



Learning from Ontology Streams with Semantic Concept Drift

Jiaoyan Chen, Freddy Lecue, Jeff Pan, Huajun Chen

► To cite this version:

Jiaoyan Chen, Freddy Lecue, Jeff Pan, Huajun Chen. Learning from Ontology Streams with Semantic Concept Drift. Twenty-Sixth International Joint Conference on Artificial Intelligence, Aug 2017, Melbourne, France. 10.24963/ijcai.2017/133 . hal-01934901

HAL Id: hal-01934901

<https://hal.inria.fr/hal-01934901>

Submitted on 26 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning from Ontology Streams with Semantic Concept Drift

Jiaoyan Chen

Zhejiang University
China
jiaoyanchen@zju.edu.cn

Freddy Lécué

INRIA, France
Accenture Labs, Ireland
freddy.lecue@inria.fr

Jeff Z. Pan

University of Aberdeen
United Kingdom
jeff.z.pan@abdn.ac.uk

Huajun Chen*

Zhejiang University
China
huajunsir@zju.edu.cn

Abstract

Data stream learning has been largely studied for extracting knowledge structures from continuous and rapid data records. In the semantic Web, data is interpreted in ontologies and its ordered sequence is represented as an ontology stream. Our work exploits the semantics of such streams to tackle the problem of concept drift i.e., unexpected changes in data distribution, causing most of models to be less accurate as time passes. To this end we revisited (i) semantic inference in the context of supervised stream learning, and (ii) models with semantic embeddings. The experiments show accurate prediction with data from Dublin and Beijing.

1 Introduction and Related Work

Stream learning, or the problem of extracting and predicting knowledge from temporal evolution of data, has been largely studied. Most of techniques in Database e.g., [Cheung *et al.*, 1996], adapting *Apriori* [Agrawal *et al.*, 1996] for streams, focus on syntactic representation of data to identify frequent associations and exploit them for prediction. [Lee *et al.*, 2003] improved its scalability by partitioning all streams using a sliding-window filtering. Approaches in Machine Learning e.g., [Gama and Kosina, 2011] focus on learning decision rules for classifying data from streams in real-time.

Although highly scalable, most approaches have been shown to be non robust to *concept drift* i.e., unexpected changes in data distribution [Coble and Cook, 2000]. Indeed their models, built on old data and then inconsistent with new data, are less accurate as time passes. Towards this challenge [Chu *et al.*, 2011] applied online active learning using customized properties of weighing. Alternatively [Gao *et al.*, 2007] prioritized recent data during the elaboration of the learning model through regular updates, assuming temporally adjacent data is the most representative information for prediction. [Bifet *et al.*, 2015] went further by considering dynamic sliding windows. More details are presented in [Gama *et al.*, 2014]. Although such approaches manage gradual changes, they fail in maintaining high accuracy for sudden, abrupt changes. This is mainly due to inconsistent

evolution of knowledge and lack of metrics to understand the semantics of its changes and concept drifts.

Towards this issue adapt the problem in semantic Web settings. Such streams, represented as ontology streams [Huang and Stuckenschmidt, 2005; Ren and Pan, 2011], are evolutive versions of ontologies where OWL (Web Ontology Language), which is underpinned by Description Logics (DL) [Baader *et al.*, 2003], is used as a rich description language. From knowledge materialization [Beck *et al.*, 2016; Galárraga *et al.*, 2013; Ren *et al.*, 2016], to predictive reasoning [Lecue and Pan., 2013; Lécué, 2015], all are inferences where dynamics, semantics of data are exploited for deriving a priori knowledge from pre-established (certain) statements. However concept drift is not handled, which limits accuracy of prediction for highly changing streams.

Our approach, exploiting the semantics of data streams, tackles the problem of learning and prediction with concept drifts. Given some continuous knowledge, how to manage its changes and their inconsistent evolution to ensure accurate prediction? Semantic reasoning and machine learning have been combined by revisiting features embeddings as semantic embeddings i.e., vectors capturing consistency and knowledge entailment in ontology streams. Such embeddings are then exploited in a context of supervised stream learning to learn models, which are robust to concept drifts i.e., sudden and inconsistent prediction changes. Our approach has been shown to be adaptable and flexible to basic learning techniques. The experiments have shown accurate prediction with live stream data from Dublin in Ireland and Beijing in China.

The next section reviews the adopted logic and ontology stream learning problem. In Section 3 we study concept drift. Section 4 presents semantic embeddings and prediction.

2 Background

The semantics of data is represented using an ontology. We focus on Description Logic (DL) to define ontologies since it offers reasoning support for most of its expressive families and compatibility to W3C standards e.g., OWL 2. Our work is illustrated using DL \mathcal{EL}^{++} [Baader *et al.*, 2005], which supports polynomial time reasoning. We review (i) DL basics of \mathcal{EL}^{++} , (ii) ontology stream, (iii) stream learning problem.

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

A signature Σ , noted $(\mathcal{N}_C, \mathcal{N}_R, \mathcal{N}_I)$ consists of 3 disjoint sets of (i) atomic concepts \mathcal{N}_C , (ii) atomic roles \mathcal{N}_R , and (iii) indi-

*Corresponding author.

viduals \mathcal{N}_I . Given a signature, the top concept \top , the bottom concept \perp , an atomic concept A , an individual a , an atomic role expression r , \mathcal{EL}^{++} concept expressions C and D in \mathcal{C} can be composed with the following constructs:

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

A DL ontology $\mathcal{O} \doteq \langle \mathcal{T}, \mathcal{A} \rangle$ is composed of TBox \mathcal{T} , ABox \mathcal{A} . A TBox is a set of concept, role axioms. \mathcal{EL}^{++} supports General Concept Inclusion axioms (GCIs e.g. $C \sqsubseteq D$), Role Inclusion axioms (RIs e.g., $r \sqsubseteq s$). An ABox is a set of concept assertion axioms, e.g., $C(a)$, role assertion axioms, e.g., $R(a, b)$, individual in/equality axioms e.g., $a \neq b$, $a = b$.

Example 1. (TBox and ABox Concept Assertion Axioms)

Figure 1 presents (i) a TBox \mathcal{T} where *DisruptedRoad* (2) denotes the concept of “roads which are adjacent to an event causing high disruption”, (ii) concept assertions (11-12) with the individual r_0 having roads r_1 and r_2 as adjunct roads.

