Generalizzazione di Clausole Basata Su Un Nuovo Criterio di Similarità

S. Ferilli, T.M.A. Basile, N. Di Mauro, M. Biba, and F. Esposito

Dipartimento di Informatica
Università di Bari
via E. Orabona, 4 - 70125 Bari - Italia
{ferilli, basile, ndm, biba, esposito}@di.uniba.it

Abstract. Pochi lavori in letteratura sono stati dedicati alla definizione di criteri di similarità tra formule della Logica del Prim'Ordine, dove la presenza di relazioni fa sì che diverse porzioni di una descrizione possano essere mappate in molti modi differenti su un'altra descrizione. Quindi, è importante individuare dei criteri generali che siano in grado di supportare il confronto tra formule. Questo potrebbe avere molte applicazioni; questo lavoro tratta il caso di due descrizioni (ad es., una definizione ed una osservazione) che devono essere generalizzate, dove il criterio di similarità potrebbe aiutare a concentrarsi sulle sottoparti delle descrizioni che sono più simili e quindi più probabilmente corrispondenti tra loro, basandosi sulla loro struttura sintattica. Sperimentazioni su dataset reali dimostrano l'efficacia dell'approccio proposto e l'efficienza di una implementazione in una procedura di generalizzazione.

1 Introduzione

La Logica del Prim'Ordine è un potente formalismo in grado di esprimere relazioni tra oggetti, il che gli consente di superare le limitazioni delle rappresentazioni proposizionali o attributo-valore. Tuttavia, la presenza di relazioni fa sì che varie porzioni di una descrizione possano essere mappate in molti possibili modi differenti su un altra descrizione. Questo pone un importante problema di sforzo computazionale quando due descrizioni devono essere confrontate tra di loro. D'altra parte, le tecniche per il confronto tra due descrizioni del prim'ordine potrebbero avere molte applicazioni, in particolare in Intelligenza Artificiale: per esempio, aiutare una procedura di sussunzione a convergere velocemente; valutare il grado di similarità tra due formule; implementare una procedure di matching flessibile; supportare tecniche di classificazione del tipo instance-based oppure il clustering concettuale.

Per quanto riguarda l'apprendimento supervisionato, molti sistemi si basano sulla generalizzazione delle definizioni rispetto alle osservazioni, ed una funzione di similarità potrebbe aiutare la procedura a concentrarsi sulle componenti che sono più simili e che quindi è più probabile che corrispondano. Ovviamente, questo riguarda gli aspetti semantici del dominio e quindi non esiste un modo

preciso (algoritmico) per riconoscere le (sotto-)formule giuste. Quindi, il problema deve essere affrontato euristicamente, sviluppando qualche metodo che può ipotizzare quali parti di una descrizione si riferiscono a quali parti di un'altra descrizione, basandosi esclusivamente sulla loro struttura sintattica. Per questo motivo, è necessario cercare similarità parziali tra le componenti delle descrizioni.

In particolare, molti sistemi di Apprendimento Automatico nel Prim'Ordine inducono teorie sotto forma di Programmi Logici, una restrizione della Logica del Prim'Ordine a insiemi di *Clausole di Horn*, ossia formule logiche della forma $l_1 \wedge \cdots \wedge l_n \Rightarrow l_0$ dove gli l_i sono *atomi*, spesso rappresentate in stile Prolog come l_0 :- $l_1, \ldots, l_n$ da interpretarsi come "l_0 (detto *testa* della clausola) è vera, se l_1 e ... e l_n (detto *corpo* della clausola) sono tutti veri". Senza perdita di generalità [8], nel seguito si tratterà il caso di clausole Datalog connesse.

Nei paragrafi seguenti verranno presentati alcuni criteri ed una formula sui quali basare le considerazioni sulla similarità tra clausole del prim'ordine. Il paragrafo 5 mostra come la formula proposta è in grado di guidare in maniera efficace una procedura di generalizzazione. Infine, il paragrafo 4 individua i lavori connessi, mentre il 6 conclude l'articolo presentando spunti per lavori futuri.

2 La Formula di Similarità

Intuitivamente, la valutazione della similarità tra due entità i' ed i'' potrebbe essere basata sia sulla presenza di caratteristiche comuni, che dovrebbero incidere positivamente sulla valutazione della similarità, sia sulle caratteristiche di ciascuna entità non possedute dall'altra (definiamo questo come *residuo* della prima entità rispetto alla seconda), che dovrebbe incidere negativamente sul valore complessivo della funzione di similarità. [5]. Quindi, possibili parametri di similarità sono:

n , il numero delle catteristiche possedute da i' ma non da i'' (*residuo* di i' rispetto a i'');
l , il numero delle catteristiche possedute sia da i' che da i'';
m ,il numero delle catteristiche possedute da i'' ma non da i' (*residuo* di i'' rispettoa i').

