Uncertainty Reasoning for the Semantic Web

Thomas Lukasiewicz

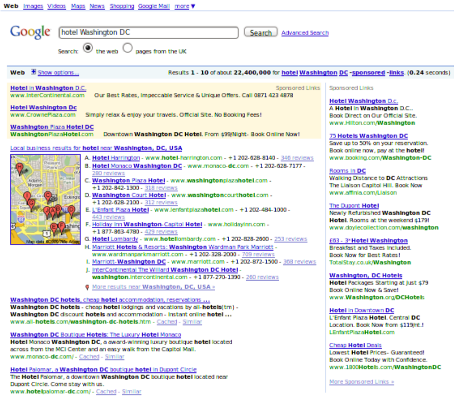
Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Outline

Outline

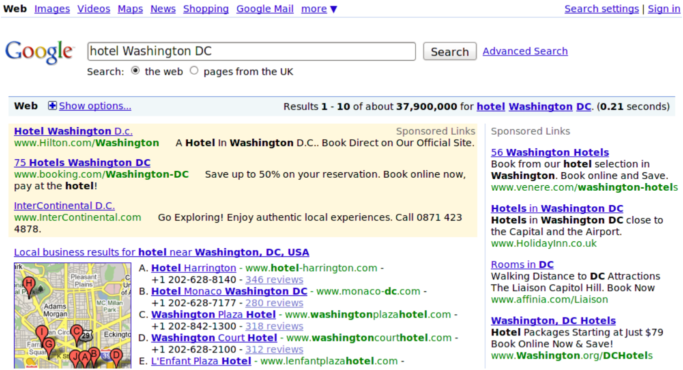
Web Search

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



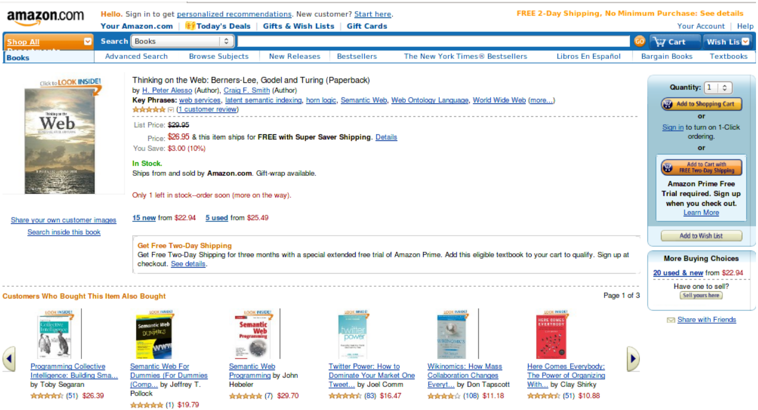
Computational Advertising

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

- Background knowledge
- 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
- ...

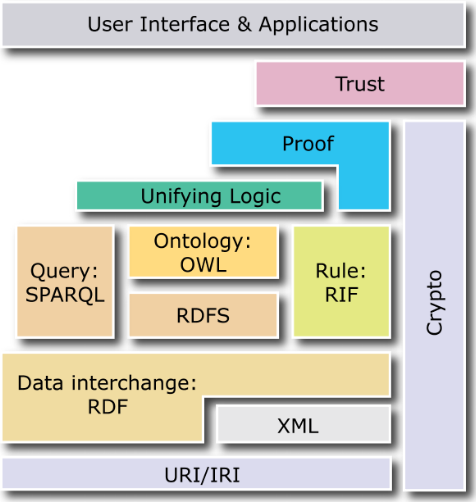
Outline

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), Semantic Web Rule Language (SWRL);
  - the Logic and Proof layers, which should offer other sophisticated representation and reasoning capabilities.

Semantic Web Stack



Challenges (from Wikipedia)

W Semantic Web - Wikipedia, the ...

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 at least 48 billion pages as of this writing (August 2, 2009). The SNOMED CT medical terminology ontology 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 which 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 which 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 which 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 ameliorate 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

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

- Uncertainty: statements are either true or false.

  But, due to lack of knowledge, we can only estimate to which probability/possibility/necessity degree, they are true or false.

  For example, one passes or one does not pass an exam. The probability/possibility/necessity degree that one passes is 0.83.

- Usually: possible world semantics with a probability/possibility distribution over possible worlds:

$$
\begin{aligned}
W &= \text{set of all classical interpretations } I, \\
&\mu\colon W \to [0,1],\quad \mu(I) \in [0,1] \\
Pr(\phi) &= \sum_{I \models \phi} \mu(I),\quad \sum_{I \in W} \mu(I) = 1 \\
Poss(\phi) &= \sup_{I \models \phi} \mu(I),\quad Poss(\bot) = 0,\quad Poss(\top) = 1 \\
Necc(\phi) &= \inf_{I \not\models \phi} \mu(I) = 1 - Poss(\neg\phi)
\end{aligned}
$$

Vagueness

- Vagueness: statements involve concepts with no exact definition, such as tall, small, close, far, cheap, and expensive.

  Statements are true to some degree, taken from a truth space.

  "John is tall to degree 0.83."
- Truth space: set of truth values L and a partial order $\leqslant$.
- Fuzzy logic: $L = [0, 1]$
- Many-valued interpretation: a function I that truth-compositionally maps formulas ϕ into L.
- Truth-compositionality: Truth of a complex formula determined by the truth of its parts and how those parts are combined.

Uncertainty and vagueness are important in the Semantic Web!

Many existing proposals for extensions of Semantic Web languages (RDF, OWL, DLs, rules, and DL rules) by uncertainty and vagueness.

In the following, some own such proposals:

- probabilistic DLs,
- probabilistic Datalog+/–,
- probabilistic ontological data exchange,
- probabilistic dl-programs,
- probabilistic fuzzy dl-programs.

Outline

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:

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

Deciding Model Existence / Satisfiability

Theorem: The probabilistic knowledge base $KB = (L, P)$ has a model Pr with $\Pr(\alpha) > 0$ iff the following system of linear constraints over the variables y_r $(r \in R)$, where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$, is solvable:

$$
\begin{aligned}
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} -l\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (1 - l)\, y_r &\geqslant 0 \quad (\forall (\psi|\phi)[l, u] \in P) \\
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} u\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (u - 1)\, y_r &\geqslant 0 \quad (\forall (\psi|\phi)[l, u] \in P) \\
\sum_{r \in R,\, r \models \alpha} y_r &= 1 \\
y_r &\geqslant 0 \quad (\text{for all } r \in R)
\end{aligned}
$$

Computing Tight Logical Consequences

Theorem: Suppose $KB = (L, P)$ has a model Pr such that $\Pr(\alpha) > 0$. Then, l (resp., u) such that $KB \models_{tight} (\beta|\alpha)[l, u]$ is given by the optimal value of the following linear program over the variables y_r ($r \in R$), where $R = \{I \in \mathcal{I}_\Phi \mid I \models L\}$:

$$\text{minimize (resp., maximize)} \sum_{r \in R,\, r \models \beta \wedge \alpha} y_r \quad \text{subject to}$$

$$\begin{aligned}
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} -l\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (1 - l)\, y_r &\geqslant 0 \quad (\forall (\psi|\phi)[l, u] \in P) \\
\sum_{r \in R,\, r \models \neg\psi \wedge \phi} u\, y_r + \sum_{r \in R,\, r \models \psi \wedge \phi} (u - 1)\, y_r &\geqslant 0 \quad (\forall (\psi|\phi)[l, u] \in P) \\
\sum_{r \in R,\, r \models \alpha} y_r &= 1 \\
y_r &\geqslant 0 \quad (\text{for all } r \in R)
\end{aligned}$$

Literature

- G. Boole. *An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probabilities*. Walton and Maberley, London, 1854.
- N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28:71–88, 1986.
- D. Dubois, H. Prade, and J.-M. Touscas. Inference with imprecise numerical quantifiers. In *Intelligent Systems*, 1990.
- R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87:78–128, 1990.
- A. M. Frisch and P. Haddawy. Anytime deduction for probabilistic logic. *Artif. Intell.*, 69:93–122, 1994.
- T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. *JAIR*, 10:199–241, 1999.
- T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM TOCL* 2(3):289–339, 2001.

