〔日本OR学会第25回事例研究賞受賞作品〕

知識発見支援ソフトウェア：MUSASHI

羽室　行信，加藤　直樹

1. MUSASHIとは

MUSASHI（Mining Utilities and System Architecture for Scalable processing of HIstorical data）とは，我々がこれまでに開発を進めてきたビジネスデータからの知識発見システムの名称である[1]．MUSASHIは，知識発見プロセスでもっとも労力を要するとされる前処理にその強みがあり，リレーショナルデータベースやデータウェアハウスを導入することなしにXMLで記述された大規模データを効率的かつ柔軟に処理できる仕組みを提供する．標準的なPC一台で数百万件～数千万件のデータ処理が可能である．また前処理だけでなく，文字列解析手法を導入した分類モデルの生成ツール[3]やネットワーク流量の推定による知識発見手法[7]など，いくつかの知識発見アルゴリズムも実装されている．

MUSASHIは現在のところ知識発見に主眼を置いた，いわゆる情報系システムでの利用を前提としているが，将来は基幹系システムへの応用も考えており，「武蔵の二刀流」という意味も込めている．

MUSASHIはオープンソースとして開発を進めており，sourceforge.jpで管理，運営されている．本誌執筆時点でのMUSASHIの最新版は1.0.4である．http://musashi.sourceforge.jp/の「ダウンロード」のメニューからインストールマニュアルに従いインストールすることができる．現在のところ動作確認が取れているOSは，Linux, FreeBSD, Solaris 9, Cygwin（Windows）である．

以降では，MUSASHIの開発目的および構成について簡単に触れた後，ブランド購買分類モデルの構築を例にしてMUSASHIの利用方法をチュートリアル形式で解説していく．ただし，UNIX系OSの利用を前提としているので，読者はUNIXの基本的な知識を有していることを前提として進めていく．

2. 開発目的

この十数年において，コンピュータの性能は飛躍的に向上し，構造化することが困難な意思決定の支援を目的とした利用が重要視されるようになってきた．その証拠に多くの企業では，これまでには捨て去っていたような非常に詳細なレベルのデータまで蓄積し始め，その膨大に蓄積されたデータを経営戦略に生かそうとする動きが活発化しており，データマイニング（もしくは大規模データベースからの知識発見）が注目されるに至っている．

データマイニングが注目され始めて以来，データベースの分野ではデータウェアハウスや多次元データベースなどの研究が盛んになり，またデータ解析の分野では「大規模データ」をキーワードにオペレーションズ・リサーチや統計学，人工知能といった分野で学際的な研究の広がりを見せている．

データマイニングを実施するにあたっては，単にデータベース上のお決まりのデータ項目を利用してモデルを作成すれば終わりというものではなく，データをクリーニングし，多様な角度からデータを集計し，そして新たなデータ項目を作りこむ（例えば，移動平均や数値データの区分化など）といった，いわゆる前処理が不可欠である．ある研究によれば，前処理はデータマイニングプロセス全体の実に8割近い時間が費やされる[8]．これは構造化されていない意思決定に特徴的な試行錯誤の結果の現れであり，それ故に，前処理のプロセスにおいては多様な処理要求に対して容易にこたえることのできる「柔軟性」を備えたシステム

はむろ　ゆきのぶ
大阪産業大学 経営学部
〒574-8530 大東市中垣内3-1-1
かとう　なおき
京都大学 大学院工学研究科
〒615-8540 京都市西京区京都大学桂

[1] MUSASHIのより詳しい論述については文献[4～6]を参考にされたい．

が求められる．MUSASHIはこの前処理を柔軟に実施することを一つの目的として開発を行ってきた．

3. MUSASHIの構成

本節では，MUSASHIの構成について，そのデータ構造およびデータ処理方式の観点から解説する．

3.1 データ構造

MUSASHIは基本データ構造として図1に例示されるようなXMLtableと呼ぶXMLによる表形式のデータ構造を採用している．

XMLtableは完全なXML文書である．ルート要素〈xmltbl〉は，〈header〉と〈body〉の二つの要素を持ち，body要素には，表形式のデータが記述され，項目区切りとしてスペース，行区切りとして改行が用いられている．そして〈header〉要素は，このデータに関する辞書として機能する．それぞれの項目に関する名前と位置情報が〈field〉要素によって記述され，データ項目への名前によるアクセスが可能となっている．