Si è quindi sviluppata una nuova funzione che esprime il grado di similarità tra i' ed i'' basato su tali parametri:

$$sf(i', i'') = \mathrm{sf}(n, l, m) = \frac{l+1}{l+n+2} + \frac{l+1}{l+m+2} \tag{1}$$

La funzione restituisce valori in $]0, 1[$, il che richiama la teoria della probabilità e quindi può aiutare gli esseri umani ad interpretare il valore risultante. Una sovrapposizione completa del modello sull'osservazione tende al limite di 1 al crescere del numero delle caratteristiche in comune. Il valore di similarità totale 1 non viene mai raggiunto, il che è coerente con l'intuizione che l'unico caso in cui questo deve succedere è l'esatta identità tra le due entità, ossia $i' = i''$ (in seguito, assumeremo $i' \neq i''$). Al contrario, nel caso di non-sovrapposizione,

la funzione tende a 0 al crescere del numero delle caratteristiche non condivise. Questo è coerente con l'intuizione che non esiste un limite al numero di caratteristiche differenti in due descrizioni che contribuiscono a renderle diverse. Inoltre, nel caso in cui non ci siano caratteristiche condivise da due descrizioni ($n = l = m = 0$, cioè il confronto di due oggetti che non hanno nessuna caratteristica in comune) la funzione assume il valore di $1/2$, che può essere considerato intuitivamente corretto per rappresentare il caso di massima incertezza. Per esempio, un tale caso si verifica quando un modello include un oggetto privo di proprietà da soddisfare: confrontandolo con un altro oggetto osservato, privo anch'esso di proprietà, non si è in grado di capire se la sovrapposizione è totale perchè le due descrizioni non hanno realmente nessuna proprietà da soddisfare, oppure perchè precedenti generalizzazioni hanno rimosso dal modello tutte le caratteristiche di cui prima disponeva. Va notato che ciascuno dei due termini si riferisce in particolare ad una delle due entità che si stanno confrontando, e quindi si potrebbe introdurre un peso per attribuire diversa importanza a ciascuna di esse (questo potrebbe essere tipicamente necessario quando il confronto riguarda un modello rispetto ad una osservazione); questo, comunque, travalica gli scopi di questo articolo.

3 Criteri di Similarità

Nelle formule del prim'ordine, i termini rappresentano oggetti specifici; i predicati unari in generale rappresentano proprietà dei, mentre i predicati n-ari esprimono relazioni fra termini. Quindi, si possono definire due livelli di similarità per coppie di descrizioni del prim'ordine: il livello degli *oggetti*, relativo alla similarità tra termini nelle descrizioni, e il livello *strutturale*, che si riferisce a come i grafi delle relazioni nelle descrizioni si sovrappongono:

Example 1. Consideriamo come esempio illustrativo nel resto dell'articolo, le seguenti due clausole (in questo caso una regola C ed una osservazione classificata E):

$$\begin{aligned} C : h(X) \text{ :- } & p(X,Y), p(X,Z), p(W,X), r(Y,U), o(Y,Z), q(W,W), s(U,V), \\ & \pi(X), \phi(X), \rho(X), \pi(Y), \sigma(Y), \tau(Y), \phi(Z), \sigma(W), \tau(W), \pi(U), \phi(U). \\ E : h(a) \text{ :- } & p(a,b), p(a,c), p(d,a), r(b,f), o(b,c), q(d,e), t(f,g), \\ & \pi(a), \phi(a), \sigma(a), \tau(a), \sigma(b), \tau(b), \phi(b), \tau(d), \rho(d), \pi(f), \phi(f), \sigma(f). \end{aligned}$$

3.1 Similarità tra Oggetti

Consideriamo due clausole C' e C''; chiamiamo $A' = \{a'_1, \ldots, a'_n\}$ l'insieme dei termini in C', e $A'' = \{a''_1, \ldots, a''_m\}$ l'insieme dei termini in C''. Quando si confrontano una coppia $(a', a'') \in A' \times A''$, ossia un oggetto preso da C' e uno da C'', rispettivamente, si possono distinguere due tipi di caratteristiche: le proprietà espresse dai predicati unari (*attributi caratteristici*), e i modi in cui questi sono relazionati ad altri oggetti in base ai predicati n-ari (*attributi relazionali*). Più precisamente, gli attributi relazionali sono definiti dalla posizione che gli

oggetti occupano tra gli argomenti dei predicati n-ari, poichè posizioni diverse si riferiscono a ruoli diversi svolti dagli oggetti. In seguito, intenderemo un *ruolo* come una coppia $R = (predicato, posizione)$ (scritto in maniera compatta come $R = predicato/arietà.posizione$). Per esempio, attributi caratteristici possono essere `maschio(X)` o `alto(X)`, mentre attributi relazionali possono essere espressi da predicati come `genitore(X,Y)`, dove in particolare l'argomento in prima posizione identifica il ruolo di genitore (*genitore*/2.1), e il secondo rappresenta il ruolo di figlio (*genitore*/2.2).

Due valori di similarità possono quindi essere associati ad a' e a'': una *similarità caratteristica*, dove (1) è applicata ai valori relativi agli attributi caratterizzanti, e una *similarità relazionale*, basata sul numero di volte che due oggetti svolgono lo stesso ruolo nei predicati n-ari.

La similarità caratteristica tra a' e a'', $\mathrm{sf}_c(a', a'')$, può essere calcolata considerando l'insieme P' delle proprietà relative ad a' e l'insieme P'' delle proprietà relative ad a'' come $\mathrm{sf}(n_c, l_c, m_c)$ per i seguenti parametri:

$n_c = |P' \setminus P''|$ è il numero delle proprietà possedute dall'oggetto rappresentato dal termine a' in C' ma non dall'oggetto rappresentato dal termine a'' in C'' (*residuo caratteristico* di a' rispetto ad a'');

$l_c = |P' \cap P''|$ è il numero di proprietà in comune tra l'oggetto rappresentato dal termine a' in C' e l'oggetto rappresentato dal termine a'' in C'';

$m_c = |P'' \setminus P'|$ è il numero di proprietà possedute dall'oggetto rappresentato dal termine a'' in C'' ma non dall'oggetto rappresentato dal termine a' in C' (*residuo caratteristico* di a'' rispetto ad a')