$SocialEvent \sqcap \exists type.Poetry \sqsubseteq Event \sqcap \exists disruption.Low$	(1)
$Road \sqcap \exists adj.(\exists occur.\exists disruption.High) \sqsubseteq DisruptedRoad$	(2)
$Road \sqcap \exists adj.(\exists occur.\exists disruption.Low) \sqsubseteq ClearedRoad$	(3)
$BusRoad \sqcap \exists travel.Long \sqsubseteq DisruptedRoad$	(4)
$BusRoad \sqcap \exists travel.OK \sqsubseteq ClearedRoad$	(5)
$Road \sqcap \exists with.Bus \sqsubseteq BusRoad$	(6)
$Road(r_1)$	(8)
$Road(r_2)$	(9)
$Bus(b_0)$	(10)
$adj(r_0, r_1)$	(11)
$adj(r_0, r_2)$	(12)
$Long \sqcap OK \sqsubseteq \perp$	(13)

Figure 1: $\mathcal{O} \doteq \langle \mathcal{T}, \mathcal{A} \rangle$. Sample of TBox \mathcal{T} and ABox \mathcal{A} .

All completion rules, which are used to classify \mathcal{EL}^{++} TBox \mathcal{T} and entail subsumption, are described in [Baader et al., 2005]. Reasoning with such rules is PTime-Complete.

2.2 Ontology Stream

We represent knowledge evolution by a dynamic, evolutive version of ontologies [Huang and Stuckenschmidt, 2005]. Data (ABox), its inferred statements (entailments) are evolving over time while its schema (TBox) remains unchanged.

Definition 1. (DL \mathcal{L} Ontology Stream)

A DL \mathcal{L} ontology stream \mathcal{P}_m^n from point of time m to point of time n is a sequence of (sets of) Abox axioms $(\mathcal{P}_m^n(m), \mathcal{P}_m^n(m+1), \dots, \mathcal{P}_m^n(n))$ with respect to a static TBox \mathcal{T} in a DL \mathcal{L} where $m, n \in \mathbb{N}$ and $m < n$.

$\mathcal{P}_m^n(i)$ is a snapshot of an ontology stream \mathcal{P}_m^n at time i , referring to ABox axioms. Thus a transition from $\mathcal{P}_m^n(i)$ to $\mathcal{P}_m^n(i+1)$ is seen as an ABox update. We denote by $\mathcal{P}_m^n[i, j]$ i.e., $\bigcup_{k=i}^j \mathcal{P}_m^n(k)$ a windowed stream of \mathcal{P}_m^n between time i and j with $i \leq j$. Any window $[i, j]$ has a fixed length. 1-length windows are denoted by (i) . We consider streams \mathcal{P}_0^n with $[\alpha] \doteq [i, j]$, $[\beta] \doteq [k, l]$ as windows in $[0, n]$ and $i < k$.

Example 2. (DL \mathcal{EL}^{++} Ontology Stream)

Figure 2 illustrates \mathcal{EL}^{++} streams \mathcal{P}_0^n , \mathcal{Q}_0^n , \mathcal{R}_0^n , related to events, travel time, buses, through snapshots at time $i \in \{0, 1, 2, 3\}$ (i.e., a view on $[0, 3]$). In our example n is any integer greater than 5. Their dynamic knowledge is captured by evolutive ABox axioms e.g., (20) captures e_1 as “a social poetry event occurring in r_2 ” at time 1 of \mathcal{P}_0^n .

By applying completion rules on static knowledge \mathcal{T} and ontology streams \mathcal{P}_0^n , snapshot-specific axioms are inferred.

The evolution of a stream is captured along its changes i.e., *new*, *obsolete* and *invariant* ABox entailments from one windowed stream to another one in Definition 2 [Lécué, 2015].

Definition 2. (ABox Entailment-based Stream Changes)

Let S_0^n be a stream; $[\alpha]$, $[\beta]$ be windows in $[0, n]$; \mathcal{T} be axioms, \mathcal{G} its ABox entailments. The changes occurring from $S_0^n[\alpha]$ to $S_0^n[\beta]$, denoted by $S_0^n[\beta] \nabla S_0^n[\alpha]$, are ABox entailments in \mathcal{G} being *new* (14), *obsolete* (15), *invariant* (16).

$$\mathcal{G}_{new}^{[\alpha],[\beta]} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup S_0^n[\beta] \models g \wedge \mathcal{T} \cup S_0^n[\alpha] \not\models g\} \quad (14)$$

$$\mathcal{G}_{obs}^{[\alpha],[\beta]} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup S_0^n[\beta] \not\models g \wedge \mathcal{T} \cup S_0^n[\alpha] \models g\} \quad (15)$$

$$\mathcal{G}_{inv}^{[\alpha],[\beta]} \doteq \{g \in \mathcal{G} \mid \mathcal{T} \cup S_0^n[\beta] \models g \wedge \mathcal{T} \cup S_0^n[\alpha] \models g\} \quad (16)$$

(14) reflects knowledge we gain by sliding window from $[\alpha]$ to $[\beta]$ while (15) and (16) denote respectively lost and stable knowledge. All duplicates are supposed removed. Definition 2 provides basics, via ABox entailments [Ren et al., 2011], for understanding how knowledge evolves over time.

$\mathcal{P}_0^n(0) : (Incident \sqcap \exists impact.Limited)(e_3), occur(r_1, e_3)$	(17)
$\mathcal{Q}_0^n(0) : (Road \sqcap \exists travel.OK)(r_1)$	(18)
$\mathcal{R}_0^n(0) : with(r_1, b_0)$	(19)
$\mathcal{P}_0^n(1) : (SocialEvent \sqcap \exists type.Poetry)(e_1), occur(r_2, e_1)$	(20)
$\mathcal{Q}_0^n(1) : (Road \sqcap \exists travel.OK)(r_2)$	(21)
$\mathcal{R}_0^n(1) : with(r_2, b_0)$	(22)
$\mathcal{P}_0^n(2) : (Event \sqcap \exists disruption.High)(e_2), occur(r_2, e_2)$	(23)
$\mathcal{Q}_0^n(2) : (Road \sqcap \exists travel.Long)(r_2)$	(24)
$\mathcal{R}_0^n(2) : with(r_2, b_0)$	(25)
$\mathcal{P}_0^n(3) : (Event \sqcap \exists disruption.High)(e_2), occur(r_2, e_2)$	(26)
$\mathcal{Q}_0^n(3) : (Road \sqcap \exists travel.Long)(r_2)$	(27)
$\mathcal{R}_0^n(3) : with(r_2, b_0)$	(28)

Figure 2: Ontology Streams $\mathcal{P}_0^n(i)$, $\mathcal{Q}_0^n(i)$, $\mathcal{R}_0^n(i)_{i \in \{0,1,2,3\}}$.

