mathcal{G} \twoheadrightarrow h$, is the proportion of snapshots in $\mathcal{O}_0^n \cup \mathcal{P}_0^n$ where both $\mathcal{G}, h$ are entailed.

The rule confidence is defined as the percentage of snapshots in $\mathcal{O}_0^n \cup \mathcal{P}_0^n$ where both $\mathcal{G}$ and h are satisfiable with regard to the overall number of snapshots where $\mathcal{G}$ is satisfiable. That is, the rule confidence can be understood as the conditional probability $p(\mathcal{T} \cup \mathcal{P}_0^n \models h \mid \mathcal{T} \cup \mathcal{O}_0^n \models \mathcal{G})$.

Example 9. ***(Rule Confidence in Ontology Streams)*** *The confidence of* (30)*, with $\mathcal{G}$:* (23), (20)*, h :* (21) *is:*

$$c((23), (20) \twoheadrightarrow (21)) \doteq \frac{supp((23), (20), (21))}{supp((23), (20))} \textit{ i.e., } \frac{^1/_4}{^1/_2} \qquad (33)$$

i.e., a score of $^1/_2$ in $\mathcal{O}_5^8 \cup \mathcal{P}_5^8 \cup \mathcal{Q}_5^8$. (30) *is correct in* 50% *of points of time* $[5, 8]$*. A confidence of $^1/_4$ is obtained for* (31)*.*

Remark 2. ***(Association Rules Restrictions)*** *Other measures e.g., lift, conviction [Geng* et al.*, 2006], all as a combination of support and confidence, can be adapted for capturing other degrees of significance. Similarly, other constraints on types of axioms in $\mathcal{G} \cup h$ could be considered for restricting rules e.g., axioms with $\equiv_{\mathcal{T}}$-similar subsumees/ers.*

4 Knowledge Prediction in Ontology Stream

We tackle the problem of prediction by (i) determining the most appropriate rules in evolutive knowledge (Algorithm 1), and (ii) exploiting the effects of injecting their consequents on streams (Algorithm 2).

4.1 Auto-Correlation-Driven Rules Selection

Algorithm 1 combines auto-correlation in lines (7-8) and association rules mining in lines (10-11), for deriving the most relevant rules in $\mathcal{O}_0^n \times \mathcal{P}_0^n$. In particular the rules are constrained to meet a minimal threshold of support and confidence (line 11). More importantly, these degrees of significance are contextualized and evaluated against only correlated snapshots (line 8). Thus, the selection is strongly driven by the factor of auto-correlation, making the rules mining knowledge evolution-aware. This constraint ensures that we learn rules that could be applied in similar context i.e., where knowledge evolution is not impacted by major changes.

