



HOKKAIDO UNIVERSITY

Title	山登りは大変なので沢登りアルゴリズム
Author(s)	宇野, 毅明
Description	ERATO 세미나2010 : No.11. 2010年8月6日
Relation	2010年度科学技術振興機構ERATO湊離散構造処理系プロジェクト講究録. p.74-78.
Issue Date	2011-06
Doc URL	https://hdl.handle.net/2115/48474
Type	conference presentation
File Information	11_all.pdf



ERATO セミナ 2010 - No. 11

山登りは大変なので沢登りアルゴリズム

宇野 毅明

国立情報学研究所

2010/8/6

概要

山登りアルゴリズムは非常に多くの問題に対して採用されているが、極小元を見つける場合、極小元とは関係ない高い山を登り切ってしまうという意味で無駄が多くなり、効率的でなくなる。本発表では、ハイパーグラフ双対化という極小元列挙問題に対して、極小解を山の頂上にして山登りアルゴリズムを適用するというアルゴリズムを提案し、効率的な計算方法も合わせて提案する。計算実験により、本アルゴリズムは多くの問題に対して非常に効率良く動くことを示す。

山登りは大変なので沢登りアルゴリズム

宇野 毅明 (国立情報学研究所
総合研究大学院大学)
村上 啓介 (国立情報学研究所)

2010年8月 ERATOの皆様へ

今日の話

- ハイパーグラフ双対化(極小ヒッティングセット列挙)という問題のアルゴリズムを発表します。
- 山登りアルゴリズムなのですが、もともと谷だったところを山にして登るので、沢登りアルゴリズムです

ヒッティングセット

- 集合族 $F \subseteq 2^E$ (マルチセットでもよし) に対して、 H がヒッティングセットであるとは、任意の $S \in F$ に対して $H \cap S \neq \emptyset$ であること (F はハイパーグラフとも呼ばれる)

- 他を含まないヒッティングセットを極小ヒッティングセットという

$F = \{1,2\}, \{1,3\}, \{2,3,4\}$
 $\{1,3,4\}$ はヒッティングセットだが、極小ではない
 $\{2,3\}$ は極小ヒッティングセット
 F の双対は $\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}$

- F の極小ヒッティングセット全てからなる集合族(ハイパーグラフ)を F の双対と呼び、 $\text{dual}(F)$ と表記する

双対化問題

- 集合族 $F \subseteq 2^E$ の双対を求める問題を双対化(ハイパーグラフ双対化)とよぶ
- 他を含まないヒッティングセットを極小ヒッティングセットという

入力: $F = \{1,2\}, \{1,3\}, \{2,3,4\}$
 出力: $\text{dual}(F) = \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}$

- 他のいろいろな問題と等価、あるいは関連している
- 出力多項式時間のアルゴリズムが存在するかどうか、未解決
- 現在の所の最速は $O(N \log N)$ ※ N は $\text{dual}(F)$ の大きさ
- 現実的にはかなり短時間、 $O(\alpha N)$ 程度、で解ける (頻出パターンマイニングと同じような感じ)

問題の難しさ

- ヒッティングセットを1つ見つけるのは簡単
 - ◀ 全体集合はヒッティングセット
- 極小ヒッティングセットを見つけるのも簡単。
 - ◀ 極小になるまで繰り返し取り除く
- ★ 任意の極小ヒッティングセットは、この方法で見つかる
- しかし、適当に探索したのでは見つけ損ないが発生し、見つけ損ないをなくそうとすると、全ての組合せを探索するようなアルゴリズムになる
- 上手に解のみを渡り歩くのは、非常に難しい

既存手法

$F = \{S_1, \dots, S_m\}, S_i \subseteq \{1, \dots, n\}$ とする。 $F_i = \{S_1, \dots, S_i\}$ とする

