OWL class learning over RDF data

David Ratcliffe
ANU CECS
david.ratcliffe@anu.edu.au

ANU COMP8410 (Data Mining)

22 May, 2018

Overview

Primer

- RDF, RDF Schema
- Web Ontology Language (OWL), Description Logics (DLs)
- OWL Profiles, Ontologies, Knowledge Bases

Data Mining and Machine Learning over RDF data

- Learning settings: Unsupervised, Supervised
- Formal definition

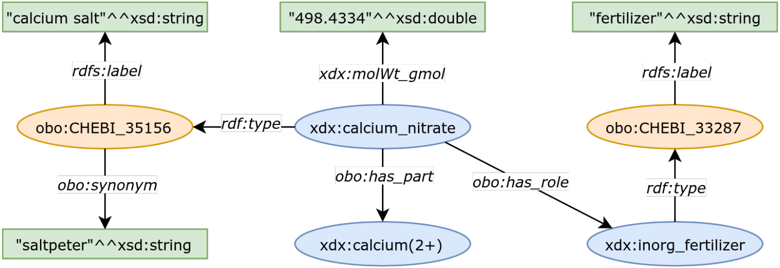
OWL Class Learning

- Class Induction by Top-Down Refinement
- Example use cases (lab exercises!)

Resource Description Framework (RDF)

RDF: Graph-based data model

e.g. Data about **chemical compounds**

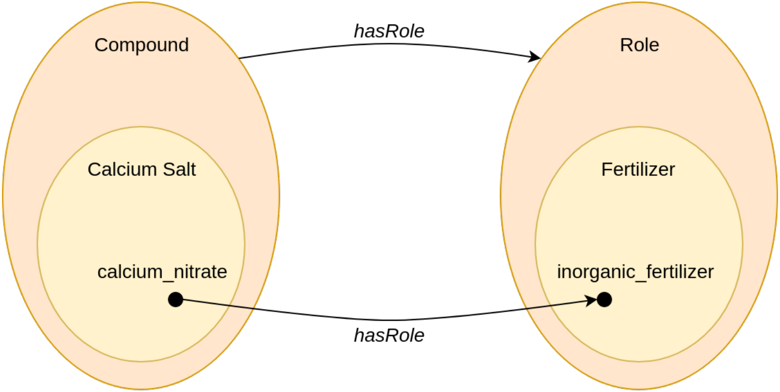


RDF: Resources (blue, orange), literals (green), properties (arcs)
RDFS: Classes (orange)

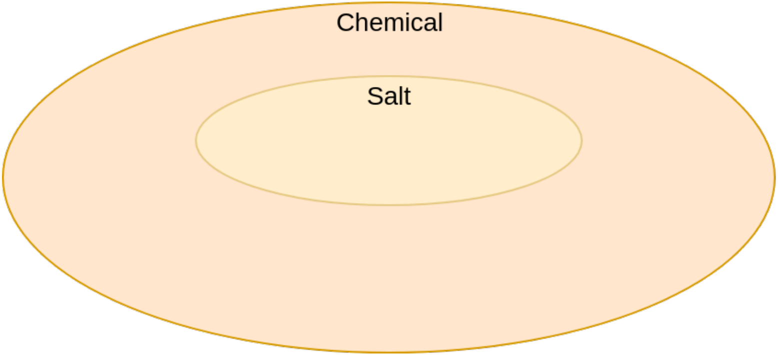
Resource Description Framework Schema (RDFS)

RDFS: Schema language for RDF graphs

e.g. Chemical roles



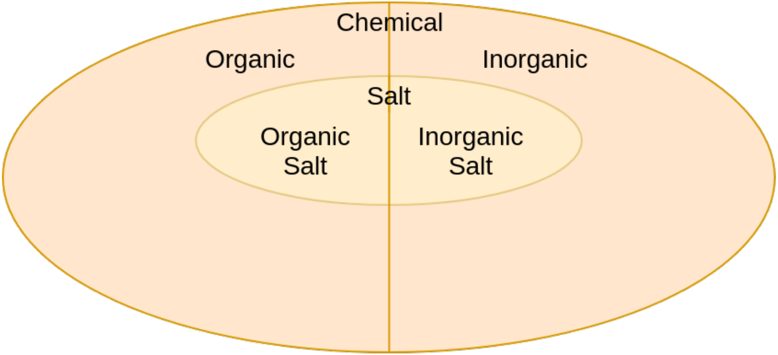
OWL: More expressive than RDFS in capturing **knowledge** with **Description Logics** (fragments of first-order logic).