Computing a solution with Algorithm 1 given a polynomial input n, the number of axioms in $\mathcal{T}$, $\mathcal{O}_0^n$, $\mathcal{P}_0^n$ and the number of initial rules $|\mathcal{R}|$ is in worst case polynomial time, due to the auto-correlation (Definition 3). However, if $|\mathcal{R}|$ is not bounded, the rules selection is in worst case NP with respect to the number of entailments in $\mathcal{O}_0^n \cup \mathcal{P}_0^n$.

```
Algorithm 1: $Rules\text{-}Selection\langle \mathcal{L}, \mathcal{T}, \mathcal{O}_0^n, \mathcal{P}_0^n, k, m_a, m_s, m_c \rangle$.
1  Input: (i) DL $\mathcal{L}$, (ii) Terminology $\mathcal{T}$, (iii) $\mathcal{L}$-streams $\mathcal{O}_0^n, \mathcal{P}_0^n$
          defined by some axioms, (iv) $k \in [0, n]$, (v) Min. thres-
          hold of auto-correlation $m_a$,support $m_s$,confidence $m_c$.
2  Result: $\mathcal{R} \doteq \{\rho \in \tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n \mid$ (i) $\tilde{\mathcal{O}}_0^n \subseteq \mathcal{O}_0^n$,
                              (ii) $\forall i, A(\tilde{\mathcal{O}}_0^n(i), \tilde{\mathcal{O}}_0^n(k)) > m_a$,
3                             (iii) $supp(\rho) > m_s, c(\rho) > m_c\}$
4  begin
5      $\tilde{\mathcal{O}}_0^n \leftarrow \emptyset$; $\mathcal{R} \leftarrow \emptyset$; % Initialization of $\tilde{\mathcal{O}}_0^n$ and result set $\mathcal{R}$.
6      % $\tilde{\mathcal{O}}_0^n$: snapshots of $\mathcal{O}_0^n$ which auto-correlates $\mathcal{O}_0^n(k)$.
7      foreach $i \in [0, n]$ do
8          if $A(\mathcal{O}_0^n(i), \mathcal{O}_0^n(k)) > m_a$ then $\tilde{\mathcal{O}}_0^n(i) \leftarrow \mathcal{O}_0^n(i)$;
9      % Min. support, confidence-constrained rules in $\tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n$.
10     foreach $\rho \in \tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n$ do
11         if $supp(\rho) > m_s \wedge c(\rho) > m_c$ then $\mathcal{R} \leftarrow \mathcal{R} \cup \{\rho\}$;
12     return $\mathcal{R}$;
```

Example 10. ***(Auto-Correlation-Driven Rules Selection)*** *$\langle \mathcal{L}, \mathcal{T}, \mathcal{X}, \mathcal{Y}, 8, 0, 1/3, 1/2 \rangle$ are inputs of Algorithm 1 for identifying rules from $\mathcal{X} \doteq \mathcal{O}_5^8 \cup \mathcal{P}_5^8$ to $\mathcal{Y} \doteq \mathcal{Q}_5^8$. Following Example 5, $\tilde{\mathcal{X}}$ (lines 8) is defined at time $i \in \{5, 7, 8\}$. Indeed $\mathcal{T} \cup \mathcal{X}(\alpha) \cup \mathcal{X}(\beta)$ is consistent for $(\alpha, \beta) \in \{(5, 8), (7, 8), (8, 8)\}$, not for $(6, 8)$, hence $A(\mathcal{X}(6), \mathcal{X}(8)) < 0$. Following lines 10-11, the support of rule (30) meets m_s i.e., $1/3$ in $\tilde{\mathcal{X}} \times \mathcal{Y}$ while it did not in $\mathcal{X} \times \mathcal{Y}$ with a score of $1/4$ (Examples 9), hence a better consideration of rules in similar context. The constraint m_c is also met in both contexts with a score of $1/2$.*

Theorem 1. ***(Consistent-based Evolution of Streams)*** *Let $\mathcal{R} \subseteq \tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n$ be output rules of Algorithm 1 with $m_a = 0$ and $\tilde{\mathcal{O}}_0^n \subseteq \mathcal{O}_0^n$. $\forall \rho \in \mathcal{R}$, supp$(\rho)$ and $c(\rho)$ have similar values in $\tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n$ and $\mathcal{O}_0^n \times \mathcal{P}_0^n$ if:*

$$\mathcal{T} \cup \mathcal{O}_0^n(i) \cup \mathcal{O}_0^n(j) \not\models \perp, \; \forall i, j \in [0, n] \qquad (34)$$

Proof. (Sketch) Algorithm 1 is applied with $m_a = 0$, thus all snapshots in $\mathcal{O}_0^n$ which are negatively correlated with $\mathcal{O}_0^n(j)$ do not appear in $\tilde{\mathcal{O}}_0^n$ (line 8). As (34) is valid $\forall i, j \in [0, n]$, all snapshots in $\mathcal{O}_0^n$ are pairwise consistent, hence all positively correlated. Thus, condition in line 8 is always true (Definition 3), hence $\tilde{\mathcal{O}}_0^n \doteq \mathcal{O}_0^n$ and $\tilde{\mathcal{O}}_0^n \times \mathcal{P}_0^n \doteq \mathcal{O}_0^n \times \mathcal{P}_0^n$ i.e., no snapshot has been removed from $\mathcal{O}_0^n$. □

The auto-correlation does not impact the rules selection if snapshots are pairwise consistent over time and $m_a = 0$.

