

Estensioni del linguaggio di rappresentazione della Programmazione Logica Induttiva

Fabrizio Riguzzi

Relatore: Prof. Aurelio Boari

Correlatori: Prof. Paola Mello, Prof. Evelina Lamma, Prof. Antonis Kakas

Data di presentazione: 30 novembre 1998

Questa tesi si occupa dello studio dell'estensione del linguaggio di rappresentazione della Programmazione Logica Induttiva. La Programmazione Logica Induttiva (Inductive Logic Programming, ILP nel seguito) è un'area di ricerca che si occupa dell'apprendimento di programmi logici partendo da una conoscenza a priori nella forma di un programma logico e da esempi nella forma di fatti ground. Sono state considerate tre diverse estensioni della programmazione logica: programmi logici abduitivi, programmi logici multi-teoria e programmi logici con negazione esplicita.

I programmi logici abduitivi consentono di effettuare ragionamento ipotetico sulla conoscenza, facendo assunzioni su alcuni predicati definiti "abducibili" in modo da spiegare il goal. Le assunzioni fatte devono inoltre rispettare dei vincoli di integrità specificati nella teoria. L'impiego di programmi logici abduitivi come linguaggio di rappresentazione della conoscenza consente di apprendere da una conoscenza di background incompleta [1,2]: l'abduzione viene usata durante l'apprendimento per completare la conoscenza disponibile. I risultati dei primi esperimenti con un sistema in grado di apprendere programmi logici abduitivi hanno dimostrato che le prestazioni per problemi di apprendimento da conoscenza incompleta sono paragonabili o superiori a quelle di sistemi quali FOIL e mFOIL.

I programmi logici multi-teoria consentono di organizzare la conoscenza di background in una tassonomia di classi e istanze. L'utilizzo di un linguaggio di rappresentazione multi-teoria consente di introdurre un modello ad oggetti dei dati in ILP, rendendo così disponibili i vantaggi della modellazione a oggetti: i domini strutturati possono essere immediatamente rappresentati tramite i costrutti del linguaggio e la suddivisione della conoscenza facilita la gestione di grandi basi di conoscenza. Questo rende più immediata l'applicazione di ILP alla generazione automatica di basi di conoscenza strutturate.

Un primo sistema è stato realizzato e sperimentato su alcuni semplici casi di studio [3]. I risultati della sperimentazione hanno confermato le ipotesi fatte circa le potenzialità rappresentative dei programmi logici multi-teoria dal punto di vista della strutturazione e non replicazione della conoscenza.

I programmi logici con negazione esplicita (o programmi logici estesi, Extended Logic Programs) consentono di rappresentare sia il concetto che si vuole imparare che il suo opposto. Solitamente in ILP viene imparata una definizione solo per il concetto target mentre il suo opposto viene definito mediante la Closed World Assumption. Questa però comporta dei problemi nel caso di apprendimento da conoscenza incompleta riguardo i predicati target, dove non si conosce la classificazione (positiva o negativa) di tutte le istanze del dominio, ma solo di una parte. Questa è la situazione tipica di un robot che esplora autonomamente l'ambiente circostante e acquisisce conoscenza apprendendo dalle osservazioni che man mano raccoglie. Per un tale robot è importante distinguere ciò che è vero da ciò che è falso e da ciò che è sconosciuto e quindi utilizzare una logica a tre valori. Infatti, nel caso utilizzi una logica a due valori, quando apprende da specifico a

generale, otterrà una definizione che copre solamente gli esempi positive e, usando la CWA, non sarà poi in grado di distinguere ciò che è falso da ciò che è sconosciuto, mentre quando apprende da generale a specifico, otterrà una definizione che copre sia gli atomi positivi che gli sconosciuti e, usando la CWA, non sarà in grado di distinguere ciò che è vero da ciò che è sconosciuto.

Nelle situazioni in cui il training set sia incompleto (ovvero non comprenda un atomo per ciascun istanza del dominio) è quindi importante utilizzare un formalismo di rappresentazione che ammetta una semantica a tre valori, come i programmi logici con negazione esplicita. Questo linguaggio mette a disposizione due forme di negazione: la negazione esplicita viene utilizzata per rappresentare l'opposto del concetto che si vuole apprendere, mentre la negazione a default viene utilizzata per evitare la contraddizione delle definizioni del concetto e del suo opposto, in modo diverso a seconda se l'istanza contraddittoria sia una eccezione oppure una istanza sconosciuta.

È stato progettato un algoritmo [4] in grado di apprendere programmi logici estesi contenenti definizioni per il concetto e per il suo opposto che possono ammettere eccezioni. Nel caso vi siano eccezioni, il sistema apprende una definizione per la classe delle eccezioni, per le eccezioni alle eccezioni, se presenti, e così via. In questo modo l'algoritmo è in grado di apprendere una gerarchia di eccezioni.

[1] A. Kakas e F. Riguzzi, "Learning with Abduction", in: Proceedings of the 7th International Workshop on Inductive Logic Programming, 1997.

[2] E. Lamma, P. Mello, M. Milano e F. Riguzzi, "Integrating induction and abduction in logic programming", apparirà sul Information Sciences, Kluwer.

[3] M. Milano, A. Omicini e F. Riguzzi, "Learning with an Object-Oriented Data Model", in: Atti dell'Incontro dei Gruppi di Lavoro di Apprendimento Automatico e Linguaggio Naturale dell'Associazione Italiana per l'Intelligenza Artificiale (AI*IA), Torino, 1997.

[4] E. Lamma, F. Riguzzi, L. M. Pereira, "Strategies for Learning with Extended Logic Programs", in: Proceedings of the Fourth International Workshop on Multistrategy Learning (MSL98), 1998.