Uncertainty in the Semantic Web

Thomas Lukasiewicz

Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Outline

Uncertainty in the Web

Semantic Web

Probabilistic DLs
- Probabilistic Logics
- P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$

Probabilistic Fuzzy DL-Programs
- Soft Shopping Agent
- Fuzzy DLs
- Fuzzy DL-Programs
- Adding Probabilistic Uncertainty

Probabilistic Datalog+/–
- Datalog+/–
- Markov Logic Networks
- Probabilistic Datalog+/–

Outline

Uncertainty in the Web

Semantic Web

Probabilistic DLs

- Probabilistic Logics
- P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$

Probabilistic Fuzzy DL-Programs

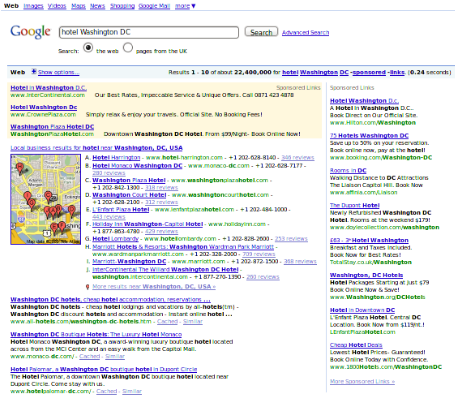
- Soft Shopping Agent
- Fuzzy DLs
- Fuzzy DL-Programs
- Adding Probabilistic Uncertainty

Probabilistic Datalog+/–

- Datalog+/–
- Markov Logic Networks
- Probabilistic Datalog+/–

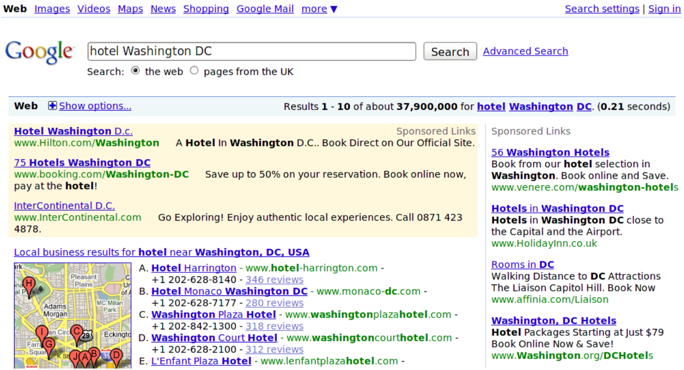
Web Search

Ranking of Web pages to be returned for a Web search query; e.g., via PageRank technique (based on statistical methods):



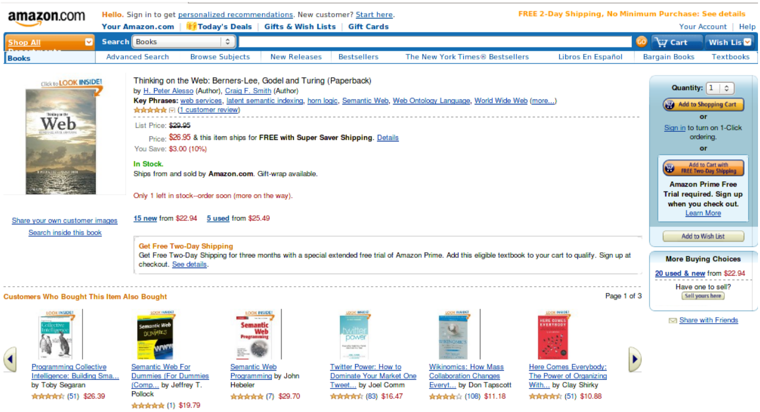
Computational Advertising

Find the best ad to present to a user in a given context, such as querying a search engine ("sponsored search"), reading a web page ("content match"), watching a movie, etc.



Recommender Systems

Present information items (movies, music, books, news, images, web pages, etc.) that may interest a user, e.g.,



Other Examples

- Web spam detection
- Information extraction
- Semantic annotation
- Trust and reputation
- User preference modeling
- Belief fusion and opinion pooling
- Machine translation
- Speech recognition
- Natural language processing
- Computer vision
- ...

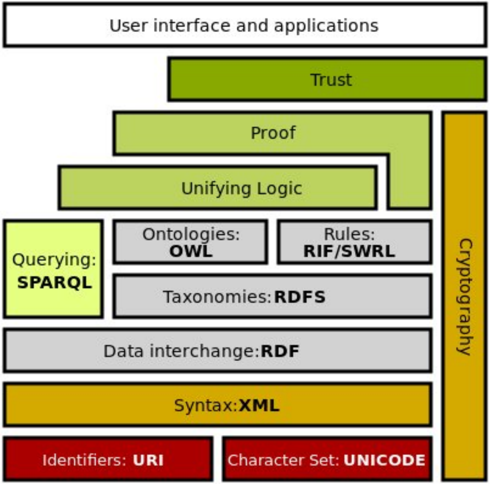
Outline

Semantic Web: Key Ideas

- Evolution of the current Web in which the meaning of information and services on the Web is defined...
- ...making it possible to understand and satisfy the requests of people and machines to use the Web content.
- Vision of the Web as a universal medium for data, information, and knowledge exchange.
- Extension of the current Web by standards and technologies that help machines to understand the information on the Web to support richer discovery, data integration, navigation, and automation of tasks.

- Use ontologies for a precise definition of shared terms in Web resources, use KR technology for automated reasoning from Web resources, and apply cooperative agent technology for processing the information of the Web.
- Consists of several *hierarchical layers*, including
  - the Ontology layer: *OWL Web Ontology Language*: *OWL Lite* $\approx \mathcal{SHIF}(\mathbf{D})$, *OWL DL* $\approx \mathcal{SHOIN}(\mathbf{D})$, *OWL Full*; recent tractable fragments: OWL EL, OWL QL, OWL RL;
  - the Rules layer: Rule Interchange Format (RIF);
  - the Logic and Proof layers, which should offer other sophisticated representation and reasoning capabilities.

Semantic Web Stack



Challenges (from Wikipedia)

Challenges [edit]

Some of the challenges for the Semantic Web include vastness, vagueness, uncertainty, inconsistency, and deceit. Automated reasoning systems will have to deal with all of these issues in order to deliver on the promise of the Semantic Web.