Una tecnica simile si può usare per calcolare la similarità relazionale tra a' ed a''. In questo caso, poichè un oggetto può svolgere più volte lo stesso ruolo in relazioni differenti (ad esempio, un genitore di molti figli), bisogna considerare i *multi*insiemi R' e R'' dei ruoli svolti da a' ed a'', rispettivamente. Quindi, la similarità relazionale tra a' ed a'', $\mathrm{sf}_r(a', a'')$, può essere calcolata come $\mathrm{sf}(n_r, l_r, m_r)$, dove

$n_r = |R' \setminus R''|$ esprime quante volte a' svolge in C' ruoli che a'' non svolge in C'', (*residuo relazionale* di a' rispetto ad a'');

$l_r = |R' \cap R''|$ è il numero di volte che sia a' in C' che a'' in C'' svolgono gli stessi ruoli;

$m_r = |R'' \setminus R'|$ esprime quante volte a'' svolge in C'' ruoli che a' non svolge in C', (*residuo relazionale* di a'' rispetto ad a').

Complessivamente, la *similarità di oggetti* tra due termini si può definire come $\mathrm{sf}_o(a', a'') = \mathrm{sf}_c(a', a'') + \mathrm{sf}_r(a', a'')$.

Example 2. Le proprietà e ruoli per lo stesso termine in C e E, e il confronto per alcune delle possibili coppie, sono riportate in Tabella 1.

Table 1. Similarità di Oggetti

C			E		
t'	P'	R'	t''	P''	R''
X	$\{\pi, \phi, \rho\}$	$\{p/2.1, p/2.1, p/2.2\}$	a	$\{\pi, \phi, \sigma, \tau\}$	$\{p/2.1, p/2.1, p/2.2\}$
Y	$\{\pi, \sigma, \tau\}$	$\{p/2.2, r/2.1, o/2.1\}$	b	$\{\sigma, \tau\}$	$\{p/2.2, r/2.1, o/2.1\}$
Z	$\{\phi\}$	$\{p/2.2, o/2.2\}$	c	$\{\phi\}$	$\{p/2.2, o/2.2\}$
W	$\{\sigma, \tau\}$	$\{p/2.1, p/2.1, p/2.2\}$	d	$\{\tau, \rho\}$	$\{p/2.1, p/2.1\}$
U	$\{\pi, \phi\}$	$\{r/2.2, s/2.1\}$	f	$\{\pi, \phi, \sigma\}$	$\{r/2.2, t/2.1\}$

t'/t''	$(P' \setminus P''), (P' \cap P''), (P'' \setminus P')$		$(R' \setminus R''), (R' \cap R''), (R'' \setminus R')$		$\mathrm{sf}_o(t', t'')$
X/a	$\{\rho\}, \{\pi, \phi\}, \{\sigma, \tau\}$	$(1, 2, 2)$	$\emptyset, \{p/2.1, p/2.1, p/2.2\}, \emptyset$	$(0, 3, 0)$	1.35
Y/b	$\{\pi\}, \{\sigma, \tau\}, \emptyset$	$(1, 2, 0)$	$\emptyset, \{p/2.2, r/2.1, o/2.1\}, \emptyset$	$(0, 4, 0)$	1.46
Y/c	$\{\pi, \sigma, \tau\}, \emptyset, \{\phi\}$	$(3, 0, 1)$	$\{r/2.1, o/2.1\}, \{p/2.2\}, \{o/2.2\}$	$(2, 1, 1)$	0.72
Z/b	$\{\phi\}, \emptyset, \{\sigma, \tau\}$	$(1, 0, 2)$	$\{o/2.2\}, \{p/2.2\}, \{r/2.1, o/2.1\}$	$(1, 1, 2)$	0.74
Z/c	$\emptyset, \{\phi\}, \emptyset$	$(0, 1, 0)$	$\emptyset, \{p/2.2, o/2.2\}, \emptyset$	$(0, 2, 0)$	1.42
W/d	$\{\sigma\}, \{\tau\}, \{\rho\}$	$(1, 1, 1)$	$\{p/2.2\}, \{p/2.1, p/2.1\}, \emptyset$	$(1, 2, 0)$	1.18
U/f	$\emptyset, \{\pi, \phi\}, \{\sigma\}$	$(0, 2, 1)$	$\{s/2.1\}, \{r/2.2\}, \{t/2.1\}$	$(1, 1, 1)$	1.18

3.2 Similarità Strutturale

Quando si cerca di valutare la similarità strutturale tra due formule, possono essere coinvolti molti oggetti, le cui relazioni reciproche rappresentano quindi un vincolo su come ognuno di quelli nella prima formula possa essere associato ad un altro nella seconda. Diversamente dal caso degli oggetti, quello che definisce la struttura di una formula è l'insieme dei predicati n-ari, ed in particolare il modo in cui questi sono applicati ai vari oggetti per metterli in relazione (un predicato applicato ad un numero di termini uguale alla sua arietà viene chiamato *atomo*). Questa è la parte più difficile, poichè le relazioni sono proprio la componente del paradigma del prim'ordine che causa indeterminazione nel mappare (parti di) una formula su (parti di) un'altra formula. D'ora in poi, chiameremo *compatibili* due (sotto-)formule del prim'ordine, che possono essere mappate tra di loro senza produrre incoerenze nell'associazione dei termini (in altre parole, un termine in una formula non può corrispondere a termini differenti in un'altra formula).

