Wheat and Chaff – Practically Feasible Interactive Ontology Revision

Nadeschda Nikitina, Birte Glimm, Sebastian Rudolph

International Semantic Web Conference 2011, Bonn
Motivation

Noisy domain statements produced by knowledge acquisition and integration methods

Should axiom $\alpha$ be a consequence of the target ontology?
**Revision Example**

**Assumption:** Deductive closure of the intended consequences must not contain unintended consequences

\[(\alpha): \text{Decision} \rightarrow \text{:DistinctEntity}\]

\[(\beta): \text{Decision} \rightarrow \text{:MentalEntity}\]

\[(\gamma): \text{Decision} \rightarrow \text{:MentalObject}\]
Revision Example

Assumption: Deductive closure of the intended consequences must not contain unintended consequences

\[(\alpha) \text{:Decision} \rightarrow \text{:DistinctEntity}\]

\[(\beta) \text{:Decision} \rightarrow \text{:MentalEntity}\]

\[(\gamma) \text{:Decision} \rightarrow \text{:MentalObject}\]
Assumption: Deductive closure of the intended consequences must not contain unintended consequences

1. (γ) :Decision ➔ :MentalObject
2. (β) :Decision ➔ :MentalEntity
3. (α) :Decision ➔ :DistinctEntity
Assumption: Deductive closure of the intended consequences must not contain unintended consequences.

1. (γ) :Decision → :MentalObject
2. (β) :Decision → :MentalEntity
3. (α) :Decision → :DistinctEntity
**Assumption:** Deductive closure of the intended consequences must not contain unintended consequences

\[
\alpha): \text{Decision} \rightarrow \text{:DistinctEntity}
\]

\[
\beta): \text{Decision} \rightarrow \text{:MentalEntity}
\]

\[
\gamma): \text{Decision} \rightarrow \text{:MentalObject}
\]
Assumption: Deductive closure of the intended consequences must not contain unintended consequences

1. \((\alpha)\): Decision \rightarrow :\text{DistinctEntity}

\((\beta)\): Decision \rightarrow :\text{MentalEntity}

\((\gamma)\): Decision \rightarrow :\text{MentalObject}
**Assumption:** Deductive closure of the intended consequences must not contain unintended consequences

1.  
   - (α): Decision → :DistinctEntity

2.  
   - (β): Decision → :MentalEntity

   - (γ): Decision → :MentalObject
Revision Example

**Assumption:** Deductive closure of the intended consequences must not contain unintended consequences

1. \( (\alpha) \) : Decision → : DistinctEntity
2. \( (\beta) \) : Decision → : MentalEntity
3. \( (\gamma) \) : Decision → : MentalObject
Assumption: Deductive closure of the intended consequences must not contain unintended consequences

1. (α): Decision → DistinctEntity

2. (β): Decision → MentalEntity

3. (γ): Decision → MentalObject
Revision Example

Assumption: Deductive closure of the intended consequences must not contain unintended consequences

- A single evaluation decision can predetermine the decision for several yet unevaluated axioms
- Order influences method effectiveness
A revision state is defined as a tuple \((\mathcal{K}, \mathcal{K}^\models, \mathcal{K}^\not\models)\) of knowledge bases with 
\[ \mathcal{K}^\models \subseteq \mathcal{K}, \emptyset \neq \mathcal{K}^\not\models \subseteq \mathcal{K}, \text{ and} \]
\[ \mathcal{K}^\models \cap \mathcal{K}^\not\models = \emptyset. \]

- \(\mathcal{K}^\models\): the set of desired consequences
- \(\mathcal{K}^\not\models\): the set of undesired consequences
A revision state is complete, if $\mathcal{K} = \mathcal{K}^\models \cup \mathcal{K}^\not\models$, and incomplete otherwise.
A revision state is complete, if $\mathcal{K} = \mathcal{K}^\vdash \cup \mathcal{K}^\not\vdash$, and incomplete otherwise.