- Vastness: The World Wide Web contains many billions of pages. The SNOMED CT medical terminology ontology alone contains 370,000 class names, and existing technology has not yet been able to eliminate all semantically duplicated terms. Any automated reasoning system will have to deal with truly huge inputs.
- Vagueness: These are imprecise concepts like "young" or "tall". This arises from the vagueness of user queries, of concepts represented by content providers, of matching query terms to provider terms and of trying to combine different knowledge bases with overlapping but subtly different concepts. Fuzzy logic is the most common technique for dealing with vagueness.
- Uncertainty: These are precise concepts with uncertain values. For example, a patient might present a set of symptoms that correspond to a number of different distinct diagnoses each with a different probability. Probabilistic reasoning techniques are generally employed to address uncertainty.
- Inconsistency: These are logical contradictions that will inevitably arise during the development of large ontologies, and when ontologies from separate sources are combined. Deductive reasoning fails catastrophically when faced with inconsistency, because "anything follows from a contradiction". Defeasible reasoning and paraconsistent reasoning are two techniques that can be employed to deal with inconsistency.
- Deceit: This is when the producer of the information is intentionally misleading the consumer of the information. Cryptography techniques are currently utilized to alleviate this threat.

This list of challenges is illustrative rather than exhaustive, and it focuses on the challenges to the "unifying logic" and "proof" layers of the Semantic Web. The World Wide Web Consortium (W3C) Incubator Group for Uncertainty Reasoning for the World Wide Web (URW3-XG) final report lumps these problems together under the single heading of "uncertainty". Many of the techniques mentioned here will require extensions to the Web Ontology Language (OWL) for example to annotate conditional probabilities. This is an area of active research.[13]

Uncertainty (and Vagueness) in the Semantic Web

- Uncertainty: statements are true or false. But, due to lack of knowledge we can only estimate to which probability / possibility / necessity degree they are true or false, e.g., "John wins in the lottery with the probability 0.01".
- Vagueness: statements involve concepts for which there is no exact definition, such as tall, small, close, far, cheap, and expensive; statements are true to some degree, e.g., "Hotel Verdi is close to the train station to degree 0.83".
- Uncertainty and vagueness are important in the SW; many existing proposals for extensions of SW languages (RDF, OWL, DLs, rules) by uncertainty and vagueness.

In the following, some own such proposals: probabilistic DLs, probabilistic fuzzy dl-programs, and probabilistic Datalog+/–.

Outline

Uncertainty in the Web

Semantic Web

Probabilistic DLs
- Probabilistic Logics
- P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$

Probabilistic Fuzzy DL-Programs
- Soft Shopping Agent
- Fuzzy DLs
- Fuzzy DL-Programs
- Adding Probabilistic Uncertainty

Probabilistic Datalog+/–
- Datalog+/–
- Markov Logic Networks
- Probabilistic Datalog+/–

Probabilistic Ontologies

Generalization of classical ontologies by probabilistic knowledge.

Main types of encoded probabilistic knowledge:

- Terminological probabilistic knowledge about concepts and roles:

  "Birds fly with a probability of at least 0.95".

- Assertional probabilistic knowledge about instances of concepts and roles:

  "Tweety is a bird with a probability of at least 0.9".

Use of Probabilistic Ontologies

- In medicine, biology, defense, astronomy, ...
- In the Semantic Web:
  - Quantifying the degrees of overlap between concepts, to use them in Semantic Web applications: information retrieval, personalization, recommender systems, ...
  - Information retrieval, for an increased recall (e.g., Udrea et al.: Probabilistic ontologies and relational databases. In *Proc. CoopIS/DOA/ODBASE-2005*).
  - Ontology matching (e.g., Mitra et al.: OMEN: A probabilistic ontology mapping tool. In *Proc. ISWC-2005*).
  - Probabilistic data integration, especially for handling ambiguous and inconsistent pieces of information.

Description Logics: Key Ideas

Description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between classes of individuals, respectively.

A description logic knowledge base encodes in particular subset relationships between concepts, subset relationships between roles, the membership of individuals to concepts, and the membership of pairs of individuals to roles.

Here, description logic knowledge bases in $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ (which are the DLs behind OWL Lite and OWL DL, respectively).

Example

Description logic knowledge base L for an online store:

(1) $\mathit{Textbook} \sqsubseteq \mathit{Book}$; (2) $\mathit{PC} \sqcup \mathit{Laptop} \sqsubseteq \mathit{Electronics}$; $\mathit{PC} \sqsubseteq \neg\mathit{Laptop}$;
(3) $\mathit{Book} \sqcup \mathit{Electronics} \sqsubseteq \mathit{Product}$; $\mathit{Book} \sqsubseteq \neg\mathit{Electronics}$;
(4) $\mathit{Sale} \sqsubseteq \mathit{Product}$;
(5) $\mathit{Product} \sqsubseteq\ \geqslant 1\, \mathit{related}$; (6) $\geqslant 1\, \mathit{related} \sqcup \geqslant 1\, \mathit{related}^{-} \sqsubseteq \mathit{Product}$;
(7) $\mathit{related} \sqsubseteq \mathit{related}^{-}$; $\mathit{related}^{-} \sqsubseteq \mathit{related}$;
(8) $\mathit{Textbook}(\mathit{tb\_ai})$; $\mathit{Textbook}(\mathit{tb\_lp})$; (9) $\mathit{related}(\mathit{tb\_ai}, \mathit{tb\_lp})$;
(10) $\mathit{PC}(\mathit{pc\_ibm})$; $\mathit{PC}(\mathit{pc\_hp})$; (11) $\mathit{related}(\mathit{pc\_ibm}, \mathit{pc\_hp})$;
(12) $\mathit{provides}(\mathit{ibm}, \mathit{pc\_ibm})$; $\mathit{provides}(\mathit{hp}, \mathit{pc\_hp})$.

Probabilistic Logics: Key Ideas

- Integration of (propositional) logic- and probability-based representation and reasoning formalisms.
- Reasoning from logical constraints and interval restrictions for conditional probabilities (also called *conditional constraints*).
- Reasoning from convex sets of probability distributions.
- Model-theoretic notion of logical entailment.

Syntax of Probabilistic Knowledge Bases

- Finite nonempty set of basic events $\Phi = \{p_1, \ldots, p_n\}$.
- Event ϕ: Boolean combination of basic events
- Logical constraint $\psi \Leftarrow \phi$: events ψ and ϕ: "ϕ implies ψ".
- Conditional constraint $(\psi|\phi)[l, u]$: events ψ and ϕ, and $l, u \in [0, 1]$: "conditional probability of ψ given ϕ is in $[l, u]$".
- Probabilistic knowledge base $KB = (L, P)$:
  - finite set of logical constraints L,
  - finite set of conditional constraints P.

