データベースシステム 113－3
（1997. 7. 14）

サンプルを用いた論理式検索機構の評価

村田 美友紀† 掛下 哲郎††
† 八代工業高等専門学校 情報電子工学科
†† 佐賀大学理工学部 知能情報システム学科

情報システムの高度化に伴うデータベースの複雑化に対応して，知識発見，データマイニングなどデータ集合を抽象化する技術が研究されて いる．データベース質問やビューもデータ集合を抽象的に表現したものである．本稿では、これらの抽象化された情報を論理式を用いて表現する．この論理式をデータベース化することはデータベースの共有や再利用を容易にする上で重要である．そこで必要になるのが論理式の検索機構である．我々はこれまでの研究で，データベース化された論理式をサンプルを用いて検索するための機構を提案した．本稿では，この機構について評価し，効率的な検索手法を導出する．

Estimating A Sample-Based Retrieval of Logical Formulae

Miyuki Murata† Tetsuro Kakeshita††
†Department of Information and Electronics Engineering,
Yatsushiro National College of Technology
††Department of Information Science, Saga University

Increasing complexity of the database due to advancement of information system encourages the research of knowledge discovery and data mining in order to develop an abstract representation of a certain data set. Database queries and views are also such examples of abstraction. We represent such type of information using logical formulae. Sharing and reuse of the database is enhanced using a database of the formulae. We have developed a retrieval mechanism for logical formulae. In this paper, we evaluate the mechanism and propose an effective retrieval strategy.

1 はじめに

情報システムの高度化に伴いデータベース (DB) の複雑化が進行している．一方で，DB の共有及び再利用を促進するためには利用者にとって DB が理解しやすいものでなければならない．このような相反するニーズを同時に満たすために知識発見 [1] やデータベース発掘技術 [2] が導入されてきた．これは DB に格納されている実体集合 (外延) を，抽象化された形式 (内包) を用いて表現することで，より簡潔に DB を表現しようとするものである．同様の考察からビューや DB 質問もデータ集合の抽象化表現と考えることができる．

DB に対する利用者の要求は多様なものであることを考慮すると，上記の抽象化情報も多様なものになることが容易に予想される．従って，これらの情報を DB に格納することは自然な要求である．本稿では DB 化された抽象化情報を検索するための機構を評価する．検索機構は特定の抽象化表現法に依存したものではないが，その評価は命題論理式を用いて行う．命題論理式により association rule，決定木，SQL クエリー，OQL クエリー [3] 等を統一的に表現することができる．全称限量子を含むような抽象化情報を表現するためには述語論理式が必要である．また，非線形データ構造 (リスト，木，集合等) を表現するための記法も提案されている [4].

データベースにおける知識発見では，データ集合を表現するための抽象化情報の構成に研究の力点が置かれていた．また，演繹オブジェクト指向データベースでは論理式を DB 化することが考慮されていたが，主な研究対象は推論機構や利用者インタフェースとしての論理式の利用にあった [5]．本研究はこれらとは異なり，抽象化情報の存在を前提として，それを効率良く検索するための手法について考察する．このような研究は著者の知る限り，あまり行われていない．

命題論理式を DB 化した場合，その検索に条件式を用いることはできない．このような検索は命題論理式の充足可能性問題に帰着されるため，NP 完全問題になる．他の種類の抽象化情報を DB 化して検索する際には，より困難な問題に直面する場合もある (例：決定不能問題)．そこで，我々は，サンプルを用いて論理式を検索するための機構を提案している [6]．サンプルは実体の集合であり，実体を全て満足する論理式を検索する．この方法は条件式を用いた検索よりも直観的であり，ユーザにとって扱い易い．また，NP 完全性等の問題を避けることができる．本手法は，初期サンプルによって検索を行ない，サンプルの追加，削除を繰り返すことで所望の論理式を段階的に検索する．初期サンプルや対象論理式の絞り込みを容易にするための追加サンプルの作成支援機構も提案されている．

本稿は，サンプルを用いた論理式検索機構の評価を行なう．サンプルの追加を 1 ステップとして，ランダムに作成した論理式を唯一に検索するまでに要するステップ数をカウントする．各論理式のステップ数を集計し，本機構の実用性を評価する．また，効率的なサンプル作成の手順についても考察する．

本稿は以下のように構成されている．2 節では，サンプルを用いた論理式検索機構とサンプルの作成法について述べる．3 節では，評価モデルと評価法について述べる．4 節では，評価結果と考察を示す．最後にまとめを行なう．

