帰納論理プログラミングのSATによる表現

近藤 誠一[1]　山本 章博[2]

[1] 京都大学 工学部 情報学科

[2] 京都大学 情報学研究科

June 23, 2012

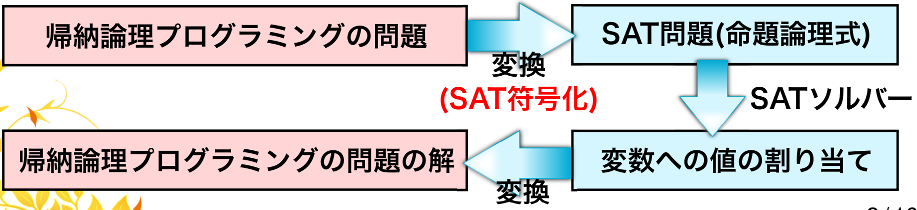
研究の概要

背景

- 機械学習手法の 1 つとして帰納論理プログラミングが提案されてきた
  - 一階述語論理に基づくため離散データに適用しやすく，明示的な規則が得られる
  - 問題が一般に NP 困難に属するため解を求めるのが難しい
- 近年になって SAT ソルバーの性能が劇的に向上している

結果

- SAT ソルバーを帰納論理プログラミングに応用した
- 分類問題において SVM との比較を行なった



SAT ソルバー

SAT ソルバー

与えられた命題論理式が充足可能であるか否かを判定するプログラム
充足可能ならば変数への真偽値の割り当てを 1 つ出力する

$$
\begin{aligned}
&(x_1 \vee \neg x_2) \\
\wedge&(x_2 \vee x_3) \\
\wedge&(\neg x_1 \vee \neg x_2 \vee x_3) \\
\wedge&(\neg x_1 \vee \neg x_2 \vee \neg x_3)
\end{aligned}
\quad \longrightarrow
$$

充足可能

$$
\begin{aligned}
x_1 &= \mathbf{F}, \\
x_2 &= \mathbf{F}, \\
x_3 &= \mathbf{T}.
\end{aligned}
$$

```
p cnf 3 4
 1 -2   0
 2   3  0
-1 -2   3 0
-1 -2 -3 0
```

- 1 行目に変数の数と節の数を記述
- 2 行目以降の各行は節を表現している
  - **0** は各節の終わりを表す
  - 正の整数は変数を表す
  - マイナス (−) は否定 (¬) を表す

帰納論理プログラミング (ILP)

ILP 問題の設定

Input 3 つの一階述語論理における節集合:
背景知識 $\mathcal{B}$, **正例** $\mathcal{E}^+$, **負例** $\mathcal{E}^-$

Output 次の条件を満たす節集合 Σ (**仮説**)

- すべての $C \in \mathcal{E}^+$ に対して $\Sigma \cup \mathcal{B} \models C$
- すべての $D \in \mathcal{E}^-$ に対して $\Sigma \cup \mathcal{B} \not\models D$

$$
\begin{aligned}
\mathcal{B} &= \emptyset, \\
\mathcal{E}^+ &= \left\{P(0), P(s^2(0)), P(s^4(0))\right\}, \\
\mathcal{E}^- &= \left\{P(s(0)), P(s^3(0))\right\}.
\end{aligned}
\longrightarrow
\Sigma = \left\{ \begin{array}{l} P(0), \\ P(s^2(x)) \leftarrow P(x) \end{array} \right\}
$$

