研究のツールボックス 【2】

大規模ビジネスデータからの知識発見システム：MUSASHI

MUSASHI: System for Knowledge Discovery in Large Business Data

羽室 行信 大阪産業大学経営学部
Yukinobu Hamuro Department of Business Administration, Osaka Sangyo University
hamuro@adm.osaka-sandai.ac.jp

加藤 直樹 京都大学大学院工学研究科
Naoki Katoh Department of Architecture and Architectural Engineering, Kyoto University
naoki@archi.kyoto-u.ac.jp

矢田 勝俊 関西大学商学部
Katsutoshi Yada Faculty of Commerce, Kansai University
yada@ipcku.kansai-u.ac.jp

鷲尾 隆 大阪大学産業科学研究所
Takashi Washio Institute for the Scientific and Industrial Research, Osaka University
washio@ar.sanken.osaka-u.ac.jp

Keywords: XML, data mining, database, PMML, preprocessing, loyal customer, classification.

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．MUSASHIは，知識発見プロセスで最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 1台で数百万～数千万件のデータ処理が可能である．

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．

以下では，MUSASHIの構成，インストール方法について簡単に触れた後，優良顧客の早期発見モデルの構築を例にしてデータ変換から決定木によるモデル構築までをチュートリアル形式で解説していく．

2. MUSASHIの構成

本章では，MUSASHIの基本コンセプトについて，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．

2・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．

```
<?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

XMLtableは完全なXML文書である．ルート要素<xmltbl>は，<header>と<body>の二つの要素をもち，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．この<body>要素内だけを見るとUNIXで標準的に用いられているテキストによる表構造で，awkなどのツール群が適用可能な構造となっている．そして<header>要

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
xtsubstr	部分文字列抜き出し	xtsep	ファイル分割
xtcal	項目間演算	xtchgstr	文字列の単純変換
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	agm	グラフマイニング
xtproduct	直積演算	xtasrule	相関ルールの生成
xtjoi	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

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．また，データのタイトルおよびコメントは<title>要素および<comment>要素で表される．

2・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もったコマンドから，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，決定木による分類モデル，クラスタリング，相関ルール，さらにはグラフマイニング[Washio 03]などのコマンドなどが含まれている．

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[Gancarz 96]．複雑なデータ要求に対しても，こうしたコマンドの組合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．

また利用頻度の高いと思われる一連の処理（例えばクロス集計やRFM分析のデータ作成など）はモジュールとして実装されており，一般のMUSASHIコマンドと同じ形式で利用できる．さらに，コマンドやモジュールを組み合わせ，一つの完結した分析（例えば，優良顧客の早期発見システム，ブランドスイッチ分析など）を「シナリオ」として実装している（図2）．

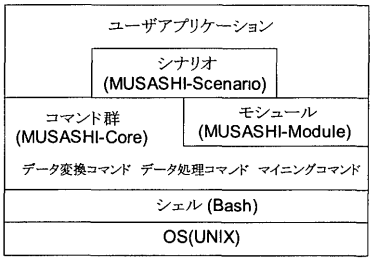

図2 MUSASHIコマンドの構成

表2 ダウンロード可能パッケージ

パッケージ名	内容
0_MUSASHI-PACKAGE	以下の1から5の全てを含んだRPMファイル(各Linuxディストリビューションに対応)
1_MUSASHI-CORE	全コマンドソース
2_MUSASHI-MAN	UNIXのman文書
3_MUSASHI-MODULE	モジュール集(図2参照)
4_MUSASHI-SCENARIO	シナリオ集(図2参照)
5_MUSASHI-CHECK	動作確認用のデータおよびスクリプト

3. MUSASHIのインストール

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」のメニューからダウンロードできる．そこには表2に示すとおり大きく分けて六つのパッケージが含まれている．

現在のところ動作確認が取れているOSは，Linux，FreeBSD，Solaris9，Cygwin（Windows），Mac OS Xである．RedHat系のLinuxであればRPMによるインストールが可能であるが，その他のOSではコンパイル作業が必要となる．詳しくはWebページ上のインストールマニュアルを参照いただきたい．

4. 優良顧客の早期発見モデルの構築

本章では，MUSASHIを利用して，データ変換から予測モデルの構築までの一連の知識発見プロセスを「優良顧客の早期発見モデルの構築[Yada 04]」を例にとり，チュートリアル形式で解説していく．

4・1 人工データのダウンロード

本章で利用するデータは人工的に生成したスーパーマーケットの顧客IDつきPOSデータである．本原稿用のデータおよびスクリプト一式を以下のURLに用意した．

```
http://www.ai-gakkai.or.jp/jsai/journal/toolbox/02/musashi.tgz
```

このデータをダウンロードした後，以下の手順で解凍する．

```
% tar zxvf musashi.tgz
```

すると，musashiディレクトリの下に，表3に示されるデータおよびスクリプトが展開されているはずである．

この購買履歴データはMUSASHIのコマンドを利用し

表3　人工データの内容

ファイル名	内容
dat csv	CSVによる購買履歴データ
cost xml	モデル構築時のコストファイル
scp sh	本章で紹介するシェルスクリプト

て人工的にそれらしく作成したもので，実際の小売店のデータではないが，機密性を考えることなく自由に利用することができる．以下では，このデータを利用して分析を進めていく．scp.shは以下で解説するすべての処理内容を含んだシェルスクリプトである．解説を読み進める前に，とにかく実行したいという読者は，scp.shを以下のようにして実行すればよい．