Example 11. ***(Consistent-based Evolution of Streams)*** *The support and confidence of any rule in $\mathcal{P}_5^8 \times \mathcal{Q}_5^8$ remain unchanged by applying Algorithm 1 with $m_a = 0$ because of the consistency of $\mathcal{P}_5^8(i) \cup \mathcal{P}_5^8(j)$, $\forall i, j \in [5, 8]$.*

4.2 Rules-based Consistent Prediction

The prediction of knowledge at point of time n in stream $\mathcal{P}_0^{n-1}$ is achieved using Algorithm 2, illustrated in Figure 5.

All significant rules are identified by Algorithm 1 under constraints of support, confidence (line 6 and ρ, ρ' in Figure 5). To enforce their applicability at time n, auto-correlation can be positively considered through m_a ($A(i, n)$, $A(j, n)$ in Figure 5). All rules $\mathcal{G} \twoheadrightarrow h$, are then filtered by confidence (line 8 and conditions (i) in Figure 5). Finally, the consistency of $\mathcal{G}$ in $\mathcal{O}_0^n$ and h in $\mathcal{P}_0^n$ (line 10 and conditions (ii-iii) in Figure 5) are checked to ensure a consistent prediction $\mathcal{P}_0^n(n)$ (line 11 and $\dashrightarrow$ in Figure 5) using filtered rules (lines 8-10 and ρ in Figure 5).

```
Algorithm 2: $Prediction\langle \mathcal{L}, \mathcal{T}, \mathcal{O}_0^n, \mathcal{P}_0^{n-1}, m_a, m_s, m_c \rangle$.
1  Input: (i) DL $\mathcal{L}$, (ii) Terminology $\mathcal{T}$, (iii) $\mathcal{L}$-streams $\mathcal{O}_0^n$,
          $\mathcal{P}_0^{n-1}$ defined by some axioms, (iv) Min. threshold of
          auto-correlation $m_a$, support $m_s$, confidence $m_c$.
2  Result: $\mathcal{P}_0^n(n)$: Knowledge predicted at point of time $n$.
3  begin
4      $\mathcal{P}_0^n(n) \leftarrow \emptyset$; % Initialization of prediction.
5      % Identification of significant rules in $\mathcal{O}_0^n \times \mathcal{P}_0^{n\text{-}1}$. ①
6      $\mathcal{R} \leftarrow Rules\text{-}Selection\langle \mathcal{L}, \mathcal{T}, \mathcal{O}_0^n, \mathcal{P}_0^{n\text{-}1}, n, m_a, m_s, m_c \rangle$;
7      % All rules $\rho : \mathcal{G} \twoheadrightarrow h$ in $\mathcal{R}$ with highest confidence.
8      foreach $\rho : \mathcal{G} \twoheadrightarrow h \mid \nexists \rho' : \mathcal{G}' \twoheadrightarrow h', c(\rho') > c(\rho)$ do
9          % Evaluation of rule $\rho \in \mathcal{R}$ at point of time $n$. ②
10         if $(\mathcal{T} \cup \mathcal{O}_0^n(n) \cup \mathcal{G} \not\models \perp) \wedge (\mathcal{T} \cup \mathcal{P}_0^n(n) \cup h \not\models \perp)$
           then % Axiom $h$ extends knowledge of $\mathcal{P}_0^n$ at time $n$.
11             $\mathcal{P}_0^n(n) \leftarrow \mathcal{P}_0^n(n) \cup h$;% ③
12     return $\mathcal{P}_0^n(n)$;
```