Example 3. (ABox Entailment-based Stream Changes)

Table 1 illustrates changes occurring from $(\mathcal{Q} \cup \mathcal{R})_0^n[0, 1]$ to $(\mathcal{Q} \cup \mathcal{R})_0^n[2, 3]$ through ABox entailments. For instance “ r_2 as a disrupted road window $[2, 3]$ of $(\mathcal{Q} \cup \mathcal{R})_0^n$ is new with respect to knowledge in $[0, 1]$. It is entailed using DL completion rules on (4), (6), (9), (24), (25), (27) and (28).

Windowed Stream Changes	$(\mathcal{Q} \cup \mathcal{R})_0^n[2, 3] \nabla (\mathcal{Q} \cup \mathcal{R})_0^n[0, 1]$		
	obsolete	invariant	new
$with(r_2, b_0)$		✓	
$ClearedRoad(r_2)$	✓		
$DisruptedRoad(r_2)$			✓

Table 1: ABox Entailment-based Stream Changes.

2.3 Ontology Stream Learning Problem

Definition 3 revisits classic supervised learning [Domingos and Hulten, 2000] for ontology stream as the problem of predicting knowledge (through entailment) in a future snapshot.

Definition 3. (Ontology Stream Learning Problem)

Let S_0^n be a stream; \mathcal{T} , \mathcal{A} be respectively TBox, ABox; $g \in \mathcal{G}$ be an ABox entailment. An Ontology Stream Learning Problem, noted $OSLP\langle S_0^n, k, \mathcal{T}, \mathcal{A}, g \rangle$, is the problem of estimating whether g can be entailed from \mathcal{T} and \mathcal{A} at time $k \in (0, n]$ of stream S_0^n , given knowledge at time $t < k$ of S_0^n .

This estimation is denoted as $p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(k) \models g)$ with values in $[0, 1]$ and $k \geq 1$. g is a class assertion entailment in

the form of $G(a)$, with G a concept expression and a an individual. The estimation, adapted from [Gao *et al.*, 1], can be elaborated using knowledge from previous snapshots of S_0^k :

$$p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(k) \models g) \doteq \frac{p_{|\mathcal{T} \cup \mathcal{A}}(S_0^{k-1} \models g)}{p_{|\mathcal{T} \cup \mathcal{A}}(a \in S_0^{k-1})} \quad (29)$$

Estimation $p_{|\mathcal{T} \cup \mathcal{A}}(S_0^{k-1} \models g)$ is the proportion of snapshots in S_0^{k-1} entailing g . The conditional probability of a in S_0^{k-1} (noted $a \in S_0^{k-1}$) given S_0^{k-1} entailing g , or $G(a)$, is 1.

Example 4. (Ontology Stream Learning Problem)

The problem of estimating whether class assertion g , defined as $DisruptedRoad(r_2)$, can be entailed from \mathcal{T} and \mathcal{A} at time 4 of $(\mathcal{Q} \cup \mathcal{R})_0^n$ is defined as $OSLP((\mathcal{Q} \cup \mathcal{R})_0^n, 4, \mathcal{T}, \mathcal{A}, g)$. The estimation can be retrieved using (29) hence $p_{|\mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n(4) \models DisruptedRoad(r_2)) \doteq 2/3$.

3 Concept Drift in An Ontology Stream

We introduce semantic concept drift, as a basis for qualifying, quantifying sudden and abrupt changes in an ontology stream.

3.1 Semantic Concept Drift

Definition 7 revisits *concept drift* [Gao *et al.*, 2007] for ontology streams as *prediction changes* (Definition 4) in ABox entailment, which are *sudden* and *abrupt* (Definitions 5, 6).

Definition 4. (Prediction Change)

Let S_0^n be a stream; \mathcal{T} , \mathcal{A} and \mathcal{G} be TBox, ABox and its entailments. A prediction change in S_0^n is occurring between time i and j in $[0, n]$ with respect to \mathcal{T} , \mathcal{A} and its entailments iff:

$$\exists g \in \mathcal{G} : \|p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(i) \models g) - p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(j) \models g)\| \geq \varepsilon \quad (30)$$

where $\varepsilon \in (0, 1]$ is a variable bounding the difference of estimation, $\|v\|$ refers to the absolute value of v , and $j > i$

ABox entailment g is called an evidence entailment of the prediction change. We denote by $\mathbb{C}_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, j, \varepsilon)$, the set of all evidence entailments of the prediction change with an ε difference between time i and j of ontology stream S_0^n .

Example 5. (Prediction Change)

$g \doteq DisruptedRoad(r_2)$ can be entailed from \mathcal{T} and \mathcal{A} at time 2 of $(\mathcal{Q} \cup \mathcal{R})_0^n$ with a zero probability following (29). Therefore a prediction change between times 2 and 4 (cf. Example 4) is captured with $g \in \mathbb{C}_{|\mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n, 2, 4, 1/3)$.

Definition 5. (α -Sudden Prediction Change)

A prediction change at point of time i in stream S_0^n , satisfying (30), is defined as α -sudden, with $\alpha \in (0, n-i]$ iff $j = i + \alpha$.

Definition 6. (Abrupt Prediction Change)

A prediction change, satisfying (30), is abrupt iff $\exists g' \in \mathcal{G}$ s.t.

$$\mathcal{T} \cup \mathcal{A} \cup g \cup g' \bigcup_{k=0}^{\max\{i, j\}} S_0^n(k) \models \perp \quad (31)$$

where $\bigcup_{k=0}^{\max\{i, j\}} S_0^n(k)$ captures all axioms from any snapshot $S_0^n(k)$ of stream S_0^n with $k \in [0, \max\{i, j\}]$.

Suddenness characterises the proximity of prediction changes in streams i.e., the lower α the closer the changes. Abruptness captures disruptive changes from a semantic perspective i.e., conflicting knowledge among snapshots $S_0^n(i)$, $S_0^n(j)$ with respect to background knowledge $\mathcal{T} \cup \mathcal{A}$.