$$Salt \sqsubseteq Chemical$$

...a subsumption *axiom* (Salt *owl:subClassOf* Chemical)

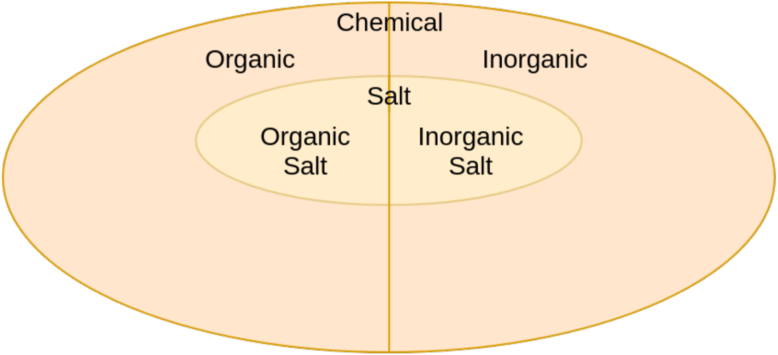
OWL: More expressive than RDFS in capturing **knowledge** with **Description Logics** (fragments of first-order logic).



$$\begin{aligned} \mathit{Chemical} &\sqsubseteq \mathit{Organic} \sqcup \mathit{Inorganic} \\ \mathit{Organic} \sqcap \mathit{Inorganic} &\sqsubseteq \bot \; (\emptyset, \mathit{Nothing}) \end{aligned}$$

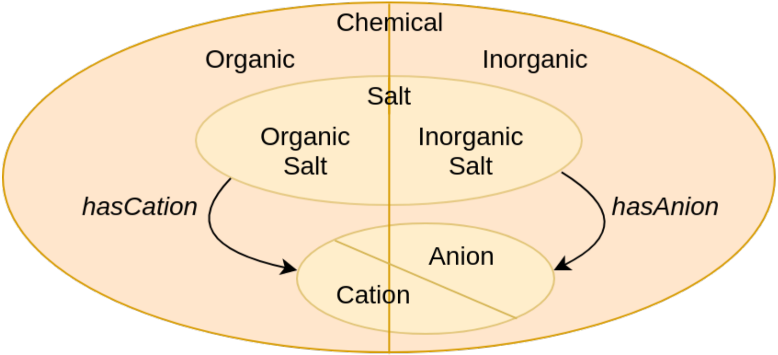
Web Ontology Language (OWL), Description Logic (DL)

OWL: More expressive than RDFS in capturing **knowledge** with **Description Logics** (fragments of first-order logic).



$$\begin{aligned} \mathit{OrganicSalt} &\sqsubseteq \mathit{Organic} \sqcap \mathit{Salt} \\ \mathit{InorganicSalt} &\sqsubseteq \mathit{Inorganic} \sqcap \mathit{Salt} \end{aligned}$$

OWL: More expressive than RDFS in capturing **knowledge** with **Description Logics** (fragments of first-order logic).



$$Salt \quad \sqsubseteq \exists hasCation.(Cation) \sqcap \exists hasAnion.(Anion)$$

($\exists / {}^{\geqslant 1}$: *Exists/some/at least one*)

OWL Ontology, RDF/OWL Knowledge Base

An OWL **ontology** is a collection of *subsumption axioms*:

$$\mathcal{T}_{box} = \{Salt \sqsubseteq Chemical, \ldots\}$$

An ontology may be associated with *data assertions*:

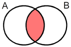
$$\mathcal{A}_{box} = \{Salt(calcium\_nitrate), \ldots\}$$

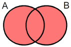
A **knowledge base** is an *ontology* with *data assertions*:

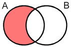
$$\mathcal{K} = \langle \mathcal{T}_{box}, \mathcal{A}_{box} \rangle$$

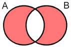
Data assertions are captured as RDF graphs (sets of *triples*).