Example

Probabilistic knowledge base $KB = (L, P)$:

- $L = \{bird \Leftarrow eagle\}$:

  "All eagles are birds".

- $P = \{(have\_legs \mid bird)[1, 1], (fly \mid bird)[0.95, 1]\}$:

  "All birds have legs".

  "Birds fly with a probability of at least 0.95".

Semantics of Probabilistic Knowledge Bases

- World I: truth assignment to all basic events in Φ.
- $\mathcal{I}_\Phi$: all worlds for Φ.
- Probabilistic interpretation Pr: probability function on $\mathcal{I}_\Phi$.
- $\Pr(\phi)$: sum of all $\Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models \phi$.
- $\Pr(\psi|\phi)$: if $\Pr(\phi) > 0$, then $\Pr(\psi|\phi) = \Pr(\psi \wedge \phi) \,/\, \Pr(\phi)$.
- Truth under Pr:
  - $\Pr \models \psi \Leftarrow \phi$ iff $\Pr(\psi \wedge \phi) = \Pr(\phi)$
    (iff $\Pr(\psi \Leftarrow \phi) = 1$).
  - $\Pr \models (\psi|\phi)[l, u]$ iff $\Pr(\psi \wedge \phi) \in [l, u] \cdot \Pr(\phi)$
    (iff either $\Pr(\phi) = 0$ or $\Pr(\psi|\phi) \in [l, u]$).

Example

- Set of basic propositions $\Phi = \{bird, fly\}$.
- $\mathcal{I}_\Phi$ contains exactly the worlds I_1, I_2, I_3, and I_4 over Φ:

	fly	$\neg$*fly*
bird	I_1	I_2
$\neg$*bird*	I_3	I_4

- Some probabilistic interpretations:

Pr_1	*fly*	$\neg$*fly*
bird	19/40	1/40
$\neg$*bird*	10/40	10/40

Pr_2	*fly*	$\neg$*fly*
bird	0	1/3
$\neg$*bird*	1/3	1/3

- $\mathrm{Pr}_1(fly \wedge bird) = 19/40$ and $\mathrm{Pr}_1(bird) = 20/40$.
- $\mathrm{Pr}_2(fly \wedge bird) = 0$ and $\mathrm{Pr}_2(bird) = 1/3$.
- $\neg fly \Leftarrow bird$ is false in Pr_1, but true in Pr_2.
- $(fly \,|\, bird)[.95, 1]$ is true in Pr_1, but false in Pr_2.

Satisfiability and Logical Entailment

- Pr is a model of $KB = (L, P)$ iff $\Pr \models F$ for all $F \in L \cup P$.
- KB is satisfiable iff a model of KB exists.
- $KB \Vdash (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a logical consequence of KB iff every model of KB is also a model of $(\psi|\phi)[l, u]$.
- $KB \models_{tight} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a tight logical consequence of KB iff l (resp., u) is the infimum (resp., supremum) of $\Pr(\psi|\phi)$ subject to all models Pr of KB with $\Pr(\phi) > 0$.

Example

- Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95,1]\}).$$

- KB is satisfiable, since

  Pr with $\Pr(bird \wedge eagle \wedge have\_legs \wedge fly) = 1$ is a model.

- Some conclusions under logical entailment:

$$KB \Vdash (have\_legs \mid bird)[0.3,1], \quad KB \Vdash (fly \mid bird)[0.6,1].$$

- Tight conclusions under logical entailment:

$$KB \models_{tight} (have\_legs \mid bird)[1,1], \quad KB \models_{tight} (fly \mid bird)[0.95,1],$$
$$KB \models_{tight} (have\_legs \mid eagle)[1,1], \quad KB \models_{tight} (fly \mid eagle)[0,1].$$

Towards Stronger Notions of Entailment

Problem: Inferential weakness of logical entailment.

Solutions:

- Probability selection techniques: Perform inference from a representative distribution of the encoded convex set of distributions rather than the whole set, e.g.,
  - distribution of maximum entropy,
  - distribution in the center of mass.
- Probabilistic default reasoning: Perform constraining rather than conditioning and apply techniques from default reasoning to resolve local inconsistencies.
- Probabilistic independencies: Further constrain the convex set of distributions by probabilistic independencies.
  ($\Rightarrow$ adds nonlinear equations to linear constraints)

Logical vs. Lexicographic Entailment

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow eagle\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[0.95,1]\}).$$

Tight conclusions under logical entailment:

$$KB \models_{tight} (have\_legs \mid bird)[1,1], \quad KB \models_{tight} (fly \mid bird)[0.95,1],$$
$$KB \models_{tight} (have\_legs \mid eagle)[1,1], \quad KB \models_{tight} (fly \mid eagle)[0,1].$$

Tight conclusions under probabilistic lexicographic entailment:

$$KB \mid\!\!\mid\!\!\!\sim^{lex}_{tight} (have\_legs \mid bird)[1,1], \quad KB \mid\!\!\mid\!\!\!\sim^{lex}_{tight} (fly \mid bird)[0.95,1],$$
$$KB \mid\!\!\mid\!\!\!\sim^{lex}_{tight} (have\_legs \mid eagle)[1,1], \quad KB \mid\!\!\mid\!\!\!\sim^{lex}_{tight} (fly \mid eagle)[0.95,1].$$

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow penguin\}, \{(have\_legs \mid bird)[1,1], (fly \mid bird)[1,1], (fly \mid penguin)[0, 0.05]\})\,.$$

Tight conclusions under logical entailment:

$$KB \models_{tight} (have\_legs \mid bird)[1,1],\ KB \models_{tight} (fly \mid bird)[1,1],$$
$$KB \models_{tight} (have\_legs \mid penguin)[1,0],\ KB \models_{tight} (fly \mid penguin)[1,0]\,.$$

Tight conclusions under probabilistic lexicographic entailment:

$$KB \Vdash\!\sim^{lex}_{tight} (have\_legs \mid bird)[1,1],\ KB \Vdash\!\sim^{lex}_{tight} (fly \mid bird)[1,1],$$
$$KB \Vdash\!\sim^{lex}_{tight} (have\_legs \mid penguin)[1,1],\ KB \Vdash\!\sim^{lex}_{tight} (fly \mid penguin)[0,0.05].$$

Probabilistic knowledge base:

$$KB = (\{bird \Leftarrow penguin\}, \{(have\_legs \mid bird)[0.99, 1], (fly \mid bird)[0.95, 1], (fly \mid penguin)[0, 0.05]\}).$$

