社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MPS－72 (8)
2008－BIO－15 (8)
2008／12／17

分枝限定法を用いた最適クロスオーバコンセプト検出法について

原口 誠　　大久保 好章

北海道大学大学院情報科学研究科

概要：内包的な類似性を根拠として２つのコンセプトを繋ぐ隠れた"クロスオーバコンセプト"を検出するための手法を，制約付の分枝限定探索アルゴリズムとして与える．

A Top-N Branch-and-Bound Algorithm for Detecting Implicit Crossover Concepts

Makoto HARAGUCHI and Yoshiaki OKUBO

Graduate School of Information Science and Technology
Hokkaido University

Abstract: This report discusses an effective depth-first mining algorithm for finding implicit crossover concepts connecting several major concepts. The algorithm is based on a dynamic ordering method depending on each search node and some search tree expansion rules. Moreover it is designed so as to find top N implicit concepts subject to the size restriction for their implicitness and some space constraints reflecting the crossoverness.

1　はじめに

密接に関連したものを纏まりあるもの，すなわち，クラスタとして抽出するクラスタリングは，雑多で個別的なデータからなるデータ群が持つ複雑さを軽減するために重要であり，多くの研究がなされてきた [1]．一方，リンク情報などの関連性を用い，相互に密結合されたページやノードの塊を見つける，コミュニティ [5] やクリーク [6] の研究も，クラスタリング同様に，一般には膨大な個別的ノード群から意味のある部分を見出すための技法であると思われる．

類似性や関連性を与えるものは，クラスタリングにおいては特徴に関わる類似性尺度，また，コミュニティ検出においてはリンク情報，などの違いはあるが，構成要素数が最多もしくは比較的に多数なものを好む傾向にある．実際，小さなクラスタの数は膨大であり，そうした細々としたものを調べることにどれ程の意味があるのかという理由に基づくと思われる．また，データマイニングにおいても，ほぼ同様の理由により，生起頻度が比較的大な，頻出パターン検出 （例えば，[3]）が多かったと思われる．

『頻出するものは重要だ』は，重要さの基準・経験則としては十分な一般性を持つものであり，本稿はこれに異議を唱えるものでは決してない．しかしながら，非頻出なパターンで個体数において小規模なものの中にも重要なものが潜んでいる可能性もこれまた無視できないものと思われる．本稿では，

『非頻出であっても，主要なクラスタを繋ぐものは重要だ』

との考えに基づき，非頻出なものの中から重要なものを絞りこんで提示できるシステムをめざし，制約付のトップ N 分枝限定アルゴリズムによって十分に高速に抽出可能なことを示す．ここで，「主要なものを繋ぐ」と言っても，個体集合としてのクラスタを摂動させ，別の近傍クラスタをたどる行為などもあるかと思われる．ここでの「繋ぐ」とは，例え外延的に離れたもの（個体集合としての交わりが空であっても良い）でも，内包的な共通性が認められる場合は，重要な繋がりの可能性があると考える．つまり，

『異なる概念間にまたがる共通概念で非頻出なもの』

をクロスオーバ概念として定める．本稿においては，形式概念を検出目標物とした議論を行うが，これは，形式概念が外延的な個体オブジェクトと内包的な属性・特徴の組からなり，外延的にはレアだが，内包

連絡先：原口 誠
〒 060-0814 札幌市北区北 14 条西 9 丁目
北海道大学大学院情報科学研究科
TEL: 011-706-7106
E-mail: mh@ist.hokudai.ac.jp

的な交わりとしてのクロスオーバを直接的に記述できるとの理由による．

2 アルゴリズム概略

$\mathcal{O}$ を**オブジェクト (個体)** の集合，$\mathcal{F}$ を**属性**の集合とする．任意のオブジェクト集合 $X \subseteq \mathcal{O}$ について，X 中のすべてのオブジェクトに共有される属性集合を φX で表す．一方，任意の属性集合 $A \subseteq \mathcal{F}$ について，A 中のすべての属性を有するオブジェクト集合を ψA で表す．オブジェクト集合 $Z, W \subseteq \mathcal{O}$ について，Z の共有する属性集合が W においても共有される，すなわち，$\varphi Z \subseteq \varphi W$ である時，Z は W **を含意する**と言い，$Z \to W$ と表記する．

