Slipcover

“Structure LearnIng of Probabilistic logic programs by searChing OVER the clause space”

# Introduzinoe

L’algoritmo SLIPCOVER esegue una ricerca “beam” nello spazio delle clausole probabilistiche e una ricerca “greedy” nello spazio delle teorie, utilizzando la verosimiglianza logaritmica dei dati come euristica guida.

Per stimare la verosimiglianza logaritmica, SLIPCOVER esegue la massimizzazione delle aspettative con EMBLEM.

SLIPCOVER raggiunge valori più elevati nelle curve di ROC e precision-recall nella maggior parte dei casi quando comparato a SLIPCASE, SEM-CP logic ed altri algoritmi.

# Corpo

L’inferenza per linguaggi PLP possono essere eseguiti con un certo numero di algoritmi che in molti casi trovano la spiegazione delle queries e ne calcolano la loro probabilità costruendo un albero di decisione binaria (BDD)

SLIPCASE apprende la struttura di un LPADs eseguendo una ricerca ‘beam’ nello spazio delle teorie probabilistiche utilizzando la verosimiglianza logaritmica dei dati come euristica guida. Per stimare la verosimiglianza logaritmica vengono eseguiti un certo numero di iterazioni di ‘Expetation Maximization’ di EMPBLEM.

Parte verde

SLIPCOVER è una evoluzione di SLIPCASE, con la differenza che SLIPCOVER per prima cosa cerca nello spazio delle clausole, tenendo da parte quelle ritenute promettenti, successivamente con approccio discriminativo le divide in:

* Clausole per i predicati ‘target’ (quelli da predire)
* Clausole per i predicati di background (i rimanenti)

Questa ricerca parte da un insieme di clausole dette ‘bottom clauses’ generate come in Prolog e cerca perfezionamenti in termini di LL. A questo punto procede con la ricerca ‘greedy’ nello spazio delle teorie. Infine, esegue l'apprendimento dei parametri con EMBLEM sulla migliore teoria target e sulle clausole per i predicati di background.

SLIPCOVER può apprendere LPAD generali inclusi programmi non-ground.