Semantic data mining for knowledge acquisition

Tutorial (K-CAP 2017)
Part I

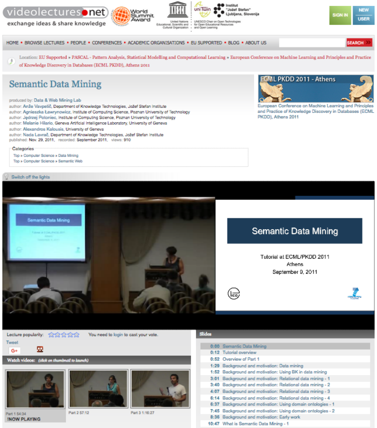
Agnieszka Lawrynowicz
Poznan University of Technology

Planned schedule

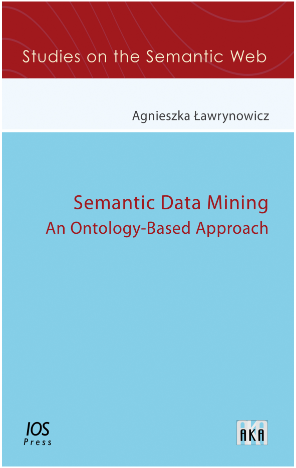
- 14:00 – 15:30 Introduction, Semantic Data Mining Tasks and Pecularities
  - Introduction (20 min.)
  - Basics of semantic data mining (20 min.)
  - Tasks of semantic data mining (20 min.)
  - Pecularities of semantic data mining (30 min.)
- 15:30 – 16:00 Coffee break
- 16:00 – 18:00 Semantic Data Mining for Knowledge Acquisition, Hands-On
  - Semantic data mining for knowledge acquisition (60 min.)
  - Hands-on: LeoLOD Swift Linked Data Miner plugin for Protégé (60 min.)

This tutorial starting points

2011



2017

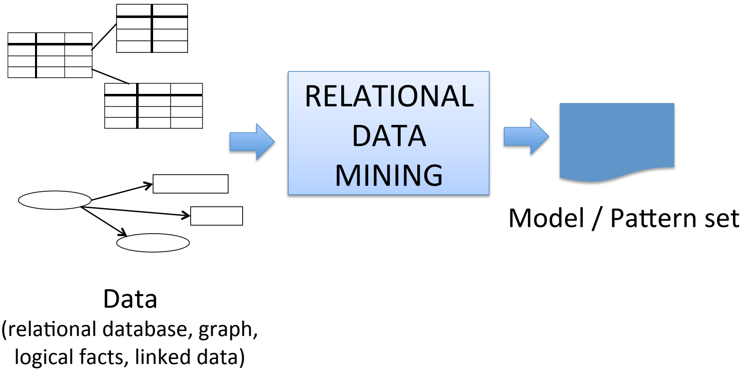


Data mining

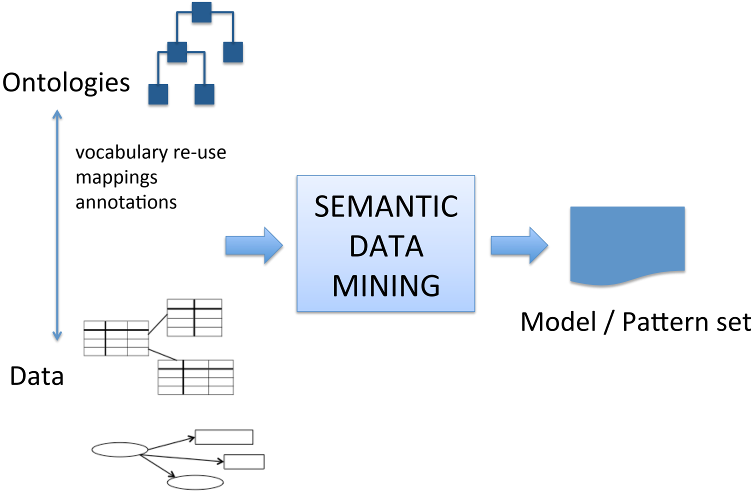
Data
(data table, text,…)

Model / Pattern set

Relational data mining



Semantic data mining



Data mining as search

- Goal: discover new and interesting patterns in large data sets
- Major components of any DM method:
  - a knowledge representation structure to store discovered knowledge in the form of a model or a pattern set, called (an inductive) hypothesis $h \in \mathcal{L}_h$, where $\mathcal{L}_h$ is the language of hypotheses
  - a method to search for hypotheses
- The discovered hypothesis needs to satisfy quality criteria posed by a performance measure that optimizes achieving a given DM task.
- The method operates on particular kinds of input data, that is training examples $e \in \mathcal{L}_e$

Model versus pattern set

- Model: *global* description, i.e. it summarizes a whole dataset and applies to all points in the space
- Pattern: *local* description, i.e. it only characterizes some subset of the space, possibly some recurring structure

