人工知能学会研究会資料
SIG-FPAI-B404-15

MaxSAT ソルバを用いた帰納論理プログラミング

Inductive Logic Programming Using a MaxSAT Solver

力 規晃 [1]　越村 三幸 [2]　藤田 博 [2]　長谷川 隆三 [2]
Noriaki Chikara[1], Miyuki Koshimura[2], Hiroshi Fujita[2], and Ryuzo Hasegawa[2]

[1] 徳山工業高等専門学校
[1]Tokuyama College of Technology
[2] 九州大学
[2] Kyushu University

Abstract: Inductive Logic Programming is a method of inductive learning based predicate logic. In this work, we propose the method that after converting a problem of inductive logic programming to MaxSAT problem, which solve the MaxSAT problem using a MaxSAT solver. MaxSAT is an optimization version of SAT which consists in finding an assignment that maximizes the number of satisfied clauses.

1 はじめに

帰納論理プログラミング(ILP)[1]とは述語論理に基づく帰納学習の一手法である．複雑なデータを直接扱い学習を行ことが可能であり，化学物質の毒性予測や自然言語処理，ウェブマイニングなど，非常に多岐にわたる応用が行われている．

一方，MaxSAT[6]とは充足可能性判定(SAT)で全ての節を満たさない場合にできるだけ多くの節を満たすようにSATを拡張したものである．また，近年SATソルバや MaxSAT ソルバが高速化され，様々な探索が必要な問題をSAT問題に変換し，求解することで，多くの成果を上げている[5,6]．本研究では，帰納論理プログラミングの問題を MaxSAT 問題に変換し，MaxSAT ソルバで解く手法を提案する．すでに SAT ソルバ—を用いた帰納論理プログラミングは提案されている[4]．本研究の手法は，先の手法とは異なり，扱う帰納論理プログラミングは制限があり，より限定されているが，さまざまな現実の問題に応用されている Progol[2]や Aleph[3]の探索手法に基づいている．また，ILP の問題を MaxSAT 符号化する際に容量が膨大にならないための前処理を行っている．

2 帰納論理プログラミング

本研究が MaxSAT ソルバで置き換えを行う帰納論理プログラミングを以下に説明する．

連絡先：徳山工業高等専門学校 情報電子工学科
〒745-8585 山口県周南市学園台
E-mail:tikara@tokuyama.ac.jp

2.1 帰納論理プログラミングの枠組み

帰納論理プログラミング(Inductive Logic Programming: ILP)[1]とは一階述語論理に基づいた機械学習の手法である．ILP の一般的な枠組みは，正例 E^+ と負例 E^-，背景知識 B が節集合で与えられ，

$$\begin{cases} B \not\models E^+ \\ B \cup E^- \not\models \square \end{cases} \quad (1)$$

であるとき，つまり，背景知識では正例が説明できず，しかも，背景知識と負例が矛盾しないとき，

$$\begin{cases} H \cup B \models E^+ \\ H \cup B \cup E^- \not\models \square \end{cases} \quad (2)$$

を満たす，つまり背景知識に仮説を加えると正例を説明でき，しかも，背景知識と仮説と負例は矛盾しない仮説 を見つけることである．一般的に帰納論理プログラミングでは仮説 H を求めるために，探索空間を探索する．

2.2 帰納論理プログラミングの探索手法

現実の ILP システムでは集合被覆アルゴリズムを用いて，仮説を複数抽出し，全ての正例を被覆する．

代表的な ILP システムである Progol[2]や Aleph[3]における仮説探索手法は以下の通りである．

B を背景知識，H=∅を仮説，E を正例集合とする．

(1) もし E=∅ならば H を出力する．
(2) e を E 中の最初の事例とする．

(3) e と B より，最弱仮説を生成し探索空間を限定する．
(4) トップダウン探索により，最適な仮説 H'を生成する．
(5) H に H'を加える，H'を B に含める．
(6) B で説明できる正例の集合を E'とする．
(7) E から E'に含まれる正例を取り除く．
(8) (1)へ戻る．

