データベースシステム 125－79
（2001．7．19）

決定木のRDB表現と知識探索支援システム

片岡浩巳† 小西 修‡

† 高知医科大学附属病院検査部
‡ 高知大学理学部数理情報科学科

この論文は，決定木生成アルゴリズムから得られた決定木をRDBで表現し，知識発見を支援するシステムを提案する．RDBで表現したテーブルをSQLクエリで探索可能とし，実際に臨床検査医学領域の知識発見支援システムに応用した事例を紹介する．

決定木は，大量な事例の中からルールを生成することができるツールとして幅広く利用されている．代表的な決定木生成アルゴリズムは，C4.5が一般的に用いられているが，近年になって，RDBMSとデータマイニングツールのシームレス統合が可能となり，さらに大規模なデータマイニングを実施できるようになった．このため，生成される決定木は巨大化し，生成された決定木の中から知識を探索することが困難となっている．本研究では，決定木をRDBで表現することにより，SQLのクエリで属性から事例の検索，あるいは，事例から属性の検索など，多角的な検索が可能なシステムを開発し，その有用性を検証する．

RDB Expression of Decision Tree, and Knowledge Discovery Supporting System

Hiromi Kataoka, † and Osamu Konishi‡

† Department of Clinical Laboratory, Kochi Medical School Hospital
‡ Department of Information Science, Graduate School of Science, Kochi University

In this paper, we propose such system that has a relational database translated from the decision tree generated by the decision tree generation algorithm and actually supports the knowledge discovery of a clinical laboratory medicine field using SQL queries. The decision tree has beenwidely used as a tool, which can generate rules from many cases. The C4.5 is a typical decision tree generation algorithm. Recently, the problem of seamless integration of data mining with DBMS is one of the key challenges. We built scalable classification in the form of relations by exploring capabilities of decision trees. The major computation required in this approach can be implemented using standatrd functions by the relational DBMS. The results of experiments conducted for performance evaluation and analysis are presented.

1 はじめに

決定木は，if-thenで表現可能なルールを生成できるシステムとして古くから研究が行われている．代表的なアルゴリズムには，CART (classification and regression tree)，CHAID (chi-squared automatic interaction detection)，C4.5などがある．これらの決定木生成アルゴリズムは，すでに金融や医学などの多くの分野に応用されている[1)-4)]．近年になって，RDBMSとデータマイニングツールがシームレスに統合されたデータマイニング支援システムの構築に関する研究[5)-6)]が始まり，エンドユーザでも容易に大規模なデータマイニングが実施できる環境が整いつつある．これらの研究の中で注目される話題は，RDBMSの標準SQL機能を利用した決定木生成アルゴリズムが提案[7)]されたことである．これは，実メモリを超えた膨大な量の属性と事例から短時間に決定木の生成ができるようになることを意味し，さらに大容量の解析が可能となっている．このように大規模な処理が容易に行えるようにな

った現在，生成されるルールも膨大な量となっている．さらに，分野ごとに検討された決定木のルールを統合して利用するなどの知識グループの管理をどのように行うかが検討課題となっている．

本研究では，決定木をRDBで表現し，巨大化した決定木から容易にルールの探索を可能とし，さらに，複数の領域の決定木を統合した知識データベースとして利用できるシステムを構築する．本システムを実装する対象領域は，医学領域の中でも臨床検査データを対象としており，膨大な事例と属性項目数を取り扱う．また，医療経済的な問題から発生する，未検査属性などの欠落データの存在に配慮したシステムの構築を目標としている．

本研究の特徴を以下にあげる．

(1) 決定木をRDBで表現した．
(2) SQLクエリで，多角的にルールの探索を可能とした．
(3) 生成されたルールの検証を行うための支援システムを構築した．

以下，2章では本研究の概要について，特に決定木の検索モデルを中心に述べ，3章では実装したシステムの構築方法について，4章では結果と考察について，5章でまとめを述べる．

2 本研究の概要

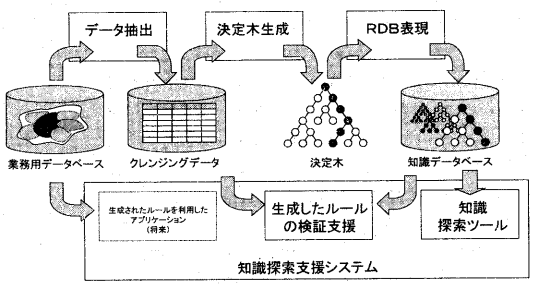

図1　本研究の概要
Fig.1 Outline of this research.