Generality relations

The structure of the space of possible hypotheses is imposed by a *generality relation* $\succeq$ between hypotheses

- θ-subsumption (syntactic generality relation) (Plotkin 1970)
- generalized subsumption (semantic generality relation) (Buntine 1988)
- taxonomical subsumption (bridges the gap between purely syntactic generality relation and relatively heavy semantic generality relation, intended for knowledge bases expressed in RDFS) (Ławrynowicz & Potoniec, IJSWIS 2014)

Generality relations

Given is background theory $\mathcal{K}$:

$hazard(x) \leftarrow dust(x)$

$hazard(x) \leftarrow nanoparticles(x)$

$substance(x) \leftarrow dust(x)$.

Consider the following clauses:

$c_1 = physical\_hazard(x) \leftarrow substance(x), physical\_agent(x), nanoparticles(x)$

$c_2 = physical\_hazard(x) \leftarrow physical\_agent(x), dust(x)$

and

$c_3 = physical\_hazard(y) \leftarrow substance(y), physical\_agent(y)$.

$c_4 = physical\_hazard(x) \leftarrow substance(x), physical\_agent(x), hazard(x)$.

Clause c_4 is θ-subsumed by clause c_3 under the substitution $\theta = \{x/y\}$.
Clause c_3 is semantically more general than each of the clauses c_1 and c_2.
But it is not more general (under θ-subsumption) than either of c_1 or c_2.

Learning in DLs

Definition

Learning in description logics: a machine learning approach that adopts Inductive Logic Programming as the methodology and description logic as the language of data and hypotheses.

Description logics theoretically underpin the state-of-art Web ontology representation language, **OWL**, so description logic learning approaches are well suited for semantic data mining.

Description logic

Definition

Description Logics, $\mathcal{DL}s$ = family of first order logic-based formalisms suitable for representing knowledge, especially terminologies, ontologies.

Description logic

Definition

Description Logics, $\mathcal{DL}s$ = family of first order logic-based formalisms suitable for representing knowledge, especially terminologies, ontologies.

- subset of first order logic (decidability, efficiency, expressivity)
- root: semantic networks, frames

Basic building blocks $\mathcal{DL}$

- concepts
- roles
- constructors
- individuals

Examples

Atomic concepts: **Artist**, **Movie**

Role: **creates**

Constructors: $\sqcap$, $\exists$

Concept definition: **Director** $\equiv$ **Artist** $\sqcap$ $\exists$**creates.Movie**

Axiom (”each director is an artist”): **Director** $\sqsubseteq$ **Artist**

Asertion: **creates(sofiaCoppola, lostInTranslation)**

DL-learning as search

- learning in DLs can be seen as search in space of concepts
- it is possible to impose ordering on this search space using subsumption as **generality relation** between concepts
  - if $D \sqsubseteq C$ then C covers all instances that are covered by D
- refinement operators may be applied to traverse the space by computing a set of specializations (resp. generalizations) of a concept

Refinement operator

Given a DL $\mathcal{L}$ and the quasi-ordered space $\langle \mathbb{S}(\mathcal{L}), \sqsubseteq \rangle$ over concepts of $\mathcal{L}$ a downward $\mathcal{L}$ refinement operator ρ is a mapping from $\mathbb{S}(\mathcal{L})$ to $2^{\mathbb{S}(\mathcal{L})}$ such that

for any $C \in \mathbb{S}(\mathcal{L})$: $D \in \rho(C)$ implies $D \sqsubseteq C$

Refinement chain

$D \in \rho(C)$ often written as $C \rightsquigarrow_\rho D$

example refinement chain: $\top \rightsquigarrow_\rho$ Director $\rightsquigarrow_\rho$ Director $\sqcap \exists$ creates.$\top$

Properties of refinement operators

Consider **downward refinement operator** ρ, and by $C \rightsquigarrow_\rho D$ denote a refinement chain from a concept C to D

- complete: each point in lattice is reachable (for $D \sqsubseteq C$ there exists E such that $E \equiv D$ and a refinement chain $C \rightsquigarrow_\rho \ldots \rightsquigarrow_\rho E$
- weakly complete: for any concept C with $C \sqsubseteq \top$, concept E with $E \equiv C$ can be reached from $\top$
- finite: finite for any concept
- redundant: there exist two different refinement chains from C to D
- proper: $C \rightsquigarrow_\rho D$ implies $C \not\equiv D$

ideal = complete + proper + finite

Combining properties

Can an operator have all of these properties?
Which properties can be combined?

