Contrazione e clausole complete

Paolo Liberatore
Dipartimento di Informatica e Sistemistica,
Università di Roma "La Sapienza"
via Salaria 113, 00198 Roma, Italia

1 Introduzione

Per belief revision si intende l'insieme delle operazioni, e delle relative proprietà, che riguardano la modifica di una teoria (insieme di formule ritenute vere) a causa di nuovi fatti.

Gli studi condotti fino a questo momento sulla belief revision riguardano essenzialmente il calcolo proposizionale, ma esistono anche alcuni metodi studiati per teorie del primo ordine. In questo lavoro si fa sempre riferimento al calcolo proposizionale.

Indichiamo con $mod(a)$ l'insieme dei modelli della formula a. Una formula si dice completa se e solo se ha uno e un solo modello.

Si dice clausola una formula costituita da un or di letterali. Chiamiamo clausola completa una clausola in cui compaiono tutte le variabili, ognuna delle quali diretta oppure negata. Si può dimostrare facilmente che il negato di una clausola completa è una formula completa, cioè ha uno e un solo modello, e che il negato di una formula completa è equivalente a una clausola completa.

Si dice clausola di Horn una clausola con al più un letterale diretto.

Nella belief revision si definiscono due operatori binari, detti contrazione e revisione. La contrazione di una teoria K rispetto a una nuova formula a si scrive $K - a$, ed è la teoria K adattata in modo da non implicare più a.

La revisione consiste nell'aggiunta di una nuova formula a ad una teoria K, facendo in modo che il risultato sia consistente. La revisione si scrive $K * a$.

Sono stati fissati dei postulati (vedi per esempio [4]) per queste due operazioni. In particolare per la contrazione sono:

1. $K - a$ è una teoria
2. $a \not\in K - a$
3. $K - a \subseteq K$
4. se $a \not\in K$ allora $K \subseteq K - a$
5. $(K - a) + a$
6. se $a = b$ allora $K - a = K - b$
7. $(K - a) \cap (K - b) \subseteq K - a \wedge b$
8. se $a \not\in K - a \wedge b$ allora $K - a \wedge b \subseteq K - a$

Esistono due formule che permettono di passare da una contrazione a una revisione e viceversa, e sono le identità di Levi e di Harper:

$$K * a = (K - \neg a) + a$$
$$K - a = (K * \neg a) \cap K$$

si può dimostrare che ogni contrazione che soddisfa gli otto postulati è ottenibile attraverso la prima formula da una revisione che soddisfa i suoi otto postulati, e viceversa.

Ci sono due teoremi che permettono di esprimere tutte e sole le contrazioni e le revisioni che soddisfano gli otto postulati.

Il primo metodo ([2]) rappresenta tutte le possibili contrazioni, ed è basato su un ordinamento fra formule $\leq_K$, detto epistemic entrenchment. Gärdenfors e Makinson hanno dimostrato che $K - a = \{b \in K | a <_K a \vee b\}$ è una contrazione che soddisfa gli otto postulati se e solo se $\leq_K$ soddisfa cinque postulati:

1. se $a \leq b$ e $b \leq c$ allora $a \leq c$
2. se $a \vdash b$ allora $a \leq b$
3. per ogni a e b, risulta $a \leq a \wedge b$ oppure $b \leq a \wedge b$
4. (se $K \neq K_\perp$) $a \notin K$ se e solo se $a \leq b$ per ogni b
5. se $b \leq a$ per ogni b, allora $\vdash a$

Inoltre, fissata una contrazione che soddisfa gli otto postulati è sempre possibile trovare un $\leq_K$ che soddisfa i cinque postulati e che la rappresenta.

Gärdenfors e Makinson hanno dimostrato che una volta fissato $\leq_K$ fra le clausole complete, questo risulta fissato anche per tutte le altre formule.

Il secondo metodo ([3]) di rappresentazione, proposto da Katsuno e Mendelzon, è basato su un ordinamento fra modelli $\leq_K$, e serve per esprimere tutte e sole le revisioni che soddisfano i postulati.