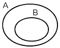
More OWL/DL language elements:

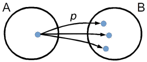
- Intersection $A \sqcap B$: 

- Union $A \sqcup B$: 

- Subtraction $A \sqcap \neg B$: 

- Difference $(A \sqcup B) \sqcap \neg(A \sqcap B)$: 

- Subsumption $B \sqsubseteq A$: 

- Min/max cardinality $A \sqsubseteq {}^{\geqslant n}p.(B)$,
  Universal quantification $A \sqsubseteq \forall p.(B)$: 

- Literals:
  - $\mathit{Teenager} \sqsubseteq \mathit{Person} \sqcap \exists \mathit{hasAge}.(\texttt{xsd:int}[13, 19])$
  - $\mathit{Chemical} \sqcap \exists \mathit{amesTest}(\texttt{xsd:boolean}[= \mathit{true}]) \sqsubseteq \mathit{Mutagenic}$

OWL 2 Profiles

OWL2-DL: Full expressivity, high computational complexity

- $\exists, \forall, {}^{\geqslant n}, {}^{\leqslant n}, \neg, \sqcap, \sqcup, \sqsubseteq, \equiv, \circ, \ldots$
- N2EXPTIME-complete

OWL2-EL: Low expressivity and computational complexity

- $\exists, \sqcap, \ldots$
- Good for large simple ontologies, PTIME

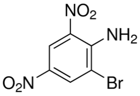
OWL2-QL: Low expressivity and computational complexity

- $\exists r.\top, \sqcap, \neg, \ldots$
- Querying relational data (UML/ER), NLogSpace-complete

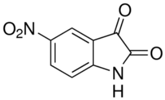
OWL2-RL: Moderate expressivity and complexity

- $\exists, {}^{\geqslant 0/1}, \sqcap, \sqcup, \ldots$
- Similar to rule-based modelling, co-NP-complete

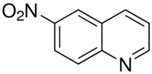
Example: Mutagenic Chemicals



2-bromo-4,6-dinitroaniline



5-nitroisatin

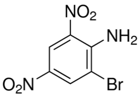


6-nitroquinoline

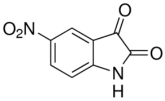
Given a chemical compound *knowledge base* consisting of:

- An OWL *onotology* with axioms describing compounds, atoms, bonds and their charges, compound properties (mutagenic or not), etc. ($\mathcal{T}_{box}$)
- RDF triples describing examples of compounds using classes and properties from the ontology ($\mathcal{A}_{box}$)

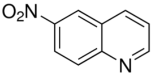
Example: Mutagenic Chemicals



2-bromo-4,6-dinitroaniline



5-nitroisatin



6-nitroquinoline

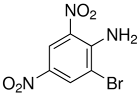
Given a chemical compound *knowledge base* consisting of:

- An OWL *onotology* with axioms describing compounds, atoms, bonds and their charges, compound properties (mutagenic or not), etc. ($\mathcal{T}_{box}$)
- RDF triples describing examples of compounds using classes and properties from the ontology ($\mathcal{A}_{box}$)

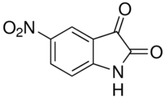
Find:

- A complex OWL class expression which describes a subset of compounds (mutagenic, structurally similar, etc.)