Refinement operators - property theorem

Lehmann & Hitzler (ILP 2007, MLJ 2010) proved that for many DLs, even simpler then those underpinning OWL, **no ideal refinement operator** exists:
learning in DLs is hard

Maximal sets of properties of $\mathcal{L}$ refinement operators which can be combined for $\mathcal{L} \in \{\mathcal{ALC}, \mathcal{ALCN}, \mathcal{SHOIN}, \mathcal{SROIQ}\}$:

1. {weakly complete, complete, finite}
2. {weakly complete, complete, proper}
3. {weakly complete, non-redundant, finite}
4. {weakly complete, non-redundant, proper}
5. {non-redundant, finite, proper}

Pattern mining

Pattern = recurring structure

Data

Pattern

itemsets, sequences, graphs, clauses,...

Frequent Itemsets

- $I = \{i_1, i_2, \ldots, i_m\}$ - a set of items
- $D_T = \{t_1, t_2, \ldots, t_n\}$, where $\forall i \; t_i \subseteq I$ -a database of transactions

- $support(X) = \frac{|\{t \in D_T : X \subseteq t\}|}{|D_T|}$

ID	Items
1	Nachos, Pepsi, Salsa
2	Nachos, Coca-Cola, Salsa
3	Nachos, Coca-Cola
4	Nachos, Pepsi, Salsa
5	Milk, Bread

Frequent Itemset	Support
{Nachos}	80%
{Salsa}	60%
{Coca-Cola}	40%
{Pepsi}	40%
{Nachos, Salsa}	60%
{Nachos, Coca-Cola}	40%
{Nachos, Pepsi}	40%
{Salsa, Pepsi}	40%

Association rule mining

- $support(X) = \frac{|\{t \in D_T : X \subseteq t\}|}{|D_T|}$
- $support(X \to Y) = support(X \cup Y)$ (relative frequency of the rule)
- $confidence(X \to Y) = \frac{support(X \cup Y)}{support(X)}$ (implication strength)

Frequent Itemset	Support
{Nachos}	80%
{Salsa}	60%
{Coca-Cola}	40%
{Pepsi}	40%
{Nachos, Salsa}	60%
{Nachos, Coca-Cola}	40%
{Nachos, Pepsi}	40%
{Salsa, Pepsi}	40%

$$confidence(\{Nachos\} \to \{Salsa\}) = \frac{60\%}{80\%} = 0.75$$

Patterns in DLs

How to represent patterns in learning from DLs?

Frequent DL concept mining

Lawrynowicz & Potoniec (ISMIS 2011)

- Fr-ONT: mining frequent patterns in the form of DL concepts C
- each C is subsumed by a reference concept $\hat{C}$ ($C \sqsubseteq \hat{C}$)
- support: ratio between the number of instances of C and $\hat{C}$ in $\mathcal{K}$

Example:

$\mathcal{T} = \{$ Director, Movie, creates $\}$
$\mathcal{A} = \{$ Director(Coppola), Director(Kieslowski) , Director(Cameron), creates(Coppola, lostInTranslation), creates(Kieslowski, Three Colors_Red), Movie(lostInTranslation), Movie(Three Colors_Red) $\}$

$\hat{C}$ = Director
C = Director $\sqcap \exists$creates.Movie

$$support(C, \hat{C}, KB) = \frac{2}{3}$$

Common approach to semantic pattern mining

A level-wise generation and testing/evaluating of candidates (idea based on the work of (Dehaspe et al. 1999) on frequent Datalog patterns:

- starting from a general pattern, e.g. concept name
- specializing patterns at each level with refinement operators to produce candidates
- evaluating and pruning the generated specializations (candidates)
- stopping when a chosen stopping criterion met

Semantic pattern mining for knowledge acquisition

- Kralj Novak et al., Lavrac et al. (IS 2009, DS 2011)
  - subgroup discovery, coined the term semantic data mining
- Lisi (IJSWIS 7(3) 2011)
  - onto-relational frequent pattern mining of the form of constrained Datalog clauses with description logic concepts as constraints in the clause body
- Ławrynowicz & Potoniec (ISMIS 2011)
  - Fr-ONT algorithm for mining frequent description logic complex concepts
- Voelker & Niepert (ESWC 2011)
  - association rules discovery from RDF data for ontology induction from scratch (no reasoning)
- Galarraga et al. (WWW 2013)
  - association rules discovery for predicting new role assertions from an RDF knowledge base (no reasoning)
- Ławrynowicz & Potoniec (IJSWIS 10(1), 2014)
  - pattern based feature construction (encoded in SPARQL) from RDFS data
- d'Amato et al. (SAC 2016, EKAW 2016)
  - discovering multi-relational association rules (encoded in SWRL) from ontological knowledge bases