Tight conclusions under logical entailment:

$$KB \models_{tight} (have\_legs \mid bird)[0.99, 1], \; KB \models_{tight} (fly \mid bird)[0.95, 1],$$
$$KB \models_{tight} (have\_legs \mid penguin)[0, 1], \; KB \models_{tight} (fly \mid penguin)[0, 0.05].$$

Tight conclusions under probabilistic lexicographic entailment:

$$KB \mathrel{\|\!\!\sim}^{lex}_{tight} (have\_legs \mid bird)[0.99, 1], \; KB \mathrel{\|\!\!\sim}^{lex}_{tight} (fly \mid bird)[0.95, 1],$$
$$KB \mathrel{\|\!\!\sim}^{lex}_{tight} (have\_legs \mid penguin)[0.99, 1], \; KB \mathrel{\|\!\!\sim}^{lex}_{tight} (fly \mid penguin)[0, 0.05].$$

P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$: Key Ideas

- probabilistic generalization of the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ behind OWL Lite and OWL DL, respectively
- terminological probabilistic knowledge about concepts and roles
- assertional probabilistic knowledge about instances of concepts and roles
- terminological probabilistic inference based on lexicographic entailment in probabilistic logic (stronger than logical entailment)
- assertional probabilistic inference based on lexicographic entailment in probabilistic logic (for combining assertional and terminological probabilistic knowledge)
- terminological and assertional probabilistic inference problems reduced to sequences of linear optimization problems

References

• T. Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6/7):852–883, April 2008.

• T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the Semantic Web. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):291–308, November 2008.

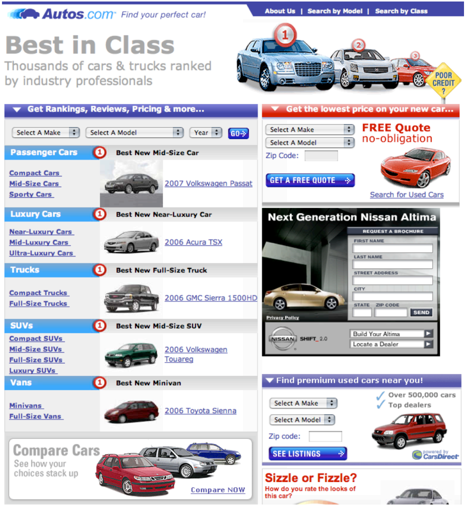
Outline

Example: Soft Shopping Agent

Suppose a person would like to buy “a sports car that costs at most about 22 000 EUR and has a power of around 150 HP”.

In todays Web, the buyer has to *manually*

- search for car selling web sites, e.g., using Google;
- select the most promising sites;
- browse through them, query them to see the cars that each site sells, and match the cars with the requirements;
- select the offers in each web site that match the requirements; and
- eventually merge all the best offers from each site and select the best ones.


Autos.com Find your perfect car!
About Us | Search by Model | Search by Class
Best in Class
Thousands of cars & trucks ranked by industry professionals
POOR CREDIT ?
Get Rankings, Reviews, Pricing & more...
Select A Make
Select A Model
Year
GO
Passenger Cars
Best New Mid-Size Car
Compact Cars
Mid-Size Cars
Sporty Cars
2007 Volkswagen Passat
Luxury Cars
Best New Near-Luxury Car
Near-Luxury Cars
Mid-Luxury Cars
Ultra-Luxury Cars
2006 Acura TSX
Trucks
Best New Full-Size Truck
Compact Trucks
Full-Size Trucks
2006 GMC Sierra 1500HD
SUVs
Best New Mid-Size SUV
Compact SUVs
Mid-Size SUVs
Full-Size SUVs
Luxury SUVs
2006 Volkswagen Touareg
Vans
Best New Minivan
Minivans
Full-Size Vans
2006 Toyota Sienna
Compare Cars
See how your choices stack up
Compare NOW
Get the lowest price on your new car...
Select A Make
Select A Model
Zip Code:
FREE Quote
no-obligation
GET A FREE QUOTE
Search for Used Cars
Next Generation Nissan Altima
REQUEST A BROCHURE
FIRST NAME
LAST NAME
STREET ADDRESS
CITY
STATE
ZIP CODE
SEND
Privacy Policy
NISSAN
SHIFT_2.0
Build Your Altima
Locate a Dealer
Find premium used cars near you!
Select A Make
Select A Model
Zip code:
Over 500,000 cars
Top dealers
SEE LISTINGS
powered by CarsDirect
Sizzle or Fizzle?
How do you rate the looks of this car?

Autos.com Find your perfect car!

About Us | Search by Model

Home > Sporty Car > Mazda > 2007 Mazda MX-5 Miata Factsheet

All Mazda Models

2007 Mazda MX-5 Miata expert reviews and lowest prices

in Sporty Car

Factsheet

Selling Point

MPG

See all

Get a FREE Price Quote!

Zip Code: GET A PRICE

Sizzle or Fizzle?

How do you rate the looks of this car?

1 2 3 4 5

Fizzle Sizzle

Vote and see how others voted!

2007 Mazda MX-5 Miata		Sporty Car Average	
SV 2dr Convertible			
Expert Reviews	unavailable	4.0 ★★★★☆	Rank all
MSRP	$20,435	$27,724	Rank all
Invoice	$18,883	$25,582	Rank all
0 to 60 Acceleration	7.8 sec	7.53 sec	Rank all
MPG	25/30	23 MPG	Rank all
Resale Value	3.0 ★★★☆☆	2.0 ★★☆☆☆	Rank all
Performance and Handling ▶ see details	4.0 ★★★★☆	4.4 ★★★★☆	Rank all
Comfort and Convenience ▶ see details	2.0 ★★☆☆☆	2.8 ★★★☆☆	Rank all
Safety Features ▶ see details	2.0 ★★☆☆☆	2.1 ★★☆☆☆	Rank all
Passenger Space ▶ see details	1.1 ★☆☆☆☆	3.0 ★★★☆☆	Rank all
Cargo Capacity ▶ see details	1.6 ★★☆☆☆	2.4 ★★☆☆☆	Rank all
Sizzle or Fizzle	2.9 ★★☆☆☆	3.0 ★★★☆☆	Rank all

A *shopping agent* may support us, *automatizing* the whole process once it receives the request/query q from the buyer:

- The agent selects some sites/resources S that it considers as *relevant* to q (represented by probabilistic rules).
- For the top-k selected sites, the agent has to reformulate q using the terminology/ontology of the specific car selling site (which is done using probabilistic rules).
- The query q may contain many so-called *vague/fuzzy* concepts such as "the prize is around 22 000 EUR or less", and thus a car may *match* q to a *degree*. So, a resource returns a ranked list of cars, where the ranks depend on the degrees to which the cars match q.
- Eventually, the agent integrates the ranked lists (using probabilities) and shows the top-n items to the buyer.