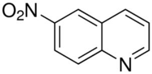
Example: Mutagenic Chemicals



2-bromo-4,6-dinitroaniline



5-nitroisatin



6-nitroquinoline

In a dataset of 125 mutagenic, 105 non-mutagenic compounds:

$$
\begin{aligned}
& \mathit{Compound} \sqcap \\
& {\geqslant}5\, \mathit{hasBond}.(\\
& \quad (\neg \mathit{Bond}_3 \sqcap \forall \mathit{inBond}.(\mathit{Carbon} \sqcap \neg \mathit{Carbon}_{10} \sqcap \\
& \qquad\qquad\qquad \exists \mathit{charge}.(\mathtt{double}[-0.204, 1.002]))) \sqcup \\
& \quad (\mathit{Bond}_7 \sqcap \forall \mathit{inBond}.(\mathit{Carbon}_{22})) \\
&)
\end{aligned}
$$

...a 85% accurate description of mutagenic compounds.

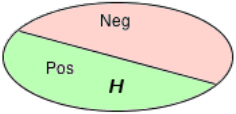
Supervised Learning

- **Predictive model:** Given a new RDF graph describing an unseen compound, label it as mutagenic or not
- **Descriptive model:** Identify an unusually distributed subset of RDF graphs describing known compounds based on certain features (e.g., structurally similar non-mutagenic chemicals)

Unsupervised Learning

- Find **clusters** of similar RDF graphs based on structure, content (e.g., any structurally similar chemicals)

Classify a new/unseen example as Pos or Neg with class/model H.



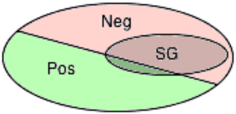
Pos $\equiv$ **H** vs. Neg

Quality measures over H: Accuracy, F1, etc.

H can be used to *predict* labels for new/unseen data.

Cluster a group of examples by:

- **Label Distribution**: a group of labelled examples with an unusual distribution relative to the full set.



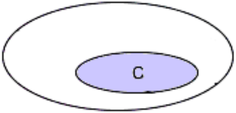
Pos/Neg (1:1) vs. **SG** (1:9)

Correlation measures: χ^2, Weighted Relative Accuracy, etc.

SG *describes* features mostly correlating with *Neg* examples.

Cluster a group of examples by:

- **Cluster size**: covers a large number of examples relative to all.



C (1:4) of all examples.

Cluster quality measures: support, etc.

Cluster **C** groups unlabelled examples by similar features.

OWL Class Learning

Given:

- An RDFS/OWL ontology describing the data ($\mathcal{T}_{box}$);
- An RDF dataset ($\mathcal{A}_{box}$);
- A *hypothesis language* $\mathcal{L}$ as choice of class constructs, e.g.:
  - Conjunction: $C \sqcap D$
  - Disjunction: $C \sqcup D$
  - Negation: $\neg C$
  - Min-qualified cardinality: ${}^{\geqslant n}p.C$ (for $1 \leq n \leq 5$)
- A quality function to assess solutions (e.g., accuracy)

Solve a learning problem by:

Finding **new** complex classes composed with language $\mathcal{L}$ over the classes and properties in $\mathcal{T}_{box}$ which meet minimum quality requirements (e.g., accuracy $\geq 95\%$).

Class Induction

Ontology contains classes $C_0, \ldots, C_m$ and properties $p_0, \ldots, p_n$, but how do we generate new classes with our language $\mathcal{L}$?

Structure the space of classes by subsumption ($\sqsubseteq$).

$\mathcal{L}$ + OWL ontology $\mathcal{T}_{box}$ defines this:

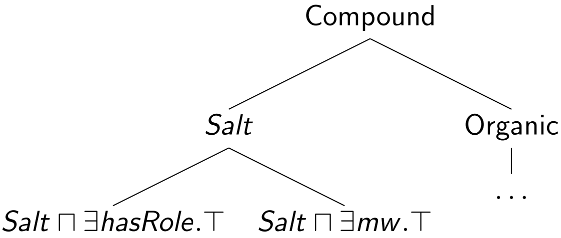
$$
\begin{array}{ll}
\mathit{CalciumNitrate} & \sqsubseteq \exists \mathit{hasRole}.(\mathit{Fertilizer}) \\
\mathit{CalciumNitrate} & \sqsubseteq \mathit{CalciumSalt} \\
\mathit{CalciumSalt} & \sqsubseteq \mathit{Salt}
\end{array}
$$

Downward Refinement Operator: $\rho(C_0) \rightarrow \{D_0, \ldots, D_k\}$

...where each $D_i \sqsubseteq C_0$ (for $0 \leq i \leq k$).