Definition 7. (Semantic Concept Drift)

A semantic concept drift in S_0^n , is defined as a 1-sudden and abrupt prediction change.

Evaluating if a concept drift occurs for a snapshot update is in worst case polynomial time with respect to acyclic TBoxes and S_0^n in \mathcal{EL}^{++} since subsumption and satisfiability in (30), (31) can be checked in polynomial time [Baader *et al.*, 2005].

Example 6. (Semantic Concept Drift)

Two prediction changes from time $i = 2$ to 3 and 3 to 4 (cf. Table 2) have occurred for $g \doteq DisruptedRoad(r_2)$ in $(\mathcal{Q} \cup \mathcal{R})_0^n$. They are semantic concept drifts as they are 1-sudden and abrupt with $g' \doteq ClearedRoad(r_2)$ in $(\mathcal{Q} \cup \mathcal{R})_0^n(1)$.

Past Points of Time	Time i	Prediction	Prediction Change	Abruptness
		$p_{ \mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n(i) \models g)$	$g \in \mathbb{C}_{ \mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n, i, i+1, 1/3)$	
$\{0\}$	1	0	\times	\times
$\{0, 1\}$	2	0	\checkmark	\checkmark
$\{0, 1, 2\}$	3	$1/2$	\checkmark	\checkmark
$\{0, 1, 2, 3\}$	4	$2/3$	N/A	N/A

Table 2: Prediction Changes in $(\mathcal{Q} \cup \mathcal{R})_0^n$ ($g \doteq Disrupted(r_2)$).

3.2 Significance of Concept Drift

Significance of semantic concept drift (Definition 8) is an indicator on its severity. It captures the homogeneity of the concept drift across ABox entailments as the proportion of ABox entailments from $S_0^n(i)$ and $S_0^n(i+1)$ causing semantic concept drift. The values of significance range in $[0, 1]$.

Definition 8. (Semantic Concept Drift Significance)

The significance of a semantic concept drift, defined between points of time $i \in (0, n)$ and $i+1$ of S_0^n with ε , \mathcal{T} , \mathcal{A} , \mathcal{G} as difference, TBox, ABox, and entailments, is $\sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, \varepsilon)$:

$$\frac{|\mathbb{C}_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, i+1, \varepsilon)|}{|\{g \in \mathcal{G} \mid \mathcal{T} \cup S_0^n(i) \models g \vee \mathcal{T} \cup S_0^n(i+1) \models g\}|} \quad (32)$$

where the expression in between $|$ refers to its cardinality.

Evaluating (32) is in worst case polynomial time cf. complexity of Definition 7.

Example 7. (Semantic Concept Drift Significance)

By applying (32) on concept drifts of Table 2 we derive that $\sigma_{|\mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n, 2, 1/3)$ is $4/7$ while $\sigma_{|\mathcal{T} \cup \mathcal{A}}((\mathcal{Q} \cup \mathcal{R})_0^n, 3, 1/3)$ is 0, hence a more significant drift between times 2, 3 than 3, 4. In other words conflicting facts $g \doteq DisruptedRoad(r_2)$ and $g' \doteq ClearedRoad(r_2)$ w.r.t. \mathcal{T} and \mathcal{A} have the most significant impact on prediction changes at times 2 and 3.

Lemma 1. (Semantic Concept Drift Evolution)

A semantic concept drift in any ontology stream S_0^n is more significant at time $i > 0$ than at time $i+1$ if $|\mathcal{G}_{new}^{[0, i], [0, i+1]}| = 0$.

Proof. (Sketch) Since $|\mathcal{G}_{new}^{[0, i], [0, i+1]}| = 0$, $S_0^n(i)$ and $S_0^n(i+1)$ are similar w.r.t. $\models_{\mathcal{T} \cup \mathcal{A}}$. Thus, the set of all entailments, predicted at $i+1$ and $i+2$ from (29), are similar but with different prediction values (30) $\forall \varepsilon \geq 0$. So $\sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, \varepsilon)$ and $\sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i+1, \varepsilon)$ in (32) have same denominators while $\mathbb{C}_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i+1, i+2, \varepsilon) \subseteq \mathbb{C}_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, i+1, \varepsilon)$ hence $\sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i+1, \varepsilon) \leq \sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, \varepsilon)$. \square

Algorithm 1 [A1] retrieves significant concept drifts in S_0^n with minimal significance σ_{\min} . [A1] iterates on all snapshots updates except those with no new ABox entailment (line 5 - lemma 1) for minimizing satisfiability and subsumption checking. Semantic concept drifts, as 1-sudden and abrupt prediction changes, are retrieved (line 7). [A1] completes the process (line 9) by filtering drifts by significance σ_{\min} .

Algorithm 1: [A1] SignificantDrift($\mathcal{O}, S_0^n, \varepsilon, \sigma_{\min}$)

```

1 Input: (i) Axioms  $\mathcal{O} : \langle \mathcal{T}, \mathcal{A}, \mathcal{G} \rangle$ , (ii) Ontology stream  $S_0^n$ ,
          (iii) Lower limit  $\varepsilon \in (0, 1]$  of prediction difference, (iv)
          Minimum threshold of drift significance  $\sigma_{\min}$ .
2 Result:  $\mathbb{S}$ : Significant concept drifts in  $S_0^n$  w.r.t.  $\sigma_{\min}$ .
3 begin
4    $\mathbb{S} \leftarrow \emptyset$ ; % Init. of the Significant concept drifts set.
5   foreach  $i \in (0, n]$  of  $S_0^n$  such that  $|\mathcal{G}_{new}^{[0,i],[0,i+1]}| \neq 0$  do
6     % Selection of 1-sudden, abrupt prediction changes.
7     if  $\exists g, g' \in \mathcal{G}$  such that:
8        $\|p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(i) \models g) - p_{|\mathcal{T} \cup \mathcal{A}}(S_0^n(i+1) \models g)\| \geq$ 
9        $\varepsilon$ 
10       $\wedge \mathcal{T} \cup \mathcal{A} \cup S_0^n(i) \cup S_0^n(i+1) \cup g \cup g' \models \perp$  then
11      % Semantic concept drift with min. significance.
12      if  $\sigma_{|\mathcal{T} \cup \mathcal{A}}(S_0^n, i, \varepsilon) \geq \sigma_{\min}$  then
13         $\mathbb{S} \leftarrow \mathbb{S} \cup \{(i, i+1)\}$  % Add snapshot update.
14 return  $\mathbb{S}$ ;
```

Computing a solution with [A1] given a polynomial input n , number of axioms, entailments in \mathcal{O} and S_0^n is in worst case polynomial time, due to the complexity of evaluating a semantic drift cf. complexity of Definition 7. However computing significant α -sudden, abrupt prediction changes following [A1] is in worst case NP w.r.t. the number snapshots.