$X = \{x \mid X \to x\}$ なるオブジェクト集合 X を，**外延**と呼ぶ．つまり，外延とは，オブジェクト含意のもとで閉じた集合のことである．**内包**についても同様に，属性含意のもとで閉じた集合と定義できる [2]．特に，任意の集合 $X \subseteq \mathcal{O}$ と $A \subseteq \mathcal{F}$ について，φX および ψA はそれぞれ，内包と外延となることに注意する．**形式概念**とは，外延 X とその対応する内包 φX の組 $(X, \varphi X)$，もしくは内包 C と対応する外延 ψC の組 $(\psi C, C)$ で定義される．以下の議論では，概念とその外延 (あるいは内包) を同一視する．

形式概念の評価を行なうために，外延と内包に関する評価関数 $eval_{\mathcal{O}}$ と $eval_{\mathcal{F}}$ を考える．特に，ここでは，これら関数は集合の包含関係のもとで単調性を有すると仮定する．すなわち，$X_1 \subseteq X_2$ $(A_1 \subseteq A_2)$ ならば，$eval_{\mathcal{O}}(X_1) \leq eval_{\mathcal{O}}(X_2)$ $(eval_{\mathcal{F}}(A_1) \leq eval_{\mathcal{F}}(A_2))$ とする．集合の要素数 (サイズ) は，こうした評価関数のひとつであり，以下の議論ではこれを仮定する．

本稿では，非頻出な概念抽出を行うが，あまりにもレアすぎる，すなわち，個体概念に近いものが興味の外にあることも自明である．下記の形式化では，内包に対する制約により非頻出なものに限定し，その中で外延評価を最大化することにより，あまりにもレアなものを結果的に排除する．

目的関数 (最適化)：以下の制約を満たす外延 X の中で，$eval_{\mathcal{O}}(X)$ による評価値が上位 N のもの．

内包制約 (必須)：閾値 $\delta > 0$ に関して，$eval_{\mathcal{F}}(\varphi X) \geq \delta$ である．

空間制約 (オプション)：X は以下を満たす．
(POS)：所与の正例オブジェクト集合 S^+ について，$S^+ \subseteq X$ である．
(NEG)：所与の負例オブジェクト集合 S^- について，$S^- \cap X = \phi$ である．
(SUB)：所与の関連属性集合 K について，$X \subseteq \psi K$ である．

(POS) は，$I = \{z \mid S^+ \to z\} \subseteq X$ であることを意味し，I は (概念の) 外延束をボトムアップに探索する場合の起点を定めている．一方，(SUB) は，外延束の上限となる外延 ψK を規定するものである．これは，$K \subseteq \varphi X$ なる制約と等価であり，抽出すべき概念は少なくとも K 中の属性をすべて含まなければならないことを要請し，これによりユーザの興味を反映させることができる．この様に，探索の対象は，I を下限，ψK を上限とする部分束に限定される．

さて，所与の主要な概念 $(X_j, \varphi X_j)$ に対し，観点毎のクロスオーバ概念 C を下記の 2 条件を満たす $(\psi C, C)$ として定める．ただし，観点は内包 K として与えるとする．

観点 K からみた類似性条件：$K \subseteq C$

外延条件：$\psi C \cap X_1 \neq \phi, \psi C \cap X_2 \neq \phi$

類似性条件は，内包 K の拡大としてクロスオーバが観測できることを要請し，(SUB) $\psi C \subseteq \varphi K$ と等価である．つまり，空間制約の特殊な場合として扱える．さらに，外延条件は，$x_j \in \psi C \cap X_j$ なる例示を (POS) 制約として与えて，自動で満たされることに注意したい．これは，主要な概念間のクロスオーバと言っても様々であり，例示をトリガーとして持つクロスオーバを限定的に求めることを意味している．