本研究は，決定木と RDB を用い，データマイニングに必要な，データの選択・クレンジング，機械学習，学習成果の検索，そして，発見された知識の検証に関する一連の操作を総合的に支援できるシステムを提案している．実装した対象領域は，大量な属性と事例が存在する医学分野の臨床検査データを用いている．

図1に，本研究の概要を示す．本システムを利用したデータマイニングの手順は，データ抽出，決定木生成，RDB 変換，知識探索，ルールの検証の順に行う．

2.1 決定木から RDB へ

図2は，決定木から RDB へ変換する処理の考え方を示している．決定木をルールに分解した後，単純な表に変換し RDB に書き込む．A,B,C,D は事例を示し，1,2,3,4,5,6 は事例を分類する属性の比較演算子を示している．表は，1 つの属性の比較演算子と事例をタプルとしており，決定木の幹に相当する属性の比較演算子でも，葉の属性と同様に 1 つの事例にリレーションした構造としている．たとえば，属性ルール 1,2,3 が条件を満たしていれば，事例 A を分類することができる．この時，属性ルール 1,3 が条件を満たし，2 が欠落データであっても事例 A を分類できる仕組みである．

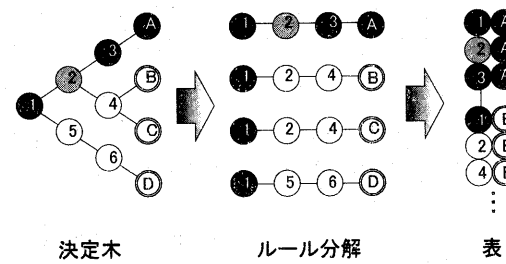

図2 木から表，そして，正規化された RDB へ変換
Fig. 2 Change into a table and RDB from a tree.

2.2 知識探索のモデル

ユーザの視点から見た探索のモデルを以下に示す．このモデルは，臨床検査医学分野において，決定木を利用した知識の探索を行う場合に必要と考えられる方法について分類したモデルである．属性は，検査項目の分析結果，事例は病名を想定している．

2.2.1 ルールに適合した事例

図3は，決定木の基本的利用法である，属性ルールから事例の探索のモデルである．複数の属性結果の判定によって目的の事例を探索する．たとえば，検査値から病名を探索する場合はこれを用いる．

2.2.2 事例を分類可能なすべてのルール

ある事例を分類するためのすべてのルールを知るモ

デルである．2.2.1 のルールに適合した事例とは逆の探索モデルである．たとえば，ある疾患を診断するためのすべてのルールを知りたい場合に用いる．

2.2.3 ある属性により分類可能な事例グループ

図 4 は，ある属性の条件によって分類できる事例はどのようなものがあるかを探索するモデルである．臨床検査医学領域では，仮説病名と呼ばれ，ある検査項目の検査結果によって分類できる疾患群を検索する時に用いる．

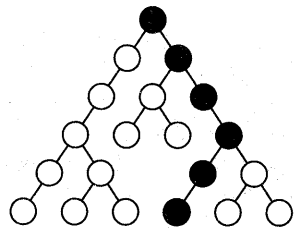

図 3 ルールに適合した事例を探索
Fig. 3 Search for the case from suited the rules.

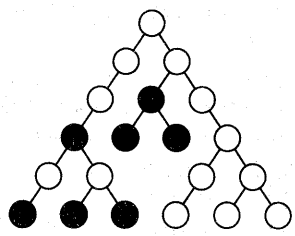

図 4 ある属性により分類可能な事例グループ
Fig. 4 Case group, classified according to a certain attribute.