また，Progol や Aleph のアルゴリズムの特徴は，一つの正例に着目し，最弱仮説(most specific hypothesis,MSH)と呼ばれるその正例を説明する仮説の中の最も弱い仮説を求め，その正例が説明できる仮説だけに限定することで探索空間を最初に枝刈りすることである．ここで，仮説が強いとは，より多くのことが説明できることであり, 仮説が弱いとは，より少ないことしか説明できないことである. また，一般的には最弱仮説は無限個のリテラルからなるが，これらの ILP システムはモード宣言を用いて，有限に抑える工夫がなされている．モード宣言は，どの述語が仮説の頭部や本体部に用いられるのかと各引数のタイプと入出力モードを指定する．

2.3 本研究で扱う ILP の制限

本研究で扱う帰納論理プログラミングの問題にはいくつかの制限を設ける．1 つ目の制限は述語の引数は構造を持たないことである．すなわち，引数部分には単純に変数が当てはまるか，定数となる．2 つ目の制限は否定の付いたリテラルは扱わない．そして，3 つ目の制限として背景知識は基礎単位節のみとする．そして最後に，モード宣言は必ず与えるものとする．

3 MaxSAT

ブール式または命題論理式の充足可能性判定(Boolean satisfiability testing;SAT)は，最初に NP 完全性が証明された問題であり，計算機工学や人工知能における数多くの問題の基本となる．近年，高速な SAT ソルバが登場し，様々な問題を SAT 問題にエンコードして求解されている. この SAT を最適化問題に拡張したのが MaxSAT である．

具体的には, CNF 形式で定義された SAT 問題の全ての節が充足不可能な場合，出来るだけ多くの節を満たす解を求めるように SAT を拡張したものである．

MaxSAT のバリエーションとして，各節の重みがすべて 1 である問題は重みなし MaxSAT(unweighted MaxSAT)であり，1 以外の重みをもつ節を含む問題は重み付き MaxSAT(weighted MaxSAT)である. また，必ず満たさなければならない節をハード節と言い，できるだけ満たせばよい(場合によっては満たさなくてもよい)節をソフト節という．このハード節とソフト節が混在した問題を部分 MaxSAT(partial MaxSAT)と呼ぶ．これらを組み合わせた問題で重み付き部分 MaxSAT(weighted partial MaxSAT)と呼ばれるものもある．

MaxSAT についても，QMaxSAT[7]等の高速な MaxSAT ソルバが開発されており，様々な問題に応用されている[8].

本研究では, 2.2 節の集合被覆アルゴリズムの探索により最適な仮説を求める部分に MaxSAT を用いる. したがって，集合被覆アルゴリズム内の繰り返しにより，複数回 MaxSAT ソルバ実行し，それぞれ仮説のルールを求める．このとき，できるだけ多くの正例を満たすできるだけ短いルール求めるため，MaxSAT を用いる．

4 ILP 問題の MaxSAT 符号化

本研究の ILP 問題の MaxSAT 符号化では最弱仮説のリテラル間の因果関係を木構造に変換する前処理を行った後，MaxSAT 符号化を行う．

4.1 前処理

単純に ILP の問題を SAT 符号化することを考えた場合，仮説のパターンの数に比例するオーダで SAT 問題内の節や変数を準備しなければならない．この仮説のパターン数は Progol や Aleph のようにモード宣言を利用して最弱仮説を事前に求めることで仮説の生成を限定した場合でも，最弱仮説内のボディ部リテラルを，求める仮説に利用するか否かの組み合わせの数となり，指数オーダとなる．この場合，比較的簡単な問題でも SAT 符号化により巨大な SAT 問題となるため，SAT 符号化することが容量的に不利となる．

本研究では仮説のリテラルの因果関係をグラフで表現した際に木構造に変換する前処理を行う．

まず，ある仮説 H を精密化演算により，本体部に述語 p3 のリテラルを追加して仮説 H'にする例について考える．

仮説 H: head(A):-p1(A,B),p2(A,C),p3(A,D)
↓精密化
仮説 H': head(A):-p1(A,B),p2(A,C),p3(A,D),p4(B,E,F).