さて，こうした概念探索をより高速に実行するために，内包制約による探索の枝刈りが有効となることを示す．内包制約は，外延の包含関係のもとでは逆単調性を有する．すなわち，$X_1 \subseteq X_2$ なる外延 X_1 と X_2 について，X_1 が内包制約を満たさない場合は，X_2 も同様に満たさない．よって，ボトムアップに深さ優先探索を行なう際，X_1 が制約を満たさないことがわかった時点で，そこから先に探索枝を張る必要はなく，直ちにバックトラックすることができる．

この様に，制約に基づいて探索空間を限定することが可能であるが，さらに効率的な探索を行なうには，概念の列挙法にも工夫が必要である．そのために，ここでは，拡張候補の動的順序付け，および，重複概念の生成を抑制するための規則を新たに導入する．

所与の制約を満たす外延 X と，あるオブジェクト $x \notin X$ を考える．いま，$cl(X, x) = \{z \mid X, x \to z\}$ が制約を満たす外延である時，x は，X **の拡張候補**と呼ばれる．つまり，拡張候補 x は，X に追加することにより，制約を満足し，かつ，X を包含する外延を生成可能なオブジェクトである．探索過程において，x は，X の次の (より大きな) 外延を求める際の新たな探索枝を形成する．

一般に X の拡張候補は複数存在するが，それらを用いて X から探索枝を張る順序は，探索効率に大きな影響を与える．ここでは，属性集合 $\varphi(X \cup \{x\})$ の大きさの昇順で処理していくものとする．こうした順序は，各 X 毎に決まるものであるから，これを**拡張候補の動的順序付け**と呼び，X における順序を $\prec_X$ と表記する．

X を x を用いて拡張することで，新たな外延 $cl(X,x)$ が生成されるが，そこには $X \cup \{x\}$ が含意する y，すなわち，$X, x \to y$ なる y が x と共に含まれる．ここで，属性集合 $\varphi(X \cup \{x\})$ が小さいことは，$X \cup \{x\}$ がより多くの y を含意する期待が持てることを意味し，結果として，より大きな外延 $cl(X,x)$ の生成が期待できる．つまり，拡張候補の動的順序付けは，より大きな外延を，より早期に見つけることを狙ったヒューリスティックであり，これにより，後に述べる分枝限定による枝刈り効果を高めることができる．

探索の基本戦略は，上述した拡張候補の動的順序付けに基づく深さ優先探索である．初期 (ルート) ノードは，(POS) 制約から定まる外延 $\{z \mid S^+ \to z\}$ とし，その拡張候補の動的順序に従って候補を選択しながら，探索木を深さ優先で拡張していく．探索過程で選択された候補の系列 $c_1, ..., c_k$ は，ルートから外延 $\{z \mid S^+, c_1, ..., c_k \to z\}$ に至るパスを表す．つまり，$S^+ \cup \{c_1, ..., c_k\}$ は，この外延のひとつの**生成元** (*generator*) となる [4]．しかし，一般に，各外延の生成元は複数存在することから，効率良い探索を行なうためには，外延の重複生成を抑制する機構が必要となる．

そのために，ここでは拡張候補を，**右候補**と**左候補**のふたつに分類する．前者は実際に探索木を拡張する際に用いる候補であり，後者は重複生成をチェックするための候補である．

いま，外延の系列 $X_0 = \{z \mid S^+ \to z\}, X_1, \ldots, X_k$ を考える．ここで，$X_i = \{z \mid S^+, c_1, ..., c_i \to z\}$ $(1 \leq i \leq k)$ であり，c_i は，X_i を生成する際に X_{i-1} で選択された候補とする．この時，X_{k+1} を生成する際に X_k で選択されたある候補 c_{k+1} に関して，X_k での候補 r が $r \in \{c_1, \cdots, c_k\}$ あるいは $r \prec_{X_{k+1}} c_{k+1}$ ならば，r は**左候補**と呼ばれる．また，これら以外の候補を**右候補**とする．