Predicting knowledge given a polynomial number of rules $\mathcal{R}$ with Algorithm 2 is in worst case polynomial time, due to line 6 (Algorithm 1) and line 10 [Baader *et al.*, 2005] in $\mathcal{EL}^{++}$. In the worst case of an exponential number of rules captured in $\mathcal{O}_0^n \times \mathcal{P}_0^n$ by Algorithm 2 (line 6), prediction is in NP. According to Theorem 1, the set of significant rules is strongly impacted by the stream inconsistency of $\mathcal{O}_0^n$. Thus the more inconsistent the evolution of stream $\mathcal{O}_0^n$ is, the more scalable will be the prediction.

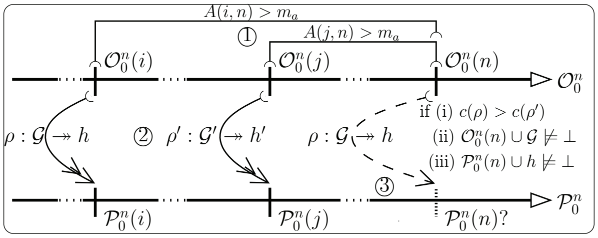

$A(i, n) > m_a$
①
$A(j, n) > m_a$
$\mathcal{O}_0^n(i)$
$\mathcal{O}_0^n(j)$
$\mathcal{O}_0^n(n)$
$\mathcal{O}_0^n$
if (i) $c(\rho) > c(\rho')$
$\rho : \mathcal{G} \twoheadrightarrow h$
② $\rho' : \mathcal{G}' \twoheadrightarrow h'$
$\rho : \mathcal{G} \twoheadrightarrow h$
(ii) $\mathcal{O}_0^n(n) \cup \mathcal{G} \not\models \perp$
(iii) $\mathcal{P}_0^n(n) \cup h \not\models \perp$
③
$\mathcal{P}_0^n(i)$
$\mathcal{P}_0^n(j)$
$\mathcal{P}_0^n(n)?$
$\mathcal{P}_0^n$

Figure 5: Predicting Knowledge in An Ontology Stream.

Example 12. ***(Rules-based Consistent Prediction)*** *Let $\mathcal{O}_5^9$, $\mathcal{P}_5^9$ be $\mathcal{L}$-stream $\mathcal{O}_5^8$, $\mathcal{P}_5^8$ extended at time 9 (Table 4). Predicting knowledge of $\mathcal{Q}_5^8$ at time 9 consists in applying Algorithm 2 e.g., $\langle \mathcal{L}, \mathcal{T}, \mathcal{X}, \mathcal{Y}, 0, 1/3, 1/2 \rangle$ with $\mathcal{X} \doteq \mathcal{O}_5^9 \cup \mathcal{P}_5^9$, $\mathcal{Y} \doteq \mathcal{Q}_5^8$. Since $m_a = 0$, line 6 (Algorithm 2) returns only rules with entailments in $\mathcal{X}(6) \times \mathcal{Y}(6)$. Indeed,*

$\mathcal{T} \cup \mathcal{X}(\alpha) \cup \mathcal{X}(\beta)$ is consistent for $(\alpha, \beta) \in \{(6, 9), (9, 9)\}$. All potential rules have a similar support and confidence of $^1/_2$*, and all reach consequent* (13) *which is consistent in any empty $\mathcal{Q}_5^9(9)$. $\mathcal{X}_5^9(9) \cup \mathcal{G} \not\models \perp$ since all entailments $\mathcal{G}$ of $\mathcal{X}_5^9$ are the same at time* 6 *and* 9. (13) *is the predicted knowledge.*

Streams	$\mathcal{O}_5^9$					$\mathcal{P}_5^9$		$\mathcal{Q}_5^9$	
Entailment $g \in \mathcal{G}$	(22)	(23)	(10)	(11)	(18)	(16)	(20)	(13)	(21)
Point of Time 9	✓		✓	✓			✓	?	?