2.2.4 ある事例群を分類するための属性

図 5 は，ある事例群をさらに分類するために必要な属性項目は何かを調べるモデルである．決定木の幹を構成する属性をテストした後，次に行うべき属性のテストを選択する場合に用いる．たとえば，スクリーニング検査を実施した後，次に考えられる疾患群を分類するために必要な検査項目は何かを知るときに用いる．

2.2.5 複数の決定木の連結

図 6 は，複数の決定木を連結して利用する場合のモデルである．一般的に，すべての属性と事例を 1 つの決定木で表現する方法と比べて，専門領域に属性と事例を分割して，決定木を作成する方が，正しい分類ができる傾向がある．そこで，分野ごとに生成した決定木をリンクして総合的に利用できるシステム設計が必要である．

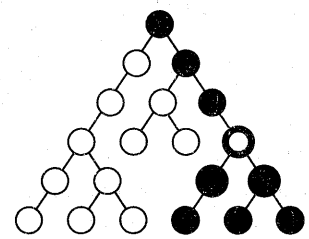

図 5 ある事例群を分類するための属性
Fig. 5 Attribute for classifying case group.

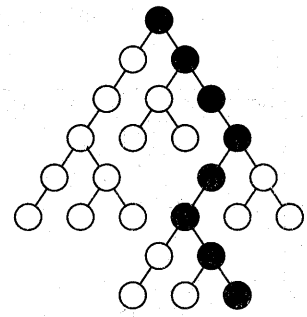

図 6 複数決定木の連結
Fig. 6 Connection of trees.

3 方法

本システムは，データ抽出，決定木生成，RDB 変換，知識探索，ルールの検証の順で，一連のデータマイニングの手順をシステム化している．以下，臨床検査医学領域のデータマイニングに適用した本システムの処理方法について述べる．

3.1 データ抽出

臨床検査で取り扱うデータは，数値情報だけでも約 5000 項目にもおよび，この中で 1 回の検査依頼で実施される属性（検査結果値の項目）数は平均で 65 項目程度である．一般的に診断に不必要な検査項目は実施されない点と，個体内で時系列変動の少ない項目は頻繁に検査が実施されない傾向がある．このため，業務用のデータベースには，多数の欠落データが存在し，決定木を作成するための属性値として直接利用するには不十分である．上記のような多数の欠落データの存在を考慮し，臨床検査情報システムでは，図 7 の構造をした RDB を用いている．

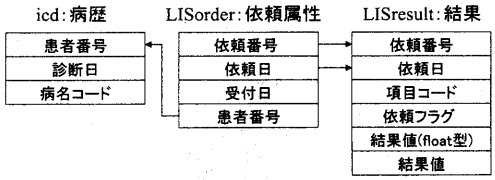

図7 臨床検査情報システムのデータベース構造
Fig. 7 Database structure of clinical laboratory information system.

図7に示すように，結果テーブルは1項目1レコードで記録し，依頼属性テーブル(LISorder)と結果テーブル(LISresult)は依頼番号＋依頼日で多値従属関係にある．

属性	2000/10/1	2000/10/12	2000/10/16	採用データ
TP	6.1	6.0	6.1	6.0
GLU	80		100	100
ALP	447			447
・・・		・・・		・・・
診断結果（事例）		Class1		Class1

1ヶ月の範囲

図8 欠落データの補填処理
Fig.8 Amends processing of the missing value.

この臨床検査情報システムの業務用データベースから決定木を生成するためには，欠落データを可能な限り補填した，属性と事例で構成したテーブルを準備する必要がある．

たとえば，図8に示すような例の場合，2000/10/12に診断された事例（Class1）の近傍検査日の属性値（検査項目）は，TP（総蛋白）の場合，当日の検査結果を用い，GLU（ブドウ糖）は10/1と10/16に分析されており，10/16のデータが近い検査日であることから，10/16の100の値を採用する．また，ALP（アルカリフォスファターゼ）は，10/12，10/16は，両日とも未検査であることから，10/1の結果の447を採用する．この検索をSQLのクエリを用いて行った一例を表1に示す．患者番号(1234567890)と検索したい項目群，そして，検索日をセットしたサブクエリを項目コードにより GROUP BY で分類し，さらに，max()または min()で検査の受付日の近い物を選択することにより，それぞれの近傍検査値が抽出できる．