3.2 データ処理方式

MUSASHIは，データ処理のためのプログラムとして，単一の機能に特化した小さなコマンド群を提供する（コマンドの一部を表1に示す）．これらのコマンドの中には，ある項目を抜き出すだけの機能を持ったコマンドから，リレーショナルデータベースで利用されている自然結合や直積演算などのコマンドまで多様なコマンドが存在する．また，マイニングコマンドとしては，相関ルール，クラスタリング，文字列パターン属性の利用が可能な決定木による分類モデルなどが含まれている．

表1 MUSASHIが提供するコマンド（抜粋）

コマンド名	機能	コマンド名	機能
xml2xt	XML→XMLtable変換	xtsed	項目の文字列変換
xt2xml	XMLtable→XML変換	xtagg	レコード集計
txt2xt	text→XMLtable変換	xtcount	行数計算
cvs2xt	cvs→XMLtable変換	xtslide	項目を一行ずらす
xtheader	ヘッダー情報の登録	xtnumber	番号付け
xtcut	項目の抜き出し	xtcombi	項目の値の組合せ出力
xtshare	シェア計算	xtsep	ファイル分割
xtcal	項目間演算	xtnulto	NULL値の置換
xtsel	行の条件選択	xtbest	行番号による選択
xtuniq	重複行の単一化	xtsort	並べ替え
xtcommon	ファイルによる行選択	xtpattern	パターン項目の作成
xtproduct	直積演算	xtasrule	相関ルールの生成
xtjoin	単純結合	xtkmean	クラスタリング
xtnjoin	自然結合	xtclassify	決定木モデル生成

```
<?xml version="1.0" encoding="euc-jp"?>
<xmltbl version="1.1">
<header>
<title>顧客購買履歴データ</title>
<comment>人工データ</comment>
<field no="1" name="店" sort="1"></field>
<field no="2" name="日付" sort="2"></field>
<field no="3" name="時間" sort="3"></field>
<field no="4" name="レシート" sort="4"></field>
<field no="5" name="顧客"></field>
<field no="6" name="商品"></field>
                :
<field no="15" name="数量"></field>
<field no="16" name="金額"></field>
<field no="17" name="仕入金額"></field>
<field no="18" name="粗利金額"></field>
</header>
<body><![CDATA[
A 20010102 134008 1000004 A00180 0000201 1 14 1402 140203 1298 129804 254 331 1 331 254 77
A 20010102 134008 1000004 A00180 0000231 1 11 1111 111105 0245 024505 339 438 1 438 339 99
A 20010102 134008 1000004 A00180 0000278 1 11 1101 110105 1453 145303 375 439 1 439 375 64
A 20010102 134008 1000004 A00180 0000294 1 14 1403 140301 0321 032105 266 373 1 373 266 107
A 20010102 134008 1000004 A00180 0000295 1 11 1107 110703 1268 126801 530 715 5 3575 2650 925
A 20010102 134008 1000004 A00180 0000323 1 11 1101 110117 0140 014003 144 212 2 424 288 136
A 20010102 134008 1000004 A00180 0000351 1 11 1104 110403 0693 069304 212 296 1 296 212 84
A 20010102 134008 1000004 A00180 0000387 1 11 1107 110701 0024 002402 441 590 1 590 441 149
A 20010102 134008 1000004 A00180 0000401 1 11 1101 110141 0905 090503 222 280 1 280 222 58
A 20010102 134008 1000004 A00180 0000522 1 14 1407 140797 0011 001103 198 258 1 258 198 60
                :
]]></body>
</xmltbl>
```

図1 XMLTableによる購買履歴の記述

MUSASHIではこれらのコマンドをパイプによって組み合わせ，シェルスクリプトとして実装することによって多様な処理を実現可能とする．この特徴は，特に新しいものではなく，UNIXで伝統的に受け継がれてきた考え方である[2]．複雑なデータ処理の要求に対しても，こうしたコマンドの組み合わせだけで柔軟に対応できるため，アプリケーションの開発時間およびコストを飛躍的に低減できる．

4. ブランド購買分類モデルの構築[2]

本節では，MUSASHIを利用して，ブランド購買分類モデルの構築を例にしてMUSASHIの利用方法をチュートリアル形式で解説していく．MUSASHIの理解を目的とするため，分析枠組みについては大幅に簡略化している．

4.1 分析の概略

企業にとって自社ブランドに対する顧客の忠誠度を高めるために顧客をどのようにマネジメントするかは極めて重要な問題である．ここでの目的はあるブランドに対する忠誠度の高い顧客に関して，過去の購買行動にどのような特徴が見られるかについてのルールを発見するというものである[3]．

ある商品分類（対象商品：商品細分類コード＝11203）におけるあるブランド（対象ブランド：ブランドコード＝000803）に対する顧客の忠誠度について考える．各顧客の2003年の一年間の対象ブランドのシェアが0.5より大きい顧客を忠誠度の高い顧客グループ，それ以外を忠誠度の低い顧客グループと定義する．人工データでは，忠誠度の高い顧客は276人，低い顧客は724人で合計1,000人の顧客が分析対象となる．