Dato un letterale n-ario, definiamo la sua *stella* come il multiset di predicati n-ari corrispondenti ai letterali ad esso connessi tramite qualche termine in comune (infatti, un predicato può apparire in più istanze multiple di questi letterali). Intuitivamente, la stella di un letterale è una vista in ampiezza di come il letterale è legato al resto della formula. La *similarità di stella* $\mathrm{sf}_s(l', l'')$ tra due letterali n-ari compatibili l' e l'' che hanno stelle S' ed S'', rispettivamente, può essere calcolata come $\mathrm{sf}(n_s, l_s, m_s)$ per i seguenti parametri:

$n_s = |S' \setminus S''|$ che esprime quante relazioni l' ha in più in C' rispetto a quelle che l'' ha in C'' (*residuo di stella* di l' rispetto a l'')

$l_s = |S' \cap S''|$ che è il numero delle relazioni che hanno in comune l' in C' ed l'' in C'';

$m_s = |S'' \setminus S'|$ che esprime quante relazioni l'' ha in più in C'' rispetto a quelle che l' ha in C' (*residuo di stella* di l'' rispetto a l').

Complessivamente, una valutazione più adeguata della similarità tra l' e l'' può essere ottenuta aggiungendo a questo risultato la similarità caratteristica e strutturale per tutte le coppie dei loro argomenti in posizioni corrispondenti:

$$\mathrm{sf}_s(l', l'') = \mathrm{sf}(n_s, l_s, m_s) + \Sigma_{t'/t'' \in \theta} \mathrm{sf}_o(t', t'')$$

dove θ è l'insieme delle associazioni dei termini che mappanno l' su l''.

Ogni formula del prim'ordine può essere rappresentata come un grafo in cui gli atomi sono i nodi e gli archi connettono due nodi se e solo se questi sono relazionati in qualche modo. Ne consegue che un confronto tra due formule per valutare la loro similarità strutturale corrisponde al calcolo di un omomorfismo fra (sotto-)grafi, un problema noto essere *NP*-difficile a causa della possibilità di mappare un (sotto-)grafo su un altro in molti modi differenti. Di conseguenza, si è interessati ad euristiche che possano dare suggerimenti significativi sulla sovrapposizione della struttura tra due formule con basso sforzo computazionale. La rappresentazione basata su grafo è ovviamente più facile per clausole che per formule generali, poichè le clausole sono composte da al massimo un unico atomo in testa e da una congiunzione di atomi nel corpo. Nel seguito si tratteranno solo clausole *connesse* (cioè il cui grafo associato è connesso), e si costruirà il grafo sulla base di una caratteristica semplice (per quanto riguarda i dettagli che esprime circa una formula), ma al tempo stesso potente (per quanto riguarda l'informazione che essa convoglia), che è la condivisione dei termini tra coppie di atomi.

Data una clausola C, definiamo il suo *grafo associato* come $G_C = (V, E)$ con

- $V = \{l_0\} \cup \{l_i | i \in \{1, \ldots, n\}, l_i$ costruito su predicati k-ari, $k > 1\}$ e
- $E \subseteq \{(a_1, a_2) \in V \times V \mid terms(a_1) \cap terms(a_2) \neq \emptyset\}$

dove $terms(a)$ denota l'insieme dei termini che appaiono come argomenti dell'atomo a. La strategia di selezione degli archi da rappresentare, schematizzata nell'Algoritmo 1, si basa sulla presenza di un unico atomo nella testa, che fornisce sia un punto di partenza che precise direzioni per attraversare il grafo, al fine di scegliere, tra le tante possibili, una prospettiva unica e ben definita sulla struttura della clausola. Più precisamente, costruiamo un grafo orientato aciclico, *stratificato* (cioè con l'insieme dei nodi partizionato) in modo tale che la testa sia l'unico nodo al livello 0 (primo elemento della partizione) ed ogni livello (elemento della partizione) successivo sia composto da nuovi nodi (non ancora raggiunti dagli archi) che hanno almeno un termine in comune con nodi del livello precedente. In particolare, ogni nodo nel nuovo livello è connesso tramite un arco entrante ad ogni nodo al livello precedente che ha tra i suoi argomenti almeno un termine in comune con esso.

Example 3. Costruiamo il grafo G_C. La testa rappresenta il livello 0 della stratificazione. Quindi, degli archi orientati possono essere introdotti da $h(X)$ a $p(X, Y)$, $p(X, Z)$ e $p(W, X)$, che sono i soli atomi ad avere X come argomento, il che produce il livello 1 della stratificazione dei termini. Adesso, il prossimo livello può essere costruito, aggiungendo archi orientati che vanno da atomi al livello 1 ad

Algorithm 1 Costruzione del grafo associato a C

Require: $C = l_0 : -l_1, \ldots, l_n$: Clause
  $i \leftarrow 0$; $Level_0 \leftarrow \{l_0\}$; $E \leftarrow \emptyset$; $Atoms \leftarrow \{l_1, \ldots, l_n\}$
  while $Atoms \neq \emptyset$ **do**
    $i \leftarrow i + 1$
    $Level_i \leftarrow \{l \in Atoms \mid \exists l' \in Level_{i-1} s.t. terms(l) \cap terms(l') \neq \emptyset\}$
    $E \leftarrow E \cup \{(l', l'') \mid l' \in Level_{i-1}, l'' \in Level_i, terms(l') \cap terms(l'') \neq \emptyset\}$
    $Atoms \leftarrow Atoms \setminus Level_i$
  end while
  return $G = (\bigcup_i Level_i, E)$: graph associated to C

atomi non ancora considerati che condividono una variabile con loro: $r(Y, U)$ – fine di un arco che parte a $p(X, Y)$ –, $o(Y, Z)$ – fine di archi che partono da $p(X, Y)$ e $p(X, Z)$ – e $q(W, W)$ – fine di un arco che parte da $p(W, X)$. Il terzo (e ultimo) livello del grafo include l'unico atomo rimasto, $s(U, V)$ – avente un arco entrante da $r(Y, U)$.