このような精密化演算は ILP の探索時に行われる．ここで p1(A,B)に背景知識のどの基礎単位節が当てはまるかは，共通の変数を持つ p4(B,E,F)に背景知識のどの基礎単位節が当てはまっているかに依存する．仮説 H'でこのような因果関係をグラフで表すと図 1

のようになる．

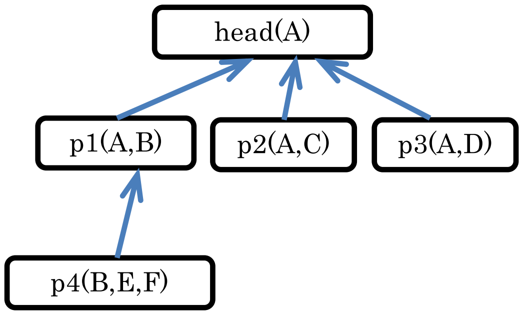

図 1 仮説内のリテラルの因果関係

このような木構造で考えると，あるノードでどの背景知識の節が当てはまるかを考える際に，その直接の子ノードでどの背景知識の節を当てはめているかを局所的に見るだけでよいということになり，この局所的な条件を SAT 符号化することにより，SAT 問題の容量を抑えることに繋がる．

ただし，図 2 の上部のようにグラフのパスがループして木構造にならない場合がある．これは先程の仮説 H'にさらに述語 p5 のリテラルを追加した場合である．述語 p5 の引数に変数 C,E を持ち，変数 C は述語 p2 にも存在し，変数 E は p4 述語にも存在している．

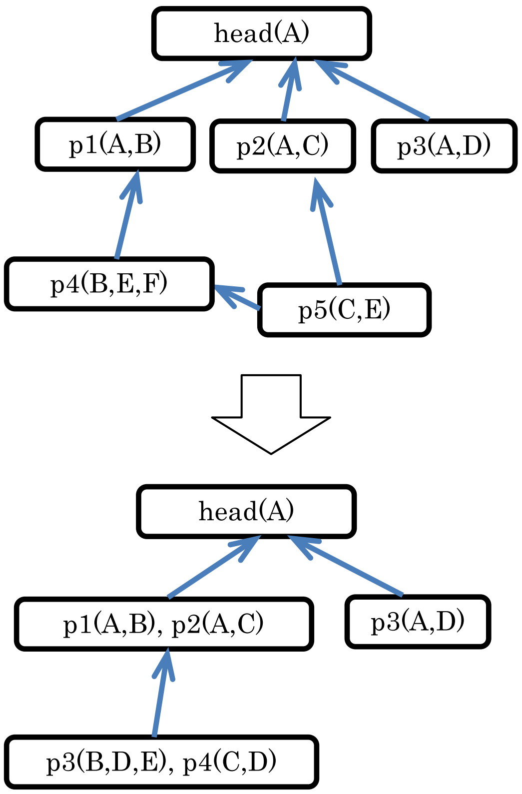

図 2 仮説内リテラルで因果のループがある場合の木構造への変換

このような場合は本来の木構造での階層ごとに述語をグループ化し，１つのノードとして扱う．これを行ったのが，図 2 の下部となる．これにより，直接の子ノードの背景知識の節の当てはめのみがそのノードの背景知識の節の当てはめのみに依存する関係が保たれる．