2 サンプルを用いた論理式検索機構

2.1 サンプルの定義

実体 e が論理式 L を満足するならば，e は L の正例という[1]．また，e が L を満足しないならば e は L の負例という．e が L の負例であるとき，これを $\bar{e}$ と表記する．O が L の正例または負例のいずれか明示されているとき，e は L のサンプルオブジェクトとなる．

サンプルは，サンプルオブジェクトを AND または OR で連結したものである．サンプルオブジェクト $S_1,\cdots,S_n$ について，AND サンプルを $\wedge\{S_1,\cdots,S_n\}$，OR サンプルを $\vee\{S_1,\cdots,S_n\}$ と表記する．S をサンプルオブジェクトとする論理式の集合を $\boldsymbol{L}_S$ とする．AND サンプルを満足する論理式の集合は $(\boldsymbol{L}_{S_1}\cap\cdots\cap\boldsymbol{L}_{S_n})$ である．また，OR サンプルを満足する論理式の集合は $(\boldsymbol{L}_{S_1}\cup\cdots\cup\boldsymbol{L}_{S_n})$ である．S_1 の NOT にはサンプルオブジェクトの負例が対応する．

[1] 実体は，RDB における組や OODB における複合オブジェクトに対応する．

2.2 サンプルオブジェクトの作成

サンプルオブジェクトを作成するための支援体制として以下の3つの機構が提案されている [5].

テンプレートによる作成

論理式を満足する実体のデータ構造をテンプレートとして保存しておき，それを編集してサンプルオブジェクトを作成する．テンプレートは関連のみが値を持ち，属性はすべて空値である仮想的な複合実体である．テンプレートは論理式を保存する時にシステムによって自動的に作成される．なお，同一のテンプレートは複数作成しない．これは，サンプルオブジェクトのデータ構造を特定するために用いる．

論理式による検索

論理式Lに対する正例を検索する際には，$L_p \rightarrow L$を満足する論理式L_pを満足する実体を検索する．負例を検索する場合，$L \rightarrow L_n$を満足する論理式L_nを満足しない実体を検索する．検索された実体を追加したANDサンプルによって検索される論理式の集合はLを含みつつ，追加前よりも要素数が減少したものとなる．この方式は検索対象論理式を絞り込む際に有効と考えられる．

サンプルオブジェクトをサンプルに追加する順序としては，次の3通りがある．

1. すべての正例をサンプルに追加した後に負例を追加する．
2. すべての負例をサンプルに追加した後に正例を追加する．
3. 正例と負例を交互にサンプルに追加する．

論理式を用いたサンプルオブジェクトの検索では，複数の実体$\boldsymbol{O}$が検索される場合がある．この場合の実体の選択方法としては，次の4通りがある．

1. $\boldsymbol{O}$の中からランダムに1つを選択する．
2. 正例 (負例) を検索した場合，検索された実体O_iとサンプル中の負例 (正例)S_jで構成される全ての組$< O_i, S_j >$の中でハミング距離が最小のO_iを選択する．ただし負例 (正例) が存在しない場合は，検索された実体O_iとサンプルの中の正例 (負例) S_jで構成される全ての組$< O_i, S_j >$の中でハミング距離が最大であるO_iを選択する．
3. 負例 (正例) を検索した場合，検索された実体O_iとサンプルの中の正例 (負例)S_jで構成される全ての組$< O_i, S_j >$中でハミング距離が最大であるO_iを選択する[2]．
4. $\boldsymbol{O}$中の全ての実体をサンプルに追加する．

実体間のハミング距離は，2つの実体間で属性値が一致しない属性数といずれか一方にしか存在しない属性数との和によって定義する．

論理式間の差演算

複数の論理式から構成される集合に対して，各論理式に固有な実体を計算する．論理式 L を満足する実体の集合を $E(L)$ とする．論理式の集合 $\{L_1, \cdots, L_n\}$ に対して論理式 L_i に固有な実体の集合 $dif(L_i)$ は以下の式で定義される．

$$dif(L_i) = E(L_i) - \cup_{i \neq j} E(L_j)$$

この操作により各 L_i を特徴づける実体が計算される．各 L_i を特徴づける実体は一つ求めれば十分であるため，$dif(L_i)$ が無限集合の場合にも対応できる．求められた実体をサンプルオブジェクトとすることで，検索対象論理式を絞り込むことができる．$L_j \rightarrow L_i$となる L_jが存在する場合には，$dif(L_i)$ は空集合であるが，$dif(L_j)$ の要素を負例としてサンプルに追加することで，L_iを識別できる．このように差演算によって，検索対象論理式を識別する実体を確実に検索できるため，目的論理式を高速に検索できる．しかしこの手法は，利用者による追加サンプルオブジェクトの決定を伴うため，検索対象論理式の集合が，比較的小さい場合に有効と考えられる．