本研究では，1ヶ月の検索範囲を設けてデータ収集を行ったが，生体における検査値は，病態に応じて時系列的に変化しており，時系列予測などの手法を取り入れるべきと考えられるが，臨床検査医学分野において，この分野の方法論が確立されていないため，個体内変動の多い検査項目は，頻繁に検査が実施される傾向を利用して，便宜的に1ヶ月の範囲で検索することとした．

表1 診断日における近傍の検査値抽出クエリの例
Table 1 Example of neighborhood value extraction query in diagnostic date.

```
select * from LISorder,LISresult
where LISorder.依頼番号 = LISresult.依頼番号
and LISorder.依頼日 = LISresult.依頼日
and 患者番号='1234567890'
and LISorder.受付日+LISresult.依頼番号
+LISresult.依頼日+項目コード
in (select max(LISorder.受付日+LISresult.依頼番号
+LISresult.依頼日)
+項目コード from LISorder,LISresult
where LISorder.依頼番号 = LISresult.依頼番号
and LISorder.依頼日 = LISresult.依頼日
and LISorder.受付日 <= '20001012'
and 患者番号='1234567890'
and 項目コード in ('010120',・・・,'010710')
and 結果値<>''
group by 項目コード)
order by 項目コード;
```

データの対象は，2万5千件（患者1万人）の蛋白電気泳動波形データを自己組織化マップ[8]によりクラスタリング[9)-10)]したクラスを事例データとし，属性データは，血液生化学検査(TP, GLU, A/G,・・・など)120項目の検査結果に年齢，性別を加えた122項目の属性を用いる．自己組織化マップでクラスタリングした波形クラスは64種類の未知のパターンに分類されており，これらのパターンにどのような診断的な意義が隠されているかを他の分野のデータを属性データとして用い調査を行う．

3.2 決定木生成

決定木生成アルゴリズムはC4.5を用いる．C4.5は、1992年にJ.R.Quinlan[1)-4)]により開発された決定木生成

のためのアルゴリズムで，分割統治法と情報エントロピを基本原理としている．

3.3 RDB 変換

決定木生成アルゴリズムによって生成される決定木は，巨大な木となる場合が多く，その決定木から知識を読み取るには，莫大な労力を必要とする問題があった．決定木から新たな知識を発見し，実践の医療に適用していくためには，生成された決定木の結果をさらにデータベース化する必要がある．

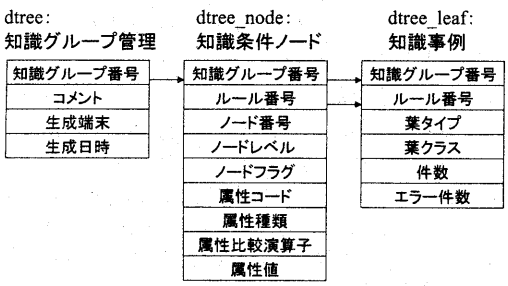

図 9 決定木による知識データベース構造
Fig.9 Structure of knowledge database by decision tree.

```
A/G <= 1.1 :
|  TP <= 7.6 :
|  |  TP <= 6.5 :
|  |  |  T-CHO > 234 :
|  |  |  |  ALB <= 2.8 : class 31 (54.3/9.3)
|  |  |  |  ALB > 2.8 : class 00 (16.1/11.1)
```

図 10 決定木の例
Fig.10 Example of decision tree.

図 9 は，決定木を RDB で表現したテーブルレイアウトである．C4.5 から生成された決定木を直接 RDB に書き込む仕組みとする．表 2 は，図 9 で示したフィールドの意味を示す．図 9 は，第 3 正規形の表現で，実際にシステムに実装した各テーブルとフィールドの概略を示す．

実際の決定木の例を図 10 に示す．図 10 の決定木を RDB で表現した場合の各フィールドの内容を表 3 に示す．

表 2 RDB で表現した決定木テーブル
Table 2 Decision tree table expressed by RDB.