4 Ontology Stream Learning

We tackle the ontology stream learning problem by (i) computing semantic embeddings, as mathematical objects exploiting the properties of concept drifts, (ii) applying all embeddings in model-based learning approaches (Algorithm 2).

4.1 Semantic Embeddings

The semantics of streams exposes two levels of knowledge which are crucial for learning with concept drift: (i) (in-)consistency evolution of knowledge, and (ii) entailment of the forecasting target from stream assertions and axioms. They are semantic embeddings, captured as: *consistency vectors* (Definition 9) and *entailment vector* (Definition 10).

Definition 9. (Consistency Vector)

A consistency vector of snapshot $S_0^n(i)$ in S_0^n , denoted by c_i , is defined $\forall j \in [0, n]$ by c_{ij} if $i < j$; c_{ji} otherwise such that:

$$c_{ij} \doteq \begin{cases} \frac{|\mathcal{G}_{inv}^{i,j}|}{|\mathcal{G}_{new}^{i,j}| + |\mathcal{G}_{inv}^{i,j}| + |\mathcal{G}_{obs}^{i,j}|} & \text{if } \mathcal{T} \cup S_0^n(i) \cup S_0^n(j) \not\models \perp \\ \frac{|\mathcal{G}_{inv}^{i,j}|}{|\mathcal{G}_{new}^{i,j}| + |\mathcal{G}_{inv}^{i,j}| + |\mathcal{G}_{obs}^{i,j}|} - 1 & \text{otherwise} \end{cases} \quad (33)$$

where the expressions in between $|$ refer to its cardinality i.e., the number of new (14), obsolete (15), invariant (16) ABox entailments from $S_0^n(i)$ to $S_0^n(j)$. $c_{ij} = c_{ji} \forall i, j \in [0, n]$.

A consistent vector, with values in $[-1, 1]^{n+1}$, encodes (i) (in-)consistency with (negative) positive values, and (ii) similarity of knowledge among $S_0^n(i)$ and any other snapshot $S_0^n(j)_{j \in [0, n]}$ of stream S_0^n w.r.t axioms \mathcal{T} and \mathcal{A} . The number of invariant entailments has a positive influence on (33). 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 instead of considering its additive inverse. This ensures that the invariant factor has always a positive impact.

Evaluating (33) is in worst case polynomial time with respect to \mathcal{T} and S_0^n in \mathcal{EL}^{++} . Indeed its evaluation requires (i) ABox entailment, and (ii) basic set theory operations from Definition 2, both in polynomial time [Baader et al., 2005].

Example 8. (Consistency Vector)

Consistency vector c_3 i.e., $(c_{03}, c_{13}, c_{23}, c_{33})$ of $(\mathcal{Q} \cup \mathcal{R})_0^n(3)$ is $(0, -0.8, 1, 1)$. Knowledge at time 3 is consistent / inconsistent / similar with knowledge at times 0 / 1 / 2 and 3.

An entailment vector (Definition 10) is adapting the concept of feature vector [Bishop, 2006] in Machine Learning to represent the (non-)presence of all ABox entailments (using \models w.r.t. \mathcal{T}, \mathcal{A}) in a given snapshot. Each dimension captures whether a particular ABox entailment is in (1) or not (0).

Definition 10. (Entailment Vector)

Let $\mathcal{G} \doteq \{g_1, \dots, g_m\}$ be all distinct ABox entailments in S_0^n . An entailment vector of a snapshot $S_0^n(i)$ in S_0^n , denoted by e_i , is a vector of dimension m such that $\forall j \in [0, m]$

$$e_{ij} \doteq 1 \text{ if } \mathcal{T} \cup \mathcal{A} \cup S_0^n(i) \models g_j, 0 \text{ otherwise} \quad (34)$$

Remark 1. (Feature vs. Entailment Vector)

Feature vectors are bounded to only raw data while entailment vectors, with much larger dimensions, embed both data and its inferred assertions from \mathcal{T} and DL completion rules. The latter ensures a larger and more contextual coverage.

4.2 Semantic Prediction

Algorithm 2 [A2] aims at learning a model (line 12) over $N \leq n+1$ snapshots of S_0^n , noted $S_0^n|_{\kappa}$, for prediction at $n+1$. κ refers to the proportion of snapshots with concept drift used for modelling. $S_0^n|_{\kappa}$ is selected to capture (i) $S_0^n(n)$ i.e., the closest (temporally) to $S_0^n(n+1)$ (line 4), (ii) knowledge in the most (lines 8-9) significant concept drifts (Definition 8 - line 5), (iii) any other snapshots to meet N (line 11).

The model is trained, following Stochastic Gradient Descent method [Zhang, 2004], using samples of the form $\{(e_i, g_i) \mid i \in \{1, \dots, N\}\}$ where e_i is the entailment vector for $S_0^n(i)$ and $v(g_i)$ is the target variable in $[0, 1]$, capturing the estimation of g_i to be entailed. The model is represented by a linear scoring function $f(e_i) = a^T e_i + b$ with model parameters $a \in \mathbf{R}^N$ and $b \in \mathbf{R}$. The goal of learning is to minimize the following objective function O_j :

$$O_j(a, b) \doteq \sum_{i=1}^{\kappa} \omega_{ij} L(v(g_i), f(e_i)) + \alpha R(a), \quad (35)$$

where L represents the loss function (e.g., hinge and log). R and α control the variance of the model in case of over fitting. R is a regularization term and $\alpha > 0$ is a non-negative hyperparameter. Each sample (e_i, g_i) in (35) is weighted by ω_{ij} in (36) (resp. (37)) for filtering out consistent (resp. inconsistent) historical snapshots w.r.t. (33).