Some of these works I will cover later in this tutorial

Supervised concept learning

Given

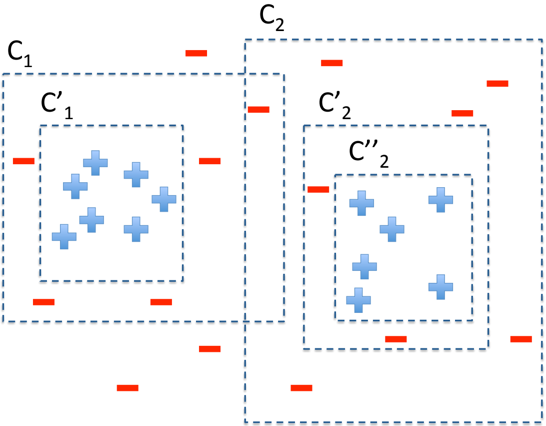
- new target concept name C
- knowledge base $\mathcal{K}$ as background knowledge
- a set E^+ of positive examples, and a set E^- of negative examples

the goal is to learn a concept definition $C \equiv D$ such that
$\mathcal{K} \cup \{C \equiv D\} \models E^+$ and $\mathcal{K} \cup \{C \equiv D\} \models E^-$

Concept learning - algorithms

- YINYANG (Iannone et al, Applied Intelligence 2007) (counterfactuals), sequential covering
- DL-Learner (Lehmann & Hitzler, ILP 2007) (genetic programming), sequential covering
- DL-FOIL (Fanizzi et al, ILP 2008) sequential covering
- TermiTIS (Fanizzi et al, ECML/PKDD 2010) (terminological decision trees), divide-and-cover
- CELOE (Lehmann et al., JWS 2011), (class expression learning for ontology engineering), sequential covering

Sequential covering



C_1 = Director $\quad C_1'$ = Director $\sqcap \exists$creates.$\top$

C_2 = Actor $\quad C_2'$ = Actor $\sqcap \exists$playsIn.$\top$ $\quad C_2''$ = Actor $\sqcap \exists$playsIn.Movie

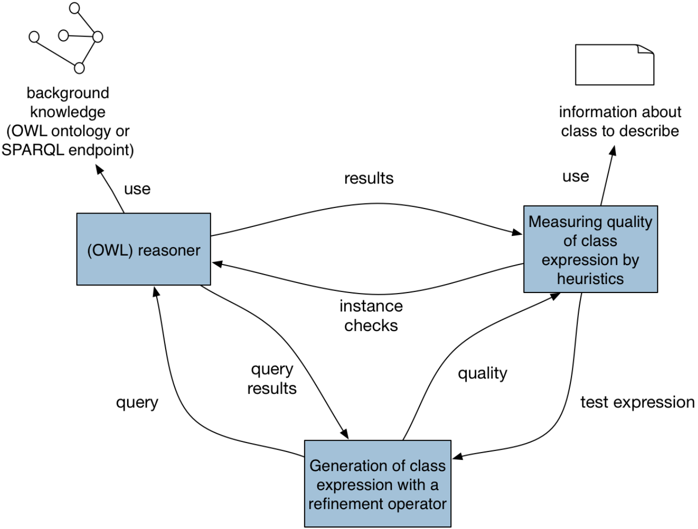
DL-FOIL

Fanizzi et al, ILP 2008

- sequential covering
- two refinement operators: one for specialization and one for generalization
- exploits positive and negative examples

CELOE

Lehmann et al., J. Web Semantics, 2011



Negative examples and Open World Assumption

But what are negative examples in the context of the Open World Assumption?

Semantics: ”closed world” vs ”open world”

- Closed world (Logic programming LP, databases)
  - *complete* knowledge of instances
  - lack of information is by default negative information (*negation-as-failure*)

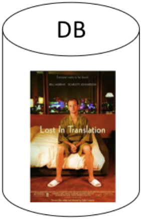
Semantics: ”closed world” vs ”open world”

- Closed world (Logic programming LP, databases)
  - *complete* knowledge of instances
  - lack of information is by default negative information (*negation-as-failure*)
- Open world (description logic DL, Semantic Web)
  - *incomplete* knowledge of instances
  - negation of some fact has to be explicitly asserted (*monotonic negation*)

"Closed world" vs "open world" example

Let data base contain the following data:

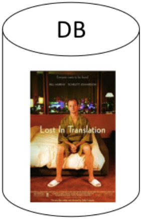
OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)

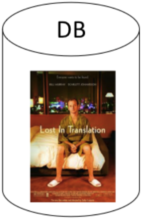


Are all of the movies of Sofia Coppola Oscar movies?

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



Are all of the movies of Sofia Coppola Oscar movies?

YES - closed world

"Closed world" vs "open world" example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)

Are all of the movies of Sofia Coppola Oscar movies?

YES - closed world   DON'T KNOW - open world

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)

Are all of the movies of Sofia Coppola Oscar movies?

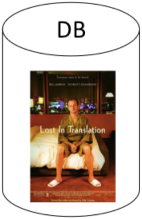
YES - closed world   DON’T KNOW - open world

Different conclusions!