Si dimostra che * soddisfa i postulati se e solo se esiste un ordinamento $\leq_K$ transitivo, totale e che soddisfi due condizioni:

1. se $I \in mod(K)$ e $J \notin mod(K)$ allora $I <_K J$
2. se $I \in mod(K)$ e $J \in mod(K)$ allora $I \leq_K J$

Un metodo proposto per effettuare la contrazione è il metodo safe. Il punto di partenza è un ordinamento $\leq$ fra formule.

Per fare $K - a$ devo eliminare da K tutte le dimostrazioni di a. Diciamo che un insieme B è una dimostrazione minima di a se $B \vdash a$ e non esiste nessun $B' \subseteq B$ tale che $B' \vdash a$. Quindi se B è una dimostrazione minima di a, allora togliendo a B anche un solo elemento non si può più dimostrare a.

Il metodo safe consiste nel prendere ognuna delle dimostrazioni minime di a, e togliere da esse l'elemento minimo rispetto a $\leq$.

In questo lavoro si vedrà che sfruttando il fatto che ogni insieme si può rappresentare attraverso l'insieme delle sue clausole complete, è possibile ottenere dei legami fra questi ultimi tre metodi. È stata inoltre trovata una condizione equivalente ai postulati 7 e 8 della contrazione, e un algoritmo efficiente di contrazione (e di revisione).

2 Legami fra metodi

E' possibile trovare un legame fra l'epistemic entrenchment e l'ordinamento fra modelli.

Se indico con $\leq'_K$ un ordinamento fra modelli e $\leq_K$ l'epistemic entrenchment, e con * e - i corrispondenti operatori di revisione e contrazione, allora vale il seguente teorema:

Theorem 1 *Le identità di Levi e di Harper sono valide fra * e - se e solo se risulta valida la seguente relazione fra $\leq'_K$ e $\leq_K$*

$mod(\neg C) \ \leq'_K \ mod(\neg D)$ se e solo se $C \ \leq_K \ D$

(D e C sono clausole complete, e il negato di una clausola completa ha un solo modello)

E' possibile trovare anche un legame fra epistemic entrenchment e safe contraction: se utilizzo lo stesso $\leq$ come epistemic entrenchment o come ordinamento safe, ottengo la stessa contrazione (se $\leq$ soddisfa i cinque postulati dell'epistemic entrenchment). Naturalmente bisogna tenere conto che nella safe contraction $\leq$ è indipendente da K (o si modifica la definizione della safe, facendo sì che $\leq$ possa dipendere da K, oppure si considera K fissato).

3 Una condizione equivalente ai postulati 7 e 8

Se indico con $I(K,a) = Cn(\neg a) \cap K$, è possibile dimostrare che i postulati 7 e 8 della contrazione equivalgono a un'unica condizione:

Theorem 2 *se sono veri i postulati 1-6 allora 7 e 8 equivalgono a:*
78. se $x \notin K - a$ *allora* $K - a \subseteq (K - x) \cup I(K,a)$

Il postulato 78 ha il vantaggio di non contenere nessun riferimento al simbolo $\wedge$, e quindi di poter venire applicata anche a una eventuale estensione della belief revision a altri tipi di logiche, diverse dal calcolo proposizionale.

E' inoltre possibile dare una diversa formulazione dell'espressione che consente di ottenere la contrazione dall'epistemic entrenchment, e cioè

$K - a = \{b \in K | a <_K b\} \cup I(K,a)$

Si noti che il postulato del recover si può esprimere come $I(K,a) \subseteq K - a$ ([4]).

4 Un algoritmo efficiente su base finita

Sfruttando l'idea che ogni teoria si può rappresentare attraverso l'insieme delle sue clausole complete, è risultato un algoritmo che risulta efficiente nel caso in cui si considera la base costituita da clausole di Horn, e di efficienza non inferiore a quella degli altri metodi (Dalal, Satoh, ecc.) nel caso generale.

L'idea centrale è la seguente: rappresentiamo la teoria T attraverso una sua base particolare, che è quella costituita dalle sue clausole complete. La formula a sarà implicata da T se e solo se tutte le clausole di a si trovano nella base di T.

Per eliminare a potrei allora scegliere una sua clausola completa ed eliminarla. Questo, per le teorie chiuse corrisponde alla maxichoice contraction.