Adesso, tutti i possibili cammini che partono dalla testa e raggiungono nodi *foglia* (quelli senza archi uscenti) possono essere interpretati come le componenti base della struttura complessiva della clausola. Essendo questi cammini univocamente determinati, si riduce la quantità di indeterminazione nel confronto. Intuitivamente, un cammino descrive in profondità una porzione delle relazioni descritte nella clausola.

Date due clausole, C' e C'', definiamo *intersezione* tra due cammini $p' =< l'_1, \ldots, l'_{n'} >$ in $G_{C'}$ e $p'' =< l''_1, \ldots, l''_{n''} >$ in $G_{C''}$ la coppia delle più lunghe sottosequenze inziali compatibili di p' e p'':

$p' \cap p'' = (p_1, p_2) = (< l'_1, \ldots, l'_k >, < l''_1, \ldots, l''_k >)$ s.t.
$\forall i = 1, \ldots, k : l'_1, \ldots, l'_i$ compatibile con $l''_1, \ldots, l''_i \wedge$
$(k = n' \vee k = n'' \vee l'_1, \ldots, l'_{k+1}$ incompatibile con $l''_1, \ldots, l''_{k+1})$
e i due residui come le parti incompatibili restanti.

$p' \setminus p'' =< l'_{k+1}, \ldots, l'_{n'} >, p'' \setminus p' =< l''_{k+1}, \ldots, l''_{n''} >)$

Quindi, la *similarità di cammini* tra p' e p'', $\mathrm{sf}_s(p', p'')$, può essere calcolata applicando (1) ai seguenti parametri:

$n_p = |p' \setminus p''| = n' - k$ è la lunghezza della sequenza finale incompatibile di p' rispetto a p'' (*residuo di cammino* di p' rispetto a p'');
$l_p = |p_1| = |p_2| = k$ è la lunghezza della massima sequenza iniziale compatibile di p' e p'';
$m_p = |p'' \setminus p'| = n'' - k$ è la lunghezza della sequenza finale incompatibile di p'' rispetto a p' (*residuo di cammino* di p'' rispetto a p').

più la similarità di stella di tutte le coppie di letterali nelle sequenze iniziali:

$$\mathrm{sf}_p(p', p'') = \mathrm{sf}(n_p, l_p, m_p) + \Sigma_{i=1,\ldots,k} \mathrm{sf}_s(l'_i, l''_i)$$

Table 2. Similarità di stelle

C		E	
l'	S'	l''	S''
$p(X,Y)$	$\{p/2, p/2, r/2, o/2\}$	$p(a,b)$	$\{p/2, p/2, r/2, o/2\}$
$p(X,Z)$	$\{p/2, p/2, o/2\}$	$p(a,c)$	$\{p/2, p/2, o/2\}$
$r(Y,U)$	$\{p/2, o/2, s/2\}$	$r(b,f)$	$\{p/2, o/2, t/2\}$

l'	l''	$(S' \setminus S''), (S' \cap S''), (S'' \setminus S')$	$\mathrm{sf}_s(l', l'')$
$p(X,Y)$	$p(a,b)$	$\emptyset, \{p/2, p/2, r/2, o/2\}, \emptyset$ $(0,4,0)$	3.52
$p(X,Y)$	$p(a,c)$	$\{r/2\}, \{p/2, p/2, o/2\}, \emptyset$ $(1,3,0)$	2.80
$p(X,Z)$	$p(a,c)$	$\emptyset, \{p/2, p/2, o/2\}, \emptyset$ $(0,3,0)$	3.57
$p(X,Z)$	$p(a,b)$	$\emptyset, \{p/2, p/2, o/2\}, \{r/2\}$ $(0,3,1)$	2.82
$r(Y,U)$	$r(b,f)$	$\{s/2\}, \{p/2, o/2\}, \{t/2\}$ $(1,2,1)$	3.24

Table 3. Similarità di cammini

Path No.	C	E
1.	$< p(X,Y), r(Y,U), s(U,V) >$	$< p(a,b), r(b,f), t(f,g) >$
2.	$< p(X,Y), o(Y,Z) >$	$< p(a,b), o(b,c) >$
3.	$< p(X,Z), o(Y,Z) >$	$< p(a,c), o(b,c) >$
4.	$< p(W,X), q(W,W) >$	$< p(d,a), q(d,e) >$

p' p''	$p' \cap p''$	$p' \setminus p''$ $p'' \setminus p'$	$\theta_{p' \cap p''}$	$(n,l,m)_p$ $\mathrm{sf}_p(p', p'')$
C.1	$< p(X,Y), r(Y,U) >$	$< s(U,V) >$	$\{X/a, Y/b, U/f\}$	$(1,2,1)$
E.1	$< p(a,b), r(b,f) >$	$< t(f,g) >$		7.36
C.1	$< p(X,Y) >$	$< r(Y,U), s(U,V) >$	$\{X/a, Y/b\}$	$(2,1,1)$
E.2	$< p(a,b) >$	$< o(b,c) >$		3.97
C.2	$< p(X,Y) >$	$< o(Y,Z) >$	$\{X/a, Y/b\}$	$(1,1,2)$
E.1	$< p(a,b) >$	$< r(b,f), t(f,g) >$		3.97
C.2	$< p(X,Y), o(Y,Z) >$	$<>$	$\{X/a, Y/b, Z/c\}$	$(0,2,0)$
E.2	$< p(a,b), o(b,c) >$	$<>$		7.95

Example 4. Poichè la testa è unica (e quindi può essere associata univocamente), in seguito tratteremo solo i letterali del corpo per mostrare i criteri strutturali. La Tabella 2 riporta i confronti delle stelle per un campione di letterali in C ed E, mentre la Tabella 3 mostra alcuni confronti tra cammini.

