2H4-OS-03a-3in

BDDに基づくALLSATソルバーを用いたアイテムセットマイニング

Itemset Mining Using BDD-based ALLSAT Solvers

戸田貴久*1　津田宏治*2
Takahisa Toda　Koji Tsuda

*1電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, the University of Electro-Communications

*2東京大学大学院新領域創成科学研究科
Graduate School of Frontier Sciences, The University of Tokyo

Recently, an ALLSAT-based itemset mining framework has been studied. A basic idea is to reduce pattern generation problems in itemset mining to ALLSAT problems and then solve them using ALLSAT solver. This framework offers declarative and flexible representation model, however it is worse than specialized approaches in terms of efficiency. In practice, we have to take care of tradeoff between flexiblity and efficiency. This paper presents an ALLSAT-based method for itemset mining, in which ALLSAT problems are efficiently solved by constructing binary decision diagrams in top-down fashion.

1. はじめに

充足可能性問題 *(SAT)* は，命題論理式が与えられるとき，その充足可能性を判定する問題である．計算機科学のさまざまな領域において現れる基本的な問題である．実用上は，充足可能性だけでなく充足変数割当も知りたいので，ほとんどのSAT ソルバーは充足変数割当を計算する．SAT は NP 完全であることが知られているが，現実的な時間内にできるだけ多くの SAT 問題例を解くための様々な高速化技法が開発されてきた．

すべての充足変数割当を計算する問題 (*ALLSAT*) もまた様々な応用において重要である．古典的な応用例として，非有界モデル検査，到達可能性解析，主項の計算などがある．ほとんどの ALLSAT ソルバーは単解探索の SAT ソルバー上で実装されている．具体的には，充足変数割当が見つかるとき，ソルバーは探索を停止せず，解が発見されなくなるまで探索を継続する．このとき，充足変数割当の再発見を防ぐために，一度発見された充足変数割当（あるいはその一部）の否定をとることで得られる**阻止節**を用いる．全域充足変数割当*1の数は変数の数に関して指数的に増大しうるので，通常は，部分充足変数割当が計算される．

近年，アイテムセットマイニングを ALLSAT に帰着する新しい枠組みが提案されている [Guns et al. 11] [Jabbour et al. 13]．基本的な考え方は，列挙対象が満たすべき条件を論理制約として記述し，CNF に符号化する．その後，既存の ALLSAT ソルバーを用いて充足可能変数割当を列挙し，結果を復号化する．ほとんどの従来手法は，問題に特化したアルゴリズムを設計するので，特定の問題に対しては効率的である．しかし，アイテムセットマイニングでは，制約の追加や削除をしばしば行うので，その度ごとにアルゴリズムを設計しなければならない．他方，上述の ALLSAT を用いた枠組みによれば，既存の ALLSAT ソルバーをそのまま使用するので，アルゴリズムの再設計は不要である．論理制約として問題をモデル化できさえすれば，原理的にはどんな問題でも解けるので，汎用性の高い計算手法といえる．

もちろん汎用性だけでなく効率性も重要であり，両者のトレードオフを考慮する必要がある．この流れの中で ALLSAT ソルバーの本質的な性能向上がますます求められている．近年の SAT ソルバーの飛躍的な性能向上により，SAT ソルバーは瞬時に充足解に収束することができるようになった．しかし，SAT ソルバーは一つの解を発見するだけである．すべての解を効率良く発見するためには，通常の SAT 技法だけでなく，本質的に ALLSAT に関係する高速化技法を開発する必要がある．

本研究では，**論理式キャッシング**（*formula caching*）[Beame et al. 10] に基づく ALLSAT ソルバーを用いたアイテムセットマイニングを提案する．具体的には，我々のソルバーは**論理積標準型** *(CNF)* の形で与えられた論理式から，論理式キャシングに基づいて**二分決定グラフ**（*BDD*）を構築する．ここで，CNF は ALLSAT 問題の標準的な入力形式であり，BDD は論理関数の圧縮データ表現である．BDD はサイクルを含まない有向グラフ（*DAG*）であり，非終端節点は論理変数に対応し，その論理変数への値割当に相当する二つの枝（*0* **枝**と *1* **枝**）が節点から出ている．二つの終端節点（*0* **終端**，*1* **終端**）が存在し，根から 1 終端節点までのパスが充足変数割当を表す．したがって，CNF から BDD 表現を構築することは ALLSAT を解くことに対応する．

2. 論理式キャッシング

論理式キャシングは，論理式の充足変数割当の個数を計算する問題 (*#SAT*) を効率的に解くための技法として提案されている [Bacchus et al. 03] [Majercik et al. 98]. 論理式キャシングと節学習は，DPLL 手続きにおけるメモ化技法（memoization）という意味において同じであるが，本質的に異なるものである：学習節は論理式を偽にする部分変数割当を計算し，メモ化することで，後の探索において同じ過ちを繰り返さない；一方で，論理式キャシングは，探索の各時点における部分問題（計算状態）に対応する論理式をメモ化し，後の探索において現れ

連絡先: 電気通信大学大学院情報システム学研究科
〒 182-8585　東京都調布市調布ヶ丘 1-5-1
E-mail: takahisa.toda@is.uec.ac.jp

*1　全ての変数への値割当

る同じ部分問題の重複計算を回避する．

3. BDD 構築

論理式キャシングと BDD 構築は密接な関係がある．DPLL 手続きにおいて探索の変数順を固定するとき，論理式をメモ化することは OBDD[*2]を DPLL 手続きに従って構築することに対応する．