以下に示す枝刈り規則は，こうした候補の区別によって，外延の重複生成を安全に排除する．

逆含意に基づく枝刈り：拡張を試みる外延を X とし，その右候補 r と左候補 ℓ を考える．この時，$X, r \to \ell$ ならば，X を r で拡張する必要はない．

これに加え，暫定的に見つかった解の評価値を利用することで，上位 N となる外延の生成が見込めない探索枝を安全に枝刈ることも可能である．

分枝限定に基づく枝刈り：拡張を試みる外延を X，その右候補を r とする．この時，$(X_r = \{w \mid X, r \to w\}) \cup \{X_r$ における右候補 $\}$ の評価値が，上位 N の暫定解における最小評価値よりも小さい場合，X を r で拡張する必要はない．

ただし，上位 N の暫定解がまだ見つかっていない場合は，この規則は無効となる．

3 実験

上述したアルゴリズムを Java で実装した．本節ではその実験結果について述べる．

ここでは，Web 文書クラスタリングのベンチマークとして公開されている *BankSearch* [7] と呼ばれるデータセットを用いた．これは，11,000 の Web 文書 (HTML 文書) から成り，次の 11 カテゴリからそれぞれ 1,000 文書を集めたものである：“*Commercial Banks*”, “*Building Societies*”, “*Insurance Agencies*”, “*Java*”, “*C/C++*”, “*Visual Basic*”, “*Astronomy*”, “*Biology*”, “*Soccer*”, “*Motor Sport*” and “*Sport*”.

前処理として，もとの HTML 文書からタグを取り除いてテキスト化し，そこから，**WordNet** [9] にある形容詞・副詞，および，(一般的な) ストップワードを除去する．ステミング処理 [8] 後，高頻度語および低頻度語を除いた 1,223 語を特徴語 (属性) とした．ここで，各文書に付随するカテゴリの情報は，属性として一切用いていないことを強調しておく．なお，実験は，Dual-Core AMD Opteron processor 2222 SE を搭載した，主記憶 16GB の PC で行なった．

抽出概念例：

BankSearch 中の Web 文書
http://www.vbsquare.com/files/association/,
を与え，$\delta = 50$ のもとで Top-3 概念を抽出した．以下は，得られた概念の一例である．

⟨{**http://www.vbsquare.com/files/association/**,
http://www.vbsquare.com/registry/tip471.html,
http://www.vb-helper.com/links.htm,
⋮
http://www.vbsquare.com/databases/dbclass/,
http://www.vbsquare.com/databases/learndb/,
http://www.vbsquare.com/mouse/context/ },
{ **API**, **component**, **resource**, ... **tips**, **VB**, **graphic** }⟩

外延は 35 の Web 文書から成るが，それらはすべて Visual Basic (VB) のリソースやチュートリアルに関するものであり，[7] においても，同一のカテゴリに属する文書であった．ただし，ここではこのカテゴリ情報を陽には用いておらず，あくまでも，文書中に現れる語のみに基づいて抽出した概念であることを再度強調しておく．この様に本手法では，事前に付与されたカテゴリ情報を陽に使わなくとも，意味的に妥当な概念の抽出が可能である．

次に，ふたつの Web 文書
http://www.citibank.com/uk/portal/
consumer/helpdesk/tc/tc1.htm,
http://vbtechniques.com/useragreement.asp
および，二つの関連語 **claim**, **Internet** を与え，$\delta = 50$ のもとで，Top-1 概念の抽出を試みた．その結果，一例として次の概念を得た．