テーブル名：TREE		
名称	フィールド名	フィールドの意味
レコード ID	ID	レコードシーケンス番号
知識グループ番号	KGROUP	知識ベースを目的別に分けるためのグループ
ルール番号	RSEQ	1 つの葉を形成するルールを同一番号とする
ノードレベル	SH	決定木の枝の深さ，(0)が幹
ノードフラグ	LFLAG	葉(1)であるか枝(0)であるか
属性コード（項目コード）	ICODE	属性の内容（項目コード）
属性比較演算子	CMP	大小関係
属性値	RESULT	検査結果値
件数	T	葉のときの事例数
エラー件数	TERR	葉のときのエラー数
葉クラス（葉，症例）	ICD	事例（病名や SOM のクラスを格納） 枝レコードにも葉の事例を格納する

表 3 レコード内容の一例
Table 3 Example of record contents.

ID	ICODE	SH	RSEQ	CMP	RESULT	LFLAG	ICD
1	A/G	0	1	<=	1.1	0	Class31
2	TP	1	1	<=	7.6	0	Class31
3	TP	2	1	<=	6.5	0	Class31
4	T-CHO	3	1	>	234	0	Class31
5	ALB	4	1	<=	2.8	1	Class31
6	A/G	0	2	<=	1.1	0	Class00
7	TP	1	2	<=	7.6	0	Class00
8	TP	2	2	<=	6.5	0	Class00
9	T-CHO	3	2	>	234	0	Class00
10	ALB	4	2	>	2.8	1	Class00

3.4 知識の探索

決定木を RDB で表現したデータベースを SQL により探索する．以下，2 章で述べた検索モデルに従い，SQL クエリの一例を示す．

3.4.1 ルールに適合した事例の検索

ルールに適合した事例の検索，または，ある属性により分類可能な事例グループを検索する場合の SQL ク

エリの一例を示す．たとえば，「TP が 7.0 で T-CHO が 250 の時，考えられる疾患，あるいはクラスを知りたい」といった質問である．クエリの内容は，各属性の条件毎にサブクエリを用いて絞込み，該当する病名を検索する．

```
select distinct ICD from TREE
where ICODE=' T-CHO'
and ((CMP = '<=' and RESULT >= ' 250' )
or (CMP = ' =' and RESULT = ' 250' )
or (CMP = ' <' and RESULT < ' 250' ))
and ICD in
(select distinct ICD from TREE
where and ICODE=' TP'
and ((CMP =' <=' and RESULT >= ' 7.0' )
or (CMP =' =' and RESULT = '7.0')
or (CMP ='<' and RESULT < '7.0'))
);
```

3.4.2 事例を分類可能なルール

ある事例を分類可能なすべてのルールを検索する場合に用いる問い合わせである．たとえば，「Class31 を診断するためのすべてのルールを検索する」という質問の場合，以下のクエリとなる．

```
select * from TREE where RSEQ
in(select distinct RSEQ from TREE
where LFLAG='1' and ICD='Class31')
order by ID ;
```

サブクエリによって，病名コードが' Class31' に属するルール番号を抽出して，そのルール番号に属するリストをレコード ID 順に求める．すなわち，Class31 を診断するためのルールを全てリストアップすることができる．

3.4.3 ある事例群を分類する属性を検索

ある事例群を分類するのに必要な属性は何かを問い合わせる問い合わせである．たとえば，「Class17 と Class41 を分割する項目で，分離に重要な役割をする項目リストを検索する」といった質問は，以下のクエリとなる．

```
select distinct ICODE+CMP+RESULT from TREE
where ICD='Class17' and LFLAG='0'
and ICODE+CMP+RESULT not in
(
select ICODE+CMP+RESULT from TREE
where ICD='Class41' and LFLAG='0'
);
```