4.2 MaxSAT 符号化

ここでは表 1 のような SAT 変数を MaxSAT 符号化(エンコード)して作成する MaxSAT 問題で用いる．

表 1 ILP 問題の MaxSAT 符号化で用いる変数

変数	意味
$a(i,bki)$	i 番目の述語グループで bki 番目の背景知識の単一化パターンが成り立つ.
$e(ia,bki,ib)$	ia 番目の述語グループと ib 番目の述語グループで引数が仮説上で同一の変数がある場合 ib 番目の述語グループで $a(ia,bki)$ が成り立つ条件が整ったことを表す.
$r(i,j)$	i 番目の述語グループの j 番目の述語がルールに現れることを表す.
$rs(i,bi)$	i 番目の述語グループに属する $r(i,j)$ の値が bi 番目のパターンとなる.
$c(i,j,k,ci)$	i 番目の述語グループの j 番目の述語の k 番目の引数が ci 番目の値となる.
$cp(head,k,ci)$	仮説(ルール)のヘッド部の k 番目の引数が ci 番目の値となる
$p(ex)$	ex 番目の正例が仮説に当てはまることを表す.
$sw(ex)$	ex 番目の正例が有効であることを表す.
$n(ex)$	ex 番目の負例が仮説に当てはまることを表す

本研究の MaxSAT 符号化により生成される節を次に示す．ただし，ソフト節に関しては明記をし併せて重みを書く，明記がないものはハード節とする．

(1) 述語グループ内の $r(i,j)$ が特定の値のパターンである時がそのパターンに対応する $rs(i,bi)$ が成り立つ

$\{\neg r(i,j) \vee \neg rs(i,bi(i,R(i,1),\ldots,\neg r(i,j),\ldots,R(i,m(i)))) \mid$
$1 \le i \le l, R(i,jx)\ is\ (r(i,jx)=True)\ or\ (r(i,jx)=False),$
$1 \le jx \le m(i), jx \ne j\}$

$\{\neg R(i,1) \vee \ldots \vee \neg R(i,m(i))) \vee rs(i,bi(i,R(i,1),\ldots,R(i,m(i)))) \mid$
$1 \le i \le l, R(i,j)\ is\ (r(i,j)=True)\ or\ (r(i,jx)=False),$
$1 \le j \le m(i)\}$

$\{r(i,j) \vee \neg rs(i,bi(i,R(i,1),\ldots,r(i,j),\ldots,R(i,m(i)))) \mid$
$1 \le i \le l, R(i,jx)\ is\ (r(i,jx)=True)\ or\ (r(i,jx)=False),$
$1 \le jx \le m(i), jx \ne j\}$

(2) $rs(i,bi)$により述語グループ内で有効な述語と無効な述語がある．これによりその$rs(i,bi)$に対応したいくつか特定の$a(i,bki)$のみが成り立つ．

$\{rs(i,bi(i,R(i,1),\ldots,R(i,m(i)))) \vee$
$\neg a(i,bki(i,\ from,z,R(i,1),\ldots,R(i,m(i)))) \mid$
$1 \le i \le l, R(i,j)\ is\ (r(i,j)=True)\ or\ (r(i,j)=False),$
$1 \le j \le m(i), \forall from, \forall z\}$

$\{\neg rs(i,bi(i,R(i,1),\ldots,R(i,m(i)))) \vee$
$a(i,bki(i,\ from,1,R(i,1),\ldots,R(i,m(i)))) \vee$
$\ldots \vee a(i,bki(i,\ from,zm,R(i,1),\ldots,R(i,m(i)))) \mid$
$1 \le i \le l, R(i,j)\ is\ (r(i,j)=True)\ or\ (r(i,j)=False),$
$1 \le j \le m(i), \forall from, 1 \le z \le zm\}$

(3) 仮説の本体部の述語の定数となるべき引数は$c(i,j,k,ci)$が成立することによりある特定の定数になる．この時，$a(i,bki)$はその定数に対応したものだけが成り立つ．また， 定数引数は同時に２つの値にならない．

$\{\neg a(i,bki) \vee c(i,j,k,ci) \mid$
$arg(i,j,k)\ is\ ConstArg,$
$atom(i,j,k,ci)\ exists\ arg(i,j,k)\ assigns\ atom(i,j,k,ci)$
$if\ \ pred(i,j)\ assigns\ bk(bki)\ \}$

$\{\neg c(i,j,k,ci(a)) \vee \neg c(i,j,k,ci(b)) \mid$
$arg(i,j,k)\ is\ ConstArg, ci(a) \ne ci(b)\}$

(4) 仮説の頭部の述語の定数引数は$cp(head,k,ci)$が成立することによりある特定の定数になる．この時，$sw(ex)$はその定数に対応したものだけが成り立つ．また， 定数引数は同時に２つの値にならない．