Top-Down Class Induction By Refinement Operator

- Start with a general class, and progressively *specialise*;
- Assess quality of each, but only continue from the best ones;
- Stop when a class is found with sufficient quality.



- ρ helps us search the space of concepts automatically;
- High expressivity of $\mathcal{L}$ + large ontology: **vast** search space!

Example Trace: Downward Refinement

Expansion of a single refinement 'path':

$$
\begin{aligned}
\top &\rightsquigarrow \mathit{Compound} \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\top) \sqcap \forall \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion}) \sqcap \forall \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion}) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal}) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion} \sqcap \exists \mathit{hasGroup}.(\top)) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal}) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion} \sqcap \exists \mathit{hasGroup}.(\mathit{Carboxyl})) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal})
\end{aligned}
$$

Example Trace: Downward Refinement

Expansion of a single refinement 'path':

$$
\begin{aligned}
\top &\rightsquigarrow \mathit{Compound} \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\top) \sqcap \forall \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion}) \sqcap \forall \mathit{hasPart}.(\top) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion}) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal}) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion} \sqcap \exists \mathit{hasGroup}.(\top)) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal}) \\
&\rightsquigarrow \mathit{Compound} \sqcap \exists \mathit{hasPart}.(\mathit{Ion} \sqcap \exists \mathit{hasGroup}.(\mathit{Carboxyl})) \sqcap \forall \mathit{hasPart}.(\neg \mathit{Metal})
\end{aligned}
$$

Potentially **too many** possible paths to compute!
How do we choose the **best** ones?

Guiding the Search

Depending on the problem being solved, define a **utility function** over:

- Quality measure: Accuracy, χ^2, etc.
- Structural quality: *short and simple* is better than *long and complex* (minimum description length principle).

Guiding the Search

Depending on the problem being solved, define a **utility function** over:

- Quality measure: Accuracy, χ^2, etc.
- Structural quality: *short and simple* is better than *long and complex* (minimum description length principle).

Example: $\mathbf{u}(C) = \text{acc}(C) - \text{length}(C)$

$\mathbf{u}$ induces an *order* over all classes: $\mathbf{u}(C_2) < \mathbf{u}(C_0) < \ldots < u(C_k)$

Select the class with 'best' utility $\mathbf{u}$ to refine next.

Basic Search Strategy

Search Loop:

- Pick the current best class expression D according to $\mathbf{u}$.
- Generate refinements with $\rho(D) \rightarrow \{C_0, \ldots, C_k\}$.
- For each *candidate* expression $C \in \{C_0, \ldots, C_k\}$, check:
  - Is C a solution?

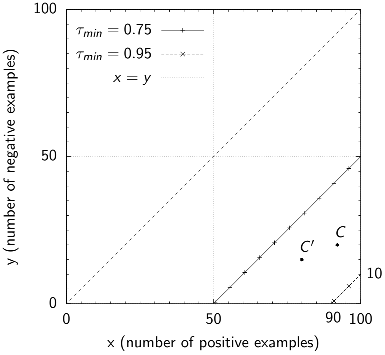
    or
  - Can *any* refinements of C *possibly* be a solution?

    or
  - Can *all* refinements of C *never* be solutions?

e.g., a 'solution' may be a candidate class C where $\mathrm{acc}(C) > 0.95$.

Assessing Candidates

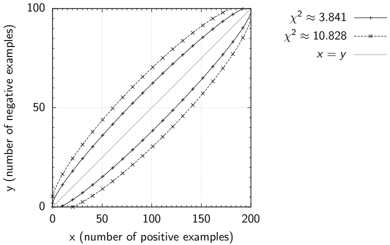
Coverage Space [Zimmerman and De Raedt, 2009]

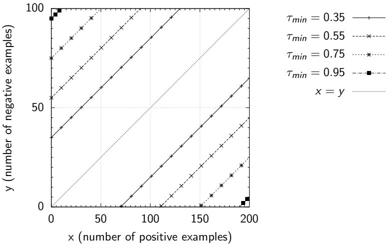


C covers x positive, y negative examples: **stamp point** (x, y).