A revision state $(\mathcal{K}, \mathcal{K}^\vdash, \mathcal{K}^\not\vdash)$ is consistent if there is no $\alpha \in \mathcal{K}^\not\vdash$ such that $\mathcal{K}^\vdash \models \alpha$. 
Revision States

Given two revision states \((K, K_1^\sqsubset, K_1^{\not\sqsubset})\) and \((K, K_2^\sqsubset, K_2^{\not\sqsubset})\), we call \((K, K_2^\sqsubset, K_2^{\not\sqsubset})\) a refinement of \((K, K_1^\sqsubset, K_1^{\not\sqsubset})\), if \(K_1^\sqsubset \subseteq K_2^\sqsubset\) and \(K_1^{\not\sqsubset} \subseteq K_2^{\not\sqsubset}\).
Revision States

Given two revision states \((\mathcal{K}, \mathcal{K}_1^\models, \mathcal{K}_1^\not\models)\) and \((\mathcal{K}, \mathcal{K}_2^\models, \mathcal{K}_2^\not\models)\), we call \((\mathcal{K}, \mathcal{K}_2^\models, \mathcal{K}_2^\not\models)\) a refinement of \((\mathcal{K}, \mathcal{K}_1^\models, \mathcal{K}_1^\not\models)\), if \(\mathcal{K}_1^\models \subseteq \mathcal{K}_2^\models\) and \(\mathcal{K}_1^\not\models \subseteq \mathcal{K}_2^\not\models\).

An incomplete revision state \((\mathcal{K}, \mathcal{K}_1^\models, \mathcal{K}_1^\not\models)\) can be refined by evaluating a further axiom \(\alpha \in \mathcal{K} \setminus (\mathcal{K}_1^\models \cup \mathcal{K}_1^\not\models)\), obtaining \((\mathcal{K}, \mathcal{K}_1^\models \cup \{\alpha\}, \mathcal{K}_1^\not\models)\) or \((\mathcal{K}, \mathcal{K}_1^\models, \mathcal{K}_1^\not\models \cup \{\alpha\})\). We call the resulting revision state an elementary refinement of \((\mathcal{K}, \mathcal{K}_1^\models, \mathcal{K}_1^\not\models)\).
The revision closure $\text{clos}(K, K^\models, K^\not\models)$ of $(K, K^\models, K^\not\models)$ is $(K, K^\models_c, K^\not\models_c)$ with

- $K^\models_c := \{ \alpha \in K \mid K^\models \models \alpha \}$ and
- $K^\not\models_c := \{ \alpha \in K \mid K^\models \cup \{ \alpha \} \models \beta \text{ for some } \beta \in K^\not\models \}$. 
Revision States

Goals:
- Obtain complete and consistent revision state
Revision States

Goals:
- Obtain complete and consistent revision state
- Reduce number of manual decisions
Revision States

Goals:
- Obtain complete and consistent revision state
- Reduce number of manual decisions

Revision closure
- insure revision state consistency
Revision States

Goals:
- Obtain complete and consistent revision state
- Reduce number of manual decisions

Revision closure
- insures revision state consistency
- automatizes revision process

Efficient computation and management of dependencies between axioms required $\leadsto$ Decision spaces – auxiliary datastructures avoiding unnecessary reasoning calls.
Axiom Impact

\((\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq)\): a consistent revision state with \(\alpha \in \mathcal{K}\)

\(?((\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq)) = |\mathcal{K} \setminus (\mathcal{K}^= \cup \mathcal{K}^\neq)|\)

- **Approval impact:** Number of automatically evaluated axioms in case \(\alpha\) is approved:
  
  \(\text{impact}^+(\alpha) = ?((\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq)) - ?(\text{clos}(\mathcal{K}, \mathcal{K}^= \cup \{\alpha\}, \mathcal{K}^\neq))\),
Axiom Impact

\((\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq)\): a consistent revision state with \(\alpha \in \mathcal{K}\)

\(?\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq\): \(|\mathcal{K} \setminus (\mathcal{K}^= \cup \mathcal{K}^\neq)|\)

- **Approval impact**: Number of automatically evaluated axioms in case \(\alpha\) is approved:
  \[
  \text{impact}^+(\alpha) = ?(\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq) - ?(\text{clos}(\mathcal{K}, \mathcal{K}^= \cup \{\alpha\}, \mathcal{K}^\neq))
  \]

- **Decline impact**: number of automatically evaluated axioms in case \(\alpha\) is declined:
  \[
  \text{impact}^-(\alpha) = ?(\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq) - ?(\text{clos}(\mathcal{K}, \mathcal{K}^=, \mathcal{K}^\neq \cup \{\alpha\}))
  \]
Axiom Impact