Table 4: Support of Entailment $g \in \mathcal{G}$ in $\mathcal{O}_5^8$, $\mathcal{P}_5^8$, $\mathcal{Q}_5^8$.

Injected in $\mathcal{P}_0^n(n)$, predictions can be used by DL reasoners for deriving new entailments, as a side-effect of Algorithm 2.

Example 13. ***(Prediction Side Effects in Ontology Streams)*** *Applying GCI* (5) *and prediction* (13) *in Example 12 using DL reasoning reaches to bus delay information on road $\{r_3\}$.*

Although Examples 12 and 13 illustrate prediction where all entailments and rules are restricted to only $\{r_1\}$, Algorithm 2 (line 8) is designed for more complex and general cases.

5 Experimental Results

We report (i) scalability of our approach and (ii) accuracy of its results. In particular we study the impact of axioms, their consistency together with support, confidence, and auto-correlation thresholds on Algorithms 1 ($A1$ for rules selection), 2 ($A2$ for knowledge prediction). Our implementation is based on (i) an extension of InfoSphere Streams [Biem *et al.*, 2010] for processing ontology streams in real-time, coupled with (ii) CEL reasoner [Baader *et al.*, 2006] for standard DL reasoning, and (iii) an adaptation of Apriori [Agrawal and Srikant, 1994] supporting subsumption for determining association rules. For scalability reasons, rules are not injected in DL reasoning but only their consequents. The experiments have been conducted on a server of 4 Intel(R) Xeon(R) X5650, 2.67GHz cores, 6GB RAM.

• **Context:** Reputable live stream data (Table 5) related to road [a] weather conditions, [b] travel times, [c] incidents together with [d] bus GPS location, delay and congestion status in Dublin City has been considered. Besides an ontology of 55 concepts, 19 role descriptions (17 concepts subsume the 38 remaining ones with a depth of 3), we inject $14,316$ $\mathcal{EL}^{++}$ GCIs (through 6 RDF triples) to describe 4772 roads, their interconnections. The objective is to predict which buses (among 300 buses) in [d] will be delayed in the next hour, using cross-stream rules selected from $A1$ and $A2$.

DataSet	Size (Mb) per day	Frequency of Update (seconds)	#Axioms per Update	#RDF Triples per Update
[a] Weather	3	300	53	318
[b] Travel	43	60	270	810
[c] Incident	0.1	600	81	324
[d] Bus	120	40	3,000	12,000

Table 5: Stream Datasets Details (average figures).

• **Scalability Result:** Figure 6 reports scalability of $A1$, $A2$ ((m_a, m_s, m_c) being $(0, {}^1/_2, {}^1/_2)$) and compares its computation time with a state-of-the-art approach [Wang *et al.*, 2003] in stream prediction, noted [W03]. They solve a classification problem over sensor raw data using statistics-based data mining techniques. The evaluation is achieved on (i) various sizes of stream windows $|w|$ (for learning/training) i.e., $\{1, 12, 48\}$ hours, and (ii) different number of streams $|s|$ i.e., $\{1, 3, 4\}$ for respectively [d], [b,c,d], [a,b,c,d] in Table 5.

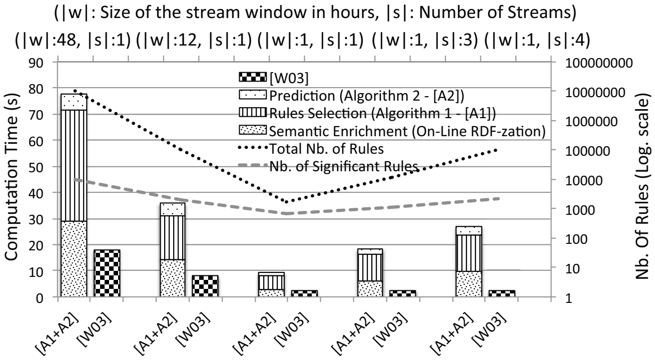


Figure 6: Scalability of Prediction Computation.