Fuzzy DLs: Key Ideas

Description logics model a domain of interest in terms of concepts and roles, which represent classes of individuals and binary relations between classes of individuals, respectively.

A description logic knowledge base encodes in particular subset relationships between concepts, subset relationships between roles, the membership of individuals to concepts, and the membership of pairs of individuals to roles.

In fuzzy description logics, these relationships and memberships then have a degree of truth in $[0, 1]$.

Example

$$Cars \sqcup Trucks \sqcup Vans \sqcup SUVs \sqsubseteq Vehicles$$
$$PassengerCars \sqcup LuxuryCars \sqsubseteq Cars$$
$$CompactCars \sqcup MidSizeCars \sqcup SportyCars \sqsubseteq PassengerCars$$

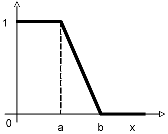
$$\begin{aligned} Cars \sqsubseteq\ & (\exists hasReview.Integer) \sqcap (\exists hasInvoice.Integer) \\ & \sqcap (\exists hasResellValue.Integer) \sqcap (\exists hasMaxSpeed.Integer) \\ & \sqcap (\exists hasHorsePower.Integer) \sqcap \ldots \end{aligned}$$

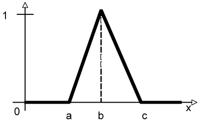
$$\begin{aligned} & MazdaMX5Miata : SportyCar \sqcap (\exists hasInvoice.18883) \\ & \qquad \sqcap (\exists hasHorsePower.166) \sqcap \ldots \\ & MitsubishiEclipseSpyder : SportyCar \sqcap (\exists hasInvoice.24029) \\ & \qquad \sqcap (\exists hasHorsePower.162) \sqcap \ldots \end{aligned}$$

We may now encode “costs at most about 22 000 EUR” and “has a power of around 150 HP” in the buyer’s request through the following concepts C and D, respectively:

$$C = \exists \mathit{hasInvoice}.\mathit{LeqAbout22000} \quad \text{and}$$
$$D = \exists \mathit{hasHorsePower}.\mathit{Around150HP},$$

where $\mathit{LeqAbout22000} = L(22000, 25000)$ and $\mathit{Around150HP} = \mathit{Tri}(125, 150, 175)$.





Fuzzy DL-Programs: Syntax

A *normal fuzzy rule r is of the form (with atoms* $a, b_1, \ldots, b_m$*):*

$$\begin{aligned} a \leftarrow_{\otimes_0} b_1 \wedge_{\otimes_1} b_2 \wedge_{\otimes_2} \cdots \wedge_{\otimes_{k-1}} b_k \wedge_{\otimes_k} \\ not_{\ominus_{k+1}} b_{k+1} \wedge_{\otimes_{k+1}} \cdots \wedge_{\otimes_{m-1}} not_{\ominus_m} b_m \geqslant v, \end{aligned} \tag{1}$$

A normal fuzzy program P is a finite set of normal fuzzy rules.

A *dl-query* $Q(\mathbf{t})$ *is of one of the following forms:*

- *a concept inclusion axiom* F *or its negation* $\neg F$*;*
- $C(t)$ *or* $\neg C(t)$*, with a concept* C *and a term* t*;*
- $R(t_1, t_2)$ *or* $\neg R(t_1, t_2)$*, with a role* R *and terms* t_1, t_2*.*

A *fuzzy dl-rule r is of form (1), where any* $b \in B(r)$ *may be a dl-atom, which is of form* $DL[S_1 op_1 p_1, \ldots, S_m op_m\, p_m; Q](\mathbf{t})$*.*

A *fuzzy dl-program* $KB = (L, P)$ *consists of a fuzzy description logic knowledge base* L *and a finite set of fuzzy dl-rules* P*.*

Example

The following fuzzy dl-rule encodes the buyer's request "a sports car that costs at most about 22 000 EUR and that has a power of around 150 HP".

$$\begin{aligned} query(x) \quad \leftarrow_{\otimes} \quad & DL[SportyCar](x) \wedge_{\otimes} \\ & DL[hasInvoice](x, y_1) \wedge_{\otimes} \\ & DL[LeqAbout22000](y_1) \wedge_{\otimes} \\ & DL[hasHorsePower](x, y_2) \wedge_{\otimes} \\ & DL[Around150HP](y_2) \geqslant 1\,. \end{aligned}$$

Here, $\otimes$ is the Gödel t-norm (that is, $x \otimes y = \min(x, y)$).

Fuzzy DL-Programs: Semantics

An *interpretation I is a mapping $I\colon HB_P \rightarrow [0,1]$.*

The truth value of $a = DL[S_1 \uplus\ p_1, \ldots, S_m \uplus p_m; Q](\mathbf{c})$ under L, denoted $I_L(a)$, is defined as the maximal truth value $v \in [0,1]$ such that $L \cup \bigcup_{i=1}^{m} A_i(I) \models Q(\mathbf{c}) \geqslant v$, where

$$A_i(I) = \{S_i(\mathbf{e}) \geqslant I(p_i(\mathbf{e})) \mid I(p_i(\mathbf{e})) > 0,\ p_i(\mathbf{e}) \in HB_P\}\,.$$

I is a model of a ground fuzzy dl-rule r of the form (1) under L, denoted $I \models_L r$, iff

$$\begin{aligned} I_L(a) \geqslant v \otimes_0 I_L(b_1) \otimes_1 I_L(b_2) \otimes_2 \cdots \otimes_{k-1} I_L(b_k) \otimes_k \\ \ominus_{k+1} I_L(b_{k+1}) \otimes_{k+1} \cdots \otimes_{m-1} \ominus_m I_L(b_m), \end{aligned}$$

I is a *model of a fuzzy dl-program $KB = (L, P)$, denoted $I \models KB$, iff $I \models_L r$ for all $r \in ground(P)$.*

Stratified Fuzzy DL-Programs

Stratified fuzzy dl-programs are composed of hierarchic layers of positive fuzzy dl-programs linked via default negation:

A *stratification of* $KB = (L, P)$ with respect to DL_P is a mapping $\lambda\colon HB_P \cup DL_P \to \{0, 1, \ldots, k\}$ such that

- $\lambda(H(r)) \geqslant \lambda(a)$ (resp., $\lambda(H(r)) > \lambda(a)$) for each $r \in ground(P)$ and $a \in B^+(r)$ (resp., $a \in B^-(r)$), and
- $\lambda(a) \geqslant \lambda(a')$ for each input atom a' of each $a \in DL_P$,

where $k \geqslant 0$ is the *length of* λ*. A fuzzy dl-program* $KB = (L, P)$ *is* stratified iff it has a stratification λ of some length $k \geqslant 0$.