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. Maximum Entropy 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 maximum entropy entailment:

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

Entailment under Maximum Entropy

- Entropy of a probabilistic interpretation Pr, denoted $H(\Pr)$:

$$H(\Pr) \quad = \quad -\sum_{I \in \mathcal{I}_\Phi} \Pr(I) \cdot \log \Pr(I)\,.$$

- The ME model of a satisfiable probabilistic knowledge base *KB* is the unique probabilistic interpretation Pr that is a model of *KB* and that has the greatest entropy among all the models of *KB*.
- $KB \,\|\!\!\!\sim^{me} (\psi|\phi)[l,u]$: $(\psi|\phi)[l,u]$ is a ME consequence of *KB* iff the ME model of *KB* is also a model of $(\psi|\phi)[l,u]$.
- $KB \,\|\!\!\!\sim^{me}_{tight} (\psi|\phi)[l,u]$: $(\psi|\phi)[l,u]$ is a tight ME consequence of *KB* iff for the ME model Pr of *KB*, it holds either (a) $\Pr(\phi) = 0$, $l = 1$, and $u = 0$, or (b) $\Pr(\phi) > 0$ and $\Pr(\psi|\phi) = l = u$.

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:

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

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:

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

Lexicographic Entailment

- Pr verifies $(\psi|\phi)[l,u]$ iff $\Pr(\phi) = 1$ and $\Pr \models (\psi|\phi)[l,u]$.
- P tolerates $(\psi|\phi)[l,u]$ under L iff $L \cup P$ has a model that verifies $(\psi|\phi)[l,u]$.
- $KB = (L,P)$ is consistent iff there exists an ordered partition $(P_0,\ldots,P_k)$ of P such that each P_i is the set of all $C \in P \setminus \bigcup_{j=0}^{i-1} P_j$ tolerated under L by $P \setminus \bigcup_{j=0}^{i-1} P_j$.
- This (unique) partition is called the z-partition of KB.

Let $KB = (L, P)$ be consistent, and $(P_0, \ldots, P_k)$ be its z-partition.

- Pr is *lex*-preferable to Pr′ iff some $i \in \{0, \ldots, k\}$ exists such that
  - $|\{C \in P_i \mid \mathrm{Pr} \models C\}| > |\{C \in P_i \mid \mathrm{Pr}' \models C\}|$ and
  - $|\{C \in P_j \mid \mathrm{Pr} \models C\}| = |\{C \in P_j \mid \mathrm{Pr}' \models C\}|$ for all $0 \leqslant j < i$.
- A model Pr of $\mathcal{F}$ is a *lex*-minimal model of $\mathcal{F}$ iff no model of $\mathcal{F}$ is *lex*-preferable to Pr.
- $KB \Vdash\!\sim^{lex} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a *lex*-consequence of KB iff every *lex*-minimal model Pr of L with $\mathrm{Pr}(\phi) = 1$ satisfies $(\psi|\phi)[l, u]$.
- $KB \Vdash\!\sim^{lex}_{tight} (\psi|\phi)[l, u]$: $(\psi|\phi)[l, u]$ is a tight *lex*-consequence of KB iff l (resp., u) is the infimum (resp., supremum) of $\mathrm{Pr}(\psi)$ subject to all *lex*-minimal models Pr of L with $\mathrm{Pr}(\phi) = 1$.

Literature

- J. B. Paris. *The Uncertain Reasoner's Companion: A Mathematical Perspective*. Cambridge University Press, 1995.
- G. Kern-Isberner and T. Lukasiewicz. Combining probabilistic logic programming with the power of maximum entropy. *Artif. Intell.*, 157(1–2):139–202, 2004.
- T. Lukasiewicz. Probabilistic Default Reasoning with Conditional Constraints. *Ann. Math. Artif. Intell.*, 34(1–3):35–88, 2002.
- J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.

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

Example

Standard terminological and assertional knowledge:

$$
\begin{aligned}
&(1)\ \mathit{MalePacemakerPatient} \sqsubseteq \mathit{PacemakerPatient},\\
&\quad\ \ \mathit{FemalePacemakerPatient} \sqsubseteq \mathit{PacemakerPatient},\\
&(2)\ \mathit{MalePacemakerPatient} \sqsubseteq \neg \mathit{FemalePacemakerPatient},\\
&(3)\ \mathit{PacemakerPatient} \sqsubseteq \mathit{HeartPatient},\\
&(4)\ \exists\, \mathit{HasIllnessSymptom}.\top \sqsubseteq \mathit{HeartPatient},\\
&\quad\ \ \exists\, \mathit{HasIllnessSymptom}^{-}.\top \sqsubseteq \mathit{IllnessSymptom},\\
&(5)\ \mathit{HeartPatient}(\mathit{Tom}),\\
&(6)\ \mathit{MalePacemakerPatient}(\mathit{John}),\\
&(7)\ \mathit{FemalePacemakerPatient}(\mathit{Maria}),\\
&(8)\ \mathit{HasIllnessSymptom}(\mathit{John}, \mathit{Arrhythmia}),\\
&\quad\ \ \mathit{HasIllnessSymptom}(\mathit{John}, \mathit{ChestPain}),\\
&\quad\ \ \mathit{HasIllnessSymptom}(\mathit{John}, \mathit{BreathingDifficulties}),\\
&\quad\ \ \mathit{HasIllnessStatus}(\mathit{John}, \mathit{Advanced}).
\end{aligned}
$$

Example

Default and probabilistic terminological knowledge:

$$
\begin{aligned}
&(9)\quad (\mathit{HighBloodPressure} \mid \mathit{HeartPatient})[1, 1],\\
&(10)\ (\neg\mathit{HighBloodPressure} \mid \mathit{PacemakerPatient})[1, 1],\\
&(11)\ (\mathit{MalePacemakerPatient} \mid \mathit{PacemakerPatient})[0.4, 1],\\
&(12)\ (\exists\, \mathit{HasHealthInsurance}.\mathit{PrivateHealthInsurance} \mid \mathit{HeartPatient})[0.9, 1],\\
&(13)\ (\exists\, \mathit{HasIllnessSymptom}.\{\mathit{Arrhythmia}\} \mid \mathit{PacemakerPatient})[0.98, 1],\\
&\qquad (\exists\, \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \mid \mathit{PacemakerPatient})[0.9, 1],\\
&\qquad (\exists\, \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \mid \mathit{PacemakerPatient})[0.6, 1].
\end{aligned}
$$

Example

Probabilistic assertional knowledge:

For individual Tom:

$$(14)\ (\mathit{PacemakerPatient} \mid \top)[0.8, 1].$$

For individual Maria:

$$(15)\ (\exists\, \mathit{HasIllnessSymptom}.\{\mathit{BreathingDifficulties}\} \mid \top)[0.6, 1],$$

$$(16)\ (\exists\, \mathit{HasIllnessSymptom}.\{\mathit{ChestPain}\} \mid \top)[0.9, 1],$$

$$(17)\ (\exists\, \mathit{HasIllnessStatus}.\{\mathit{Final}\} \mid \top)[0.2, 0.8].$$

Complexity Results

SAT: Satisfiability

PTCON: Probabilistic TBox consistency

PKBCON: Probabilistic knowledge base consistency

TLOGENT: Tight logical entailement

TLEXENT: Tight lexicographic entailment

	P-*DL-Lite*	P-$\mathcal{SHIF}$(**D**)	P-$\mathcal{SHOIN}$(**D**)
SAT	NP	EXP	NEXP
PTCON	NP	EXP	NEXP
PKBCON	NP	EXP	NEXP

	P-*DL-Lite*	P-$\mathcal{SHIF}$(**D**)	P-$\mathcal{SHOIN}$(**D**)
TLOGENT	$\mathrm{FP}^{\mathrm{NP}}$	FEXP	in $\mathrm{FP}^{\mathrm{NEXP}}$
TLEXENT	$\mathrm{FP}^{\mathrm{NP}}$	FEXP	in $\mathrm{FP}^{\mathrm{NEXP}}$

References

- T. Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6/7):852-883, 2008.