Unsurprisingly, [W03] scales much better than our approach in all contexts. Our approach requires some non-negligible computation time for the semantic enrichment of streams. In addition, as the number of potential rules is exponential with the number of entailments in streams (secondary vertical axis), the identification of significant rules is time consuming specially when the window size is growing. Once all rules are identified, the pure prediction part performs from $1.1s$ to $6.2s$. As [W03] is mainly designed for one stream, the computation time remains unchanged if multiple streams are considered.

• **Accuracy Result:** Figure 7 reports the prediction accuracy of both approaches where Table 6 is used only to configure the parameters values (m_a, m_c, m_s) of our approach $A2$. Prediction is computed within a window of 48 hours with all streams (Table 5). Accuracy is measured by comparing predictions (delayed buses) with real-time situations in Dublin City, where results can be easily extracted and compared from the raw and semantic data in respectively [W03] and our approach. Negative auto-correlation (c_1-c_4) strongly alters the accuracy while support and confidence have a positive effect. The confidence has a stronger impact on (i) the reduction of significant rules and (ii) accuracy. If positive auto-correlation, the accuracy of $A2$ (with a minimum confidence and support of .4) results outperforms results of [W03].

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
m_a	< 0				> 0			
m_c	.4	.4	.8	.8	.4	.4	.8	.8
m_s	.4	.8	.4	.8	.4	.8	.4	.8

Table 6: (m_a, m_c, m_s) Configuration.

• **Lessons Learnt:** Although inferring cross-streams rules has a positive impact on consistent prediction and its accuracy, it alters its computation time specially if streams derived numerous entailments. It is even worst with more expressive DLs because of the auto-correlation evaluation (line 7 in Algorithm 1) and consistency check (line 10 in Algorithm 2). The scalability (resp. accuracy) of prediction is negatively (resp. positively) impacted by the ordered settings: (1) stream window size, (2), expressivity of DL, (3) update frequency. The window size is probably the most critical as it drives some heterogeneity in the rules elaboration, which could improve accuracy, but not scalability. The uncertainty related to sensor data e.g., through mis-calibration, noise has also some impacts on association rules hence their accuracy. However accurate prediction can be achieved in real-time using a stream window size $|w|$ of 12 hours in our context of $\mathcal{EL}^{++}$ DL and minimal update frequency of 40 seconds.

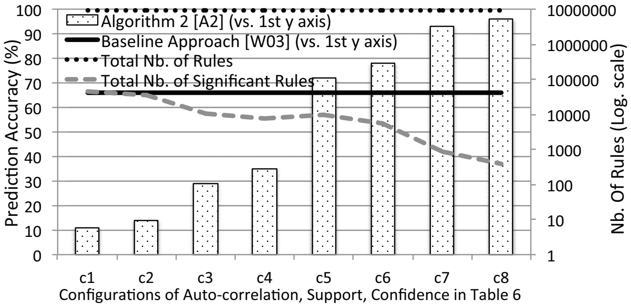


Figure 7: Accuracy of Prediction.

6 Related Work

Prediction, or the problem of estimating future observations given some historical information, spans many research fields, from Statistics to Database and various fields of Artificial Intelligence e.g., planning, knowledge representation and reasoning. Depending on the level of data representation which is considered, a prediction problem [Han and Kamber, 2006] can be formulated as a standard machine learning classification (for symbolic values) [Wang *et al.*, 2003] or regression (for numeric values) task [Härdle, 1992].