ILP 問題の SAT 符号化

ILP 問題の主張に命題変数の真偽を対応付ける

- 節 C に対して $C \in \Sigma \Leftrightarrow [C \in \Sigma] = \mathbf{T}$
- 節 D に対して $\Sigma \models D \Leftrightarrow [\Sigma \models D] = \mathbf{T}$

$$
\begin{array}{ll}
[P(0) \in \Sigma] = \mathbf{F}, & [\Sigma \models P(0)] = \mathbf{F}, \\
[P(x) \in \Sigma] = \mathbf{F}, & [\Sigma \models P(s(0))] = \mathbf{T}, \\
[P(s(0)) \in \Sigma] = \mathbf{T}, & [\Sigma \models P(s^2(0))] = \mathbf{T}, \\
[P(s(x)) \in \Sigma] = \mathbf{F}, & [\Sigma \models P(s^3(0))] = \mathbf{T}, \\
[P(s(x)) \leftarrow P(x) \in \Sigma] = \mathbf{T}, & [\Sigma \models P(s^4(0))] = \mathbf{T}, \\
\cdots & \cdots
\end{array}
$$

$$
\Sigma = \left\{ \begin{array}{l} P(s(0)), \\ P(s(x)) \leftarrow P(x) \end{array} \right\}
$$

ILP の SAT 符号化の問題点

- 仮説 Σ の論理的帰結となる基礎単位節をどう計算するか
  ⇒ T_P **オペレータ**を使って計算する
- SAT 符号化のために準備する命題変数が無限個必要である
  - $C \in \Sigma$ の真偽を考慮する必要のある節 C が無限に必要
  - $\Sigma \models D$ の真偽を考慮する必要のある節 D が無限に必要

  ⇒ 仮説 Σ は**縮小確定節**の有限集合とする

T_P オペレータ

T_P オペレータ

- 写像 T_P は 基礎単位節の集合 I から新な基礎単位節の集合を生成する

$$T_P(I) = \{A \mid A \leftarrow B_1 \wedge \ldots \wedge B_n \text{は } P \text{ に含まれる節に代入を行って得られる基礎確定節,} \text{ 各 } B_1, \ldots, B_n \text{ は } I \text{ に含まれる}\}$$

論理プログラム $\Sigma = \{P(0),\ P(s(x)) \leftarrow P(x)\}$ に対して

- $T_\Sigma(\emptyset) = \{P(0)\}$
- $T_\Sigma(T_\Sigma(\emptyset)) = \{P(0), P(s(0))\}$
- $T_\Sigma(T_\Sigma(T_\Sigma(\emptyset))) = \{P(0), P(s(0)), P(s(s(0)))\}$

縮小確定節

論理式 A に対して次の 2 つの式を定義する

$|A|$ ：A に含まれる変数記号，定数記号，関数記号の総数

$o(x, A)$ ：A に含まれる変数 x の個数

縮小確定節

確定節 $A \leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_n$ は，任意の代入 θ と $i = 1, \ldots, n$ に対して

$$|A\theta| > |B_i\theta|$$

が成り立つとき，**縮小確定節**と呼ぶ

確定節 $A \leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_n$ が縮小確定節となるのは，任意の変数 x と $i = 1, \ldots, n$ に対して以下の式が成り立つとき，かつそのときに限る

$$|A| > |B_i|$$
$$o(x, A) \geq o(x, B_i)$$

縮小確定節の有限集合の論理的帰結

A より記号数が小さい B のうち，Σ の論理的帰結であるものを集めて T_P オペレータを適用すれば $\Sigma \models A$ を判定できる

$$\Sigma \models A \Leftrightarrow A \in T_\Sigma\left(\left\{B \;\middle|\; |B| < |A|,\ \ \Sigma \models B\right\}\right)$$

$\Sigma = \left\{P(0), P(s^2(x)) \leftarrow P(x)\right\}$ のとき $\Sigma \models P(s^3(0))$ の真偽を判定する

$|B| < |P(s^3(0))|$ かつ $\Sigma \models B$ を満たす基礎単位節 B は以下の 2 つ

- $P(0)$
- $P(s^2(0))$

$T_\Sigma\left(\left\{P(0), P(s^2(0))\right\}\right) = \left\{P(0), P(s^2(0)), P(s^4(0))\right\}$ であり $\Sigma \not\models P(s^3(0))$