- DL法: $\text{dual}(F_i)$ を逐次的に求める

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $\text{dual}(F_1) = \{1\}, \{2\}, \{3\}$
 $\text{dual}(F_2) = \{1\}, \{3\}$
 $\text{dual}(F_3) = \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$

- 任意の $X \in \text{dual}(F_i)$ に対し、 $X \in \text{dual}(F_{i-1})$ か $X \setminus e \in \text{dual}(F_{i-1})$ ($\{e\} = X \setminus F_i$ 、必ず成り立つ)であるので、各 $Y \in \text{dual}(F_{i-1})$ に対して、 $Y \in \text{dual}(F_i)$ であるかチェックし、そうでなければ各 $e \in F_i$ に対して $Y \cup e$ (必ずヒッティングセット)が極小であるかどうかをチェックする (Y を大きさ順に処理して含むものを検索)

DL法続き

- DL法のボトルネックは極小性の計算
 - ★ Y を大きさ順に処理して、各候補に対して、それがより小さなヒッティングセットを含むかどうか検索
- 解が大きく、多くなると時間がかかる
- かわりに、探索を深さ優先的に行うようにしたのがKS法
 - ★ 各候補 $Y' = Y \cup e$ に対して、各 $Y' \setminus f$ がヒッティングセットかどうかチェック

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $\text{dual}(F_1) = \{1\}, \{2\}, \{3\}$
 $\text{dual}(F_2) = \{1\}, \{3\}$
 $\text{dual}(F_3) = \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$

- 解が大きく、データが大きくなると時間がかかる

KS法

- KS法は、逆探索的に解釈できる。親を定義してみる
- $X \in \text{dual}(F_i)$ が $\text{dual}(F_{i-1})$ に含まれる(変わらず極小)
 - X の親は $X \setminus (F_{i-1})$ (どの e も $\text{crit}(X,e) \neq \emptyset$)
- そうでないとき、(F_i に対し、ある e が $\text{crit}(X,e) = \{S_i\}$)
 - X の親は $X \setminus e$ (必ず極小)

$(S_i$ は e に対してクリティカルなので、他の e' に対してはクリティカルでないため、 $\text{crit}(X,e')$ は変化しない)

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $\text{dual}(F_1) = \{1\}, \{2\}, \{3\}$
 $\text{dual}(F_2) = \{1\}, \{3\}$
 $\text{dual}(F_3) = \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$

子供の計算

- 親は唯一的。親 X の子供は、
 - $X \cap S_i \neq \emptyset$ なら $X \in \text{dual}(F_i)$ であり、 X が子供になる
 - そうでないなら $X \cup e, e \in S_i$ が子供になるかもしれない
子供になるための必要十分条件は、 $X \cup e$ が極小であること
- ①のチェックは、集合の共通部分なので簡単。
②はがボトルネック
 $O(|X| \times ||F||)$ 時間
($O(||F||)$ にできる)

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $\text{dual}(F_1) = \{1\}, \{2\}, \{3\}$
 $\text{dual}(F_2) = \{1\}, \{3\}$
 $\text{dual}(F_3) = \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$

apriori

- 大きさが1の non-ヒッティングセットを求め、大きさが2の、と求め、ヒッティングセットになった瞬間に極小性をチェック
 - ★ 1つ小さいヒッティングセットの集合を参照すればいい
 - 解が大きく、多くなると時間がかかる
- 効率良い枝刈りを実現したのが Mtminer
 - ★ ガロワコネクションという条件を提案
 $\text{uncov}(X) \neq \text{uncov}(X \setminus e)$ for all e が成り立つときのみ、極小
($\text{uncov}(X) = \{S \in F \mid X \cap S = \emptyset\}$ 、 $\text{uncov}(X) = \emptyset$ なら X はヒッティングセット)
 - 探索範囲が小さくなり、効率化ができる
 - 実は、今回提案するものの焼き直しになっている