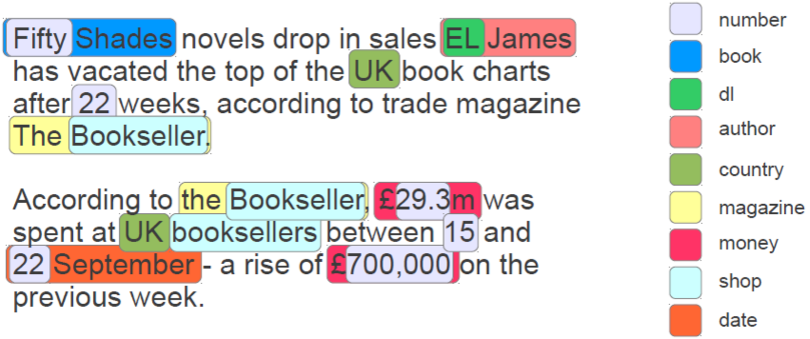
Outline

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 $\mathit{hasFirstAuthor}$).

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: $\exists P\ \mathit{directs}(M, P) \leftarrow \mathit{manager}(M)$;

- negative constraints:

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

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

- 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:

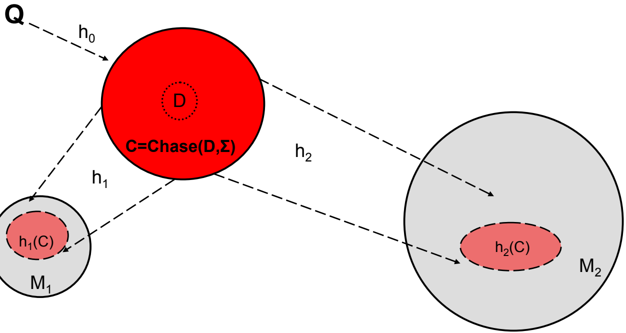
- D: database over $\text{dom}(D)$.
- Σ: set of TGDs and/or EGDs

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 $\text{dom}(D)$ 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 $\Sigma = \{p(x) \to \exists y\, q(y);\ p(x) \to q(x)\}$:

  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 $\Sigma = \{p(x, y) \to \exists z\, p(y, z)\}$:

  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.

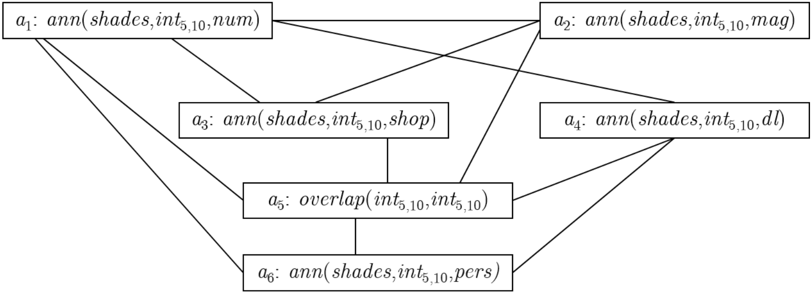
- A particularly costly step is the computation of Z, but this is a one-time calculation.

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:

$$book(X) \rightarrow editorialProd(X) \quad : \{\}$$

$$magazine(X) \rightarrow editorialProd(X) \quad : \{\}$$

$$author(X) \rightarrow person(X,P) \quad : \{\}$$

$$descLogic(X) \wedge author(X) \rightarrow \bot \quad : \{ann(\mathrm{X},I_1,dl) = 1 \wedge ann(\mathrm{X},I_2,pers) = 1 \; overlap(I_1,I_2) = 0\}$$

$$shop(X) \wedge editorialProd(X) \rightarrow \bot \quad : \{ann(\mathrm{X},I_1,shop) = 1 \wedge ann(\mathrm{X},I_2,mag) = 1 \; overlap(I_1,I_2) = 0\}$$

$$number(X) \wedge date(X) \rightarrow \bot \quad : \{ann(\mathrm{X},I_1,num) = 1 \wedge ann(\mathrm{X},I_1,date) = 1 \; overlap(I_1,I_2) = 0\}$$

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.

- In cMLNs, the worlds can be enumerated with decreasing probabilities.
- 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?
- Studied the tractability of all three kinds of queries under Monte Carlo sampling and top-down enumeration.
- Also considering tractable MLNs (TPMs), such as Tractable Markov Logic from the literature.

Summary of approximation and special-case algorithms:

Problem	**Monte Carlo Sampling**	**Top-down Enumeration**
Ranking	General MLNs: Tractable, but no sound/complete guarantees TPM KBs: Bounded error and partial rankings can be guaranteed	cMLNs: Error is bounded and partial rankings guaranteed TPM KBs: Bounded error and partial rankings can be guaranteed
Threshold	General MLNs: *#P-Hard* TPM KBs: Sound, complete under certain conditions	cMLNs: Sound and complete under certain conditions TPM KBs: Sound and complete under certain conditions
CQs	General MLNs: *#P-Hard* TPM KBs: Sound	cMLNs: *#P-Hard* TPM KBs: Tightest possible interval is guaranteed

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.

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.

Outline

Motivation

Probabilistic ontological data exchange

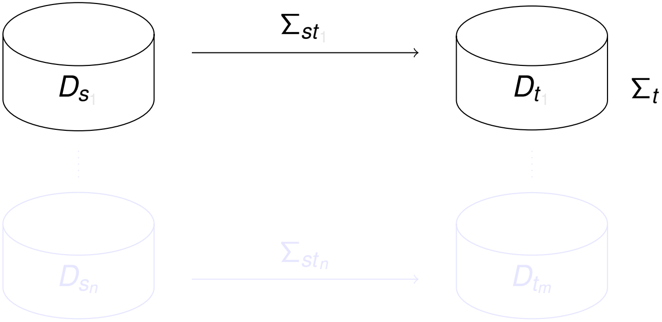
- Ontological data exchange for integrated query answering over distributed ontologies on the Semantic Web.
- Ontological data exchange extending distributed ontology-based data access (OBDA).

Probabilities

- Automatically gathered and processed data (e.g., via information extraction, financial risk assessment)
  $\Rightarrow$ probabilistic databases
- Uncertainty about the proper correspondence between items in distributed databases and ontologies (e.g., due to automatic generation)
  $\Rightarrow$ probabilistic mappings

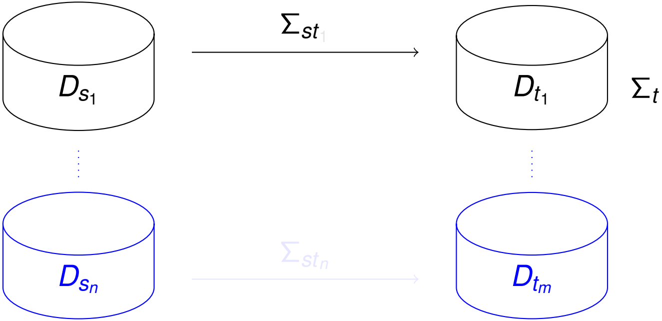
Overview

Probabilistic data exchange:



$\Sigma_{st} \cup \Sigma_t$: TGDs from WA

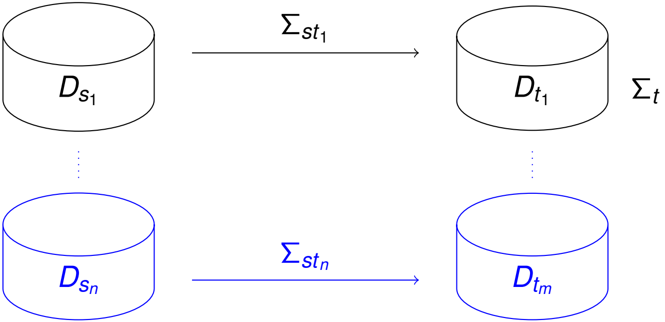
Probabilistic data exchange:



$\Sigma_{st} \cup \Sigma_t$: TGDs from WA

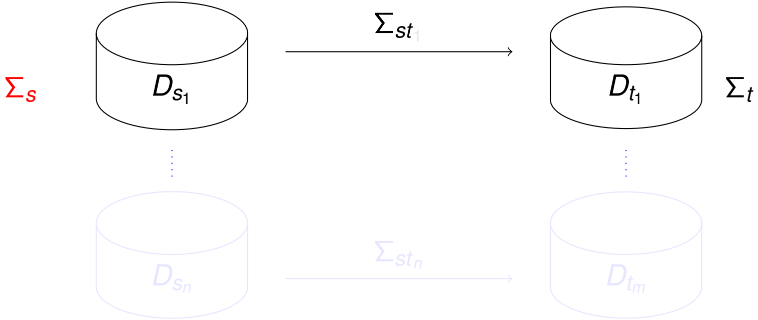
Overview

Probabilistic data exchange:



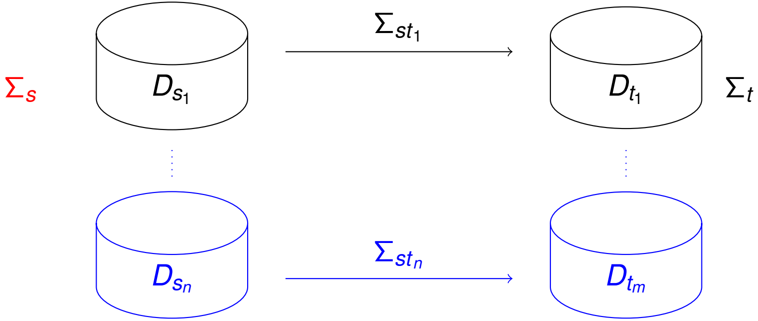
$\Sigma_{st} \cup \Sigma_t$: TGDs from WA

Probabilistic ontological data exchange: (PODE)



$\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$:
NCs and TGDs from WA, A, G, WG, S, WS, L, F, LF, AF, SF, GF

Probabilistic ontological data exchange: (PODE)



$\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$:
NCs and TGDs from WA, A, G, WG, S, WS, L, F, LF, AF, SF, GF

Probabilistic Databases

Probabilistic databases/instances:

- A probabilistic database (resp., probabilistic instance) is a probability space $Pr = (\mathcal{I}, \mu)$ such that $\mathcal{I}$ is the set of all databases (resp., instances) over a schema **S**, and $\mu\colon \mathcal{I} \to [0, 1]$ is a function that satisfies $\sum_{I \in \mathcal{I}} \mu(I) = 1$.

Example:

	Possible database facts
r_a	*Researcher*(Alice, UniversityOfOxford)
r_p	*Researcher*(Paul, UniversityOfOxford)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Probabilistic database $Pr = (\mathcal{I}, \mu)$	
$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}\}$	0.075

Compact Encoding of Probabilistic Databases

Annotations and annotated atoms:

- Elementary events e_i: $e_1, \dots, e_n$ with $n \geqslant 1$.
- World w: conjunctions $\ell_1 \wedge \dots \wedge \ell_n$ of literals $\ell_i \in \{e_i, \neg e_i\}$.
- Annotations λ: Boolean combinations of elementary events:
  - each e_i is an annotation λ;
  - if λ_1 and λ_2 are annotations, then also $\neg\lambda_1$ and $\lambda_1 \wedge \lambda_2$.
- Annotated atoms $a : \lambda$: atoms a and annotations λ.

Uncertainty model:

- Bayesian network over n binary random variables $E_1, \dots, E_n$ with the domains $dom(E_i) = \{e_i, \neg e_i\}$.

Compact Encoding of Probabilistic Databases

A set **A** of annotated atoms $\{a_1 : \lambda_1, \ldots, a_l : \lambda_l\}$ along with a Bayesian network B compactly encodes a probabilistic database $Pr = (\mathcal{I}, \mu)$:

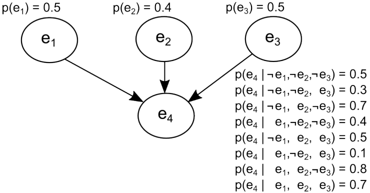
1. probability $\mu(\lambda)$, for every annotation λ: sum of the probabilities of all worlds in B in which λ is true;
2. probability $\mu(D)$, for every database $D = \{a_1, \ldots, a_m\} \in \mathcal{I}$: probability of the conjunction $\lambda = \lambda_1 \wedge \cdots \wedge \lambda_m$ of the annotations of its atoms. (Note that D is maximal with λ.)

Compact Encoding of Probabilistic Databases

Example:

Possible database facts and their encoding

r_a	*Researcher*(Alice, UniversityOfOxford)	true
r_p	*Researcher*(Paul, UniversityOfOxford)	$e_1 \vee e_2 \vee e_3 \vee e_4$
p_{aml}	*Publication*(Alice, ML, JMLR)	$e_1 \vee e_2$
p_{adb}	*Publication*(Alice, DB, TODS)	$\neg e_1 \wedge \neg e_2$
p_{pdb}	*Publication*(Paul, DB, TODS)	$e_1 \vee (\neg e_2 \wedge \neg e_3 \wedge e_4)$
p_{pai}	*Publication*(Paul, AI, AIJ)	$(\neg e_1 \wedge e_2) \vee (\neg e_1 \wedge e_3)$



Probabilistic database $Pr = (\mathcal{I}, \mu)$	
$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}\}$	0.075

Ontological Data Exchange (Syntax)

Ontological data exchange (ODE) problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$:

- source schema **S**,
- target schema **T**, disjoint from **S**,
- source ontology Σ_s: finite set of TGDs and NCs over **S**,
- target ontology Σ_t: finite set of TGDs and NCs over **T**,
- (source-to-target) mapping Σ_{st}: finite set of TGDs and NCs over $\mathbf{S} \cup \mathbf{T}$ with *body*(σ) and *head*(σ) over $\mathbf{S} \cup \mathbf{T}$ and **T**, resp..

Probabilistic ODE (PODE) problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$:

- probabilistic (source-to-target) mapping μ_{st}: function $\mu_{st} : 2^{\Sigma_{st}} \to [0, 1]$ such that $\sum_{\Sigma' \subseteq \Sigma_{st}} \mu_{st}(\Sigma') = 1$.

Ontological Data Exchange (Semantics)

- J is a solution (resp., universal solution) of I w.r.t. Σ: $I \in \text{ins}(\mathbf{S})$, $J \in \text{inst}(\mathbf{T})$, and (I, J) is a model (resp., universal model) of $\Sigma = \Sigma_s \cup \Sigma_t \cup \Sigma_{st}$
- $Sol_{\mathcal{M}}$ (resp., $USol_{\mathcal{M}}$): set of all pairs (I, J) with J being a solution (resp., universal solution) for I w.r.t. Σ
- A probabilistic target instance $Pr_t = (\mathcal{J}, \mu_t)$ is a probabilistic solution (resp., universal solution) for a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$ w.r.t. $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ iff there exists a probability space $Pr = (\mathcal{I} \times \mathcal{J}, \mu)$ such that:
  - The left and right marginals of Pr are Pr_s and Pr_t, resp.:
    - $\sum_{J \in \mathcal{J}}(\mu(I, J)) = \mu_s(I)$ for all $I \in \mathcal{I}$ and
    - $\sum_{I \in \mathcal{I}}(\mu(I, J)) = \mu_t(J)$ for all $J \in \mathcal{J}$;
  - $\mu(I, J) = 0$ for all $(I, J) \notin Sol_{\mathcal{M}}$ (resp., $(I, J) \notin USol_{\mathcal{M}}$).