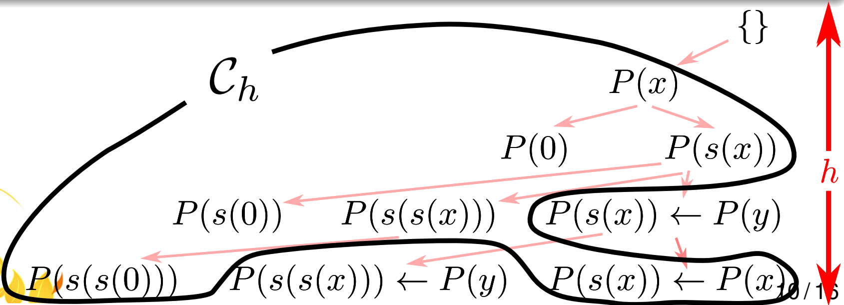
精密化グラフ

空節を用意し以下の操作から１つを選び適用する

- 変数を定数で置き換え
- 変数を別の変数で置き換え
- 変数を関数記号から成る項で置き換え (引数は全て新しい変数)
- リテラルを本体に追加 (引数は全て新しい変数)

空節に手続きを h 回以下の適用して得られる節のうち，
縮小確定節の集合を $\mathcal{C}_h$ とする

仮説 Σ は $\mathcal{C}_h$ の部分集合とする ($\Sigma \subseteq \mathcal{C}_h$)



SAT 問題の作成 (1/2)

ILP の問題の解となる仮説 Σ が満たすべき条件は次の 2 つ

- すべての $A \in \mathcal{E}^+$ に対して $\Sigma \models A$
- すべての $B \in \mathcal{E}^-$ に対して $\Sigma \not\models B$

作成した命題変数を用いて条件を命題論理式で記述する

$$\begin{array}{ll} \text{すべての基礎単位節 } A \in \mathcal{E}^+ \text{ に対して} & [\Sigma \models A], \\ \text{すべての基礎単位節 } B \in \mathcal{E}^- \text{ に対して} & \neg[\Sigma \models B]. \end{array}$$

$\mathcal{E}^+ = \left\{P(0), P(s^2(0))\right\}$, $\mathcal{E}^- = \left\{P(s(0)), P(s^3(0))\right\}$ のときは,

次の命題論理式が真であれば良い

$$[\Sigma \models P(0)] \wedge [\Sigma \models P(s^2(0))] \wedge \neg[\Sigma \models P(s(0))] \wedge \neg[\Sigma \models P(s^3(0))].$$

SAT 問題の作成 (2/2)

基礎単位節 A が仮説 Σ の論理的帰結となる条件を考える

$$\begin{aligned}
\Sigma \models A &\leftrightarrow A \in T_\Sigma\left(\left\{B \,\middle|\, |B| < |A|,\ \Sigma \models B\right\}\right) \\
&\leftrightarrow \bigvee_{C \in C_h} (C \in \Sigma) \wedge A \in T_{\{C\}}\left(\left\{B \,\middle|\, |B| < |A|,\ \Sigma \models B\right\}\right) \\
&\leftrightarrow \bigvee_{C \in C_h} (C \in \Sigma) \wedge \bigvee_{\theta : C\theta = A \leftarrow B_1 \wedge \ldots \wedge B_n} \bigwedge_{i=1,\ldots 2} (\Sigma \models B_i)
\end{aligned}$$