Another question

Let data base contain the following data:

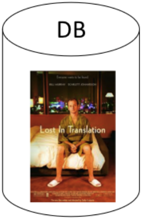
OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



Another question

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)

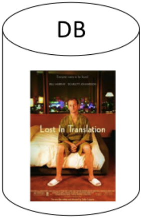


Is "Avatar" a movie created by Sofia Coppola?

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



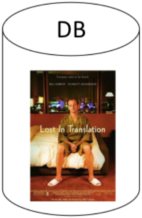
Is “Avatar” a movie created by Sofia Coppola?

NO - closed world   DON’T KNOW - open world

”Closed world” vs ”open world” example

Let data base contain the following data:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)



Is “Avatar” a movie created by Sofia Coppola?

NO - closed world   DON’T KNOW - open world

Different conclusions!

”Closed world” vs ”open world” example

We need to explicitely assert negative knowledge:

OscarMovie(lostInTranslation)
Director(sofiaCoppola)
creates(sofiaCoppola, lostInTranslation)
¬creates(sofiaCoppola, avatar)

OWA and machine learning

OWA is problematic for machine learning since an individual is rarely deduced to belong to a complement of a concept unless explicitely asserted so.

Dealing with OWA in learning

Solution1: alternative problem setting

Solution2: **K** operator

Solution3: new performance measures

Dealing with OWA in learning: alternative problem setting

”Closing” the knowledge base to allow performing instance checks under the Closed World Assumption (CWA).

By default:
Positive examples of the form $C(a)$, and negative examples of the form $\neg C(a)$, where a is an individual and holding:
$\mathcal{K} \cup \{C \equiv D\} \models E^{+}$ and $\mathcal{K} \cup \{C \equiv D\} \models E^{-}$

Alternatively:
Examples of the form $C(a)$ and holding: $\mathcal{K} \cup \{C \equiv D\} \models E^{+}$ and $\mathcal{K} \cup \{C \equiv D\} \not\models E^{-}$

Dealing with OWA in learning: **K** operator

- epistemic **K**–operator allows for querying for known properties of known individuals w.r.t. the given knowlege base $\mathcal{K}$
- the **K** operator alters constructs like $\forall$ in a way that they operate on a Closed World Assumption.

Consider two queries:
Q1: $\mathcal{K} \models \{(\forall \text{creates.OscarMovie})\ (\text{sofiaCoppola})\}$
Q2: $\mathcal{K} \models \{(\forall \mathbf{K}\text{creates.OscarMovie})\ (\text{sofiaCoppola})\}$

- Badea and Nienhuys-Cheng (ILP 2000) considered the **K** operator from a theoretical point of view.
- not easy to implement in reasoning systems, non-standard

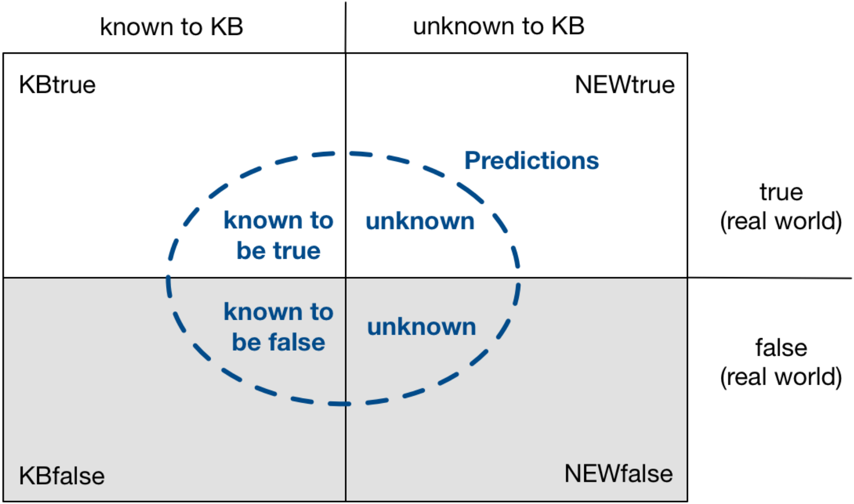
Dealing with OWA in learning: new performance measures

d'Amato et al. (ESWC 2008)
– overcoming unknown answers from the reasoner (as a reference system)
– correspondence between the classification by the reasoner for the instances w.r.t. the test concept C and the definition induced by a learning system

- match rate: number of individuals with exactly the same classification by both the inductive and the deductive classifier w.r.t the overall number of individuals;
- omission error rate: number of individuals not classified by inductive method, relevant to the query w.r.t. the reasoner;
- commission error rate: number of individuals found relevant to C, while they (logically) belong to its negation or vice-versa;
- induction rate: number of individuals found relevant to C or to its negation, while either case not logically derivable from $\mathcal{K}$;