これら二つのグループに属する顧客について，2001年から2002年の2年間における，対象商品のブランド購入パターン（ブランドの購入順序を表す文字列リスト）にどのような違いが見られるかについて検討する．ここでは，ブランド忠誠度を目的変数とし，ブランド購入パターンを説明変数とする決定木による分類モデルを構築することによりルールの発見を試みる．

一般的なモデル構築において利用される説明変数は数値型もしくはカテゴリ型の属性に限定されるが，MUSASHIに実装されている決定木による分類モデル生成コマンド（xtclassify）では文字列パターンを説明変数として扱うことが可能である．これは分子生物学の分野で開発されたBONSAI[10]の機能をビジネスデータの解析の目的のために拡張したものである[3]．

4.2 スクリプトの概要

目的の分析を行うスクリプトを図2に示す．各行末の縦棒(|)は，シェル（BASH）におけるパイプの機能を表しており，パイプの前に実行されたコマンドの出力データをそのまま次行のコマンドの入力データとして受け渡す．スクリプトは，目的変数の作成，ブランド購入パターンの作成，データセットの作成，分類モデルの構築，の四つのブロックから構成されており，次では，それぞれのブロックごとに解説を進めていく．

4.3 目的変数の作成

節4.1で定義した顧客別ブランド忠誠度を計算する(図2①～⑦)．まず元データ（図1）より対象期間(2003年1月～12月）と対象商品（111203）の行を選択する（①，②)．

続いて③～⑤の処理によって，顧客別のブランドシェアを計算している．③のxtcutにより必要な項目(顧客，ブランド，数量）を抜き出し，④のxtaggにより，顧客別のブランド別購買数量合計を計算してい

```
  # 目的変数の作成
①xtsel -c '$日付 >= 20030101 && $日付 <= 20031231' -i dat.xt |
②xtselstr -f 細分類 -v 111203 |
③xtcut -f 顧客,ブランド,数量 |
④xtagg -k 顧客,ブランド -f 数量 -c sum |
⑤xtshare -k 顧客 -f 数量:数量シェア |
⑥xtsel -c '$数量シェア>0.5 && $ブランド -eq 000803' |
⑦xtsetchr -v 高 -a class -o class.xt

  # ブランド購入パターンの作成
⑧xtsel -c '$日付 >= 20010101 && $日付 <= 20021231' -i dat.xt |
⑨xtselstr -f 細分類 -v 111203 |
⑩xtcut -f 顧客,日付,ブランド |
⑪xtuniq -k 顧客,日付 |
⑫xtbest -k 顧客 -s 日付%r -R 1_10 |
⑬xtpattern  -k 顧客 -s 日付 -f ブランド:ブランドパターン -d , |
⑭xtcut -f 顧客,ブランドパターン -o brandPat.xt

  # データセットの作成
⑮xtjoin -k 顧客 -m class.xt -f class -n -i brandPat.xt |
⑯xtnulto -f class -v 低  -o datSet.xt

  # 分類モデルの作成
⑰xtclassify -D 8 -c class -p ブランドパターン
                -i train.xt -I test.xt  -D, -C cost.xml -o model.txt
```

図2 スクリプト

[2] 本原稿用のデータおよびスクリプト一式を次のURLに用意した．
http://musashi.sourceforge.jp/OR/musashi.tgz

顧客	ブランド	数量	数量シェア	Class
A00003	000803	5	0.833	高
A00005	000803	3	1	高
A00008	000803	1	1	高
A00022	000803	1	1	高
A00024	000803	7	1	高
A00026	000803	5	1	高
A00027	000803	3	0.75	高
A00029	000803	8	0.8	高
A00031	000803	8	0.615	高
:	:	:	:	:

図3　目的変数class.xtの内容

顧客	ブランド購入パターン
A00001	000803,000803
A00002	000803,000803
A00003	099101,041602,000803,041602
A00004	000803
A00007	121906,000803,000803
A00008	000803,000803,099101,000803,000803
A00016	000803
A00022	000803,121906,099101,000803
A00023	121906
A00024	121906,000803,000803,041602
:	:

図4　ブランド購入パターンの内容（抜粋）

る．その結果から⑤のxtshareコマンドにより各顧客の中でのブランドシェアを計算している．計算された値は，「数量シェア」という新しい項目名で出力される．

最後に対象ブランド（000803）のシェアが0.5を超える顧客を⑥のxtselコマンドによって選択している．⑦のxtsetchrコマンドは，新しい項目（class）として全行に「高」という文字をセットする．この項目が目的変数となる（「低」については節4.6にて設定する）．