すべての $A \in \mathcal{G}$ に対して

$$[\Sigma \models A] \leftrightarrow \bigvee_{C \in C_h} [C \in \Sigma] \wedge \bigvee_{\theta : C\theta = A \leftarrow B_1 \wedge \ldots \wedge B_n} \bigwedge_{i=1,\ldots 2} [\Sigma \models B_i]$$

$$\Sigma = \left\{C \,\middle|\, [C \in \Sigma] = \mathrm{T}\right\}$$

⇒ 全ての論理式の連言 (∧) を CNF 式に変換して SAT 問題とする

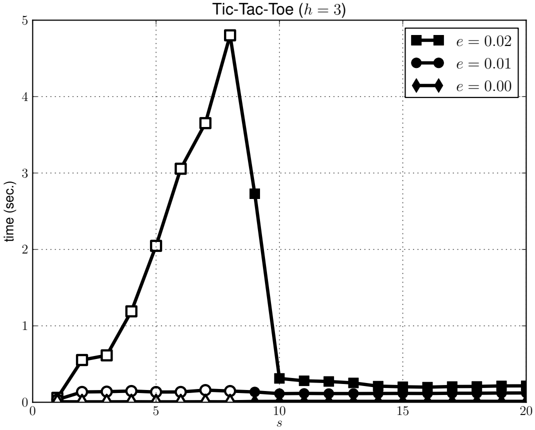
提案手法の評価

- 環境: Mac OS X 10.7.2 (2.66GHz Intel Core i5, 16GB)
- SAT ソルバーには clasp [Gebser 07] を利用
- UCI Machine Learning Repositry から 3 種類のデータセット: Adult, Mushroom, Tic-Tac-Toe を入手し，実験に用いた
- 2 クラス分類問題の精度と時間を計測
- 比較対象として SVM の実装の 1 つである libsvm を利用した
    - パラメータの決定には付属のスクリプトを利用した

			Accuracy		time (sec.)	
	train	test	libsvm	提案手法	libsvm	提案手法
Tic-Tac-Toe	479	479	98.633	100.00	21.093	1.616
Mushroom	4062	4062	99.991	100.00	792.13	233.90
a1a(Adult)	1605	30956	84.087	81.971	159.59	6294
a2a(Adult)	2265	30296	84.512	82.074	324.62	6710

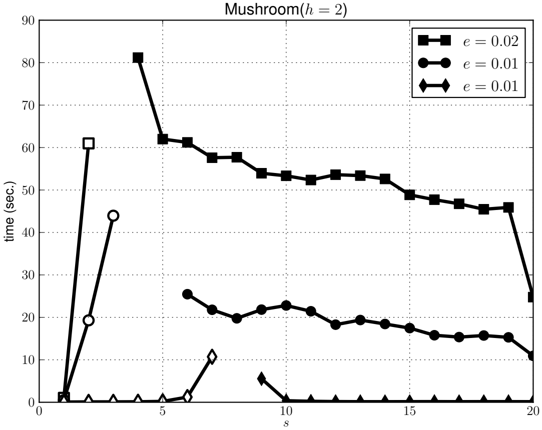
実験結果 (計算時間: Tic-Tac-Toe)

- 横軸は仮説のサイズ $|\Sigma|$ の上限値 s，縦軸は計算時間
- 塗り潰されている点は充足可能，塗り潰されていない点は充足不能
- h の値は $\mathbf{3}$，e の値は $\mathbf{0.00}$，$\mathbf{0.01}$，$\mathbf{0.02}$ の 3 つを試した
- 充足可能と充足不可の境界では計算時間が大幅に増加している



実験結果 (計算時間: Mushroom)

- 横軸は仮説のサイズ $|\Sigma|$ の上限値 s，縦軸は計算時間
- 塗り潰されている点は充足可能，塗り潰されていない点は充足不能
- h の値は 3，e の値は 0.00，0.01，0.02 の 3 つを試した
- 充足可能と充足不可の境界では計算時間が大幅に増加している



まとめと展望

まとめ

- 帰納論理プログラミング問題の解を SAT ソルバーを用いて求めた
- 仮説を縮小確定節の有限集合に限定することで,
  仮説の論理的帰結となる節の計算を命題論理式で表現した
- 分類問題においては SVM との実験を行った

今後の展望

- SAT ソルバーに用いられているテクニックを応用
  (e.g. インクリメンタル探索)
- 節ではなくリテラルに命題変数を対応させる