Dealing with OWA in learning: new performance measures

Galárraga et al. (2013, 2015): a rule mining method called *AMIE*



Dealing with OWA in learning: new performance measures

Sample facts belonging to four groups

KBtrue creates(sofiaCoppola, lostInTranslation)	**NEWtrue** creates(sofiaCoppola, marieAntoinette) creates(sofiaCoppola, somewhere)
¬creates(sofiaCoppola, avatar) **KBfalse**	¬creates(sofiaCoppola, theGodfather) **NEWfalse**

Dealing with OWA in learning: new performance measures

Galárraga et al. (2013, 2015)

- to generate negative evidence with *Partial Completeness Assumption (PCA)*, namely assuming that if the KB knows some property p of x, then it knows all properties p of x
- this assumption holds true for functional properties p, e.g. date of birth, as well as for inverse-functional properties
- also a reasonable assumption for cases where properties have a high functionality, and for knowledge bases which were extracted automatically from a single source (contain either all property assertions or none)

Dealing with OWA in learning: new performance measures

Confidence:

$$confidence(B_1 \wedge ... \wedge B_n \implies p(x,y), KB) = \frac{support(B_1 \wedge ... \wedge B_n \implies p(x,y), KB)}{|\{(x,y) \colon \exists z_1, ..., z_m : B_1 \wedge ... \wedge B_n\}|} \quad (1)$$

PCA (Partial Completeness Assumption) confidence:

$$pcaconfidence(B_1 \wedge ... \wedge B_n \implies p(x,y), KB) = \frac{support(B_1 \wedge ... \wedge B_n \implies p(x,y), KB)}{|\{(x,y) \colon \exists z_1, ..., z_m, y' : B_1 \wedge ... \wedge B_n \wedge p(x,y')\}|} \quad (2)$$

Dealing with OWA in learning: new performance measures

Sazonau et al. (2015)

- general terminology induction – learning sets of general class inclusions (GCIs) having on input both data and background knowledge,
- major objective to induce such new knowledge (hypotheses) which respects the existing knowledge along with the data in order to be both informative and non-contradictory,
- a set of axioms (GCIs) constitutes a hypothesis h when they do not contradict the input ontology

Dealing with OWA in learning: new performance measures

Sazonau et al. (2015)

- general terminology induction – learning sets of general class inclusions (GCIs) having on input both data and background knowledge,
- major objective to induce such new knowledge (hypotheses) which respects the existing knowledge along with the data in order to be both informative and non-contradictory,
- a set of axioms (GCIs) constitutes a hypothesis h when they do not contradict the input ontology

Dealing with OWA in learning: new performance measures

Sazonau et al. (2015)

Table: Projection, i.e. a set of positive and negative class assertions over a set of classes entailed by $\mathcal{K}$

	OscarMovie	$\neg$OscarMovie	$\exists$creates.OscarMovie	$\exists$creates.Director
sofiaCoppola	?	?	1	?
lostInTranslation	1	0	?	?
avatar	?	?	?	?

Dealing with OWA in learning: new performance measures

Sazonau et al. (2015)

- Projection used to assess how well hypothesis h fits the *known* data with an *assumption* that it is also correct on the *unknown* data
- Due to OWA, a hypothesis h can only make *assumptions* with regard to unknown data what corresponds to turning question marks into 1s and 0s

Dealing with OWA in learning: new performance measures

Sazonau et al. (2015)

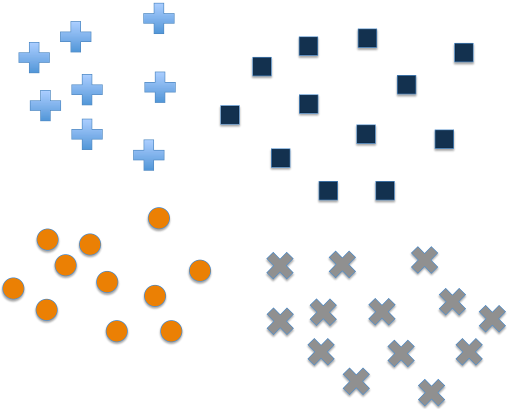
How well hypotheses represent data given background knowledge and OWA?