Va notato che nessun criterio è di per sé nettamente discriminante, ma la loro cooperazione riesce con successo a distribuire i valori di similarità e a rendere le differenze sempre più chiare man mano che ciascuno viene composto con quelli precedenti.

Una generalizzazione può essere ora calcolata considerando le intersezioni dei cammini in base alla similarità decrescente, e aggiungendo alla generalizzazione parziale generata finora i letterali in comune per ogni coppia ogni volta che questi sono compatibili (vedere l'Algoritmo 2). Ulteriori generalizzazioni possono essere ottenute tramite backtracking. Questo permette anche, nel caso lo si desideri, di tagliare la generalizzazione quando si raggiunge una certa soglia

Algorithm 2 Generalizzazione basata sulla similarità

Require: C: Rule; E: Example
  $P_C \leftarrow paths(C)$; $P_E \leftarrow paths(E)$;
  $P \leftarrow \{(p_C, p_E) \in P_C \times P_E | p_C \cap p_E \neq (<>, <>)\}$;
  $G \leftarrow \emptyset$; $\theta \leftarrow \emptyset$
  while $P \neq \emptyset$ **do**
    $(\overline{p}_C, \overline{p}_E) \leftarrow \text{argmax}_{(p_C, p_E) \in P}(sf(p_C, p_E))$
    $P \leftarrow P \setminus \{(\overline{p}_C, \overline{p}_E)\}$
    $(\overline{q}_C, \overline{q}_E) \leftarrow p_C \cap p_E$
    $\theta_q \leftarrow$ substitution s.t. $q_C = q_E$
    if θ_q compatible with θ **then**
      $G \leftarrow G \cup q_C$; $\theta \leftarrow \theta \cup \theta_q$
    end if
  end while
  return G: generalization between C and E

di lunghezza, assicurando che solo le porzioni con similarità meno significative vengano rimosse.

Example 5. L'intersezione dei cammini con il più alto valore di similarità è C.3/E.3, e quindi la prima generalizzazione parziale risulta $\{p(X, Z), o(Y, Z)\}$, con associazioni $\{X/a, Y/b, Z/c\}$. Subito dopo si considera C.2/E.2, le cui associazioni sono compatibili con quelle attuali, cosicchè esso contribuisce con $\{p(X, Y)\}$ alla generalizzazione (non ci sono nuove associazioni). Poi viene C.1/E.1, che essendo compatibile estende la generalizzazione aggiungendo $\{r(Y, U)\}$ e le associazioni con $\{U/f\}$. A questo punto tocca a C.1/E.2 e quindi a C.2/E.1, che sono compatibili ma ridondanti, e quindi non aggiungono niente alla generalizzazione attuale (né alle associazioni). In seguito viene considerato C.4/E.4, che è compatibile ed estende la generalizzazione e le associazioni attuali, rispettivamente, con $\{p(W, X)\}$ e $\{W/d\}$. Infine, vengono considerati C.3/E.2, C.2/E.3, C.3/E.1 e C.1/E.3, che però vengono scartati perchè le relative associazioni sono incompatibili.

4 Lavori correlati

Molte misure di distanza sono state sviluppate per rappresentazioni attributo-valore, ma pochi lavori hanno affrontato la definizione di misure di similarità o di distanza per descrizioni del prim'ordine. In [4], viene proposta una misura di distanza basata sulla teoria delle probabilità applicata alle componenti della formula. Rispetto a questa misura, la nostra funzione non richiede assunzioni o ipotesi semplificative (come per esempio indipendenza statistica, mutua esclusione), il che facilita la gestione delle probabilità, né alcuna conoscenza *a priori* del linguaggio di rappresentazione (come per esempio i tipi del dominio). Inoltre, il nostro approccio non necessita che l'utente imposti dei pesi sull'importanza dei predicati e non è basato sulla presenza di relazioni obbligatorie, come per la sottoclausola $G1$ in [4].

Molti sistemi di apprendimento supervisionato dimostrano l'importanza di una misura di similarità. Per esempio, *KGB* [1] usa una funzione di similarità, parametrizzata dall'utente, per guidare la generalizzazione; le nostre idee di similarità caratteristica e relazionale sono molto simili a questa, ma il calcolo della similarità è più immediato. Mentre *KGB* non può gestire informazione negativa nelle clausole, il nostro approccio può essere facilmente esteso a questo scopo. Il classificatore k-Nearest Neighbor *RIBL* [2] è basato su una versione modificata della funzione proposta in [1]. L'idea di base è che la similarità di oggetti dipende dalla similarità dei valori dei loro attributi (la similarità della loro dimensione), e, ricorsivamente, dalla similarità degli oggetti a questi connessi. Tale propagazione pone il problema della indeterminazione (incertezza) nelle associazioni, che il nostro approccio evita grazie al diverso approccio strutturale.