Ontological Data Exchange (Example)

- σ_s : $\mathit{Publication}(X, Y, Z) \rightarrow \mathit{ResearchArea}(X, Y)$
- σ_{st} : $\mathit{ResearchArea}(N, T) \wedge \mathit{Researcher}(N, U) \rightarrow \exists D\ \mathit{UResearchArea}(U, D, T)$
- σ_t : $\mathit{UResearchArea}(U, D, T) \rightarrow \exists Z\ \mathit{Lecturer}(T, Z)$

Possible source database facts

r_a	*Researcher*(Alice, UoO)
r_p	*Researcher*(Paul, UoO)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Possible target instance facts

u_{ml}	*UResearchArea*(UoO, N_1, ML)
u_{ai}	*UResearchArea*(UoO, N_2, AI)
u_{db}	*UResearchArea*(UoO, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic source instance $Pr_S = (\mathcal{I}, \mu_S)$

$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, ra_{aml}, ra_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, ra_{aml}, ra_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, ra_{adb}, ra_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, ra_{adb}, ra_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}, ra_{adb}\}$	0.075

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$

$J_1 = \{u_{ml}, u_{db},$	0.5
$J_2 = \{u_{ml}, u_{ai},$	0.2
$J_3 = \{u_{ai}, u_{db},$	0.15
$J_4 = \{u_{db},$	0.15

Ontological Data Exchange (Example)

- σ_s : $\mathit{Publication}(X, Y, Z) \rightarrow \mathit{ResearchArea}(X, Y)$
- σ_{st} : $\mathit{ResearchArea}(N, T) \wedge \mathit{Researcher}(N, U) \rightarrow \exists D\ \mathit{UResearchArea}(U, D, T)$
- σ_t : $\mathit{UResearchArea}(U, D, T) \rightarrow \exists Z\ \mathit{Lecturer}(T, Z)$

Possible source database facts

r_a	*Researcher*(Alice, UoO)
r_p	*Researcher*(Paul, UoO)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Probabilistic source instance $Pr_S = (\mathcal{I}, \mu_S)$

$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, ra_{aml}, ra_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, ra_{aml}, ra_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, ra_{adb}, ra_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, ra_{adb}, ra_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}, ra_{adb}\}$	0.075

Possible target instance facts

u_{ml}	*UResearchArea*(UoO, N_1, ML)
u_{ai}	*UResearchArea*(UoO, N_2, AI)
u_{db}	*UResearchArea*(UoO, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$

$J_1 = \{u_{ml}, u_{db},$	0.5
$J_2 = \{u_{ml}, u_{ai},$	0.2
$J_3 = \{u_{ai}, u_{db},$	0.15
$J_4 = \{u_{db},$	0.15

Ontological Data Exchange (Example)

- σ_s : $\textit{Publication}(X, Y, Z) \rightarrow \textit{ResearchArea}(X, Y)$

- σ_{st} : $\textit{ResearchArea}(N, T) \wedge \textit{Researcher}(N, U) \rightarrow$
  $\exists D\ \textit{UResearchArea}(U, D, T)$

- σ_t : $\textit{UResearchArea}(U, D, T) \rightarrow \exists Z\ \textit{Lecturer}(T, Z)$

Possible source database facts

r_a	*Researcher*(Alice, UoO)
r_p	*Researcher*(Paul, UoO)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Probabilistic source instance $Pr_S = (\mathcal{I}, \mu_s)$

$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, ra_{aml}, ra_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, ra_{aml}, ra_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, ra_{adb}, ra_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, ra_{adb}, ra_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}, ra_{adb}\}$	0.075

Possible target instance facts

u_{ml}	*UResearchArea*(UoO, N_1, ML)
u_{ai}	*UResearchArea*(UoO, N_2, AI)
u_{db}	*UResearchArea*(UoO, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$

$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.5
$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.15
$J_4 = \{u_{db}, l_{db}\}$	0.15

Ontological Data Exchange (Example)

- σ_s : $\mathit{Publication}(X, Y, Z) \rightarrow \mathit{ResearchArea}(X, Y)$
- σ_{st} : $\mathit{ResearchArea}(N, T) \wedge \mathit{Researcher}(N, U) \rightarrow \exists D\ \mathit{UResearchArea}(U, D, T)$
- σ_t : $\mathit{UResearchArea}(U, D, T) \rightarrow \exists Z\ \mathit{Lecturer}(T, Z)$

Possible source database facts

r_a	*Researcher*(Alice, UoO)
r_p	*Researcher*(Paul, UoO)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Probabilistic source instance $Pr_S = (\mathcal{I}, \mu_s)$

$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, ra_{aml}, ra_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, ra_{aml}, ra_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, ra_{adb}, ra_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, ra_{adb}, ra_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}, ra_{adb}\}$	0.075

Possible target instance facts

u_{ml}	*UResearchArea*(UoO, N_1, ML)
u_{ai}	*UResearchArea*(UoO, N_2, AI)
u_{db}	*UResearchArea*(UoO, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$

$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.5
$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.15
$J_4 = \{u_{db}, l_{db}\}$	0.15

Ontological Data Exchange (Example)

- σ_s : $Publication(X, Y, Z) \rightarrow ResearchArea(X, Y)$
- σ_{st} : $ResearchArea(N, T) \wedge Researcher(N, U) \rightarrow \exists D\ UResearchArea(U, D, T)$
- σ_t : $UResearchArea(U, D, T) \rightarrow \exists Z\ Lecturer(T, Z)$

Possible source database facts

r_a	*Researcher*(Alice, UoO)
r_p	*Researcher*(Paul, UoO)
p_{aml}	*Publication*(Alice, ML, JMLR)
p_{adb}	*Publication*(Alice, DB, TODS)
p_{pdb}	*Publication*(Paul, DB, TODS)
p_{pai}	*Publication*(Paul, AI, AIJ)

Probabilistic source instance $Pr_S = (\mathcal{I}, \mu_S)$

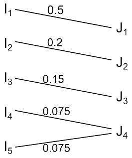
$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, ra_{aml}, ra_{pdb}\}$	0.5
$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, ra_{aml}, ra_{pai}\}$	0.2
$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, ra_{adb}, ra_{pai}\}$	0.15
$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, ra_{adb}, ra_{pdb}\}$	0.075
$I_5 = \{r_a, p_{adb}, ra_{adb}\}$	0.075

Possible target instance facts

u_{ml}	*UResearchArea*(UoO, N_1, ML)
u_{ai}	*UResearchArea*(UoO, N_2, AI)
u_{db}	*UResearchArea*(UoO, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$

$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.5
$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.15
$J_4 = \{u_{db}, l_{db}\}$	0.15



UCQs

Given:

- ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$;
- probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$;
- UCQ $q(\mathbf{X}) = \bigvee_{i=1}^{k} \exists \mathbf{Y}_i \, \Phi_i(\mathbf{X}, \mathbf{Y}_i,)$ over target schema.

Then, confidence of a tuple:

- $Pr_t(q(\mathbf{t}))$ for $Pr_t = (\mathcal{J}, \mu_t)$: sum of all $\mu_t(J)$ such that $q(\mathbf{t})$ evaluates to true in the instance $J \in \mathcal{J}$;
- $conf_q(\mathbf{t})$: confidence of a tuple $\mathbf{t}$ for q in Pr_s relative to $\mathcal{M}$: infimum of $Pr_t(q(\mathbf{t}))$ subject to all probabilistic solutions Pr_t for Pr_s relative to $\mathcal{M}$.

UCQs (Example)

	Possible target instance facts
u_{ml}	*UResearchArea*(University of Oxford, N_1, ML)
u_{ai}	*UResearchArea*(University of Oxford, N_2, AI)
u_{db}	*UResearchArea*(University of Oxford, N_3, DB)
l_{ml}	*Lecturer*(ML, N_4)
l_{ai}	*Lecturer*(AI, N_5)
l_{db}	*Lecturer*(DB, N_6)

Probabilistic universal solution $Pr_t = (\mathcal{J}, \mu_t)$	
$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.5
$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.15
$J_4 = \{u_{db}, l_{db}\}$	0.15

$$Pr = \{(I_1, J_1), .5), ((I_2, J_2), .2), ((I_3, J_3), .15), ((I_4, J_4), .075), ((I_5, J_4), .075)\}$$

A student wants to know whether she can study both machine learning and databases at the University of Oxford:

$$\begin{aligned} q() = \exists \mathrm{X}, \mathrm{Y}(&\exists \mathrm{Z}(\mathit{Lecturer}(\mathit{AI}, \mathrm{X}) \wedge \mathit{UResearchArea}\,(\mathit{UnivOx}, \mathrm{Z}, \mathrm{AI})) \\ &\vee \exists \mathrm{Z}(\mathit{Lecturer}(\mathit{ML}, \mathrm{Y}) \wedge \mathit{UResearchArea}(\mathit{UnivOx}, \mathrm{Z}, \mathit{ML}))). \end{aligned}$$

Then, q yields the probability 0.85.