$$\omega_{ij} \doteq \begin{cases} 0, & \text{if } c_{ij} > 0 \\ -c_{ij} & \text{else,} \end{cases} \quad (36) \quad \omega_{ij} \doteq \begin{cases} 0, & \text{if } c_{ij} < 0 \\ c_{ij} & \text{else,} \end{cases} \quad (37)$$

Algorithm 2: [A2]PredictionModel($\mathcal{O}, \mathcal{S}_0^n, \varepsilon, \sigma_{\min}, \kappa, \mathbf{N}$)

```

1 Input: (i) Axioms  $\mathcal{O} : \langle \mathcal{T}, \mathcal{A}, \mathcal{G} \rangle$ , (ii) Stream  $\mathcal{S}_0^n$ , (iii) Lower
    limit  $\varepsilon \in (0, 1]$ , (iv) Minimum drift significance  $\sigma_{\min}$ ,
    (v) Proportion  $\kappa$  of snapshots with concept drift used
    for modelling, (vi) Number of snapshots  $\mathbf{N}$ .
2 Result:  $f$ : Model for prediction at time  $n+1$  of  $\mathcal{S}_0^n$ .
3 begin
4    $\mathcal{S}_0^n|_{\kappa} \leftarrow \{n\}$ ; % Initial snapshots' set for learning model.
5   % Computation of the most significant drifts w.r.t.  $\varepsilon, \sigma_{\min}$ .
6    $\mathbb{S} \leftarrow \text{SignificantDrift}(\mathcal{O}, \mathcal{S}_0^n, \varepsilon, \sigma_{\min})$ ;
7   % Selection of  $\kappa/\mathbf{N}$  snapshots involved in concept drifts  $\mathbb{S}$ .
8   foreach  $i \in [0, n]$  s.t.  $(i, i+1) \in \mathbb{S} \wedge |\mathcal{S}_0^n|_{\kappa}| < \kappa/\mathbf{N}$  do
9      $\mathcal{S}_0^n|_{\kappa} \leftarrow \mathcal{S}_0^n|_{\kappa} \cup \{i\}$ ;
10  % Expand  $|\mathcal{S}_0^n|_{\kappa}|$  with snapshots not involved in  $\mathbb{S}$ .
11  add  $1-\kappa/\mathbf{N}$  point(s) of time  $i$  to  $\mathcal{S}_0^n|_{\kappa}$  s.t.  $(i, i+1) \notin \mathbb{S}$ ;
12  % Learning model  $f$  using (35) with weight (36) or (37).
    (i)  $\min_{(a,b) \in \mathbf{R}^N \times \mathbf{R}} \sum_{i=1}^N \omega_{ij} L(\mathbf{v}(g_i), f(\mathbf{e}_i)) + \alpha R(a)$ 
    (ii)  $f(\mathbf{e}_i) = a^T \mathbf{e}_i + b$ 
13  return  $f$ ;

```

[A1–2] parameterized with low ε , σ_{\min} , high κ and (36) as weight (line 12 (i)) favours models with significant concept drifts for prediction, which supports diversity and prediction changes in the model. Parameterized with high ε , σ_{\min} , low κ and (37) as weight, it will capture more consistent models.

The linear scoring function f in (35) has the following advantages compared to more complex structures such as artificial neural network: (i) better handling over-fitting with reduced sample size - due to filtering of snapshots not involved in significant concept drifts (lines 8–9 in [A2]), (ii) ensuring efficient, scalable learning and prediction for online contexts.

5 Experimental Results

We report accuracy by (i) studying the impact of [A2] and semantic embeddings on concept drift for *Dublin-Ireland*, *Beijing-China* applications, and (ii) comparing its results with state-of-the-art approaches. The system is tested on: 16 Intel(R) Xeon(R) CPU E5-2680, 2.80GHz cores, 32GB RAM.

• **Beijing Air Quality (BAQ) Context:** BAQ index, ranging from Good (value 5), Moderate (4), Unhealthy (3), Very Unhealthy (2) to Hazardous (1), can be forecasted using data streams of B_1 : air pollutants and meteorology elements B_2 : wind speed, B_3 : humidity observed in 12 sites. The context variation, characterising a concept drift problem, makes BAQ index difficult to be forecasted. The semantics of context is based on a DL \mathcal{A}_{CC} ontology, including 48 concepts, 13 roles, 598 axioms. An average of 6, 500 RDF triples are generated at each update (i.e., every 600 seconds) for the streams.

• **Dublin Bus Delay (DBD) Context:** DBD, classified as Free (value 5), Low (4), Moderate (3), Heavy (2), Stopped (1) can be forecasted using reputable live stream contextual data (Table 3) related to D_1 : bus GPS location, delay, congestion

status, D_2 : weather conditions, D_3 : road incidents. However bus delay is subject to major changes due the high degree of context variation. The latter, responsible for the concept drift problem, impacts accuracy the most. We consider an extended settings by enriching data using a DL \mathcal{EL}^{++} domain ontology (55 concepts, 19 roles and 25, 456 axioms).

Feature DataSet	Size (Mb) per day	Frequency of Update (seconds)	#Axioms per Update	#RDF Triples per Update
D_1 : Bus	120	40	3,000	12,000
D_2 : Weather	3	300	53	318
D_3 : Incident	0.1	600	81	324

Table 3: Datasets Details of Dublin Bus Delay Context.

• **Validation:** Accuracy is measured by comparing predictions with real-world observations in cities.

• **Semantic Impact:** Table 4 reports the positive impact of using semantic embeddings (cf. columns with \checkmark) on all forecasting tasks, with an average improvement of 26.6%. The embeddings naturally identify semantically (dis-)similar contexts by capturing temporal (in-)consistency(ies). Thus, they help in building discriminating models, even for long-term-ahead forecasting as shown for $\Delta = 18$ -hours with a 33.1% gain. The difference of results between Beijing and Dublin confirms the importance of semantic expressivity i.e., 40+ times more axioms with a 71.5% gain of accuracy for Dublin. Computing semantic embedding (i.e., consistent vector / entailment vector) is on average 2.95s / 1.01s and 1.82 / 0.65s in respectively Dublin and Beijing contexts.