In most of data stream mining applications, prediction is estimated by correlating current and past data patterns using different distance metrics [Gehrke *et al.*, 2001]. More sophisticated approaches e.g., [Zaki, 2001] investigated sequential pattern mining for capturing the time-based evolution of data in streams. [Papadimitriou *et al.*, 2005] goes further by augmenting existing models with cross-correlation over streams. These approaches are designed for very fast processing and mining of (syntactic) raw data from sensor networks. However they all fail in using and interpreting underlying semantics of data, making prediction in ontology streams not as accurate and consistent as it could be, specially in a context of concept drift (i.e., streams characterized by many changes over time). [Lisi and Malerba, 2004] partially tackles this issue by mining multi-level association rules following hierarchies-based representation of data. However (i) multi-streams dimension, (ii) their inconsistent evolution and (iii) predictive inference are not addressed.

From an ontology stream perspective, (i) reasoning [Valle *et al.*, 2009] for capturing real-time knowledge, or (ii) diagnosing [Lécué, 2012] for explaining anomalies in streams have been recently addressed. Similarly to prediction, the latter requires metrics for evaluating similarities over time e.g., through the detection of changes [Noy and Musen, 2002] in ontologies. These techniques among others [Shvaiko and Euzenat, 2013] could be adapted for correlating past and current semantics-augmented observations and then deriving prediction. However the prediction would solely depends on few observations, which can be noisy, not necessarily accurate, and also ignore potential (logics-based) correlations with other streams. This may prevent the identification of relevant and accurate patterns and rules.

Alternative approaches [Bloehdorn and Sure, 2007], combining machine learning and semantic web principles, are able to predict high level classes of data by mining data instances in ontologies [Stumme *et al.*, 2006]. While promising for learning structure and class-membership on the semantic web, such methods would require major adaptations for predicting knowledge in streams.

7 Conclusion and Future Work

We studied ontology stream i.e., a dynamic and an evolutive versions of ontologies over time, and addressed the problem of predicting its future (or missing) knowledge. We envisioned predictive reasoning as a correlation and interpretation of past semantics-augmented data over multiple ontology streams. We inferred and mined rules across exogenous streams, where their DL-based semantic representations are captured and interpreted through some background knowledge. Based on an analysis of stream evolution, consistent predictions are constructed by selecting and applying relevant cross-streams association rules. The step of lifting raw stream data at semantic (i.e., ontology stream) level was beneficial for (i) easy integration and adaptation of heterogenous data streams with static knowledge, (ii) cross-correlating and mining knowledge as a basis of the prediction approach and (iii) bootstrapping knowledge. In particular the semantic representation of the domain was crucial for pruning the search space of association rules and identify those which are relevant (in the sense of semantic consistency) for prediction. Our experiments have shown accurate prediction with real and live stream data from Dublin City.

In future work we will improve the scalability for supporting high throughput sensors. One direction is to apply novel reasoning services, e.g. by using summarization techniques [Fokoue *et al.*, 2006; 2012] to reduce the size of window of observations while maintaining its knowledge or by using native support for stream reasoning and parallel reasoning in TrOWL [Thomas *et al.*, 2010]. The classification and generalization of rules using background knowledge is another direction.

Acknowledgments

The authors would like to thank Alice Marascu for helpful discussions, and the anonymous reviewers for their valuable comments and suggestions to improve the paper.

References

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.

[Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[Auer *et al.*, 2006] Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki - a tool for social, semantic collaboration. In *ISWC*, pages 736–749, 2006.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*, 2003.

[Baader *et al.*, 2005] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the el envelope. In *IJCAI*, pages 364–369, 2005.

[Baader *et al.*, 2006] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL - A Polynomial-Time Reasoner for Life Science Ontologies. In *IJCAR*, pages 287–291, 2006.

[Baader *et al.*, 2008a] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the el envelope further. In *OWLED*, 2008.

[Baader *et al.*, 2008b] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the el envelope further. In *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[Biem *et al.*, 2010] Alain Biem, Eric Bouillet, Hanhua Feng, Anand Ranganathan, Anton Riabov, Olivier Verscheure, Haris N. Koutsopoulos, and Carlos Moran. Ibm infosphere streams for scalable, real-time, intelligent transportation services. In *SIGMOD Conference*, pages 1093–1104, 2010.