Theorem: Every stratified fuzzy dl-program KB is satisfiable and has a canonical minimal model via a finite number of iterative least models (which does not depend on the stratification of KB).

Adding Probabilistic Uncertainty: Example

The buyer's request, but in a "different" terminology:

$$\begin{aligned} \mathit{query}(x) \leftarrow_{\otimes} \; & \mathit{SportsCar}(x) \wedge_{\otimes} \mathit{hasPrize}(x, y_1) \wedge_{\otimes} \mathit{hasPower}(x, y_2) \wedge_{\otimes} \\ & \mathit{DL}[\mathit{LeqAbout22000}](y_1) \wedge_{\otimes} \mathit{DL}[\mathit{Around150HP}](y_2) \geqslant 1 \end{aligned}$$

Ontology alignment mapping rules:

$$\begin{aligned} \mathit{SportsCar}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{SportyCar}](x) \wedge_{\otimes} \mathit{sc}_{\mathit{pos}} \geqslant 1 \\ \mathit{hasPrize}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{hasInvoice}](x) \wedge_{\otimes} \mathit{hi}_{\mathit{pos}} \geqslant 1 \\ \mathit{hasPower}(x) &\leftarrow_{\otimes} \mathit{DL}[\mathit{hasHorsePower}](x) \wedge_{\otimes} \mathit{hhp}_{\mathit{pos}} \geqslant 1\,, \end{aligned}$$

Probability distribution μ:

$$\begin{array}{ll} \mu(\mathit{sc}_{\mathit{pos}}) = 0.91 & \mu(\mathit{sc}_{\mathit{neg}}) = 0.09 \\ \mu(\mathit{hi}_{\mathit{pos}}) = 0.78 & \mu(\mathit{hi}_{\mathit{neg}}) = 0.22 \\ \mu(\mathit{hhp}_{\mathit{pos}}) = 0.83 & \mu(\mathit{hhp}_{\mathit{neg}}) = 0.17\,. \end{array}$$

The following are some tight consequences:

$$KB \;\Vdash\!\!\sim_{\mathit{tight}}\; (\mathbf{E}[q(\mathit{MazdaMX5Miata})])[0.21, 0.21]$$
$$KB \;\Vdash\!\!\sim_{\mathit{tight}}\; (\mathbf{E}[q(\mathit{MitsubishiEclipseSpyder})])[0.19, 0.19]\,.$$

Informally, the expected degree to which *MazdaMX5Miata* matches the query q is 0.21, while the expected degree to which *MitsubishiEclipseSpyder* matches the query q is 0.19,

Thus, the shopping agent ranks the retrieved items as follows:

rank	item	degree
1.	*MazdaMX5Miata*	0.21
2.	*MitsubishiEclipseSpyder*	0.19

Summary

- Description logic programs that allow for dealing with probabilistic uncertainty and fuzzy vagueness.
- Semantically, probabilistic uncertainty can be used for data integration and ontology mapping, and fuzzy vagueness can be used for expressing vague concepts.
- Query processing based on fixpoint iterations.

References

• T. Lukasiewicz and U. Straccia. Description logic programs under probabilistic uncertainty and fuzzy vagueness. *International Journal of Approximate Reasoning*, 50(6):837–853, June 2009.

• T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. *Artificial Intelligence*, 172(12/13):1495–1539, August 2008. (AIJ Prominent Paper Award 2013.)

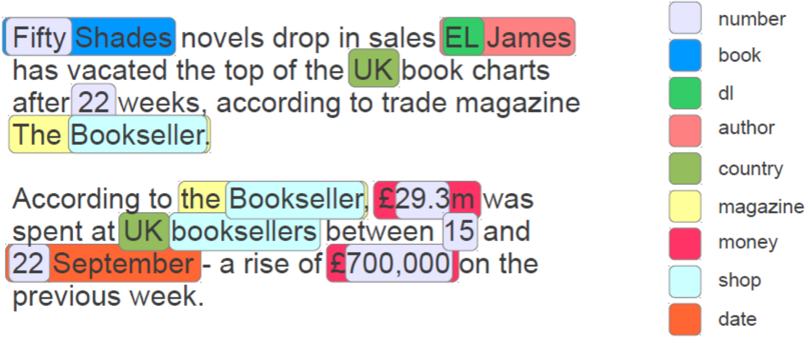
Outline

Probabilistic Datalog+/–: Key Ideas

- Probabilistic Datalog+/– ontologies combine "classical" Datalog+/– with Markov logic networks (MLNs).
- The basic idea is that formulas (TGDs, EGDs, and NCs) are annotated with a set of probabilistic events.
- Event annotations mean that the formula in question only applies when the associated event holds.
- The probability distribution associated with the events is described in the MLN.
- Key computational problems: answering ranking queries, conjunctive queries, and threshold queries.
- Application in data extraction from the Web, where Datalog+/– is used as data extraction language (DIADEM).

Example

Consider the problem of entity extraction over the following text snippet:



Datalog+/−: Encoding Ontologies in Datalog

Plain Datalog allows for encoding some ontological axioms:

- concept inclusion axioms:

  $person(X) \leftarrow employee(X)$ iff $employee \sqsubseteq person$;

- role inclusion axioms:

  $manages(X, Y) \leftarrow reportsTo(Y, X)$ iff $reportsTo^{-1} \sqsubseteq manages$;

- concept and role membership axioms:

  $person(\text{John}) \leftarrow$ iff $person(\text{John})$;

  $manages(\text{Bill}, \text{John}) \leftarrow$ iff $manages(\text{Bill}, \text{John})$.