City	$\mathcal{ID} : \text{Features}$	$\Delta=6$ hours		$\Delta=12$ hours		$\Delta=18$ hours	
		\times	\checkmark	\times	\checkmark	\times	\checkmark
Beijing	$B_1 : B_1$.351	.398	.344	.441	.261	.342
	$B_2 : B_1 + B_2$.398	.449	.350	.453	.279	.371
	$B_3 : B_1 + B_3$.421	.508	.373	.459	.282	.379
	$B_4 : B_1 + B_2 + B_3$.501	.611	.389	.478	.286	.393
	Average Improvement (%)	17.206		25.890		33.954	
Dublin	$D_1 : D_1$.455	.514	.387	.441	.321	.387
	$D_2 : D_1 + D_2$.534	.688	.499	.553	.361	.497
	$D_3 : D_1 + D_3$.601	.701	.513	.645	.371	.547
	$D_4 : D_1 + D_2 + D_3$.659	.921	.533	.834	.601	.745
	Average Improvement (%)	24.550		26.744		32.408	

Table 4: Forecasting Accuracy of [A1] with (\checkmark) and without (\times) Semantic Embeddings in Beijing and Dublin Contexts.

• **Feature Impact:** Table 4 emphasises an extra accuracy gain when increasing the number of features i.e., average gain of 68.5% accuracy from 1 to 3 features.

• **Concept Drift** is characterised by 48% and 51% of stream updates in respectively BAQ and DBD. We focus on 4 levels of concept drifts, ranging from a .2 to .8 significance $\forall \Delta \in \{6, 12, 18\}$. Level 0 does not capture any change. Figure 3 reports the proportion of severity levels in concept drift for BAQ and DBD e.g., 7% are level-.4 for BAQ while 19% are level-.8 for DBD. Although accuracy clearly declined by increasing the severity level e.g., from 96% (level-.2) to 21% (level-.8) in DBD, semantic embeddings has shown to significantly boost accuracy. More interestingly the more severity the higher improvement i.e., (average) 36% to 56% on level-.4 to .8. Thus integrating semantics is a way forward to build machine learning models which are robust to changes, potential erroneous sensor data and concept drifts.

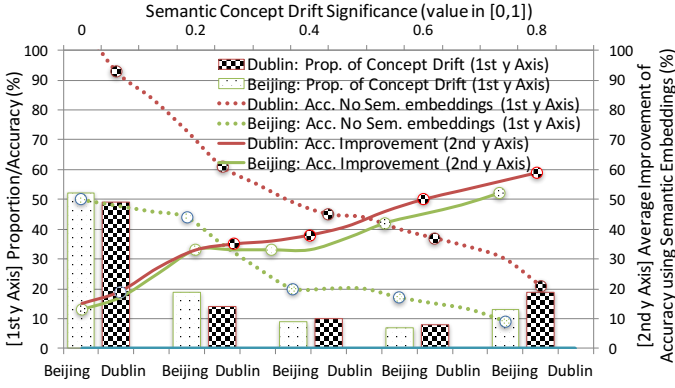


Figure 3: Forecasting Accuracy vs. Drift Significance.

• **Model Consistency Impact:** Figures 4 and 5 report accuracy of forecasting tasks on a High and Low Concept Drift versions of the Dublin and Beijing problems, noted HCD and LCD. 85% and 15% of snapshots are impacted by concept drift respectively in HCD and LCD.

[A1-2] is evaluated with 3 variances of $(\varepsilon, \sigma_{\min}, \kappa)$: (i) consistent model with $(.9, .9, .1)$, (ii) mixed model with $(.5, .5, .5)$, (iii) inconsistent model with $(.1, .1, .9)$. $N = 1,500$. Figure 4 (resp. 5) reports that prediction with consistent (resp. inconsistent) samples outperforms models with inconsistent (resp. consistent) samples by about 318% (resp. 456%) and 254% (resp. 322%) in respectively Beijing and Dublin for LCD (resp. HCD). These results confirm the importance of semantic encoding, which support the encoding of concept drift and consistency properties in our approach.

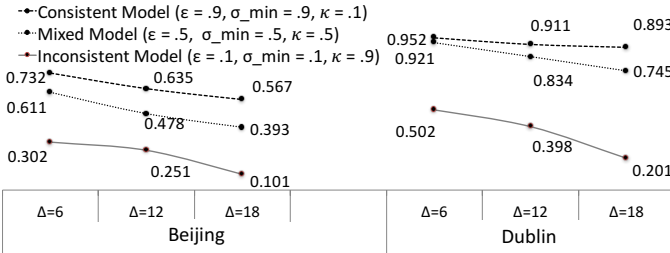


Figure 4: Model Consistency & Forecasting Accuracy. Low Concept Drift. (15% of snapshots impacted by concept drift).

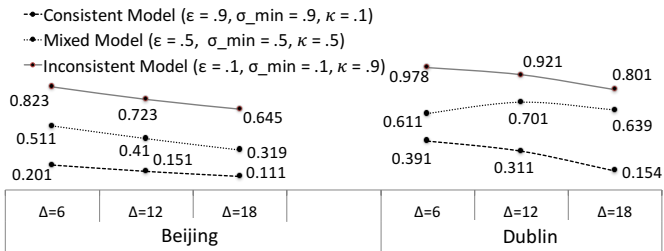


Figure 5: Model Consistency & Forecasting Accuracy. High Concept Drift. (85% of snapshots impacted by concept drift).

• **Baseline:** We compare our approach $\mathcal{B}_i, \mathcal{D}_{i,1 \leq i \leq 4}$ in Table 4 with (i) weighted Stochastic Gradient Descent (SGD), (ii) Auto-Regressive Integrated Moving Average (ARIMA), a standard time-series forecasting model [Saboia, 1977], and two methods addressing concept drift: (iii) Hoeffding Adaptive Tree (HAT), (iv) Leveraging Bagging (LB) [Bifet

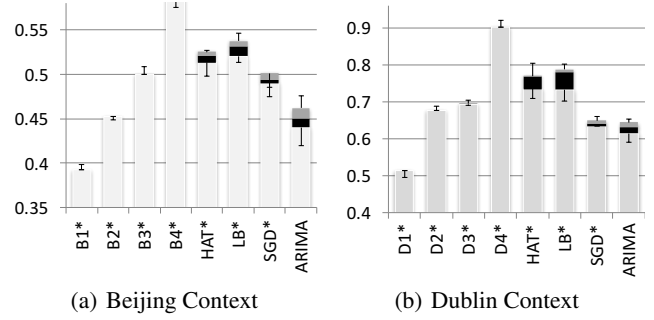


Figure 6: Baseline Comparison of Forecasting Accuracy. ($\Delta = 6$, Approach*: Approach with all features).

et al., 2010b; 2010a]. ARIMA considers one stream variable: BAQ index for Beijing and DBD for Dublin while SGD, HAT and LB use all features of $\mathcal{B}_4, \mathcal{D}_4$ and favour recent snapshots during learning. The forecasted real value in $[0, 5]$ is discretised back using our categories. Results with optimum parameters for [A1-2] are reported. Figure 6 emphasises that our approach (with 3 levels of features: $\mathcal{B}_4, \mathcal{D}_4$) over-performs state-of-the-art methods. The more features the more accurate. More interestingly classic learning algorithms do not generalise as well as [A1-2] in presence of semantics although SGD, HAT, LB integrate all features. [A1-2] shows to be very robust with less variance. Experiments also demonstrate that semantic (in-)consistency [Flouris *et al.*, 2006] matters more than recentness during learning.