複数の事例を分離する決定木の中から，2 つの事例を分離するルールを選び，その共通のルールだけを除外したものが，この 2 つの事例を分割するのに適した属性と比較条件と言える．

3.5 ルールの検証

決定木は，相関性のある属性を組み合わせた場合，多数のルールが生成される．また，事例数が少ない場合でも，細かいルールを生成することが可能な特徴がある．このため，決定木を読み取る場合，事例数の少ないルールに気を取られ，重要なルールを見逃してしまう危険性がある．

これを防止するためには，生成したルールを容易に検証できるようにする必要がある．RDBを利用した場合，GROUP BYによるクエリを利用して度数分布を作成し，そのルールによって分類できる出現頻度を確認できる．その他，ピボットテーブルを利用して，情報の可視化を行うことも可能である．

4 結果と考察

4.1 データ抽出

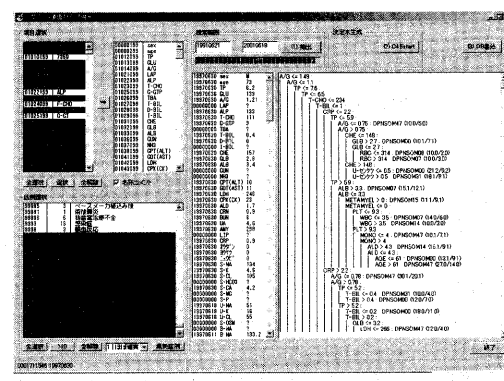

図 11 データ抽出画面の一例
Fig. 11 Example of data extraction processing.

データ抽出処理から決定木の RDB 変換処理までの処理画面を図 11 に示す．属性数（項目数）が 5000 項目にもおよぶ業務用のデータベースから，解析に必要な属性を選択できるようにし，さらに，事例（症例）を自由に選択できるようにした．データ抽出処理の処理スピードは，手続き的に 1 患者 1 項目単位で検索する方法よりも，表 1 で示したクエリを用いた方が，約 1/20 に短縮された．

4.2 決定木生成の結果

64 種類のクラス，122 項目の属性を用い，2 万 5 千件の事例から，14,483 個のノードで構成された決定木が生成できた．また，枝刈り処理後でも 11,993 個のノードとなった．エラー率は，5,931 ノード(24.6%)で，臨床検査データを用いた決定木としては比較的エラー率の低い結果が得られた．これは，教師データとなる事例データの分類が正しく行われていることを示し，SOM による教師無し学習と決定木を組み合わせたデータマイニングの優位性を示している．

4.3 知識発見の一例

図 12 は，巨大な決定木の中から，自明でないパターンが発見できた事例を示す．蛋白電気泳動波形を SOM により 64 種類にクラスタリングしたパターンの中で，Class41 は健常者のグループで，Class17 は，Class41 に非常に良く似た波形であり，どのような診断的な意義が隠されているか解らなかった症例である．波形のパターンの特徴は，アルブミンのグロブリン側のテールと α 2 分画の上昇，γ グロブリンの減少である．

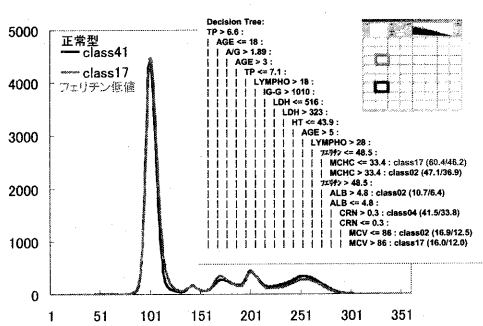

図 12 発見された症例

Fig. 12 Discovered case.