$\{\neg sw(ex) \vee e(head,ex,i) \mid$
$The\ Assignents\ of\ \ head(ex)\ and\ \ preds(i)$
$has\ common\ value, ex \in PosEx\}$

$\{\neg cp(head,k,ci(a)) \vee \neg cp(head,k,ci(b)) \mid$
$arg(head,k)\ is\ ConstArg\}$

(5) 仮説の頭部述語の正例割当$sw(ex)$に対応する$e(head,ex,ib)$の全てが成り立てば， $sw(ex)$が成り立つ．逆に， $sw(ex)$と対応する$e(head,ex,ib)$のうち１つでも成立しなければ， $sw(ex)$は成立しない．

$\{\neg sw(ex) \vee e(head,ex,i) \mid$
$The\ Assignents\ of\ \ head(ex)\ and\ \ preds(i)$
$has\ common\ value, ex \in PosEx\}$

$\{sw(ex) \vee \neg e(head,ex,i(1)) \vee$
$\ldots \vee \neg e(head,ex,i(g(ex))) \mid$
$The\ Assignents\ of\ \ head(ex)\ \ and\ \ preds(i(x))$
$have\ common\ value., 1 \le x \le g(ex), ex \in PosEx\}$

(6) 仮説の頭部述語の負例割当$n(ex)$に対応する$e(head,ex,ib)$の全てが成り立てば，$n(ex)$が成り立つ．逆に，$n(ex)$と対応する$e(head,ex,ib)$のうち１つでも成立しなければ， $sw(ex)$は成立しない．

$\{\neg n(ex) \vee e(head,ex,i) \mid$
$The\ Assignents\ of\ \ head(ex)\ and\ \ preds(i)$
$has\ common\ value, ex \in NegEx\}$

$\{n(ex) \vee \neg e(head,ex,i(1)) \vee \ldots \vee \neg e(head,ex,i(g(ey))) \mid$
$The\ Assignents\ of\ \ head(ex)\ \ and\ \ preds(i(x))$
$have\ common\ value., 1 \le x \le g(ey), ex \in NegEx\}$

(7) $e(head,ex,ib)$に関連する$a(ib,bki)$が一つでも成り立っていれば，$e(head,ex,ib)$は成り立つ．$rs(ib,bi)$の状態で$a(ib,bki)$が有効であるにもかかわらずその有効な$a(ib,bki)$の全てが成り立たないとき，$e(head,ex,ib)$は成り立たない．

$\{e(head,ex,i) \vee \neg a(i,bki) \mid$
$The\ Assignments\ of\ \ head(ex)\ and\ a(i,bki)$
$have\ common\ value., ex \in PosEx \cup NegEx\}$

$\{\neg e(head,ex,i) \vee \neg rs(i,bi(R(i,1),\ldots,R(i,m(i)))) \vee$
$a(i,bki(i,\ fromHead(ex),1,R(i,1),\ldots,R(i,m(i)))) \vee \ldots \vee$
$a(i,bki(i,\ fromHead(ex),zm,R(i,1),\ldots,R(i,m(i)))) \mid$
$\forall e(head,ex,i),$
$The\ Assignents\ of\ \ head(ex)\ and$
$a(i,bki(i,\ fromHead(ex),z,(R(i,1),\ldots,R(i,m(i)))))$
$has\ common\ value,$
$If\ \ rs(i,bi(i,R(i,1),\ldots,R(i,m(i)))) = True\ then\ \ possible$
$a(i,bki(i,\ fromHead(ex),z,(R(i,1),\ldots,R(i,m(i))))) = True,$
$1 \le z \le zm,$
$R(i,j)\ is\ (r(i,j)=True\ or\ \ r(i,j)=False)\ except\ \ for$
$(R(i,1),\ldots,R(i,m(i)))$
$=(r(i,1)=False,\ldots,r(i,m(i))=False),$
$ex \in PosEx \cup NegEx\}$