\((\mathcal{K}, \mathcal{K} \models, \mathcal{K} \not\models)\): a consistent revision state with \(\alpha \in \mathcal{K}\)

\(?((\mathcal{K}, \mathcal{K} \models, \mathcal{K} \not\models))\): \(|\mathcal{K} \setminus (\mathcal{K} \models \cup \mathcal{K} \not\models)|

- Approval impact: Number of automatically evaluated axioms in case \(\alpha\) is approved:
  \(\text{impact}^+(\alpha) = ?((\mathcal{K}, \mathcal{K} \models, \mathcal{K} \not\models)) - ?(\text{clos}(\mathcal{K}, \mathcal{K} \models \cup \{\alpha\}, \mathcal{K} \not\models))\),

- Decline impact: number of automatically evaluated axioms in case \(\alpha\) is declined:
  \(\text{impact}^-(\alpha) = ?((\mathcal{K}, \mathcal{K} \models, \mathcal{K} \not\models)) - ?(\text{clos}(\mathcal{K}, \mathcal{K} \models, \mathcal{K} \not\models \cup \{\alpha\}))\),

- Guaranteed impact:
  \(\text{guaranteed}(\alpha) = \min(\text{impact}^+(\alpha), \text{impact}^-(\alpha))\).
Impact Computation

\( (\alpha): \text{Decision} \rightarrow \text{DistinctEntity} \)

\( (\beta): \text{Decision} \rightarrow \text{MentalEntity} \)

\( (\gamma): \text{Decision} \rightarrow \text{MentalObject} \)

\[\begin{array}{ccc}
\text{impact}^+ & \text{impact}^- & \text{guaranteed} \\
0 & 2 & 0 \\
1 & 1 & 1 \\
2 & 0 & 0 \\
\end{array}\]
Parametrizing the Impact Function

- Validity ratio: proportion of approved axioms in a dataset
- Axiom ranking functions tailored towards validity ratios of 100% and 0%

Develop a ranking function parametrized by the validity ratio

Idea: Privilege axioms with the smallest deviation from the expected proportion of accepted or declined axioms
Parametrizing the Impact Function

\[ \text{impact}_N^{+a}(\alpha) = \frac{1 + \text{impact}_{+a}(\alpha)}{|\mathcal{K}^\alpha|}, \]

\[ \text{impact}_N^{+d}(\alpha) = \frac{\text{impact}_{+d}(\alpha)}{|\mathcal{K}^\alpha|}, \]

\[ \text{impact}_N^{-}(\alpha) = \frac{1 + \text{impact}^{-}(\alpha)}{|\mathcal{K}^\alpha|}. \]
Parametrizing the Impact Function

\[ \text{impact}_N^+(\alpha) = \frac{1 + \text{impact}_N^+(\alpha)}{|\mathcal{K}|}, \]

\[ \text{impact}_N^d(\alpha) = \frac{\text{impact}_N^d(\alpha)}{|\mathcal{K}|}, \]

\[ \text{impact}_N^-(\alpha) = \frac{1 + \text{impact}_N^-(\alpha)}{|\mathcal{K}|}. \]

\[ \text{norm}_R^+(\alpha) = -|R - \text{impact}_N^+(\alpha)|, \]
Parametrizing the Impact Function

\[
\begin{align*}
\text{impact}_{N}^{+a}(\alpha) &= \frac{1 + \text{impact}^{+a}(\alpha)}{|\mathcal{K}^?|}, \\
\text{impact}_{N}^{+d}(\alpha) &= \frac{\text{impact}^{+d}(\alpha)}{|\mathcal{K}^?|}, \\
\text{impact}_{N}^{-}(\alpha) &= \frac{1 + \text{impact}^{-}(\alpha)}{|\mathcal{K}^?|}.
\end{align*}
\]

\[
\begin{align*}
\text{norm}_{R}^{+a}(\alpha) &= -|R - \text{impact}_{N}^{+a}(\alpha)|, \\
\text{norm}_{R}^{+d}(\alpha) &= -|1 - R - \text{impact}_{N}^{+d}(\alpha)|,
\end{align*}
\]
Parametrizing the Impact Function