$acc(C) > 0.75$, $acc(C) > 0.95$ induce **isometric lines** in the space.

Method relies on quality measure **convexity** in coverage space.





Isometric lines for χ^2, WRAcc are convex in coverage space.

Weighted Relative Accuracy (WRACC): $\frac{p}{P} - \frac{n}{N}$

Summary

RDF data and OWL ontologies:

- RDF captures structure, categorical and numerical features.
- OWL captures highly expressive domain knowledge over RDF.

Data mining and machine learning over RDF and OWL:

- Class learning
  - Produces compact, readable descriptions of data
  - Naturally combines categorical, numerical data and ontological knowledge
- Several other methods
  - Graph kernels, neural graph embeddings
  - SVM, neural nets

Example Domains for RDF/OWL Learning Methods

Life Sciences

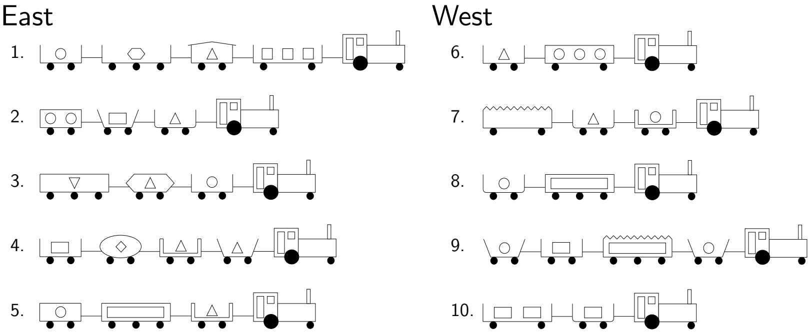
- Chemistry, genetics, phenomics, drug design, etc.
- Learn new scientific knowledge by finding patterns in data
- Very many rich OWL ontologies in this space already

Recommender Systems

- Shopping, music, videos, adertisements, etc.
- Learn classes to predict someone's preferences

Many more areas to explore!

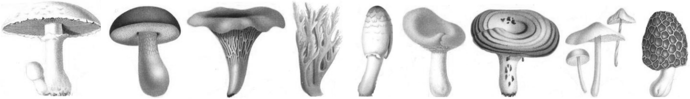
Michalski Trains



Mushrooms

Binary classification: **edible**, or **poisonous** mushroom?

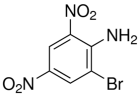
- OWL ontology: 84 classes, 21 properties, 40,679 individuals



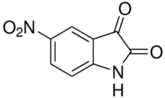
Mutagenic or Carcinogenic Chemicals

Binary classification: **mutagenic/carcinogenic**, or not?

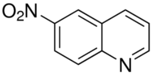
- OWL: 100+ classes, 10+ properties, 20,000+ individuals



2-bromo-4,6-dinitroaniline



5-nitroisatin



6-nitroquinoline

Poker

A♠ K♠ Q♠ J♠ 10♠	(royal flush)
3♦ 4♦ 5♦ 6♦ 7♦	(straight flush)
7♦ 7♠ 7♣ 7♥ 3♦	(four of a kind)
Q♠ Q♣ Q♥ 9♠ 9♥	(full house)
J♣ 10♣ 8♣ 3♣ 2♣	(flush)
6♥ 5♦ 4♥ 3♥ 2♠	(straight)
5♦ 5♣ 5♠ K♦ 7♠	(three of a kind)
4♥ 4♦ K♠ K♥ 3♠	(two pair)
9♥ 9♠ 10♦ 4♦ 2♠	(one pair)
K♦ Q♠ 6♣ 7♠ 3♦	(nothing)

- Learning problem: one-versus-all (OvA) models for each class
- Ontology contains 27,546 assertions, 40 classes, 6 properties

Questions?

Try out our OWL class learning tool
OWL-Miner in the lab!