- A two-fold statistical quality criterion consisting of two measures:
  - Fitness: measuring how a hypothesis fits data along with background knowledge, in other terms, how well the projection can be shrunk using the hypothesis and background knowledge
  - Braveness: measuring how cautious a hypothesis is, i.e., how many assumptions it makes

Clustering

Clustering: unsupervised methods with the goal to organize a collection of unlabeled examples into clusters such that:

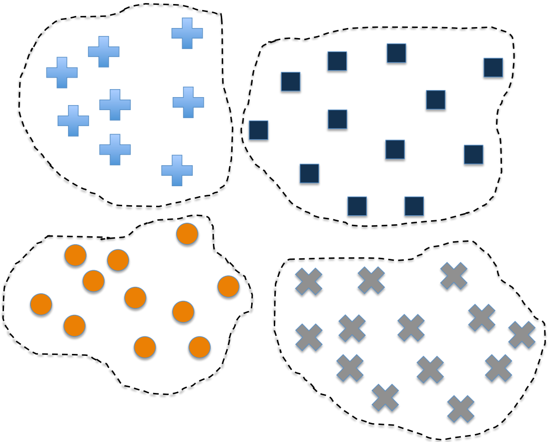
- intra-cluster similarity is high
- inter-cluster similarity is low



Clustering

Clustering: unsupervised methods with the goal to organize a collection of unlabeled examples into clusters such that:

- intra-cluster similarity is high
- inter-cluster similarity is low



Clustering in DLs

Classically:

- objects represented as feature vectors in an n-dimensional space
- features may be of different types, but many algorithms are designed to cluster interval-based (numerical) data
  - such algorithms may employ centroid to represent a cluster

DLs:

- individuals in DL knowledge bases are objects to be clustered
- DL individuals need to be logically manipulated
- similarity measures for DLs need to be defined
- DL specific cluster representative may be necessary

(Dis)-similarity measures for DLs

- Structural, intensional:
  - often decompose concepts structurally, and try to assess an overlap function for each construtor of the considered logic, then aggregate the results of the overlap functions
  - often a new measure has to be defined for each logic, this does not easily scale to more expressive DLs
- Extensional
  - based on the ABox, checking individual membership to concepts

Intensional measures

- simple DL, allowing only disjunction (Borgida et al., 2005)
- $\mathcal{ALC}$ (d'Amato et al., 2005, SAC 2006)
- $\mathcal{ALCNR}$ (Janowicz 2006)
- $\mathcal{EL}^{++}$ (Jozefowski et al., COLISD at ECML/PKDD 2011)
- generic DLs: using features which are the subsumers of the concepts being compared (Alsubait, Parsia, Sattler, EKAW 2014)

Extensional measures: example

(Fanizzi et al. DL 2007)

- basic idea inspired by (Sebag 1997): individuals compared on the grounds of their behavior w.r.t. a set of discriminating features
- on a semantic level, similar individuals should behave similarly w.r.t. the same concepts
- $F = F_1, F_2, ..., F_m$ - a collection of (primitive or defined) concept descriptions
- checking whether an individual belongs to F_i, $\neg F_i$ or none of them
- aggregating the results in a way inspired to Minkowski's norms L_p

Semantic similarity measure

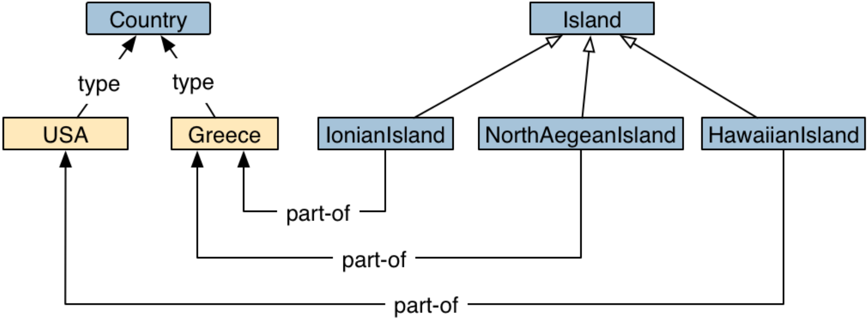
But what is a truly ”semantic” similarity measure?

Semantic similarity measure properties - example

$\mathcal{DL}$ knowledge base:

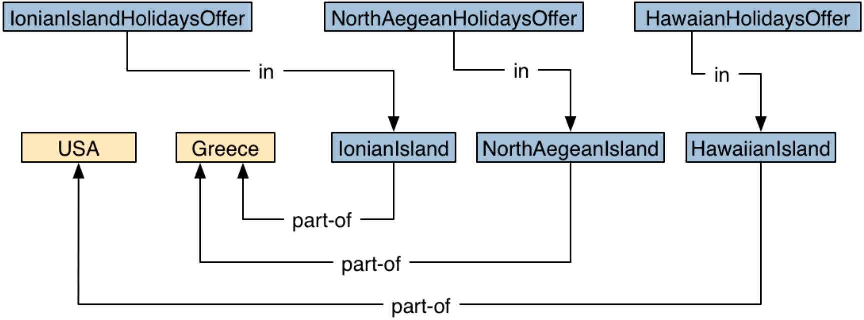
$$\begin{gathered}
\mathcal{T} = \{ \text{IonianIsland} \sqsubseteq \text{Island} \sqcap \text{part-of.}\{\text{Greece}\}, \\
\text{NorthAegeanIsland} \sqsubseteq \text{Island} \sqcap \text{part-of.}\{\text{Greece}\}, \\
\text{HawaiianIsland} \sqsubseteq \text{Island} \sqcap \text{part-of.}\{\text{USA}\}, \\
\text{IonianIslandHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.IonianIsland}, \\
\text{NorthAegeanHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.NorthAegeanIsland}, \\
\text{HawaiianHolidaysOffer} \equiv \text{Offer} \sqcap \exists\ \text{in.HawaiianIsland} \} \\
\mathcal{A} = \{ \text{Country(Greece), Country(USA)} \}
\end{gathered}$$