Computational Problems

Consistency:

- Given a (P)ODE problem $\mathcal{M}$ and a probabilistic source database Pr_s, decide whether there exists a (universal) probabilistic solution for Pr_s relative to $\mathcal{M}$.

Threshold UCQ answering:

- Given a (P)ODE problem $\mathcal{M}$, a probabilistic source database Pr_s, a UCQ $q(\mathbf{X})$, a tuple $\mathbf{t}$ of constants, and $\theta > 0$, decide whether $conf_Q(\mathbf{t}) \geqslant \theta$ in Pr_s w.r.t. $\mathcal{M}$.

Computational Problems

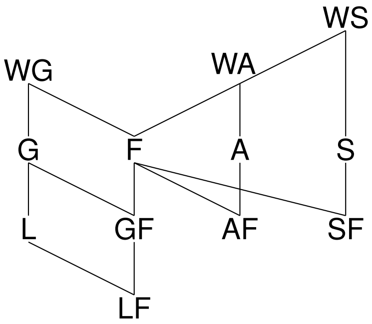
Classes of existential rules:

- linear full (LF), guarded full (GF), acyclic full (AF), sticky full (SF), full (F)
- acyclic (A), weakly acyclic (WA)
- linear (L), guarded (G), weakly guarded (WG)
- sticky (S), weakly sticky (WS)

Types of complexity:

- data complexity,
- fixed-program combined (fp-combined) complexity,
- bounded-arity combined (ba-combined) complexity,
- combined complexity

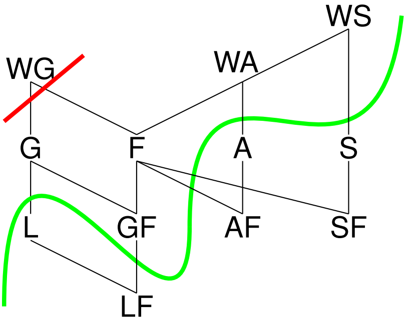
Relationships between Classes of Existential Rules



Complexity Results: Data Complexity

Data complexity of standard BCQ answering

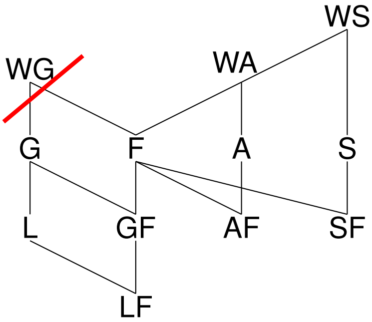
	BCQs
L, LF, AF	in AC^0
G	P
WG	EXP
S, SF	in AC^0
F, GF	P
A	in AC^0
WS, WA	P



Complexity Results: fp-Combined Complexity

fp-combined complexity of standard BCQ answering

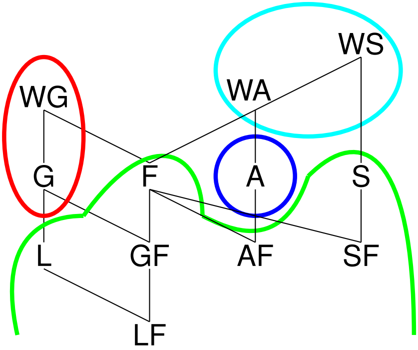
	BCQs
L, LF, AF	NP
G	NP
WG	EXP
S, SF	NP
F, GF	NP
A	NP
WS, WA	NP



Complexity Results: ba-Combined Complexity

ba-combined complexity of standard BCQ answering

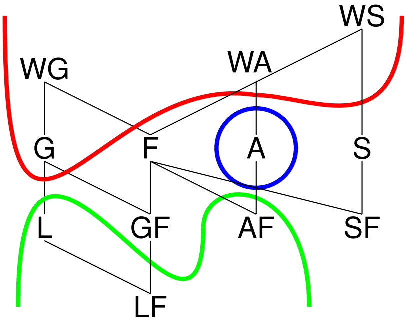
	BCQs
L, LF, AF	NP
G	EXP
WG	EXP
S, SF	NP
F, GF	NP
A	NEXP
WS, WA	2EXP



Complexity Results: Combined Complexity

combined complexity of standard BCQ answering

	BCQs
L, LF, AF	PSPACE
G	2EXP
WG	2EXP
S, SF	EXP
F, GF	EXP
A	NEXP
WS, WA	2EXP



Summary of Complexity Results (Consistency)

Complexity of deciding the existence of a (universal) probabilistic solution (for both ODE and PODE problems):

	Data	***fp*-comb.**	***ba*-comb.**	**Comb.**
L, LF, AF	coNP	coNP	coNP	PSPACE
G	coNP	coNP	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	coNP	coNP	coNP	EXP
F, GF	coNP	coNP	coNP	EXP
A	coNP	coNP	coNEXP	coNEXP
WS, WA	coNP	coNP	2EXP	2EXP

All entries are completeness results; hardness holds even when any two variables are independent from each other.

Complexity of deciding threshold query entailment (for both ODE and PODE problems; annotations are Boolean events under Bayesian networks).

	Data	***fp*-comb.**	***ba*-comb.**	**Comb.**
L, LF, AF	PP	PP^{NP}	PP^{NP}	PSPACE
G	PP	PP^{NP}	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	PP	PP^{NP}	PP^{NP}	EXP
F, GF	PP	PP^{NP}	PP^{NP}	EXP
A	PP	PP^{NP}	NEXP	NEXP
WS, WA	PP	PP^{NP}	2EXP	2EXP

All entries are completeness results; hardness holds even when any two variables are independent from each other.

Inconsistency-Tolerant Threshold UCQ Entailment

Repairing errors in probabilistic databases/instances; existential rules have no errors.

- repair of a deterministic database D relative to Σ: maximal subset of D that is consistent relative to Σ.
- repair of a probabilistic database $(\mathcal{I}, \mu)$ relative to Σ: consists of a repair of each $I \in \mathcal{I}$ with its probability $\mu(I)$
- $conf_q(\mathbf{t})$: confidence of a tuple $\mathbf{t}$ for q in Pr_s relative to $\mathcal{M}$: infimum of $Pr_t(q(\mathbf{t}))$ subject to all repairs of probabilistic solutions Pr_t for Pr_s relative to $\mathcal{M}$.

Complexity of deciding inconsistency-tolerant threshold query entailment (for both ODE and PODE problems; annotations are Boolean events under Bayesian networks).

	Data	***fp*-comb.**	***ba*-comb.**	**Comb.**
$L_\bot$, $LF_\bot$, $AF_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	$PP^{\Sigma_2^p}$	PSPACE
$G_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	EXP	2EXP
$WG_\bot$	EXP	EXP	EXP	2EXP
$S_\bot$, $SF_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	$PP^{\Sigma_2^p}$	EXP
$F_\bot$, $GF_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	$PP^{\Sigma_2^p}$	EXP
$A_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	in PP^{NEXP}	in PP^{NEXP}
$WS_\bot$, $WA_\bot$	PP^{NP}	$PP^{\Sigma_2^p}$	2EXP	2EXP

All entries but the “in” ones are completeness results; hardness holds even when any two variables are independent from each other.

Summary

- ontological data exchange with probabilistic data
- ontological data exchange with probabilistic mappings
- compact encoding of probabilities via Boolean annotations under Bayesian networks as uncertainty models
- for the main classes of existential rules: data, fp-combined, ba-combined, and combined complexity for:
  - consistency
  - UCQ threshold entailment
  - inconsistency-tolerant UCQ threshold entailment

References

- T. Lukasiewicz, M. V. Martinez, L. Predoiu, G. I. Simari. Existential rules and Bayesian networks for probabilistic ontological data exchange. In *Proc. RuleML*, 2015.
- T. Lukasiewicz, L. Predoiu. Complexity of threshold query answering in probabilistic ontological data exchange. In *Proc. ECAI*, 2016.
- İ.İ. Ceylan, T. Lukasiewicz, R. Peñaloza. Complexity results for probabilistic Datalog+/–. In *Proc. ECAI*, 2016.

Outline

Overview

One of the major challenges of the Semantic Web: aligning heterogeneous ontologies via semantic mappings.