(8) 仮説の本体部述語グループの背景知識割当$a(ia,bki)$に対応する$e(ia,bki,ib)$の全てが成り立てば$a(ia,bki)$は成り立つ．逆に$a(ia,bki)$と対応する

$e(ia,bki,ib)$ のうち１つでも成立しないければ，$a(ia,bki)$ は成立しない．

$$\{\neg a(ia,bki) \vee e(ia,bki,ib) \mid \text{The Assignents of } a(ia,bki) \text{ and } preds(ib) \text{ has common value}\}$$

$$\{a(ia,bki) \vee \neg e(ia,bki,i(1)) \vee \ldots \vee \neg e(ia,bki,i(g(ia,bki))) \mid \text{The Assignents of } a(ia,bki) \text{ and } preds(i(x)) \text{ have common value.}, 1 \le x \le g(ia,bki)\}$$

(9) $e(ia,bki,ib)$ に関連する $a(ib,bki)$ が一つでも成り立っていれば，$e(ia,bki,ib)$ は成り立つ．$rs(ib,bi)$ の状態で $a(ib,bki)$ が有効であるにもかかわらずその有効な $a(ib,bki)$ の全てが成り立たないとき，$e(ia,bki,ib)$ は成り立たない．

$$\{e(ia,bkia,ib) \vee \neg a(ib,bkib) \mid \text{The Assignments of } a(ia,bkia) \text{ and } a(ib,bkib) \text{ have common value.}, \forall e(ia,bkia,ib)\}$$

$$\{\neg e(ia,bkia,ib) \vee \neg rs(ib,bi(R(ib,1),\ldots,R(ib,m(ib)))) \vee a(ib,bki(ib, fromBody(ia,bkia),1,R(ib,1),\ldots,R(ib,m(ib)))) \vee \ldots \vee a(ib,bki(ib, fromBody(ia,bkia),zm,R(i,1),\ldots,R(ib,m(ib)))) \mid \forall e(ia,bkia,ib), \text{The Assignents of } a(ia,bkia) \text{ and } a(i,bki(i, fromBody(ia,bkia),z,(R(ib,1),\ldots,R(ib,m(ib))))) \text{ has common value}, 1 \le z \le zm, \text{If } rs(ib,bi(ib,R(ib,1),\ldots,R(ib,m(ib)))) = True \text{ then possible } a(ib,bki(ib, fromBody(ia,bkia),z,(R(ib,1),\ldots,R(ib,m(ib))))) = True, R(ib,j) \text{ is } (r(ib,j) = True \text{ or } r(ib,j) = False) \text{ except for } (R(ib,1),\ldots,R(ib,m(ib))) = (r(ib,1) = False,\ldots,r(ib,m(ib)) = False)\}$$

(10) 最弱仮説の木構造の末端では $rs(ib,bi)$ と $c(i,j,k,ci)$ の条件に問題なければ，$a(i,bki)$ は成立する．

$$\{\neg rs(i,bi(i,R(i,1),\ldots,R(i,m(i)))) \vee \neg c(i,j(1),k(1),ci(1)) \vee \ldots \vee \neg c(i,j(mc),k(mc),ci(mc)) \vee a(i,bki(i,back(i),z,R(i,1),\ldots,R(i,m(i)))) \mid Node(i) \text{ is } Leaf, R(i,j) \text{ is } (r(i,j) = True) \text{ or } (r(i,j) = False), 1 \le j \le m(i), \forall back(i), \forall z, arg(i,j(y),k(y)) \text{ is } ConstArg, \text{If } a(i,bki(i,back(i),z,R(i,1),\ldots,R(i,m(i)))) = True \text{ Then } k(y)th \text{ Argument of } Pred(i,j(y)) \text{ is } ci(y)th \text{ const value.} 1 \le y \le mc, mc \ge 0\}$$