- transitivity axioms:

  $manages(X, Y) \leftarrow manages(X, Z), manages(Z, Y)$ iff (Trans $manages$)

However, it cannot express other important ontological axioms:

- concept inclusion axioms involving existential restrictions on roles in the head:

  $\mathit{Scientist} \sqsubseteq \exists \mathit{isAuthorOf}$;

- concept inclusion axioms stating concept disjointness:

  $\mathit{JournalPaper} \sqsubseteq \neg \mathit{ConferencePaper}$;

- functionality axioms:

  (funct *hasFirstAuthor*).

Question: Can Datalog be extended in such a way that it can be used as ontology language?

Answer: Yes, by introducing:

- tuple-generating dependencies (TGDs):

  $\forall \mathbf{X} \forall \mathbf{Y} \exists \mathbf{Z}\ \Psi(\mathbf{X}, \mathbf{Z}) \leftarrow \Phi(\mathbf{X}, \mathbf{Y})$,
  where $\Phi(\mathbf{X}, \mathbf{Y})$ and $\Psi(\mathbf{X}, \mathbf{Z})$ are conjunctions of atoms;

  Example: $\exists P\, directs(M, P) \leftarrow manager(M)$;

- negative constraints:

  $\forall \mathbf{X}\ \perp \leftarrow \Phi(\mathbf{X})$,
  where $\Phi(\mathbf{X})$ is a conjunction of atoms;

  Example: $\perp \leftarrow c(X), c'(X)$;

- equality-generating dependencies (EGDs):

  $\forall \mathbf{X}\ X_i = X_j \leftarrow \Phi(\mathbf{X})$,
  where $X_i, X_j \in \mathbf{X}$, and $\Phi(\mathbf{X})$ is a conjunction of atoms

  Example: $Y = Z \leftarrow r_1(X, Y), r_2(Y, Z)$.

The Chase

Given:

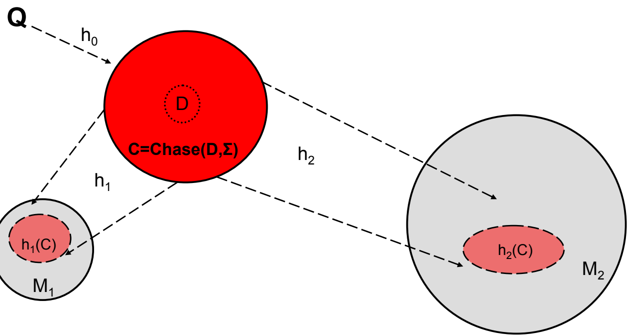
- D: database over $\mathrm{dom}(D)$.
- Σ: set of TGDs and/or EGDs

Question: How do we perform query answering?

Answer: Via the chase: If $D \not\models \Sigma$, then

- either $D \cup \Sigma$ is unsatisfiable due to a "hard" EGD violation, or
- the rules in Σ can be enforced via the chase by
  - adding facts in order to satisfy TGDs, where null values are introduced for $\exists$-variables
  - equating nulls with other nulls or with $\mathrm{dom}(D)$ elements in order to satisfy EGDs.

The Chase is a Universal Model



For each other model M of D and Σ,
there is a homomorphism from chase(D, Σ) to M.

$\Rightarrow$ conjunctive queries to $D \cup \Sigma$ can be evaluated on chase(D, Σ):

$$D \cup \Sigma \models Q \;\text{ iff }\; \text{chase}(D, \Sigma) \models Q$$

Facts about the Chase

- Depends on the order of rule applications:

  Example: $D = \{p(a)\}$ and $\Sigma = \{p(x) \rightarrow \exists y\, q(y);\ p(x) \rightarrow q(x)\}$:

  Solution 1 $= \{p(a), q(u), q(a)\}$
  Solution 2 $= \{p(a), q(a)\}$

  $\Rightarrow$ Assume a canonical ordering.

- Can be infinite:

  Example: $D = \{p(a, b)\}$ and $\Sigma = \{p(x, y) \rightarrow \exists z\, p(y, z)\}$:

  Solution $= \{p(a, b),\ p(b, u_1),\ p(u_1, u_2),\ p(u_2, u_3),\ \ldots\}$

  $\Rightarrow$ Query answering for D and TGDs alone is undecidable.
  $\Rightarrow$ Restrictions on TGDs and their interplay with EGDs.

Guarded and Linear Datalog+/–

A TGD σ is guarded iff it contains an atom in its body that contains all universally quantified variables of σ.

Example:

- $r(X, Y), s(Y, X, Z) \rightarrow \exists W\, s(Z, X, W)$ is guarded, where $s(Y, X, Z)$ is the guard, and $r(X, Y)$ is a side atom;
- $r(X, Y), r(Y, Z) \rightarrow r(X, Z)$ is not guarded.

A TGD is linear iff it contains only a singleton body atom.

Example:

- $manager(M) \rightarrow \exists P\, directs(M, P)$ is linear;
- $r(X, Y), s(Y, X, Z) \rightarrow \exists W\, s(Z, X, W)$ is not linear.

Markov Logic Networks

- We use Markov logic networks (MLNs) to represent uncertainty in Datalog+/–.
- MLNs combine classical Markov networks (a.k.a. Markov random fields) with first-order logic (FOL).
- We assume a set of random variables $X = \{X_1, \ldots, X_n\}$, where each X_i can take values in $Dom(X_i)$.
- A value for X is a mapping $x\colon X \to \bigcup_{i=1}^{n} Dom(X_i)$ such that $x(X_i) \in Dom(X_i)$.
- MLN: set of pairs (F, w), where F is a FO formula, and w is a real number.

- The probability distribution represented by the MLN is:

$$P(X = x) = \frac{1}{Z} \cdot exp(\sum_j w_j \cdot n_j(x)),$$

  where n_j is the number of ground instances of formula F_j made true by x, w_j is the weight of formula F_j, and $Z = \sum_{x \in X} exp(\sum_j w_j \cdot n_j(x))$ (normalization constant).

- Exact inference is $\#$P-complete, but MCMC methods obtain good approximations in practice.

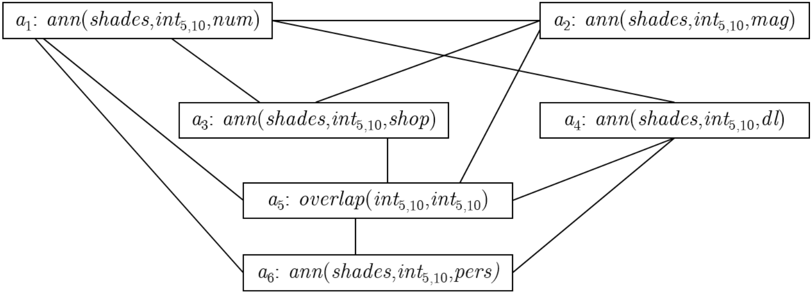
- A particularly costly step is the computation of Z, but this is a one-time calculation.