今回の作戦

- critical hyperedge という概念を導入
- 集合 X に含まれる要素 e に対して、 X の中で e のみを含む F のメンバーが critical hyperedge
- 直観的に言えば、 X から e を取り除くと critical hyperedge がカバーされなくなる

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $X = \{2,4\}$
 $\{1,2,3\}$ は2の critical hyperedge
 4 の critical hyperedge はない

今回の作戦

- 集合 X および e に対して、 $\text{crit}(X,e) = \{S \in F \mid X \cap S = \{e\}\}$
- 極小性条件:** ヒッティングセット X が極小 $\Leftrightarrow \text{crit}(X,e) = \emptyset$ for all e (ガロワコネクションと同じ)

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$
 $X = \{2,4\}$ $\text{crit}(X,2) = \{1,2,3\}$
 $\text{crit}(X,4) = \emptyset$
 $X = \{1,2\}$ $\text{crit}(X,1) = \{1,3\}$
 $\text{crit}(X,1) = \{2,4\}$

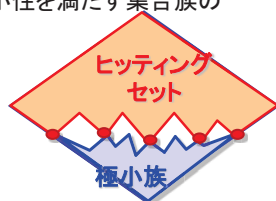
- $\text{crit}(X,e) = \emptyset$ を枝刈りに使って、効率良く探索する

極小性の単調性

- 集合 $X, X', X \subseteq X'$ および e に対して、
 $\text{crit}(X', e) \subseteq \text{crit}(X, e)$ 、つまり、**crit** は単調非増加
- 極小性条件は、(ヒッティングセットだけでなく) 一般の集合でも成り立つ → **crit** による極小性条件は unti-monotone である
 ★ ということは、枝刈りに使える
- さらに、「極小ヒッティングセットは、極小性を満たす集合族の極大元になっている」こともわかる

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$

$X = \{1\} \quad \text{crit}(X, 1) = \{\{1,2,3\}, \{1,3\}\}$
 $X = \{1,2\} \quad \text{crit}(X, 1) = \{\{1,3\}\}$



Crit の良いところ

- 全ての e について **crit** の大きさを足しても $|F| - |\text{uncov}(X)|$ 以下
 ← 任意の $S \in F$ は、高々1つの e に対してのみクリティカル
 → メモリを喰わないし、計算も速くなる
- crit** は、現在の解に一つ要素を追加した際の更新が楽
 → $\text{crit}(X \cup f, e) = \text{crit}(X, e) \setminus F(f)$
 e を含む F の元の集合
- 集合の演算でできるので、極小性判定が高速にできる。特に、データがスパースな場合は速い
 (とても密な場合も、補集合を使うことで高速化)

Crit + 探索手法

- crit** による極小性判定 + 既存の探索手法で高速手法を作る

KS法: 極小性の判定が短時間でできるようになる

→ $O(|X| \times |F|)$ がおよそ $O(|X| \times |F(e)|)$ 、あるいは $O(|F|)$ に

apriori法: 深さ優先型の探索が可能に

(crit が空になった要素は以後無視する、という枝刈りが可能)

- いずれの解法も、極小性の判定がボトルネックなので、高速化が期待される。ただし、以下の場合にはそれほどでもない

- + 解が少ない
- + $|F|$ が小さい
- + F が疎でも密でもない

疎性の利用

- ★ 逆探索の探索木の、ノードの数を減らす

- 列挙するもの: ある i に対して $X \in \text{dual}(F_i)$ であるような X
- X の親: $X \in \text{dual}(F_i)$ である最小の i に対する、先ほどの親
- X の子: $X \cap S_{i+1} = \emptyset$ である最小の i に対する、先ほどの子

- 子供を計算するときに、多少簡単になる

- + $\text{uncov}(X)$ の最小添え字
 のものが S_{i+1}

- uncov の更新だけで
 更新の必要がある S_{i+1} を
 高速で見つけられる

$F = \{1,2,3\}, \{1,3\}, \{2,4\}$

$\text{dual}(F_1) = \{1\}, \{2\}, \{3\}$

$\text{dual}(F_2) = \{1\}, \{3\}$

$\text{dual}(F_3) = \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$

実験(データ)