2.3 サンプルを用いた論理式検索の手順

図1にサンプルを用いた一般的な論理式検索の手順を示す．まず，テンプレートを用いて初期サンプルを作成し，論理式を検索する．所望の結果が得られなければ，サンプルオブジェクトを作成し，サンプルに追加する．改良されたサンプルを用いて論理式の検索を行ない，所望の結果が得られるまで繰り返す．

[2]正例 (負例) が存在しない場合は2と同様に決定する．

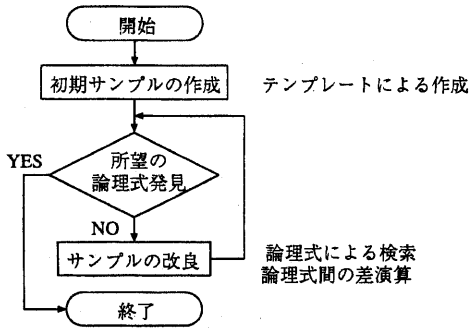

図 1: サンプルを用いた論理式検索の流れ

3 サンプルを用いた論理式検索の評価

3.1 評価モデル

実体を表現するために，スキーマSを考える．Sは10個の属性$\{A_i|i=0,1,\cdots,9\}$を持ち，各属性A_iの定義域は$\{0,1\}$とする．S中には可能な全てのインスタンスが存在するものとする．従って，Sの実体は$<0,\cdots,0>, \cdots, <1,\cdots,1>$の$2^{10}$個である．

上記で定義した実体を要素とする集合を表現するために，以下に示す形式の命題論理式をランダムに100通り生成する．評価実験はこれによって生成された論理式の集合を用いて行う．属性A_iに対応した項として$A_i=0$, $A_i=1$, $True$の3種類を定義する．論理式は，各A_iについて対応する項を1つずつ選択し，それらをANDによって結合することで生成する．上記の定義を満たす論理式は3^{10}通り存在する．この中から選択された論理式は高い確率で共通の項を持つため[3]，論理式間の分別が困難な場合に対応している．

各属性は二値しか取らないが，属性に対応する項が'='のみを用いて定義されているため，項を満たす属性値，満たさない属性値，don't careの3通りを表現できる．従って，対象インスタンスを構成するためには十分である．

なお，検索対象の論理式はAND条件のみを含むが，$(A_i=0)\vee(A_i=1)=True$であることから，限定された形ではあるがOR条件を考慮している．同様に，$\neg(A_i=0)=(A_i=1)$であることから，NOT条件も考慮されている．

[3] 2つの論理式間の共通項数の期待値は3.3である．

3.2 サンプル生成コストの評価

生成された100個の論理式集合に対して，各論理式を唯一に決定するためのサンプルを生成する．サンプルの生成は2.3節で示した手順に従って行なう．具体的詳細を以下で説明する．各論理式の検索コストは，初期サンプルに対するサンプルオブジェクトの追加回数によって定義する．検索コスト毎に論理式を分類し，度数分布を求めることによって，サンプル構成方式の評価を行う．

初期サンプルの作成

初期サンプルは，スキーマの実体の中からランダムに1つ選択する．後のサンプルの改良法に応じて正例、負例のいずれかを決定する．初期サンプルはテンプレートを用いて作成するが，本稿の評価モデルでは単一の実体クラスを考えているため，実体の構造は唯一である．このため，全てのインスタンスがサンプルオブジェクトになることができる．

サンプルの改良

サンプルの改良は，新たなサンプルオブジェクトをANDサンプルに追加することで行う．論理式の検索コストを低減するために，ORサンプルは考慮しないが，この制限によって検索できない論理式は存在しない．新たなサンプルオブジェクトを求める方法としては，論理式を用いた検索と論理式間の差演算を用いた方式がある．これらを用いたサンプルの改良については，以下の手順が考えられる．

1. 論理式を用いた検索のみを使用する．
2. 論理式間の差演算のみを使用する．
3. 両者を交互に使用する．

論理式を用いたサンプルオブジェクトの検索

論理式を用いたサンプルオブジェクトの検索を行う際には，検索目的の論理式Lについて，$L'\rightarrow L$または$L\rightarrow L'$を満足する論理式L'を導く必要がある．評価実験ではLが既知であることを活用して以下のようにL'を求める．