[Bloehdorn and Sure, 2007] Stephan Bloehdorn and York Sure. Kernel methods for mining instance data in ontologies. In *ISWC/ASWC*, pages 58–71, 2007.

[Bracewell, 1999] Ronald Bracewell. *The Autocorrelation Function in The Fourier Transform and Its Applications*. McGraw-Hill, New York, 3rd edition, 1999.

[Calbimonte *et al.*, 2010] Jean-Paul Calbimonte, Óscar Corcho, and Alasdair J. G. Gray. Enabling ontology-based access to streaming data sources. In *International Semantic Web Conference (1)*, pages 96–111, 2010.

[Fokoue *et al.*, 2006] Achille Fokoue, Aaron Kershenbaum, Li Ma, Edith Schonberg, and Kavitha Srinivas. The summary abox: Cutting ontologies down to size. In *International Semantic Web Conference*, pages 343–356, 2006.

[Fokoue *et al.*, 2012] A. Fokoue, F. Meneguzzi, M. Sensoy, and J. Z. Pan. Querying linked ontological data through distributed summarization. In *AAAI2012*, 2012.

[Gehrke *et al.*, 2001] Johannes Gehrke, Flip Korn, and Divesh Srivastava. On computing correlated aggregates over continual data streams. In *SIGMOD Conference*, pages 13–24, 2001.

[Geng *et al.*, 2006] Liqiang Geng, Howard J Hamilton, and Leah. Findlater. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3), September 2006.

[Han and Kamber, 2006] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[Härdle, 1992] W. Härdle. *Applied Nonparametric Regression*. Econometric Society Monographs. Cambridge University Press, 1992.

[Horrocks and Sattler, 2001] Ian Horrocks and Ulrike Sattler. Ontology reasoning in the shoq(d) description logic. In *IJCAI*, pages 199–204, 2001.

[Huang and Stuckenschmidt, 2005] Zhisheng Huang and Heiner Stuckenschmidt. Reasoning with multi-version ontologies: A temporal logic approach. In *International Semantic Web Conference*, pages 398–412, 2005.

[Lécué, 2012] Freddy Lécué. Diagnosing changes in an ontology stream: A dl reasoning approach. In *AAAI*, 2012.

[Lisi and Malerba, 2004] Francesca A. Lisi and Donato Malerba. Inducing multi-level association rules from multiple relations. *Machine Learning*, 55(2):175–210, 2004.

[Noy and Musen, 2002] N. F. Noy and M. A. Musen. Promptdiff: A fixed-point algorithm for comparing ontology versions. In *AAAI/IAAI*, pages 744–750, 2002.

[Papadimitriou *et al.*, 2005] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. Streaming pattern discovery in multiple time-series. In *VLDB*, pages 697–708, 2005.

[Ren and Pan, 2011] Yuan Ren and Jeff Z. Pan. Optimising ontology stream reasoning with truth maintenance system. In *CIKM*, pages 831–836, 2011.

[Sheth, 2010] Amit P. Sheth. Computing for human experience: Semantics-empowered sensors, services, and social computing on the ubiquitous web. *IEEE Internet Computing*, 14(1):88–91, 2010.

[Shvaiko and Euzenat, 2013] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013.

[Stumme *et al.*, 2006] Gerd Stumme, Andreas Hotho, and Bettina Berendt. Semantic web mining: State of the art and future directions. *J. Web Sem.*, 4(2):124–143, 2006.

[Thomas *et al.*, 2010] E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *Extended Semantic Web Conference*, 2010.

[Valle *et al.*, 2009] Emanuele Della Valle, Stefano Ceri, Frank van Harmelen, and Dieter Fensel. It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, 24(6):83–89, 2009.

[Wang *et al.*, 2003] Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD*, pages 226–235, 2003.

[Zaki, 2001] Mohammed Javeed Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.