①～⑦の処理による出力結果はclass. xtファイルに出力され，内容は図3に示す通りである．後の処理で必要な項目は「顧客」と「class」のみで，その他の項目は，class項目を計算するために利用したものである．

4.4　ブランド購入パターンの作成

本節では図4に示される顧客別のブランド購入パターンを作成する（図2⑧～⑭）．⑧～⑩は前節と同様に必要な行と項目を選択している．

⑪のxtuniqでは「顧客」と「日付」項目について，値が重複している行を単一化する．すなわち，ある顧客が同一日に異なるブランドを複数購入していても，ある単一のブランドを購入したものと見なす[3]．そして次の⑫のxtbestコマンドにより，各顧客について，日付で降順に並べたときの上位10行を選択している．この処理により，対象商品を頻繁に購入している人については最新10回の購買ブランドのみが選択されることになる．

そして最後に，⑬のxtpatternコマンドによってブランドの購入パターンを作成している．ここでは，ブランドをコンマで区切った文字列リストとしてパターンを表現している．⑭で必要項目である「顧客」，「ブランドパターン」の二項目を抜き出している．

⑧～⑭の処理による出力結果はbrandPat. xtファイルに出力され，内容は図4に示す通りである．

4.5　データセットの作成

ここまでの処理で，結果変数（class. xt）および説明変数（brandPat. xt）の作成が完了した．これら二つのファイルを結合し，最終的なデータセットを作成する（図2⑮～⑯）．

⑮のxtjoinコマンドは，入力データ（brandPat. xt）に参照データ（class. xt）の「class」項目を結合する．結合の条件としては，両データのキー項目（顧客）の値が等しい場合に結合処理を行う．また-nを指定することにより，キー項目の値が入力データにあって参照データにない場合にでも，NULL値をセットする（SQLのOuter Joinに同じ）．説明変数データ（入力データ）には全顧客の情報が含まれているが，参照データ（目的変数）には，ブランド忠誠度の高い顧客のみ含まれている．よって，Outer Joinによって忠誠度の低い顧客の「class」項目にはNULL値（アスタリスク）がセットされることになる．そして⑯のxtnultoコマンドにより，この「class」項目のNULL値を「低」に置換している．出力結果はdatSet. xtファイルに出力され，内容は図5に示す通りである．

4.6　分類モデルの構築

ここまでに作成したデータセットを用いて決定木による分類モデルを構築する（図2⑰）．xtclassifyは，決定木による分類モデルを作成するコマンドである．このコマンドではCART[1]と同様に，枝の分岐基準としてGini Indexを用い二進木を生成する．枝刈り

[3] 分析目的からすると全く厳密性を欠くものであるが，ここではMUSASHIの利用方法の解説のために簡略化している．

顧客	ブランド購入パターン	class
A00001	000803,000803	低
A00002	000803,000803	低
A00003	099101,041602,000803,041602	高
A00004	000803	低
A00007	121906,000803,000803	低
A00008	000803,000803,099101,000803,000803	高
A00016	000803	低
A00022	000803,121906,099101,000803	高
A00023	121906	低
A00024	121906,000803,000803,041602	高
:	:	:

図5　データセット（抜粋）

```
<?xml version="1.0" encoding="euc-jp"?>
<missClassificationCost>
  <cost class="高" predict="低" value="2.62"/>
</missClassificationCost>
```

図6　コストファイル

については C 4.5[9]で採用されている Error Based Pruning を用いている．

⑰の xtclassify では，「ブランドパターン」項目を説明変数とし，「class」項目を目的変数として指定し決定木による分類モデルを生成している．またここではコストファイルを利用している（-c cost. xml）．本ケースにおけるデータセットでは，忠誠度の高い顧客と低い顧客の人数比が 1：2.62 と偏りのある分布であるため，コストファイルを指定することでこの偏りを吸収している．コストファイルの内容は図 6 に示されるように XML にて記述する．

実行結果は，テキストとして model. txt に出力され，その結果が図 7 に示されている．また図 8 に決定木を分かりやすく図示している．

BONSAI はオリジナルのアルファベットを少数のグループに分割することで，より精度が高く可読性の高いモデルの構築を試みる．図 7 の先頭に示されている Alphabet-Index にそのグループが示されている．ここでは四つのブランドを二つのグループに分割しており，“099101” と “121906” をグループ「1」に，“000803” と “041602” をグループ「2」に分割し（これらの新しいグループ番号をインデックスと呼ぶ），これらのインデックスに基づいて決定木を作成している．