Noi vogliamo però una contrazione su base finita. Questo significa che se la teoria T è costituita da un certo numero (finito) di formule $T = \{t_1, , t_n\}$, allora $T - a$ deve essere costituito da un sottoinsieme di T (per la contrazione su teorie chiuse quello che imponevo era che $Cn(T - a) \subseteq Cn(T)$).

Una volta scelta la clausola C di a che voglio eliminare, quello che devo fare e cercare in T tutte le formule t_i che implicano C, e toglierle dalla base.

Quindi, una volta scelta una clausola completa C di a, la contrazione si fa

$T - a = \{t_i \in T | t_i \not\vdash C\}$

Una volta nota la clausola C, la determinazione di $t_i \vdash C$ si può fare facilmente come $mod(\neg C) \subseteq mod(\neg t_i)$. Dato che $\neg C$ ha un solo modello (C è una clausola completa), si tratta di vedere se $\neg t_i$ in tale modello è vero oppure no.

Data una interpretazione, la verifica se una formula è vera o falsa è una operazione che si può fare in tempo lineare. Quindi, una volta che sia stata fissata la clausola C, la realizzazione della nuova base $K - a$ si può fare in tempo lineare (per ogni formula di T devo vedere se è vera nel modello di $\neg C$).

Le operazione computazionalmente difficili sono

1. Determinare se $a \notin Cn(T)$.

2. Scegliere (trovare) una clausola completa di a. Come si è visto, trovare una clausola completa di una formula equivale a trovare una interpretazione in cui tale formula è vera, quindi è un problema NP completo.

3. In genere il confronto di complessità fra i metodi si fa valutando il problema di determinare se $K * p \vdash q$ (o nel nostro caso, $K - p \vdash q$). Quindi ai costi precedenti devo aggiungere il costo della inferenza $T - a \vdash q$.

Si tenga presente che i costi delle operazioni 1 e 3 si hanno per tutte le possibili revisioni (o contrazioni) che soddisfano i primi sei postulati: ogni revisione è un problema sia NP che CoNP hard. Infatti per determinare se a è soddisfacibile basta vedere se $a * \top \vdash a$, mentre per vedere se è una contraddizione basta fare $\top * a \vdash \bot$ (vedi dimostrazione seguita in [5] sulla full meet revision).

Per la contrazione, il fatto che sia una operazione NP hard lo si vede dal fatto che a è soddisfacibile se e solo se $a \in Cn(a) - \bot$. E' anche CoNP hard: infatti per verificare se a è un teorema basta fare $a \in Cn(C) - C$ dove C è una qualsiasi clausola completa.

In ogni caso, per formule generali, non esiste un algoritmo efficiente per trovare una clausola completa (una interpretazione) di una formula.

Il problema si semplifica se si prendono in esame solo teorie e formule costituite da clausole (non necessariamente complete), e in particolare clausole di Horn. Infatti, in questo caso le operazioni 1 e 3, cioè $a \notin Cn(T)$ e $T - a \vdash q$ diventano lineari nelle dimensioni del problema.

Per quello che riguarda la fase due, cioè la determinazione di una clausola completa di a, possiamo distinguere due casi: quello in cui a è una clausola e quello in cui a è il negato di una clausola. Il primo caso corrisponde a quello che si ha quando devo fare $T - a$ con a clausola, il secondo è quello che si ottiene se devo fare $T * a$, in cui a è ancora una clausola. Infatti, la revisione si può fare come $T * a = (T - \neg a) + a$, e quindi occorre contrarre T rispetto a $\neg a$, che è appunto il negato di una clausola.

La determinazione di una clausola completa di una clausola è ovviamente un problema banale (si tratta di "completare" la clausola, cioè di aggiungere ad essa i letterali che mancano).

Anche il problema di trovare una clausola completa del negato di una clausola è un problema computazionalmente facile. Infatti, dato che le clausole complete di una formula negata sono tutte e solo le clausole (complete) che non appartengono alla formula, si può semplicemente determinare una clausola completa che differisca per almeno un letterale dalla clausola di cui voglio eliminare il negato.

Quindi in entrambi i casi il problema della determinazione della clausola completa diventa banale.