• **Lessons Learnt:** Adding semantics to classic learning model has clearly shown the positive impact on accuracy, specially in presence of concept drifts. Our approach also demonstrates that the more semantic axioms the more robust is the model and hence the higher the accuracy. Axiom numbers are critical as they drive and control the semantics of data in streams, which improve accuracy, concept drift detection but not scalability (not reported in the paper). It is worst with more expressive DLs due to consistency checks, and with limited impact on accuracy. Lightweight semantics such as RDF-S would highly limit the scope of our model given the omission of inconsistency checking cf. Figures 4-5.

6 Conclusion

We tackled the problem of learning with concept drifts. Semantic reasoning and machine learning have been combined by revisiting features embeddings as semantic embeddings. Such embeddings are exploited in a context of supervised stream learning to learn models, which are robust to concept drifts. Our approach has been shown to be adaptable and flexible to basic learning algorithms. Besides demonstrating accurate prediction with concept drifts in Dublin and Beijing forecasting applications, experiments have shown that encoding semantics in models is a way towards outperforming state-of-the-art approaches. We will investigate the impact of semantic embeddings in other Machine Learning models.

Acknowledgments

This work is funded by the Alibaba-ZJU joint project on e-Business Knowledge Graph, NSFC 61473260/61673338/61672393 and the EU Marie Curie IAPP K-Drive project (286348).

References

- [Agrawal *et al.*, 1996] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.
- [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.
- [Beck *et al.*, 2016] Harald Beck, Minh Dao-Tran, and Thomas Eiter. Equivalent stream reasoning programs. In *IJCAI*, pages 929–935, 2016.
- [Bifet *et al.*, 2010a] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.
- [Bifet *et al.*, 2010b] Albert Bifet, Geoff Holmes, and Bernhard Pfahringer. Leveraging bagging for evolving data streams. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, ECML PKDD’10, pages 135–150, Berlin, Heidelberg, 2010. Springer-Verlag.
- [Bifet *et al.*, 2015] Albert Bifet, Gianmarco de Francisci Morales, Jesse Read, Geoff Holmes, and Bernhard Pfahringer. Efficient online evaluation of big data stream classifiers. In *SIGKDD*, pages 59–68. ACM, 2015.
- [Bishop, 2006] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.
- [Cheung *et al.*, 1996] David W Cheung, Jiawei Han, Vincent T Ng, and CY Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In *ICDE*, pages 106–114. IEEE, 1996.
- [Chu *et al.*, 2011] Wei Chu, Martin Zinkevich, Lihong Li, Achint Thomas, and Belle Tseng. Unbiased online active learning in data streams. In *SIGKDD*, pages 195–203. ACM, 2011.
- [Coble and Cook, 2000] Jeffrey Coble and Diane J. Cook. Real-time learning when concepts shift. In *Proceedings of the 13th International Florida Artificial Intelligence Research Society Conference*, pages 192–196, 2000.
- [Domingos and Hulten, 2000] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *SIGKDD*, pages 71–80. ACM, 2000.
- [Flouris *et al.*, 2006] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, Negations and Changes in Ontologies. In *AAAI*, pages 1295–1300, 2006.
- [Galárraga *et al.*, 2013] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, pages 413–422. ACM, 2013.
- [Gama and Kosina, 2011] João Gama and Petr Kosina. Learning decision rules from data streams. In *IJCAI*, pages 1255–1260, 2011.
- [Gama *et al.*, 2014] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.
- [Gao *et al.*,] Jing Gao, Wei Fan, and Jiawei Han. On appropriate assumptions to mine data streams: Analysis and practice. In *ICDM*.
- [Gao *et al.*, 2007] Jing Gao, Wei Fan, Jiawei Han, and S Yu Philip. A general framework for mining concept-drifting data streams with skewed distributions. In *SDM*, pages 3–14. SIAM, 2007.
- [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.
- [Lecue and Pan., 2013] Freddy Lecue and Jeff Z. Pan. Predicting Knowledge in An Ontology Stream. In *IJCAI*, 2013.
- [Lécué, 2015] Freddy Lécué. Scalable maintenance of knowledge discovery in an ontology stream. In *IJCAI*, pages 1457–1463, 2015.
- [Lee *et al.*, 2003] Chang-Hung Lee, Ming-Syan Chen, and Cheng-Ru Lin. Progressive partition miner: An efficient algorithm for mining general temporal association rules. *IEEE Trans. Knowl. Data Eng.*, 15(4):1004–1017, 2003.
- [Ren and Pan, 2011] Yuan Ren and Jeff Z. Pan. Optimising Ontology Stream Reasoning with Truth Maintenance System. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM 2011)*, 2011.
- [Ren *et al.*, 2011] Yuan Ren, Jeff Z. Pan, and Kevin Lee. Parallel ABox Reasoning of EL Ontologies. In *Proc. of the First Joint International Conference of Semantic Technology (JIST 2011)*, 2011.
- [Ren *et al.*, 2016] Yuan Ren, Jeff Z. Pan, Isa Guclu, and Martin Kollingbaum. A Combined approach to Incremental Reasoning for EL Ontologies. In *Proc. of the 10th International Conference on Web Reasoning and Rule Systems (RR2016)*, 2016.
- [Saboia, 1977] Joao Luiz Maurity Saboia. Autoregressive integrated moving average (arima) models for birth forecasting. *Journal of the American Statistical Association*, 72(358):264–270, 1977.
- [Zhang, 2004] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, page 116. ACM, 2004.