正例のサンプルオブジェクトを検索する際には，検索目的の論理式Lに対して$True$に対応する項を

ランダムに (複数個) 選択し，$A_i = 0$ または $A_i = 1$ に変更した論理式 L' を構成する[4]．L' を用いて検索される実体から L の正例を選択する．

負例のサンプルオブジェクトを検索する際には，検索目的の論理式 L に対して $A_i = 0$ または $A_i = 1$ に対応する項をランダムに (複数個) 選択し，$True$ に変更した論理式 L' を構成する．ただし，$A_i = 0$ または $A_i = 1$ に対応するすべての項を同時に選択しない．$\neg L'$ を用いて検索される実体から L の負例を選択する．

4 評価結果

本節では評価結果を示す．ステップ数は 30 ステップまでしか示していない．実際にこのシステムを使用する際には，サンプルの改良はユーザ自身が行なうため，これを 30 ステップ以上繰り返すことは現実的でないと考える．

図 2に論理式のみを用いてサンプルを追加する各方法の比較結果を示す．追加する実体はサンプル間距離の最も短いものを選択する[5]．各グラフの横軸はステップ数，縦軸は検索結果残った論理式の数を表している．初期サンプルは，負例から追加する場合は，負例を選択し，その他の場合は正例を選択した．

これより，正例から追加する方法，交互に追加する方法で，初期の段階で大幅に絞り込まれていることが分かる．これに対し負例から追加する方法では，徐々に絞り込まれていることが分かる．これは論理式 L に対して正例となる実体数は、2^T(Tは L 中の $True$ である属性数) であり，スキーマの全実体数のたかだか半数であるためと考えられる．よって、正例の追加は論理式の絞り込みのために有効である．正例と負例を交互に追加する方法が正例のみの方法とほとんど同等なのは，初期サンプル (正例) によって検索対象の論理式数が 5 個以下まで絞り込まれるためである．

図 3に論理式を使用してサンプルオブジェクトを検索する場合の各選択方法の比較結果を示す．各グラフの横軸はステップ数，縦軸は検索される論理式の累積数を表している．検索されたすべての実体を選択した場合を除き，ハミング距離の最も短いサンプルオブジェクトを選択した場合が少ないステップ

[4] L が $True$ に対応する項を持たない場合には，L 自身を L' として使用する．負例を検索する際もこれに準じる．

[5] 後にこの選択法が最も効率的であることを示す．

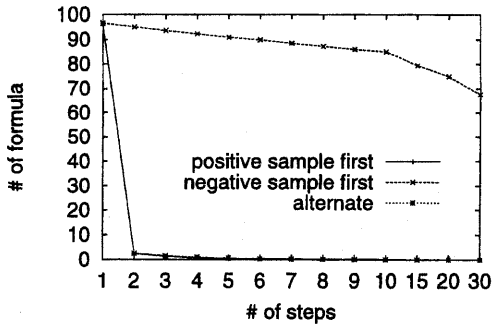

図 2: 論理式による検索におけるサンプルオブジェクト作成順の比較

数で最も多くの論理式を検索している．これは，ハミング距離が最小となる実体の追加によって，検索領域を正確に指定できるためと考えられる．すべてを選択した場合ではステップ数は少なくなるが，負例を検索した場合は少なくともスキーマの実体の半数が追加される．このため，必要メモリ領域や一回の検索に要する計算量が増大するなどシステムのオーバーヘッドが増大する．

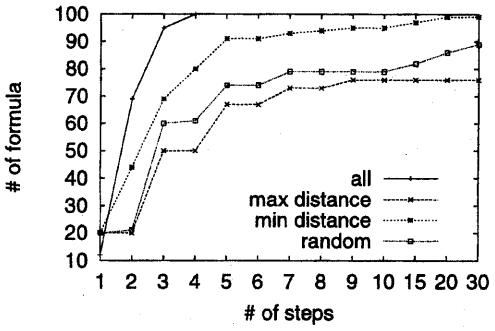

図 3: 論理式による検索におけるサンプルオブジェクト選択法の比較

図 4に属性数を 10 に固定して，論理式数を 200 に増やした場合 (200, 10) と属性数を 5 に減らした場合 (100, 5) の評価結果を示す．グラフの横軸はステップ数，縦軸は検索された論理式数の比率である．論理式数を 200 とした場合でも，ステップ数 10 の時点で 95%の論理式が検索され，27 ステップで全ての論理式が計算された．論理式数を増やしたことで，実体の検索やハミング距離が最小となる実体の