Semantic similarity measure properties - example



IonianIsland $\sqsubseteq$ Island $\sqcap$ part-of.{Greece}
NorthAegeanIsland $\sqsubseteq$ Island $\sqcap$ part-of.{Greece}
HawaiianIsland $\sqsubseteq$ Island $\sqcap$ part-of.{USA}

Semantic similarity measure properties - example



IonianIslandHolidaysOffer $\equiv$ Offer $\sqcap \exists$ in.IonianIsland
NorthAegeanHolidaysOffer $\equiv$ Offer $\sqcap \exists$ in.NorthAegeanIsland
HawaiianHolidaysOffer $\equiv$ Offer $\sqcap \exists$ in.HawaiianIsland

Soundness

d’Amato et al. (EKAW 2008)

IonianIslandHolidaysOffer should be assesed more similar to **NorthAegeanHolidaysOffer** than to **HawaiianHolidaysOffer** since both are located in Greece

Equivalence soundness

d'Amato et al. (EKAW 2008)
Let us assume there exist two concept definitions:
SantoriniHolidaysOffer $\equiv$ Offer $\sqcap \exists$ in.Santorini $\sqcap \forall$ in.Santorini
ThiraHolidaysOffer $\equiv$ Offer $\sqcap \exists$ in.Santorini $\sqcap \forall$ in.Santorini

Since concept names **SantoriniHolidaysOffer** and **ThiraHolidaysOffer** represent semantically equivalent concepts, it should hold:

sim(SantoriniHolidaysOffer, HawaianHolidaysOffer) =
sim(ThiraHolidaysOffer, HawaianHolidaysOffer)

Santorini (officially **Thira)**

Disjointness compatibility

d'Amato et al. (EKAW 2008) Let us assume we assert in $\mathcal{K}$:

$$\text{IonianIslandHolidaysOffer} \sqcap \text{NorthAegeanHolidaysOffer} \equiv \perp$$

This should not necessarily mean the offers are totally different.

They both represented offers located in Greece, and thus have more commonalities then arbitrary offers. That's why it should hold:

sim(NorthAegeanHolidaysOffer, IonianIslandHolidaysOffer) > sim(NorthAegeanHolidaysOffer, Offer)

Semantic similarity measure properties

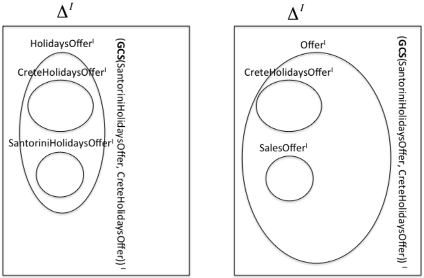
A truly semantic similarity measure should take into account compatibility of a concept (dis-)similarity with the semantics of background ontologies.
A set of criteria for a measure to satisfy to correctly handling ontological representations:

- soundness: ability to take the semantics of $\mathcal{K}$ (e.g. subsumption hierarchy) into account (d'Amato et al., EKAW 2008)
- equivalence soundness, equivalence invariance: ability to recognize semantically equivalent concepts as equal (d'Amato et al., EKAW 2008), (Lehmann and Turhan, Jelia 2012)
- equivalence closedness: two concepts are totally similar if and only if they are equivalent (Lehmann and Turhan, Jelia 2012)
- disjointness compatibility: ability to recognize similarities between disjoint concepts (d'Amato et al., EKAW 2008)

GCS-based semantic measure

d'Amato et al. (EKAW 2008)

- many of the "traditional" measures when applied to DLs, and also DL-specific measures fail to meet these semantic criteria
- "semantic" measure based on common super-concept (Good Common Subsumer, GCS of the concepts)
- two concepts are more similar as much their extensions are similar



Problem: GCS not defined for most expressive DLs

Acknowledgements

- Some presentation ideas inspired on/borrowed from: Nada Lavrac, Claudia d'Amato, Nicola Fanizzi, Jens Lehmann, Simon Razniewski, Fabian Suchanek, Heiko Paulheim, Johanna Voelker, and Jedrzej Potoniec
- Polish National Science Center (Grant No 2014/13/D/ST6/02076), grant entitled "ARISTOTELES: Methodology and algorithms for automatic revision of ontologies in task based scenarios" (2015-2018)