Mappings are automatically produced by matching systems.

Automatically created mappings often contain uncertain hypotheses and errors:

- mapping hypotheses are often oversimplifying;
- there may be conflicts between different hypotheses for semantic relations;
- semantic relations are only given with a degree of confidence in their correctness.

In the following, I survey a logic-based language (close to semantic web languages) for representing, combining, and reasoning about such ontology mappings.

Basics

- Ontologies are encoded in L (here: OWL DL or OWL Lite).
- $Q(O)$ denotes the matchable elements of the ontology O.
- Matching: Given two ontologies O and O', determine correspondences between $Q(O)$ and $Q(O')$.
- Correspondences are 5-tuples (id, e, e', r, n) such that
  - id is a unique identifier;
  - $e \in Q(O)$ and $e' \in Q(O')$;
  - $r \in R$ is a semantic relation (here: implication);
  - n is a degree of confidence in the correctness.

Representation Requirements

- Tight integration of mapping and ontology language
- Support for mappings refinement
- Support for repairing inconsistencies
- Representation and combination of confidence
- Decidability and efficiency of instance reasoning

Description Logics

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).

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) $\mathit{provides}(\mathit{ibm}, \mathit{pc\_ibm})$; $\mathit{provides}(\mathit{hp}, \mathit{pc\_hp})$.

Disjunctive Programs

Disjunctive program P for an online store:

(1) $pc(pc_1)$; $pc(pc_2)$; $pc(obj_3) \vee laptop(obj_3)$;
(2) $brand\_new(pc_1)$; $brand\_new(obj_3)$;
(3) $vendor(dell, pc_1)$; $vendor(dell, pc_2)$;
(4) $avoid(X) \leftarrow camera(X), not\ sale(X)$;
(5) $sale(X) \leftarrow electronics(X), not\ brand\_new(X)$;
(6) $provider(V) \leftarrow vendor(V, X), product(X)$;
(7) $provider(V) \leftarrow provides(V, X), product(X)$;
(8) $similar(X, Y) \leftarrow related(X, Y)$;
(9) $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z)$;
(10) $similar(X, Y) \leftarrow similar(Y, X)$;
(11) $brand\_new(X) \vee high\_quality(X) \leftarrow expensive(X)$.

Syntax

- Sets $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, $\mathbf{I}$, and $\mathbf{V}$ of atomic concepts, abstract roles, datatype roles, individuals, and data values, respectively.
- Finite sets Φ_p and Φ_c of constant and predicate symbols with: (i) Φ_p not necessarily disjoint to $\mathbf{A}$, $\mathbf{R}_A$, and $\mathbf{R}_D$, and (ii) $\Phi_c \subseteq \mathbf{I} \cup \mathbf{V}$.
- A tightly integrated disjunctive dl-program $KB = (L, P)$ consists of a description logic knowledge base L and a disjunctive program P.

Semantics

- An interpretation I is any subset of the Herbrand base HB_Φ.
- I is a model of P is defined as usual.
- I is a model of L iff $L \cup I \cup \{\neg a \mid a \in HB_\Phi - I\}$ is satisfiable.
- I is a model of KB iff I is a model of both L and P.
- The Gelfond-Lifschitz reduct of a disjunctive program P relative to $I \subseteq HB_\Phi$, denoted P^I, is the ground positive disjunctive program obtained from $ground(P)$ by (i) deleting every rule r s.t. $B^-(r) \cap I \neq \emptyset$, and (ii) deleting the negative body from each remaining rule.
- The Gelfond-Lifschitz reduct of $KB = (L, P)$ w.r.t. $I \subseteq HB_\Phi$, denoted KB^I, is defined as the disjunctive dl-program (L, P^I), where P^I is the standard Gelfond-Lifschitz reduct of P w.r.t. I.
- $I \subseteq HB_\Phi$ is an answer set of KB iff I is a minimal model of KB^I.
- KB is consistent iff it has an answer set.
- A ground atom $a \in HB_\Phi$ is a cautious (resp., brave) consequence of a disjunctive dl-program KB under the answer set semantics iff every (resp., some) answer set of KB satisfies a.

Examples

A disjunctive dl-program $KB = (L, P)$ is given by the above description logic knowledge base L and disjunctive program P.

Another disjunctive dl-program $KB' = (L', P')$ is obtained from KB by adding to L the axiom $\geqslant 1\, \mathit{similar} \sqcup \geqslant 1\, \mathit{similar}^{-} \sqsubseteq \mathit{Product}$, which expresses that only products are similar:

The predicate symbol *similar* in P' is also a role in L', and it freely occurs in both rule bodies and rule heads in P'.

Properties

Every answer set of a disjunctive program *KB* is also a minimal model of *KB*, and the converse holds when *KB* is positive.

The answer set semantics of disjunctive dl-programs faithfully extends its ordinary counterpart and the first-order semantics of description logic knowledge bases.

The tight integration of ontologies and rules semantically behaves very differently from the loose integration: $KB = (L, P)$, where

$$\begin{aligned} L &= \{person(a),\ person \sqsubseteq male \sqcup female\} \text{ and} \\ P &= \{client(X) \leftarrow male(X),\ client(X) \leftarrow female(X)\}\,, \end{aligned}$$

implies $client(a)$, while $KB' = (L', P')$, where

$$\begin{aligned} L' &= \{person(a),\ person \sqsubseteq male \sqcup female\} \text{ and} \\ P' &= \{client(X) \leftarrow DL[male](X),\ client(X) \leftarrow DL[female](X)\}\,, \end{aligned}$$

does *not* imply $client(a)$.

Basics

Tightly integrated disjunctive dl-programs $KB = (L, P)$ can be used for representing (possibly inconsistent) mappings (without confidence values) between two ontologies.

Intuitively, L encodes the union of the two ontologies, while P encodes the mappings between the ontologies.

Here, disjunctions in rule heads and nonmonotonic negations in rule bodies in P can be used to resolve inconsistencies.

Example

The following two mappings have been created by the hmatch system for mapping the CRS Ontology (O_1) on the EKAW Ontology (O_2):

$$\mathit{EarlyRegisteredParticipant}(X) \leftarrow \mathit{Participant}(X)\,;$$
$$\mathit{LateRegisteredParticipant}(X) \leftarrow \mathit{Participant}(X)\,.$$

L is the union of two description logic knowledge bases L_1 and L_2 encoding the ontologies O_1 resp. O_2, while P encodes the mappings.

However, we cannot directly use the two mapping relationships as two rules in P, since this would introduce an inconsistency in KB.

Resolving Inconsistencies

By disjunctions in rule heads:

$$EarlyRegisteredParticipant(X) \vee LateRegisteredParticipant(X) \leftarrow Participant(X)\,.$$

By nonmonotonic negations in rule bodies (using additional background information):

$$EarlyRegisteredParticipant(X) \leftarrow Participant(X) \wedge RegisterdbeforeDeadline(X)\,;$$
$$LateRegisteredParticipant(X) \leftarrow Participant(X) \wedge not\ RegisteredbeforeDeadline(X)\,.$$

Syntax and Semantics

Tightly integrated probabilistic dl-program $KB = (L, P, C, \mu)$:

- description logic knowledge base L,
- disjunctive program P with values of random variables $A \in C$ as "switches" in rule bodies,
- probability distribution μ over all joint instantiations B of the random variables $A \in C$.

They specify a set of probability distributions over first-order models: Every joint instantiation B of the random variables along with the generalized normal program specifies a set of first-order models of which the probabilities sum up to $\mu(B)$.