[9] presenta un approccio per l'induzione di distanze su esempi in logica del prim'ordine, che dipende dal pattern che discrimina i concetti-obiettivo. Si selezionano k clausole e i valori di verità che esprimono se la clausola copre o meno l'esempio sono utilizzati come k componenti di una distanza tra gli esempi nello spazio $\{0,1\}^k$. [6] organizza i termini in una gerarchia in base all'importanza e propone una distanza tra termini basata su interpretazioni ed una funzione che mappa ogni espressione semplice in un numero naturale. [7] presenta una funzione di distanza tra atomi basata sulla differenza rispetto alla loro lgg e usa la funzione per calcolare distanze tra clausole. La funzione consiste in una coppia: la prima componente estende la distanza in [6] ed è basata sulle differenze tra i funtori nei due termini, mentre la seconda componente è basata sulle differenze nelle occorrenze delle variabili e permette di differenziare casi che la prima componente non può discriminare.

5 Sperimenazioni

All scopo di verificare se i criteri proposti siano in grado di produrre indizi significativi sulla similarità tra due strutture, la procedura di generalizzazione guidata dalla similarità è stata confrontata con una versione precedente (non guidata) utilizzata nel sistema INTHELEX [3]. Il sistema è stato impostato in maniera tale che, quando la prima generalizzazione prodotta dalla procedura guidata non era coerente con tutti gli esempi negativi, il sistema poteva cercare generalizzazioni più specifiche in backtracking[1]. Per valutare l'accuratezza predittiva è stata usata la 10-fold cross-validation.

Un primo confronto ha riguardato il classico dataset ILP della Mutagenesi, sul quale il sistema ha raggiunto un'accuratezza predittiva leggermente migliore (87%) rispetto alla versione non guidata (86%) sfruttando solo il 30% del tempo di esecuzione (4035 secondi anzich 13720). E' importante notare che la prima

[1] Per evitare che il sistema impiegasse troppo tempo in generalizzazioni difficili, è stato impostato in maniera da abbandonare la generalizzazione attuale e cercare di generalizzare un'altra clausola (se presente) non appena venivano trovate in backtracking 500 (sotto-)generalizzazioni ridondanti oppure ne venivano calcolate 50 nuove ma incoerenti

Table 4. Risultati sperimentali

		Ratio	Time (sec.)	Cl	Gen	Exc^+	$Spec^+$	$Spec^-$	Exc^-	Acc
Classificazione	SF	90.52	579	8	47(+0)	0	2	0	0	0.94
	I	70.22	137	7	33(+100)	0	1	1	1	0.97
	S80	73.63	206	7	33(+13)	0	0	1	1	0.97
Etichettatura	SF	91.09	22220	36	180(+0)	0	8	3	3	0.89
	I	68.85	33060	39	137(+5808)	0	15	11	12	0.93
	S80	71.75	15941	54	172(+220)	0	14	8	2	0.93

generalizzazione trovata era sempre corretta, e quindi non è stato mai necessario fare backtracking per cercare altre alternative, mentre la versione non guidata ha dovuto calcolare 5815 ulteriori generalizzazioni per gestire quei casi in cui la prima generalizzazione trovata era incoerente con gli esempi pregressi. Questo conferma che la misura di similarità è in grado di guidare la corretta identificazione delle sotto-parti corrispondenti nelle descrizioni chimiche.

Altre sperimentazioni sono state condotte su un dataset[2] contenente descrizioni del layout della prima pagina di 122 articoli scientifici appartenenti a 4 classi diverse. Da questi sono stati ricavati 488 esempi posititi/negativi per apprendere regole di classificazione dei documenti, e 1488 esempi per apprendere regole che identificano 12 tipi di (etichette per le) componenti logiche dei documenti (per esempio, titolo, autore, abstract etc). I risultati sono riportati nella Tabella 4.

La prima questione è se la funzione di similarità proposta fosse in grado di guidare la procedura di ricerca verso l'identificazione delle sotto-parti appropriate da mettere in corrispondenza nelle due descrizioni che si stavano confrontando. Poichè l'associazione corretta non è nota, questo può essere valutato solo indirettamente. Un modo per far ciò è valutare il fattore di compressione della generalizzazione guidata, ossia la porzione di letterali nelle clausole da generalizzare che è preservata dalla generalizzazione. Infatti, poiché ogni generalizzazione in INTHELEX deve essere un sottoinsieme di ciascuna delle clausole da generalizzare (a causa della assunzione di Object Identity), più letterali vengono preservati da queste clausole dalla generalizzazione, meno generale sarà la generalizzazione. Più formalmente, valutiamo la compressione come il rapporto tra la lunghezza della generalizzazione e quella della clausola più corta da generalizzare: più grande è questo valore, più sicuri si può essere che le associazioni corrette siano state trovate grazie ai criteri e alla formula di similarità. Ovviamente, più grande è la differenza in lunghezza tra due clausole da generalizzare, più grande sarà l'incertezza presente, e quindi più difficile sarà identificare appropriatamente le parti corrispondenti tra loro. È interessante notare che sul dataset dei documenti la generalizzazione guidata dalla similarità (SF) ha preservato in media più del 90% dei letterali della clausola più corta, con un massimo di 99,48% (193 letterali su 194, contro un esempio da 247) e solo 0,006 di varianza. Di conseguenza, ci si può aspettare che le generalizzazioni

[2] `http://lacam.di.uniba.it:8000/systems/inthelex/index.htm#datasets`

prodotte siano effettivamente le meno generali (o le approssimino molto). Per verificarlo, invece di calcolare la vera generalizzazione meno generale per ogni generalizzazione effettuata e confrontarla con quella prodotta dalla procedura guidata, il che sarebbe stato computazionalmente pesante e avrebbe richiesto di modificare il comportamento del sistema, si è calcolato quanti backtracking il sistema doveva calcolare per raggiungerne una più specifica. Il risultato è stato che, entro il limite dato, non è stata trovata nessuna altra generalizzazione, il che suggerisce che la prima generalizzazione trovata è probabilmente molto vicina a (ed in effetti molto spesso è) la generalizzazione meno generale. Da notare che applicando della formula di similarità di Tverski [10] (che rappresenta lo stato dell'arte nell'attuale letteratura) allo stesso problema, si sono ottenute sempre generalizzazioni più corte di quelle ottenute utilizzando la nostra formula.