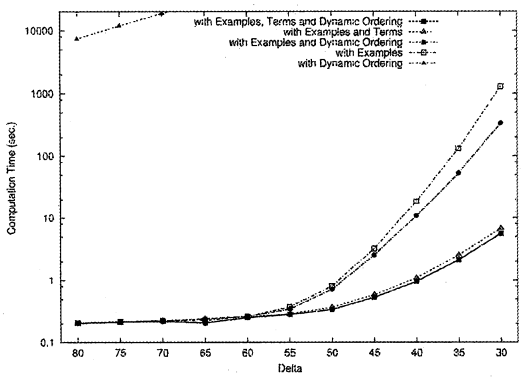

図 1: 正例・関連属性・動的順序付けによる計算時間の改善

⟨ { http://www.citibank.com/uk/portal/
consumer/helpdesk/tc/tc1.htm,
http://vbtechniques.com/useragreement.asp,
http://www.hrbs.co.uk/cashisatandcapply.htm,
⋮
http://www.hrbs.co.uk/panthertandconline.htm,
http://www.hrbs.co.uk/rewardsixtandcapply.htm,
http://www.lloyds.com/un/en/
termsandconditions/category/article/
},
{ claim, Internet, accept, law, condition, ...
reason, right, term, transfer } ⟩ .

外延は 22 の文書から成り，これらは契約や条項に関するものである．また，これらは 4 つの異なるカテゴリ "*Commercial Banks*", "*Visual Basic*", "*Building Society*" および "*Insurance Agency*" に属するものであり，クロスオーバコンセプトの具体例となっている．

この様に，本手法は，クロスオーバコンセプトを含む様々なタイプの概念を抽出可能な極めて柔軟な枠組であると考える．

計算効率：

先に述べた通り，空間制約として与える正例・負例・関連語 (属性) の集合は，ユーザの意図を反映しつつ探索空間を限定する．さらに，拡張候補の動的順序付けも計算効率の改善に寄与している．ここでは，先のクロスオーバコンセプト抽出において与えた正例と関連語について，その効果を具体的に示す．

図 1 より，正例・関連語 (属性)，および，動的順序付けが，計算効率の改善に極めて効果的であることが見て取れる．正例と動的順序付けのみでも大きな改善が見られるが，関連語によって，さらに大幅な改善が得られる．こうした様子は，δ の値が低い領域で特に顕著であり，大規模データに対しても有望なものとなろう．

4 おわりに

非頻出のクロスオーバ概念を高速に求めるために，制約を組み合わせた分枝限定手法を論じた．課題としては，非頻出性を第 1 制約として形式化したことから，一般には長大な内包が提示されるという難点がある．これを避けるために，本稿で与えたものと双対な手法（トップダウン法）を与えることができ，別の機会に報告したい．

参考文献

[1] G.Gan, C.Ma and J.Wu. Data Clustering – Theory, Algorithms, and Applications. SIAM, 2007.

[2] B.Ganter and R.Wille. Formal Concept Analysis - Mathematical Foundations. Springer, 284 pages, 1999.

[3] T. Uno, M. Kiyomi and H. Arimura. LCM ver. 2: Efficient Mining Algorithm for Frequent/Closed/Maximal Itemsets. Proc. of IEEE ICDM'04 Workshop on Frequent Itemset Mining Implementations - FIMI'04, http://sunsite.informatik.rwth-aachen.de/verb+Publications/CEUR-WS//Vol-126/, 2004.

[4] L. Lakhal and G. Stumme. Efficient mining of association rules based on formal concept analysis. Formal Concept Analysis – Foundations and Applications, LNCS-3626, 180–195, 2005.

[5] Y.Zhang, J.X.Yu and J.Hou. Web Communities - Analysis and Construction. Springer, 187 pages, 2006.

[6] E. Tomita and T. Seki. An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique with Computational Experiments. Journal of Global Optimization, 37(1), 95 – 111, Springer, 2007.

[7] M. P. Sinka and D. W. Corne. A Large Benchmark Dataset for Web Document Clustering. Soft Computing Systems: Design, Manegement and Applications, Series of Frontiers in Artificial Intelligence and Applications, 87, 881 – 890, http://www.pedal.reading.ac.uk/banksearchdataset/, 2002.

[8] M.F. Porter. An Algorithm for Suffix Stripping. Program, 14(3), 130 – 137, 1980..

[9] C. Fellbaum (Ed.). WordNet - An Electronic Lexical Database. The MIT Press, 1998.