\[
\begin{align*}
\text{impact}^+_{N}(\alpha) &= \frac{1+\text{impact}^+_{N}(\alpha)}{|\mathcal{K}|}, \\
\text{impact}^+_{D}(\alpha) &= \frac{\text{impact}^+_{D}(\alpha)}{|\mathcal{K}|}, \\
\text{impact}^-_{N}(\alpha) &= \frac{1+\text{impact}^-_{N}(\alpha)}{|\mathcal{K}|}.
\end{align*}
\]

\[
\begin{align*}
\text{norm}^+_{R}(\alpha) &= -|R - \text{impact}^+_{N}(\alpha)|, \\
\text{norm}^+_{D}(\alpha) &= -|1 - R - \text{impact}^+_{D}(\alpha)|, \\
\text{norm}^-_{R}(\alpha) &= -|1 - R - \text{impact}^-_{N}(\alpha)|.
\end{align*}
\]
Evaluation Results: Parametrized Ranking

![Graph showing evaluation results for different parameters.

- Optimal
- Norm
- Impact +
- Guaranteed
- Impact -
- Random

The graph illustrates the performance of different ranking strategies across various parameter values.

15 27.10.2011 Nadeschda Nikitina - Wheat and Chaff – Practically Feasible Interactive Ontology Revision

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
Learning Validity Ratio

- The validity ratio is rarely known in advance
Learning Validity Ratio

- The validity ratio is rarely known in advance
- Learn the validity ratio for the parametrized ranking function

For larger datasets (e.g., 5,000 axioms and more) the learned validity ratio deviates only 0.3% from the known one.
Learning Validity Ratio

- The validity ratio is rarely known in advance
- Learn the validity ratio for the parametrized ranking function
- Start with an initial value
Learning Validity Ratio

- The validity ratio is rarely known in advance
- Learn the validity ratio for the parametrized ranking function
- Start with an initial value
- Set expected validity ratio to the current validity ratio after each application of revision closure

Even for small datasets (50–100 axioms) validity ratio can be learned effectively. For larger datasets (e.g., 5,000 axioms and more) the learned validity ratio deviates only 0.3% from the known one.
Learning Validity Ratio

- The validity ratio is rarely known in advance
- Learn the validity ratio for the parametrized ranking function
- Start with an initial value
- Set expected validity ratio to the current validity ratio after each application of revision closure
- Even for small datasets (50–100 axioms) validity ratio can be learned effectively
Learning Validity Ratio

- The validity ratio is rarely known in advance
- Learn the validity ratio for the parametrized ranking function
- Start with an initial value
- Set expected validity ratio to the current validity ratio after each application of revision closure
- Even for small datasets (50–100 axioms) validity ratio can be learned effectively
- For larger datasets (e.g., 5,000 axioms and more) the learned validity ratio deviates only 0.3% from the known one
Evaluation Results with Learned Parametrized Ranking

![Graph showing deviation from optimal cost reduction vs. size of dataset]

- **norm**
- **dynnorm\(_{1.0}\)**
- **dynnorm\(_{0.5}\)**
- **dynnorm\(_{0.0}\)**
- **random**

Size of dataset:
- small (~50)
- medium (~500)
- large (~5000)
Summary

Addressed problems:

- Existing impact measures not working well for medium quality datasets
Summary

Addressed problems:
- Existing impact measures not working well for medium quality datasets
- Dataset quality not known in advance

Contributions:
- Parametrized impact function taking into account expected validity ratio
- Learning validity ratio over the course of the revision
- Further optimizations for computing axiom dependencies (partitioning)
Summary

Addressed problems:

- Existing impact measures not working well for medium quality datasets
- Dataset quality not known in advance

Contributions:

- Parametrized impact function taking into account expected validity ratio
- Learning validity ratio over the course of the revision
- Further optimizations for computing axiom dependencies (partitioning)
Summary

Addressed problems:
- Existing impact measures not working well for medium quality datasets
- Dataset quality not known in advance

Contributions:
- Parametrized impact function taking into account expected validity ratio
- Learning validity ratio over the course of the revision
Summary

Addressed problems:
- Existing impact measures not working well for medium quality datasets
- Dataset quality not known in advance

Contributions:
- Parametrized impact function taking into account expected validity ratio
- Learning validity ratio over the course of the revision
- Further optimizations for computing axiom dependencies (partitioning)