知識編纂の文脈において，SAT ソルバーを動かしながら，CNF から BDD を構築する手法（**トップダウン法**）が提案されている [Darwiche et al. 04] [Toda et al. 15]．基本的なアルゴリズムの振る舞いは，SAT ソルバーの探索順に従い，BDD 節点を作成していく．ただし，変数の選択順は固定する．このとき，サイズが指数的に増大しないように，同一の部分グラフ同士は共有させる必要がある．トップダウンに構築するので，部分グラフを構築する前に節点同士の共有判定を行わなければならない．このための基本アイディアは，探索の各時点での計算状態を表す論理式をメモ化し，もし同じ計算状態が存在するなら（節点を新規に作成しないで）既存の節点と共有することである．ただし，厳密な共有判定は困難なので，実用上用いられる方法は，同一の計算状態を持つならば等価な節点（すなわち、部分グラフが同じ）であるが，逆は必ずしも成り立たない，という意味での弱い共有判定を用いる．

4. アイテムセットマイニング

頻出アイテムセット列挙問題は以下のように定式化される．$I = \{i_1, \ldots, i_n\}$ をアイテムの集合とし，$\mathcal{T} = \{T_1, \ldots, T_m\}$ をトランザクションの集合とする．ここで，各トランザクションは I の部分集合とする．I の部分集合を**アイテムセット**という．アイテムセット S が閾値 θ に関して**頻出**であるとは，S を含むトランザクションの個数が θ 以上であることを意味する．アイテムセットマイニングにおいてもっとも基本的なタスクの一つは，与えられた閾値に関するすべての頻出アイテムセットを計算することである．

5. ALLSAT への帰着

各 x_j は j 番目のアイテム i_j を選ぶか否かを表す論理変数を表すとき，頻出アイテムセットは以下の二種類の論理制約によって特徴付けられる．

$$R_i: \quad t_i \leftrightarrow \bigwedge_{i_j \notin T_i} \neg x_j \tag{1}$$

$$S: \quad \sum_{1 \leq i \leq m} t_i \leq \theta \tag{2}$$

ここで，一つ目の論理制約は各トランザクションにつき一つ定まる．また，選択されているアイテムを全て含むトランザクションの個数を取り扱うために補助的な論理変数 $t_1, \ldots, t_m$ を導入している．

各制約を CNF に符号化する様々な方法が提案されている [田村ほか 10]．これらの論理制約を CNF に符号化し，ALLSAT ソルバーですべての充足変数割当を計算することで，すべての頻出アイテムセットを求めることができる．

6. 関連研究

トランザクション集合から，各アイテムセットの出現頻度を表す ZDD 表現[*3]を計算し，構築された ZDD の深さ優先探索により頻出アイテムセットを列挙する手法が提案されている [Minato et al. 07]．本研究と同様のアプローチで，論理制約を CNF 符号化し，BDD 演算を繰り返し適用して BDD を構築する手法も提案されている [Cambazard et al. 10]．BDD 演算に基づく BDD 構築法は，実装が容易だが，多くの中間 BDD が生成され，最終的な BDD サイズに比べ中間サイズが爆発的に増大する傾向がある．上述の二つの手法では，各出現頻度ごとに対応するアイテムセットを格納するため，ZDD ベクトル，複数の終端節点のある BDD をそれぞれ用いている．

∗2 完全には簡約化されていない BDD

∗3 BDD の派生データ構造で，疎な集合族を効率的に表現する．

参考文献

[田村ほか 10] 田村直之，丹生智也，番原睦則: 制約最適化問題と SAT 符号化, 人工知能学会会誌 25 巻 1 号（2010 年 1 月）．

[Guns et al. 11] Guns, T., Nijssen, S. and De Raedt, L.: Itemset mining: A constraint programming perspective, Artificial Intelligence, Vol. 175, pp. 1951–1983 (2011).

[Jabbour et al. 13] Jabbour, S., Sais, L. and Salhi, Y.: Boolean Satisfiability for Sequence Mining, In Proc. of the 22nd ACM international conference on Conference on information & knowledge management, pp. 649–658 (2013).

[Bacchus et al. 03] Bacchus, F., Dalmao, S. and Pitassi, T.: Algorithms and Complexity Results for #SAT and Bayesian Inference, In Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science, pp. 340–351 (2003).

[Beame et al. 10] Beame, P., Impagliazzo, R., Pitassi, T. and Segerlind, N.: Formula Caching in DPLL, ACM Transaction on Computation Theory, Vol.1, No.3, Article 9 (2010).

[Majercik et al. 98] Majercik, S.M. and Littman, M.L.: Using Caching to Solving Larger Probabilistic Planning Problems, In Proc. of the 15th National Conference on Artificial Intelligence, pp. 954–959 (1998).

[Cambazard et al. 10] Cambazard, H., Hadzic, T. and Sullivan, B.O.: Knowledge Compilation for Itemset Mining, In Proc. of the 19th European Conference on Artificial Intelligence, pp. 1109–1110 (2010).

[Minato et al. 07] Minato, S. and Arimura, H.: Frequent Pattern Mining and Knowledge Indexing Based on Zero-Suppressed BDDs, In Proc. of the 5th international conference on Knowledge discovery in inductive databases, pp. 152–169 (2007).

[Darwiche et al. 04] Huang, J. and Darwiche, A.: Using DPLL for Efficient OBDD Construction, In Proc. of the 7th international conference on Theory and Applications of Satisfiability Testing, pp. 157–172 (2004).

[Toda et al. 15] Toda, T. and Tsuda, K.: BDD Construction for All Solutions SAT and Efficient Caching Mechanism, In Proc. of the 30th ACM/SIGAPP Symposium On Applied Computing, to appear (2015).