計算など，システムの計算量は増大する．しかし，ステップ数の増大は比較的少ないことからユーザの負担に対する影響は少ない．また，属性数を 5 個とした場合でも，ステップ数 11 で 95%の論理式が検索された．この場合，生成可能な論理式の約半数を検索の対象としているため，属性数 10 の場合よりも対象論理式間の重複は多い．このため，検索ステップ数は増加している．

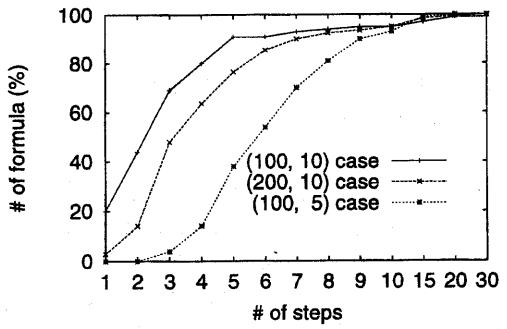

図 4: 異なる検索対象論理式の検索コスト

論理式間の差演算を使用することで，100 個の論理式のうち 76 個について，それを特定する実体が検索できる．しかし，100 個の論理式に対して差演算を直接適用すると，$dif(L)$ の数が論理式数と等しくなるため，ユーザが適切な実体を選択する手間が増大する．これを避けるためには，差演算を適用する論理式集合をある程度小さくすることが不可欠である．図 2によると，正例の初期サンプルを用いることによって，最初の一回の追加で検索対象の論理式を 5 個以下まで絞り込めることが分かる．検索対象論理式の数がこれより多い場合でも，論理式による絞り込みを行い，この後に差演算を適用することでユーザの負担を減らすことができる．

5 おわりに

本稿では，先に提案したサンプルを用いた論理式検索を評価し，効率的なサンプルの作成法を導出した．この結果，以下に示す手順が最良であると分かった．(1) 初期サンプルは正例を用いる．(2) まず、論理式を用いた検索で検索対象論理式を絞り込む．このときには，正例を追加する．複数の実体が検索された場合には，ハミング距離が最小のものを選択する．(3) 絞り込みが進行しなくなった時点で論理式間の差演算を用いる．これらの手順の内，ハミング距離が最小の実体の計算や，絞り込みの進行度の検査などはシステムによる支援が可能である．また，論理式数を増やしてもユーザの負担増加は比較的少なく、システム側の負担に帰着される．

今後の課題としては，論理式を編集することで対応する実体集合を一括編集することが挙げられる．これは，これまでのビュー機能に関する研究成果が活用できる [7, 8]．本稿では，命題論理を検索対象としたが，述語論理など他の抽象情報に関する適用及び評価を進める予定である．また，抽象 (内包) 情報と実体 (外延) 情報が混在した DB の操作に関する研究構想も持っている．

謝辞 本研究の一部は，平成 9 年度科研費重点領域研究 (No.08244105) の援助を受けている．

参考文献

[1] U. Fayyad, R. Uthurusamy, Ed. , "Data mining and knowledge discovery in databases", *Comm. ACM*, Vol. 39, No. 11, pp.24–68, Nov. 1996.

[2] M-S. Chen, J. Hans, P. S. Yu, "Data mining: an overview from a database perspective", *IEEE Trans. Knowledge and Data Engineering*, Vol. 8, No. 6, pp. 866–883, 1996.

[3] R. G. G. Cattell編著, 河込他訳, "オブジェクト・データベース標準 ODMG-93", 共立出版, 1995.

[4] T. Miura, "A logical framework for deductive object", *Information Systems*, Vol. 17, No. 5, pp. 395–414 , 1992.

[5] 近谷, "知的データベース – オブジェクト指向・演繹・ハイパーメディア", オーム社, 1992.

[6] 村田，掛下, "データベース化されたビューに対するサンプルを用いた検索", 電子情報通信学会 DEWS'97 論文集, pp.67–72, 1997.

[7] 掛下, 村田, "複合オブジェクトを宣言的に操作するためのビュー機能", 電子情報通信学会論文誌, J79-D-I, 10, pp.811–819, 1996.

[8] 村田, "宣言的なデータベース操作のためのビュー機能", 佐賀大学大学院修士論文, 1996.