Logic-Based Ranking of Assertions in Inconsistent ABoxes.

Horacio Tellez Jef Wijsen

Département d'Informatique
University of Mons, Belgium

Dutch-Belgian Database Day, Brussels Belgium (Online)

Description Logics

Knowledge base framework

- ABox: contains the data;
- TBox: group of axioms that provide structure and allow for inference on the ABox;
- interesting complexity properties;
- can vary between OWA and CWA;

Motivation

Description Logics and Inconsistency

Informally, inconsistency can occur in two forms in DL knowledge bases:

- TBox unsatisfiability (i.e., errors in the axioms), and
- ABox inconsistency (i.e., errors in concept/role assertions).

Our work assumes that the TBox is error-free, and deals exclusively with errors in ABoxes:

The ABox Inconsistency Problem

How to solve inconsistency present in an ABox?

Motivation

Description Logics and Inconsistency

Informally, inconsistency can occur in two forms in DL knowledge bases:

- TBox unsatisfiability (i.e., errors in the axioms), and
- ABox inconsistency (i.e., errors in concept/role assertions).

Our work assumes that the TBox is error-free, and deals exclusively with errors in ABoxes:

The ABox Inconsistency Problem

How to solve inconsistency present in an ABox?

Same questions

Two main approaches

- Data cleaning
- Inconsistency-tolerant semantics

What is 'good' data ?

How to choose which piece of data is more trustful ?
How can we quantify the quality of assertions in an ABox ?

Same questions

Two main approaches

- Data cleaning
- Inconsistency-tolerant semantics

What is 'good' data ?

How to choose which piece of data is more trustful ?
How can we quantify the quality of assertions in an ABox ?

Toward a tool for elucidating choices

To choose means to compare

- We need a flexible comparison tool,
- that is independent of the underlying DL, and
- that, ideally, has tractable complexity.

ABox Assessment

For a fixed TBox $\mathcal{T}$, an ABox assessment of an ABox $\mathcal{A}$ is a function

$$\nu : \mathcal{A} \to \mathbb{R}.$$

We are merely interested in the preorder on $\mathcal{A}$ induced by ν: whenever we have to choose among two assertions, the one with the higher ν-value is preferred.

Toward a tool for elucidating choices

To choose means to compare

- We need a flexible comparison tool,
- that is independent of the underlying DL, and
- that, ideally, has tractable complexity.

ABox Assessment

For a fixed TBox $\mathcal{T}$, an ABox assessment of an ABox $\mathcal{A}$ is a function

$$\nu : \mathcal{A} \to \mathbb{R}.$$

We are merely interested in the preorder on $\mathcal{A}$ induced by ν: whenever we have to choose among two assertions, the one with the higher ν-value is preferred.

Rank-equivalence

Two ABox assessments ν_1 and ν_2 are rank-equivalent if for all $\alpha, \beta \in \mathcal{A}$,

$$\nu_1(\alpha) > \nu_1(\beta) \text{ if and only if } \nu_2(\alpha) > \nu_2(\beta).$$

We are merely interested in the family of ABox assessments modulo rank-equivalence.

Desirable properties of ABox assessments

ABox assessments should

- be capable of assessing [the quality of] individual assertions (possibly extensible to sets of assertions);
- be capable of "booting" from an initial assessment provided by domain experts (called credibility function); and
- be capable of using the TBox axioms to find arguments pro/contra assertions.

Desirable properties of ABox assessments

ABox assessments should

- be capable of assessing [the quality of] individual assertions (possibly extensible to sets of assertions);
- be capable of “booting” from an initial assessment provided by domain experts (called credibility function); and
- be capable of using the TBox axioms to find arguments pro/contra assertions.

Desirable properties of ABox assessments

ABox assessments should

- be capable of assessing [the quality of] individual assertions (possibly extensible to sets of assertions);
- be capable of "booting" from an initial assessment provided by domain experts (called credibility function); and
- be capable of using the TBox axioms to find arguments pro/contra assertions.