Example

Consider the following MLN:

$$\phi_1 : ann(S_1, I_1, num) \wedge ann(S_2, I_2, X) \wedge overlap(I_1, I_2) : 3$$
$$\phi_2 : ann(S_1, I_1, shop) \wedge ann(S_2, I_2, mag) \wedge overlap(I_1, I_2) : 1$$
$$\phi_3 : ann(S_1, I_1, dl) \wedge ann(S_2, I_2, pers) \wedge overlap(I_1, I_2) : 0.25$$

Graph representation (for a specific set of constants):



Computing probabilities w.r.t. this MLN:

λ_i	a_1	a_2	a_3	a_4	a_5	a_6	SAT	Probability
1	False	False	False	False	False	False	–	e^0 / Z
2	False	False	False	True	True	True	ϕ_3	$e^{0.25}$ / Z
3	True	False	False	True	True	True	ϕ_1, ϕ_3	$e^{3+0.25}$ / Z
4	True	False	True	True	True	True	ϕ_1, ϕ_3	$e^{3+0.25}$ / Z
5	False	True	False	False	True	False	–	e^0 / Z
6	False	True	True	False	True	True	ϕ_2	e^1 / Z
7	False	True	True	True	True	True	ϕ_2, ϕ_3	$e^{1+0.25}$ / Z
8	True	True	True	True	True	True	ϕ_1, ϕ_2, ϕ_3	$e^{3+1+0.25}$ / Z

. . . (64 possible settings for the binary random variables)

Probabilistic Datalog+/– Ontologies

- A probabilistic Datalog+/– ontology consists of a classical Datalog+/– ontology O along with an MLN M.

  Notation: $KB = (O, M)$

- Formulas in O are annotated with a set of pairs $\langle X_i = x_i \rangle$, with $x_i \in \{true, false\}$ (we also use 0 and 1, respectively).

  Variables that do not appear in the annotation are unconstrained.

  Possible world: a set of pairs $\langle X_i = x_i \rangle$ where each $X_i \in X$ has a corresponding pair.

- Basic intuition: given a possible world, a subset of the formulas in O is induced.

Example Revisited

The following formulas were adapted from the previous examples to give rise to a probabilistic Datalog+/– ontology:

$$
\begin{array}{ll}
book(X) \rightarrow editorialProd(X) & : \{\} \\
magazine(X) \rightarrow editorialProd(X) & : \{\} \\
author(X) \rightarrow person(X,P) & : \{\} \\
descLogic(X) \wedge author(X) \rightarrow \bot & : \{ann(\mathrm{X},I_1,dl) = 1 \wedge ann(\mathrm{X},I_2,pers) = 1 \\
 & \ \ overlap(I_1,I_2) = 0\} \\
shop(X) \wedge editorialProd(X) \rightarrow \bot & : \{ann(\mathrm{X},I_1,shop) = 1 \wedge ann(\mathrm{X},I_2,mag) = 1 \\
 & \ \ overlap(I_1,I_2) = 0\} \\
number(X) \wedge date(X) \rightarrow \bot & : \{ann(\mathrm{X},I_1,num) = 1 \wedge ann(\mathrm{X},I_1,date) = 1 \\
 & \ \ overlap(I_1,I_2) = 0\}
\end{array}
$$

Formulas with an empty annotation always hold.

Ranking Queries

- Ranking Query (RQ): what are the ground atoms inferred from a KB, in decreasing order of probability?
- Semantics: the probability that a ground atom a is true is equal to the sum of the probabilities of possible worlds where the resulting KB entails the CQ a.
- Recall that possible worlds are disjoint events.
- Unfortunately, computing probabilities of atoms is intractable:

  Theorem: Computing $Pr(a)$ w.r.t. a given probabilistic ontology is #P-hard in the data complexity.
- We now explore ways to tackle this uncertainty.

Conjunctive MLNs

- First, we propose a special class of MLNs:

  A conjunctive MLN (cMLN) is an MLN in which all formulas (F, w) in the set are such that F is a conjunction of atoms.

- This restriction allows us to define equivalence classes over the set of possible worlds w.r.t. M :
  - Informally, two worlds are equivalent iff they satisfy the same formulas in M.
  - Though there are still an exponential number of classes, there are some properties that we can leverage.
- Proposition 1: Given cMLN M, deciding if an equivalence class C is empty is in PTIME.

Conjunctive MLNs: Properties

- Proposition 2: Given cMLN M, and equivalence class C, all elements in C can be obtained in linear time w.r.t. the size of the output.
- Proposition 3: Given cMLN M, and worlds λ_1 and λ_2, we have that if $\lambda_1 \sim_M \lambda_2$ then $Pr(\lambda_1) = Pr(\lambda_2)$.
- Proposition 4: Given cMLN M, and worlds λ_1 and λ_2, deciding if $Pr(\lambda_1) \leq Pr(\lambda_2)$ is in PTIME.
- Computing exact probabilities in cMLNs, however, remains intractable:

  Theorem: Let a be an atom; deciding if $Pr(a) \geq k$ is PP-hard in the data complexity.

Summary

- Presented an extension of the Datalog+/- family of languages with probabilistic uncertainty.
- Uncertainty in rules is expressed by means of annotations that refer to an underlying Markov Logic Network.
- The goal is to develop a language and algorithms capable of managing uncertainty in a principled and scalable way.
- Scalability in our framework rests on two pillars:
  - We combine scalable rule-based approaches from the DB literature with annotations reflecting uncertainty;
  - Many possibilities for heuristic algorithms; MLNs are flexible, and sampling techniques may be leveraged.

- Also studying other kinds of probabilistic queries:
  - Threshold queries: what is the set of atoms that are inferred with probability at least p?
  - Conjunctive queries: what is the probability with which a conjunction of atoms is inferred?
- We are studying the tractability of all three kinds of queries under both sampling techniques.
- Also considering different kinds of restrictions on MLNs.

References

• T. Lukasiewicz, M. V. Martinez, G. Orsi, and G. I. Simari. Heuristic ranking in tightly coupled probabilistic description logics. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI 2012)*, pp. 554–563, 2012.

• G. Gottlob, T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Query answering under probabilistic uncertainty in Datalog+/– ontologies. *Annals of Mathematics and Artificial Intelligence*, 69(1):37–72, Sept. 2013.