- 規則的な手法を使って、人工的な問題を生成

- + 入力の数、あるいは解の数がとても小さい

- パターンマイニングを使って問題を生成

① トランザクションデータ D を用意し、閾値 σ に対して
 極大頻出アイテムセットを全て求める

② 各極大頻出アイテムセットの補集合を集めたものをインスタンスとする

- + 極小ヒッティングセットは、「どの極大頻出アイテムセットにも含まれないもの」 → 極小非頻出アイテムセット

- + とても密な問題になるが(3000アイテム中、2995アイテムを含む、など)、現実の問題設定としては妥当と思われる

実験(入力→小、解数→大)

M(n)	in out	BEGK(FK)	DL	BMR	HBC	shd 0	shd D	shd 9(KS)
20	10 1024	0.35	0.03	0.031	6.115	0.015	0	0
24	12 4096	5.62	0.078	0.124	70.871	0.015	0.015	0.015
28	14 16384	11.33	0.795	1.17		0.015	0.015	0.015
30	15 32768	37.79	2.964	3.884		0.077	0.046	0.015
32	16 65536	42.01	10.85	15.06		0.124	0.062	0.046
34	17 131072	fail	41.77	55.89		0.265	0.155	0.093
36	18 262144	fail	164.53	211.53		0.514	0.296	0.202
38	19 524288	fail	700.03	845.529		1.014	0.608	0.39
40	20 1048576	fail	2787.31			4.056	2.87	2.448

実験(入力→大、解数→小)

DM(n)	in out	BEGK(FK)	DL	BMR	HBC	shd 0	shd D	shd 9(KS)
20	1024 10	13.97	0.14	0.421		0.031	0.046	0.217
24	4096 12	41.11	1.138	4.056		0.32	0.468	4.82
28	16384 14	65.74	5.538	38.469		3.63	5.16	94.24
30	32768 15	130.8	10.825	119.902		11.88	17.40	440.57
32	65536 16	fail	24.694	365.494		39.51	54.89	
34	131072 17		55.832	1056.64		136.06	205.59	
36	262144 18		125.736			432.60	678.15	
38	524288 19		276.09			1666.91		
40	1048576 20		1918.562			16599.73		

実験(入力→中、解数→中)

SDTH(n)	in out	BEGK(FK)	DL	BMR	HBC	shd 0	shd D	shd 9(KS)
42	422 422	5.52	0.187	0.202		0	0.015	0.015
62	932 932	18.79	0.811	0.967		0.015	0.046	0.062
82	1642 1642	56.43	2.34	3.073		0.062	0.093	0.156
102	2552 2552	142.96	5.475	7.581		0.124	0.187	0.405
122	3662 3662	354.03	11.2	15.693		0.249	0.421	0.764
142	4972 4972	775.35	19.078	27.58		0.374	0.702	1.326
162	6482 6482		33.368	46.908		0.624	1.294	2.277
182	8192 8192		52.681	75.254		0.982	1.918	3.525
202	10102 10102		83.257	121.415		1.341	2.713	5.116

ランダムデータ(入力→中、解数→大)

dense	in out	BEGK(FK)	DL	BMR	HBC	shd 0	shd D	shd 9(KS)
0.9	1000 30429	1265.98	2556.60	249.72	599.21	0.34	0.29	31.43
0.8	1000 364902					2.82	2.76	411.15
0.7	1000 2509944					20.07	18.11	3310.99
0.6	1000 16809231					153.53	133.31	

まとめ

- 極小解が極大解になるような、極小性の導入 (crit)
- crit の高速更新法を開発し、それを使って高速アルゴリズムを開発

今後の課題

- 実装の改良と計算実験、論文書き
- k匿名化列挙への応用