Example

Probabilistic rules in P along with the probability μ on the choice space C of a probabilistic dl-program $KB = (L, P, C, \mu)$:

- $avoid(X) \leftarrow Camera(X), not\ offer(X), avoid\_pos;$
- $offer(X) \leftarrow Electronics(X), not\ brand\_new(X), offer\_pos;$
- $buy(C, X) \leftarrow needs(C, X), view(X), not\ avoid(X), v\_buy\_pos;$
- $buy(C, X) \leftarrow needs(C, X), buy(C, Y), also\_buy(Y, X), a\_buy\_pos.$

μ: $avoid\_pos,\ avoid\_neg \mapsto 0.9\,, 0.1;\ \ offer\_pos,\ offer\_neg \mapsto 0.9\,, 0.1;$
$v\_buy\_pos,\ v\_buy\_neg \mapsto 0.7\,, 0.3;\ \ a\_buy\_pos,\ a\_buy\_neg \mapsto 0.7\,, 0.3.$

$\{avoid\_pos, offer\_pos, v\_buy\_pos, a\_buy\_pos\} : 0.9 \times 0.9 \times 0.7 \times 0.7, \ldots$

Probabilistic query: $\exists(buy(john, ixus500))[L, U]$

Basics

Tightly integrated probabilistic dl-programs $KB = (L, P, C, \mu)$ can be used for representing (possibly inconsistent) mappings with confidence values between two ontologies.

Intuitively, L encodes the union of the two ontologies, while P, C, and μ encode the mappings between the ontologies.

Here, confidence values can be encoded as error probabilities, and inconsistencies can also be resolved via trust probabilities (in addition to using disjunctions and negations in P).

Example

Mapping the publication ontology in test 101 (O_1) on the ontology of test 302 (O_2) of the Ontology Alignment Evaluation Initiative:

Encoding two mappings produced by hmatch:

$$\begin{aligned} &Book(X) \leftarrow Collection(X) \wedge hmatch_1 \; ; \\ &Proceedings_2(X) \leftarrow Proceedings_1(X) \wedge hmatch_2 \; . \end{aligned}$$

$C = \{\{hmatch_i, not\_hmatch_i\} \mid i \in \{1, 2\}\}$
$\mu(hmatch_1) = 0.62$ and $\mu(hmatch_2) = 0.73$.

Encoding two mappings produced by falcon:

$$\begin{aligned} &InCollection(X) \leftarrow Collection(X) \wedge falcon_1 \; ; \\ &Proceedings_2(X) \leftarrow Proceedings_1(X) \wedge falcon_2 \; . \end{aligned}$$

$C' = \{\{falcon_i, not\_falcon_i\} \mid i \in \{1, 2\}\}$
$\mu'(falcon_1) = 0.94$ and $\mu'(falcon_2) = 0.96$.

Merging the two encodings:

$$\begin{aligned}
&\mathit{Book}(X) \leftarrow \mathit{Collection}(X) \wedge \mathit{hmatch}_1 \wedge \mathit{sel_hmatch}_1\ ;\\
&\mathit{InCollection}(X) \leftarrow \mathit{Collection}(X) \wedge \mathit{falcon}_1 \wedge \mathit{sel_falcon}_1\ ;\\
&\mathit{Proceedings}_2(X) \leftarrow \mathit{Proceedings}_1(X) \wedge \mathit{hmatch}_2\ ;\\
&\mathit{Proceedings}_2(X) \leftarrow \mathit{Proceedings}_1(X) \wedge \mathit{falcon}_2\ .
\end{aligned}$$

$C'' = C \cup C' \cup \{\mathit{sel\_hmatch}_1, \mathit{sel\_falcon}_1\}$
$\mu'' = \mu \cdot \mu' \cdot \mu^\star$, where $\mu^\star$: $\mathit{sel\_hmatch}_1, \mathit{sel\_falcon}_1 \mapsto 0.55, 0.45$.

Any randomly chosen instance of *Proceedings* of O_1 is also an instance of *Proceedings* of O_2 with the probability 0.9892.

Probabilistic query $Q = \exists(\mathit{Book}(\mathit{pub}))[R, S]$:
The tight answer θ to Q is $\theta = \{R/0, S/0\}$ (resp., $\theta = \{R/0.341, S/0.341\}$), if *pub* is not (resp., is) an instance of *Collection* in O_1.

- Tightly integrated probabilistic (disjunctive) dl-programs for representing ontology mappings.
- Resolving inconsistencies via disjunctions in rule heads and nonmonotonic negations in rule bodies.
- Explicitly representing numeric confidence values as error probabilities, resolving inconsistencies via trust probabilities, and reasoning about these on a numeric level.
- Expressive, well-integrated with description logic ontologies, still decidable, and data-tractable subsets.

References

- A. Calí, T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt. Tightly coupled probabilistic description logic programs for the Semantic Web. *J. Data Semantics* 12:95–130, 2009.
- T. Lukasiewicz, L. Predoiu, H. Stuckenschmidt. Tightly integrated probabilistic description logic programs for representing ontology mappings. *Ann. Math. Artif. Intell.* 63(3/4):385–425, 2011.

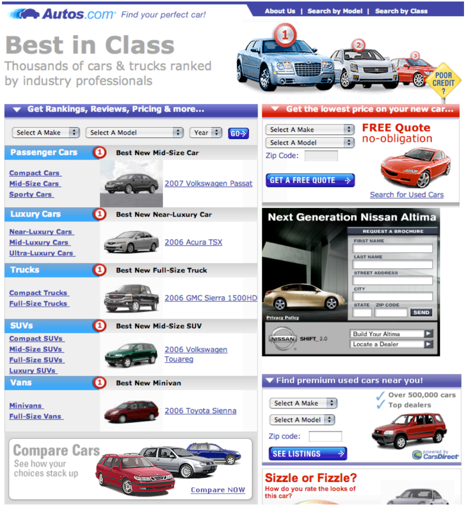
Outline

Example

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!
Over 500,000 cars
Top dealers
Select A Make
Select A Model
Zip code:
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.

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

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

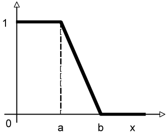
$$\begin{aligned} Cars \sqsubseteq\ & (\exists hasReview.Integer) \sqcap (\exists hasInvoice.Integer) \\ & \sqcap (\exists hasResellValue.Integer) \sqcap (\exists hasMaxSpeed.Integer) \\ & \sqcap (\exists hasHorsePower.Integer) \sqcap \ \ldots \end{aligned}$$

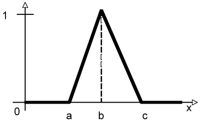
$$\begin{aligned} & MazdaMX5Miata : SportyCar \sqcap (\exists hasInvoice.18883) \\ & \qquad \sqcap (\exists hasHorsePower.166) \sqcap \ldots \\ & MitsubishiEclipseSpyder : SportyCar \sqcap (\exists hasInvoice.24029) \\ & \qquad \sqcap (\exists hasHorsePower.162) \sqcap \ldots \end{aligned}$$

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:

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

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





Syntax

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

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

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".

$$
\begin{aligned}
\mathit{query}(x) \quad \leftarrow_{\otimes} \quad & DL[\mathit{SportyCar}](x) \wedge_{\otimes} \\
& DL[\mathit{hasInvoice}](x, y_1) \wedge_{\otimes} \\
& DL[\mathit{LeqAbout}22000](y_1) \wedge_{\otimes} \\
& DL[\mathit{hasHorsePower}](x, y_2) \wedge_{\otimes} \\
& DL[\mathit{Around}150\mathit{HP}](y_2) \geqslant 1 \,.
\end{aligned}
$$

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

Semantics

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

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).

Example

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

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

Ontology alignment mapping rules:

$$\begin{aligned} SportsCar(x) &\leftarrow_{\otimes} DL[SportyCar](x) \wedge_{\otimes} sc_{pos} \geqslant 1 \\ hasPrize(x) &\leftarrow_{\otimes} DL[hasInvoice](x) \wedge_{\otimes} hi_{pos} \geqslant 1 \\ hasPower(x) &\leftarrow_{\otimes} DL[hasHorsePower](x) \wedge_{\otimes} hhp_{pos} \geqslant 1\,, \end{aligned}$$

Probability distribution μ:

$$\begin{array}{ll} \mu(sc_{pos}) = 0.91 & \mu(sc_{neg}) = 0.09 \\ \mu(hi_{pos}) = 0.78 & \mu(hi_{neg}) = 0.22 \\ \mu(hhp_{pos}) = 0.83 & \mu(hhp_{neg}) = 0.17\,. \end{array}$$

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

- 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, U. Straccia. Description logic programs under probabilistic uncertainty and fuzzy vagueness. *Int. J. Approx. Reasoning* 50(6):837–853, 2009.