Specification of ABox assessments I

Booting from users' expertise ⇝ Credibility function

A credibility function maps ABox assertions to real numbers, and can be extended, using some form of aggregation, to a mapping

$$f : 2^{\mathcal{A}} \rightarrow \mathbb{R}.$$

Using TBox axioms ⇝ Supporters and refuters

Let $(\mathcal{T}, \mathcal{A})$ be a knowledge base in some DL language that supports negation. Let $\alpha \in \mathcal{A}$ and $\alpha \notin B \subseteq \mathcal{A}$. Then,

- B is called a supporter of α if B is an inclusion-minimal subset of $\mathcal{A}$ such that $(\mathcal{T}, B)$ is consistent and entails α;
- B is a refuter of α if B is an inclusion-minimal subset of $\mathcal{A}$ such that $(\mathcal{T}, B)$ is consistent and entails $\neg\alpha$.

Specification of ABox assessments I

Booting from users' expertise $\rightsquigarrow$ Credibility function

A credibility function maps ABox assertions to real numbers, and can be extended, using some form of aggregation, to a mapping

$$f : 2^{\mathcal{A}} \rightarrow \mathbb{R}.$$

Using TBox axioms $\rightsquigarrow$ Supporters and refuters

Let $(\mathcal{T}, \mathcal{A})$ be a knowledge base in some DL language that supports negation. Let $\alpha \in \mathcal{A}$ and $\alpha \notin B \subseteq \mathcal{A}$. Then,

- B is called a supporter of α if B is an inclusion-minimal subset of $\mathcal{A}$ such that $(\mathcal{T}, B)$ is consistent and entails α;
- B is a refuter of α if B is an inclusion-minimal subset of $\mathcal{A}$ such that $(\mathcal{T}, B)$ is consistent and entails $\neg\alpha$.

Specification of ABox assessments II

$\rightsquigarrow$ An affine relationship

Given knowledge base $(\mathcal{T}, \mathcal{A})$, we want ν to satisfy, for every $\alpha \in \mathcal{A}$:

$$a * \nu(\alpha) = c + b * \begin{pmatrix} \Sigma_{B \text{ is a supporter}} f(B) * (\Sigma_{\beta \in B} \nu(\beta)) \\ -\Sigma_{B \text{ is a refuter}} f(B) * (\Sigma_{\beta \in B} \nu(\beta)) \end{pmatrix}$$

where $a, b, c \in \mathbb{R}$ are control parameters.

Goal: Find an ABox assessment ν from this system of equalities

Construction of a system of linear equations

- Let $\mathcal{A} = \{\alpha_1, \ldots, \alpha_N\}$;
- we can define a matrix A such that solution vector x of the following system:

$$(a * \mathbb{1} - b * A) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \qquad (1)$$

has the following property:

$$(x_1, ..., x_N) = (\nu(\alpha_1), ..., \nu(\alpha_N)).$$

Goal(redefined): solve the linear system

Construction of a system of linear equations

- Let $\mathcal{A} = \{\alpha_1, \ldots, \alpha_N\}$;
- we can define a matrix A such that solution vector x of the following system:

$$(a * \mathbb{1} - b * A) \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} c \\ c \\ \vdots \\ c \end{bmatrix} \tag{1}$$

has the following property:

$$(x_1, ..., x_N) = (\nu(\alpha_1), ..., \nu(\alpha_N)).$$

Goal(redefined): solve the linear system

Research questions

- About the system of linear equations:
  - When does the system (1) have a [unique] solution?
  - Are solutions obtained for different values of the parameters (a, b, c) rank-equivalent?
- About the matrix A:
  - What is the computational complexity of computing A?
  - How does this complexity depend on the underlying DL?
  - Can we identify tractable cases of practical interest?

We provide partial answers to these questions.