決定木の RDB 表現を図 3 の「事例を分類可能なすべてのルールを知る」クエリで検索すると Class17 のルールだけを即時に参照できる．また，図 5 の「ある事例群を分類するための属性を知る」モデルで検索すると，分離に重要な役割を果たす属性が何であるかを把握できる．得られた特徴的なルールは，5<AGE<=18, HT<=43.3, フェリチン<=48.5, MCHC<=33.4 であることが解る．

決定木は，少量の間違った事例が含まれた教師データを用いた場合でも，その間違いを正確に分類してしまうことから，決定木だけで，最終的な決断をするのは，非常に危険である．そこで，情報エントロピの低い属性の順に，GROUP BY 表，あるいは，ピボットテーブルを用いて，各属性の出現頻度を図で確認する方法が有用である．

図 13 は，図 12 の症例の年齢分布を確認したグラフである．Class17 は，決定木で示される若い年齢層の症例であることが，一目で確認できる．

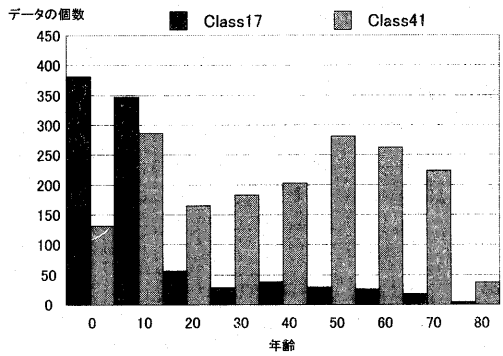

図 13 Class17 と Class41 の年齢分布

Fig. 13 Age distribution of Class17 and Class41.

図 5 で示した，ある事例群を分類するための属性を検索するモデルを，SQL により検索した場合，関係する属性項目を列挙する目的には良いが，具体的に，大小関係まで求めるには，手続き的な処理が必要であり，改良の余地があると考えられる．

5 まとめ

情報システムの高性能化と大容量化に伴い，大規模なデータマイニングの実施が容易に行えるようになっている．データマイニングの手法の一つである決定木に関しても，RDB とデータマイニングツールのシーム

レス統合や決定木生成アルゴリズムの改良の研究が進んでおり，さらに，大容量のデータを用いたデータマイニングが実施できるようになった．このような大容量のデータから決定木を生成した場合，生成された決定木も膨大な量に達し，その特徴を短時間で把握することは困難である．

本研究では，このような背景から，決定木の探索モデルを整理し，さらに，決定木を RDB で表現することにより，決定木の特徴を短時間で探索できるシステムの構築を行った．また，データマイニングに必要な，データの選択，クレンジング，機械学習，情報の可視化といった一連の操作をシステムに実装し，臨床検査医学領域の知識発見の補助を行うことを可能とした．

本研究では，図 6 で示した，複数決定木の連結に関するモデルの具体的な検討が残されている．今後の課題は，この複数の決定木にまたがるルールの探索方法の確立と，さらに，情報の可視化処理を強化し，得られたルールの検証を短時間で行えるシステムを構築することが目標である．

参考文献

1) J.R.Quinlan: Induction of Decision Trees, Machine Learning, vol1, No.1, pp.81-106 (1986).
2) J.R.Quinlan: C4.5 Programs for Machine Learning, Morgan KaufmannPublishers 2929 Campus Drive, Suite 260 San Mateo, CA94403 (1993).
3) J.R.Quinlan: Unknown Attribute Values in Induction, Proc. 6th International Conference on Machine Learning, pp.164-168 (1989).
4) J.R.Quinlan, 古川康一訳：Aiによるデータ解析, 株式会社トッパン, (1995).
5) Amir Netz. Surajit Chaudhuri, Jeff Bernhardt, Usama Fayyad: Integration of Data Mining and Relational Databases, VLDB-2000 , pp.719-722(2000).
6) Hongjun Lu: Seamless Integration of Data Mining with DBMS and Applications, LNCS-2035 (2001).
7) Hongiun Lu, Hongyan Liu: Decision Tables, Scalable Classification Exploring RDBMS Capabilities, VLDB-2000, pp.373-384(2000).
8) Kohonen.T: Self-Organizing Maps, Springer, (1995).
9) 片岡浩巳，小西修，ほか：蛋白泳動波形情報のデータマイニングシステム，日本臨床検査自動化学会誌，Vol26,No3,pp.170-175(2001).
10) 片岡浩巳，小西修：動的計画法-SOMに基づく類似波形検索システム,情報処理学会，(2001).（投稿中）