決定木のルートノードの条件は，「もしインデックス 2 で示されるブランド（000803 もしくは 041602）

```
[Alphabet-Index]
Field Name: $ブランドパターン
Index[1]={"099101","121906"}
Index[2]={"000803","041602"}

[Decision Tree]
if($ブランドパターン has substring "2 2")
  then if($ブランドパターン has substring "1 2")
    then $class="高" (hit/sup)=(171/368)
    else if($ブランドパターン has substring "2 2 2 2")
      then $class="高" (hit/sup)=(64/138)
      else $class="低" (hit/sup)=(164/187)
  else $class="低" (hit/sup)=(289/307)

[Confusion Matrix]
          Predicted As ...
          低      高      Total
低        453     271     724
高        41      235     276
Total     494     506     1000
```

図7　分類モデル（抜粋）

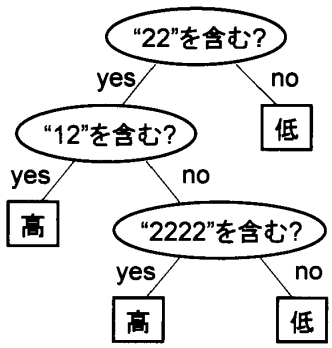

図8　決定木

を 2 回連続購入していれば」と解釈する．人工データを利用しているため，ルールそのものに意味はないが，文字列パターンを属性として扱うことにより，数値やカテゴリ属性を中心とした従来の分類モデルとは異なった観点からの知識発見が可能となる．

5．おわりに

コマンドを組み合わせてスクリプトを作成するのは，初めのうちは少し複雑に思えるかもしれない．しかし慣れてくれば，どのような複雑なデータでも，ブロック遊びの感覚でコマンドを組み合わせ作成することができるようになり，MUSASHI の柔軟性を体感できるであろう．また処理速度についても，インデックスなどの特別なデータ構造を採用しておらず，基本はシーケンシャル処理のみであるにも関わらず，その効率性は高く，実際に大量データを処理することにより，その効率性を実感していただきたい．

MUSASHI プロジェクトはまだ初期段階にあり，

今後も精力的に開発を進めていく計画である．特にMUSASHIは前処理の効率化を目的に開発を進めてきた経緯から，統計解析手法やマイニングアルゴリズムの実装はまだまだ少ない．MUSASHIプロジェクトはAPIもオープンにしており，研究者が独自に開発したデータ解析手法を実装するためのプラットフォームとして活用していただけることを期待している．MUSASHIの開発に興味をお持ちの方は是非ともMUSASHIプロジェクトに参加いただければと願っている．

参考文献

[1] Breiman, L. Friedman, J. H., Olshen, R. A., and Stone, C. J.: *Classification and Regression Trees*, Chapman & Hall (1984).

[2] Gancarz, M.: *The Unix Philosophy*, Butterworth-Heinemann (1996).

[3] Y. Hamuro, H. Kawata, N. Katoh, K. Yada: A Machine Learning Algorithm for Analyzing String Patterns Helps to Discover Simple and Interpretable Business Rules from Purchase History, Lecture Notes in Artificial Intelligence, Vol. 2281, pp. 565-575 (2001).

[4] Hamuro, Y., Katoh, N., Yada, K.: MUSASHI: Flexible and Efficient Data Preprocessing Tool for KDD based on XML, *Proc. of the First International Workshop on Data Cleaning and Preprocessing*, pp. 38-49 (2003)

[5] 羽室行信, 加藤直樹, 矢田勝俊, 鷲尾隆: MUSASHIでらくらくデータマイニング, Software Design, vol. 222, 技術評論社, pp. 95-108 (2004).

[6] 羽室行信, 加藤直樹, 矢田勝俊, 鷲尾隆: 大規模ビジネスデータからの知識発見システムMUSASHI, 人工知能学会誌, Vol. 20, No. 1, pp. 59-66 (2005).

[7] 羽室行信, 加藤直樹: ネットワーク流量推定によるブランド購買パターンからの知識発見, オペレーションズ・リサーチ, Vol. 50, No. 2, pp. 84-91 (2005).

[8] Pyle, D.: *Data Preparation for Data Mining*, Morgan Kaufmann, 1999.

[9] Quinlan, J. R.: *C4.5 : Programs for Machine Learning*, Morgan Kaufmann Publishers (1993).

[10] Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S. and Arikawa, S.: Knowledge Acquisition from Amino Acid Sequences by Machine Learning System BONSAI, *Trans. Information Processing Society of Japan*, Vol. 35, pp. 2009-2018 (1994).