Research questions

- About the system of linear equations:
  - When does the system (1) have a [unique] solution?
  - Are solutions obtained for different values of the parameters (a, b, c) rank-equivalent?
- About the matrix A:
  - What is the computational complexity of computing A?
  - How does this complexity depend on the underlying DL?
  - Can we identify tractable cases of practical interest?

We provide partial answers to these questions.

Main Technical Results

- Existence: for every fixed value of $b > 0$, there is a value for a with $a > b$ such that the above system of equalities has a unique ABox assessment ν as its solution.
- Convergence: there is a computable threshold value $t^* > 0$ such that all parametrizations satisfying $a/b \geq t^*$ yield a unique ABox assessment *modulo rank-equivalence*.
- Polynomial-time computable: If the TBox or DL allows for an upper bound on the cardinality of supporters/refuters, then this unique ABox assessment *modulo rank-equivalence* can be computed in polynomial time.

Main Technical Results

- Existence: for every fixed value of $b > 0$, there is a value for a with $a > b$ such that the above system of equalities has a unique ABox assessment ν as its solution.
- Convergence: there is a computable threshold value $t^* > 0$ such that all parametrizations satisfying $a/b \geq t^*$ yield a unique ABox assessment *modulo rank-equivalence*.
- Polynomial-time computable: If the TBox or DL allows for an upper bound on the cardinality of supporters/refuters, then this unique ABox assessment *modulo rank-equivalence* can be computed in polynomial time.

Main Technical Results

- Existence: for every fixed value of $b > 0$, there is a value for a with $a > b$ such that the above system of equalities has a unique ABox assessment ν as its solution.
- Convergence: there is a computable threshold value $t^* > 0$ such that all parametrizations satisfying $a/b \geq t^*$ yield a unique ABox assessment *modulo rank-equivalence*.
- Polynomial-time computable: If the TBox or DL allows for an upper bound on the cardinality of supporters/refuters, then this unique ABox assessment *modulo rank-equivalence* can be computed in polynomial time.

Effortless Adaptation to the Database Case

Key points on ontologies

- data: ABox $\mathcal{A}$;
- conflict modeling: TBox $\mathcal{T}$.

Database side

- data: relational database db;
- conflict modeling: integrity constraints Σ.

Bonus

All results previously mentioned hold.

Effortless Adaptation to the Database Case

Key points on ontologies

- data: ABox $\mathcal{A}$;
- conflict modeling: TBox $\mathcal{T}$.

Database side

- data: relational database db;
- conflict modeling: integrity constraints Σ.

Bonus

All results previously mentioned hold.

Effortless Adaptation to the Database Case

Key points on ontologies

- data: ABox $\mathcal{A}$;
- conflict modeling: TBox $\mathcal{T}$.

Database side

- data: relational database db;
- conflict modeling: integrity constraints Σ.

Bonus

All results previously mentioned hold.

Contributions and Future Work

Contributions

- A generic framework for ranking ABox assertions in terms of their "quality" (consistency, truthfulness. . .)
  - booted from any expert-provided credibility function; and
  - taking into account TBox axioms in any logic.
- Founded in linear algebra.
- Shown to be computationally tractable in a setting of practical relevance.

Future work

- Implement and test our framework in real-life applications;
- investigate relationships between our framework and existing inconsistency-tolerant semantics;
- explore the possibility of hybrid semantics, combining our framework with existing semantics.

Contributions and Future Work

Contributions

- A generic framework for ranking ABox assertions in terms of their "quality" (consistency, truthfulness...)
  - booted from any expert-provided credibility function; and
  - taking into account TBox axioms in any logic.
- Founded in linear algebra.
- Shown to be computationally tractable in a setting of practical relevance.

Future work

- Implement and test our framework in real-life applications;
- investigate relationships between our framework and existing inconsistency-tolerant semantics;
- explore the possibility of hybrid semantics, combining our framework with existing semantics.

Thanks!