Un possibile criterio di scelta di questa clausola completa può essere quello relativo al numero di formule che vanno eliminate. Infatti, si può considerare una cosa positiva, per una contrazione, il fatto di eliminare il minore numero possibile di formule dalla base.

Una euristica di scelta della clausola potrebbe essere la seguente: per ogni letterale memorizzo un valore $uso(letterale)$, che indica il numero di clausole della base in cui tale letterale è presente. Per esempio se ho la base $\{x \vee y, \neg x\}$ allora xy e $\neg x$ hanno uso uno, mentre $\neg y$ ha uso zero.

Nel caso in cui voglio contrarre rispetto a una clausola, quello che devo fare è semplicemente completare la clausola con i letterali meno usati.

Nel caso in cui la formula da eliminare era invece il negato di una clausola, si può procedere nel modo seguente: si prende la clausola composta da tutti i letterali meno usati (cioè si costruisce una clausola completa scegliendo per ogni variabile il letterale meno usato: se tutti e due hanno lo stesso valore se ne sceglie uno a caso). A questo punto la clausola completa può non appartenere alla formula da eliminare.

Questo può accadere solo se la clausola completa appartiene invece al negato della formula da eliminare. Quello che occorre fare è semplicemente invertire uno dei letterali comuni fra la clausola completa e il negato della formula da eliminare.

Fra tutti i letterali converrà ovviamente scegliere quello di uso maggiore (in modo che il negato sia quello di uso minore).

Con questi criteri di scelta si dimostra per esempio che non elimino mai più della metà delle formule che hanno qualche clausola in comune con la formula da eliminare. Questo non toglie che potrebbe esistere una altra clausola completa la cui eliminazione mi porta a cancellare meno formule dalla base.

Questa proprietà migliora se invece di tenere un uso per ogni letterale, tengo un uso per le coppie di letterali (intendo $uso(x_1 x_2)$ poi $uso(x_3 x_4)$ ecc, non tutte le possibili coppie).

In alternativa, potrei considerare ogni formula della base con un valore differente: mentre nel metodo basato sull'uso quando trovavo che un letterale stava in una clausola sommavo 1 al suo uso, ora sommo il valore che ho attribuito alla formula.

Per quello che riguarda gli aspetti teorici, si può notare che questa contrazione non soddisfa il recover. Del resto nessuna contrazione su base finita che soddisfi la condizione $T - a \subseteq T$ può soddisfare il recover, se T è la base finita della teoria (questa condizione si trova per esempio in [1], e del resto è naturale).

Per far sì che una contrazione su basi finita verifichi il recover bisogna fare

$T -' a = (T - a) \cup \{a \rightarrow t | t \in T, t \notin T - a\}$

Questo però può portare ad alcune difficoltà: per esempio se parto da una base T composta esclusivamente da clausole di Horn, anche se a è a sua volta una clausola di Horn, gli elementi $a \rightarrow t$ possono non esserlo.

Per quello che riguarda i modelli, con una dimostrazione simile a quella del teorema 1, si può provare che l'eliminazione di una clausola C dalla teoria $K - a$ equivale ad aggiungere il suo (unico) modello alla teoria. Quindi, dal punto di vista dei modelli, la revisione corrispondente equivale a prendere tutte le formule che contengono quel modello.

Si può dimostrare che il metodo di contrazione che utilizza i letterali di uso minimo soddisfa i postulati 7 e 8 della contrazione. Quindi la corrispondente revisione soddisfa tutti e otto i relativi postulati (anche se la revisione non soddisfa il recover).

References

[1] A. Fuhrman. Theory contraction through base contraction. *Journal of Philosophical Logic*, 20:175–203, 1991.

[2] P. Gärdenfors and D. Makinson. Revision of knowledge systems using epistemic entrenchment. In *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge (TARK-88)*, pages 83–95, 1988.

[3] H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence Journal*, 52:263–294, 1991.

[4] D. Makinson. How to give up: A survey of some formal aspects of the logic of theory change. *Synthese*, 62:347–363, 1985.

[5] B. Nebel. Belief revision and default reasoning: Syntax-based approaches. In *Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning (KR-91)*, pages 417–428, 1991.