(11) 最弱仮説の木構造の末端以外のノードで $rs(ib,bi)$ の値により，末端となる場合，$c(ia,j,k,ci)$ と $rs(ia,bi)$ に問題なければ，$e(ia,bki,ib)$ を成立される．仮説頭部の $e(head,ex,ib)$ についても同様の扱いをする．

$$\{e(ia,bkia,ib) \vee \neg rs(ia,bi(ia,bkia)) \vee \neg c(ia,j(1),k(1),ci(1)) \vee \ldots \vee \neg c(ia,j(mc),k(mc),ci(mc)) \vee \neg rs(ib,bi(ib,False,\ldots,False)) \mid \forall e(ia,bkia,ib) \text{ If } a(i,bki) = True \text{ then } k(y)th \text{ Argument of } Pred(i,j(y)) \text{ is } ci(y)th \text{ const value.} 1 \le y \le mc, mc \ge 0\}$$

$$\{e(head,ex,i) \vee \neg cp(ia,j(1),k(1),ci(1)) \vee \ldots \vee \neg cp(ia,j(mc(head)),k(mc(head)),ci(mc(head))) \vee \neg rs(i,bi(i,False,\ldots,False)) \mid \forall e(head,ex,i) \text{ If } p(ex) = True \text{ or } n(ex) = True \text{ then } k(y)th \text{ Argument of } Pred(head,j(y)) \text{ is } ci(y)th \text{ const value.} 1 \le y \le mc(head), mc(head) \ge 0, ex \in PosEx \cup NegEx\}$$

(12) ルールをできるだけ短くするため，$r(i,j)$ はできるだけ成り立たない．(ソフト節,重み:1)

$$\{\neg r(i,j) \mid 1 \le i \le l, 1 \le j \le m(i)\}$$

(13) ルールはできるだけ多くの正例が当てはまるように，$p(ex)$ はできるだけ成り立つ．
(ソフト節,重み:最弱仮説のボディ部のリテラル数+1)

$$\{p(ex) \mid ex \in PosEx\}$$

(14) 負例は否定される

$$\{\neg n(ey) \mid ey \in NegEx\}$$

(15) $sw(ex)$ が成り立たないときは $p(ex)$ は成立しない．

$$\{\neg p(ex) \vee sw(ex) \mid ex \in PosEx\}$$

(16) 集合被覆アルゴリズムで着目した正例は無条件で成立するものとする

$$\{p(ex(xi)) \mid ex(xi) \text{ is specific}\}$$

5 まとめ

本研究では，MaxSAT を用いる帰納論理プログラミングを行う手法を提案した．前処理として木構造に変換するため，ILP の問題を変換した際に生成される MaxSAT 問題の容量は比較的小さくなる．今後の課題はいくつかの実用的な ILP の問題を本手法で試し，比較検討を行うことである．

謝辞

本研究の一部は JSPS 科研費 25330085 の助成を受けたものである．

参考文献

[1] 古川康一，尾崎知伸，植野研: 帰納論理プログラミング，共立出版, (2001)

[2] S. Muggleton: Inverse Entailment and Progol, New Generation Computing Journal, Vol. 13, pp. 245-286, (1995)

[3] Ashwin Srinivasan: The Aleph Manual, http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/ ,(1999)

[4] 近藤誠一，山本章博: SAT ソルバーを用いた帰納論理プログラミング, 人工知能学会研究会資料 (SIG-FAI-B104),pp.75-80,(2012)

[5] 井上克己，田村直之: SAT ソルバ―の基礎，人工知能学会誌 25(1),pp.57-67,(2010)

[6] 平山勝敏，横尾真: *-SAT:SAT の拡張，人工知能学会誌 25(1),pp.105-113,(2010)

[7] Miyuki Koshimura, Tong Zhang, Hiroshi Fujita, Ryuzo Hasegawa: QMaxSAT: A Partial Max-SAT Solver, JSAT, Vol.8 pp. 95-100 (2012)

[8] Xiaojuan Liao, Hui Zhang, Miyuki Koshimura, Hiroshi Fujita, Ryuzo Hasegawa:Using MaxSAT to Correct Errors in AES Key Schedule Images, Proceedings of IEEE 25th International Conference on Tools with Artificial Intelligence(ICTAI2013),pp.284-291,(2013)