Comunque, essendo tali generalizzazioni molto specifiche, la loro accuratezza predittiva risulta più bassa di quelle ottenute dall'algoritmo INTHELEX non guidato (I), il che probabilmente è dovuto alla necessità di un maggior numero di esempi per convergere a definizioni predittive oppure all'overfitting. Per questo motivo, la procedura di generalizzazione guidata dalla similarità è stata impostata per scartare almeno il 20% dei letterali della più corta clausola originale (S80): questo ha portato alla stessa accuratezza predittiva di I, ed ha ridotto drasticamente il tempo di esecuzione rispetto alla versione senza vincoli (ed anche rispetto ad I nel caso del compito di etichettatura). Anche il numero delle generalizzazioni decresce, sebbene al costo di uno sforzo maggiore di specializzazione (anche per mezzo di letterali negativi che sono gestiti come spiegato nella sezione precedente), che è in ogni caso più efficace rispetto a I (in particolare per quanto riguarda le eccezioni negative). Nel compito di classificazione il numero delle clausole decresce leggermente, mentre in quello di etichettatura il numero cresce di 15, ma viene equilibrato da 10 eccezioni negative in meno. È importante notare che le generalizzazioni trovate sono ancora molto aderenti agli esempi, come suggerisce il fatto che nelle sperimentazioni di classificazione sono state trovate in backtracking (e testate per verificarne la correttezza) solo 13 generalizzazioni più specifiche (delle quali solo 1 corretta), mentre I ne ha trovate 100, di cui 6 corrette. Nel task di etichettatura, $S80$ ha trovato 220 ulteriori generalizzazioni candidate (di cui 6 corrette) contro 5808 (di cui 39 corrette) di I. Questo dimostra che si potrebbe anche evitare il backtracking con poca perdita in $S80$, ma non in I.

Un altro importante parametro per valutare l'utilità della funzione proposta e della relativa strategia è il tempo di esecuzione. La Tabella 4 mostra che, nel task di etichettatura, utilizzando la funzione di similarità si raggiunge un risparmio che va da 1/3 fino a 1/2 del tempo totale, dell'ordine di ore. Le prestazioni nel task di classificazione sono in ogni caso confrontabili (differenze dell'ordine di decimi di secondo) il che si può probabilmente spiegare col fatto che tale task è molto più facile, per cui resta poco spazio per miglioramenti e il tempo necessario per cacolare la formula annulla il risparmio.

6 Conclusions

Le relazioni, in Logica del Prim'Ordine, portano ad indeterminazione nel mappare porzioni di una descrizione su un'altra descrizione. In questo articolo si sono identificati un insieme di criteri ed una formula per il confronto tra clausole, e sono stati utilizzati per guidare una procedura di generalizzazione indicando le sottoparti delle descrizioni che più verosimilmente corrispondo l'una all'altra, basandosi solo sulla loro struttura sintattica.

Stando ai risultati sperimentali, la generalizzazione basata sulla similarità è in grado di catturare le associazioni corrette e può produrre la stessa accuratezza predittiva della versione non guidata, ma producendo teorie più 'pulite' e riducendo drammaticamente la quantità di tempo necessario per convergere (il risparmio di tempo cresce al crescere della complessità del problema).

Come lavoro futuro si potrebbe raffinare la metodologia di calcolo della similarità. Inoltre, la nuova misura di similarità potrebbe essere messa alla prova con altre applicazioni, quali il matching flessibile, il clustering concettuale oppure l'instance-based learning.

References

[1] G. Bisson. Learning in FOL with a similarity measure. In W.R. Swartout, editor, *Proc. of AAAI-92*, pages 82–87, 1992.

[2] W. Emde and D. Wettschereck. Relational instance based learning. In L. Saitta, editor, *Proc. of ICML-96*, pages 122–130, 1996.

[3] F. Esposito, S. Ferilli, N. Fanizzi, T. Basile, and N. Di Mauro. Incremental multistrategy learning for document processing. *Applied Artificial Intelligence Journal*, 17(8/9):859–883, 2003.

[4] F. Esposito, D. Malerba, and G. Semeraro. Classification in noisy environments using a distance measure between structural symbolic descriptions. *IEEE Transactions on PAMI*, 14(3):390–402, 1992.

[5] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[6] S. Nienhuys-Cheng. Distances and limits on herbrand interpretations. In D. Page, editor, *Proc. of ILP-98*, volume 1446 of *LNAI*, pages 250–260. Springer, 1998.

[7] J. Ramon. *Clustering and instance based learning in first order logic.* PhD thesis, Dept. of Computer Science, K.U.Leuven, Belgium, 2002.

[8] C. Rouveirol. Extensions of inversion of resolution applied to theory completion. In *Inductive Logic Programming*, pages 64–90. Academic Press, 1992.

[9] M. Sebag. Distance induction in first order logic. In N. Lavrač and S. Džeroski, editors, *Proc. of ILP-97*, volume 1297 of *LNAI*, pages 264–272. Springer, 1997.

[10] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.