```
% bash scp.sh
```

実行後，model.pmmlには決定木による予測モデルが，predict.txtには未知データの関する予測結果が出力される．

4・2　分析の概略

企業にとって優良顧客は多くの利益をもたらす最も大切な顧客群のことであり，こうした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．ここでの目的は新規に来店した顧客から将来，優良顧客に育つ顧客群を早期に発見する予測モデルを構築し，マーケティングに活用する有用な知見を得ようとするものである．

優良顧客の定義は，業種や分析目的によって異なってくるが，ここでは優良顧客を「継続的に利益をもたらしくれる顧客」と考える．そこでまず初回来店から6か月後の3か月間における利益率（購入金額に占める「粗利額（売上－原価）の割合），および来店回数の二つの次元について，5×5のマトリックスを作成する（図3）．

各セルにはなるべく等しい人数の顧客が分類されるように配慮する．図3に示されるように，このマトリックスにおいて，利益率，来店頻度の高い右上五つのセルに分類される顧客を優良顧客と定義する．

以上のように定義した優良顧客を新規来店から3か月間の購買行動から予測するためのモデルを構築する．ここでは説明を簡単にするために，優良顧客を定義した際に利用した利益率と来店頻度を説明属性とし，決定木を用いた予測モデルを構築する．すなわち，顧客の初回来店から3か月間の利益率と来店頻度から，6か月後にその顧客が優良顧客になるかどうかを予測しようというものである．

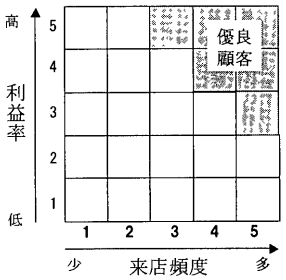

図3　優良顧客の定義

4・3　データ変換

本節より，scp.shに記述された処理内容について詳細に解説を進めていく．

まず4・1節にて用意したCSV形式のデータ（dat.csv）をXMLtable形式に変換する．リレーショナルデータベースや表計算ソフトなど多くのソフトウェアは，データをCSVファイルにエクスポートする機能を有しており，CSV形式のデータから始めるケースは多いと考えられる．

CSVからXMLtableに変換するためには，csv2xtコマンドを利用する．コマンドラインにて，以下のように入力する．

```
% csv2xt -F -i dat.csv -o dat.xt
```

“-F”オプションを指定することによって，1行目を項目名として解釈する．項目名をコマンドラインから指定する場合には“-a 項目名”とすればよい．“-i”，“-o”オプションは，ほとんどすべてのコマンドに共通したパラメータで，入力ファイル名と出力ファイル名を指定する．ここでは，dat.csvを入力ファイルとし，変換結果がdat.xtに出力される．図1に変換されたファイルdat.xtの内容が示されている．

XMLtableではスペースを項目区切り文字として利用しているため，CSVデータにスペースが含まれていれば，自動的に“_”に変換される．

またすべてのコマンドに共通して“-h”オプションのみを与えることによって，簡単なヘルプを参照することができる．

```
% csv2xt -h
```

4・4　基本操作

ここではコマンドの基本動作を見るために，店別売上合計を求めてみる．次のようにコマンドラインから入力する．

```
% xtcut -f 店,金額 -1 dat.xt | xtagg -k 店 -f 金額
-c sum -o storeTotal.xt
```

ここではxtcutとxtaggの二つのコマンドをパイプで連結している．まずxtcutコマンドにより図1に示されるデータdat.xtから「店」と「金額」の二項目を抜き出し，その結果はパイプライン（“|”）を通じてxtaggコマンドに受け渡される．xtaggでは，「店」をキーにして（店別に）-fで指定された「金額」項目を集計する．集計方法は“-c sum”で示されており，ここでは合計である．そして集計結果は，“-o”で指定された“storeTotal.xt”ファイルに出力される．その内容は図4に示されている（見やすさのため，簡略化して示している．以下同様）．

このようにMUSASHIでは単一の機能をもったコマン

店	金額
A	56585371
B	72105393
C	53950632
D	47134643

図4 店別金額合計

ドを組み合わせることによって多様な処理を実現することができる．複数のコマンドを組み合わせる場合，コマンドラインから入力するのではなく，シェルスクリプトとして記述すれば効率的に作業を進めることができる．

4・5 初回来店日を求める

モデルの作成にあたって対象となる顧客は，利用可能なデータ（dat.xt）において初回来店が記録された顧客である．顧客の会員登録日が別に記録されていれば，その情報を利用すればよいが，ここではそのような情報がなく，購買履歴データから初回来店日を推定する方法をとることにする．

今回利用するデータは2002年1月からの3年間のデータである．そこで2002年1月から6月までの6か月間に来店記録のない顧客を選び，それらの顧客は2002年7月以降に初回来店したと推定する．実際には6か月より長い期間を設定するべきであろう．

```
xtcut -f 顧客,日付 -i dat.xt |                                    ····· ①
xtagg -k 顧客 -f 日付 初回来店日 -c min |                          ····· ②
xtsel -c '$初回来店日 >= 20010700' -o firstVisit.xt               ····· ③

xtcut -f 顧客,日付,金額,粗利金額 -i dat.xt |                      ····· ④
xtagg -k 顧客,日付 -f 金額,粗利金額 -c sum |                      ····· ⑤
xtjoin -k 顧客 -m firstVisit.xt -f 初回来店日 |                    ····· ⑥
xtcal -c 'day($日付,$初回来店日)' -a 日数 -o base.xt               ····· ⑦
```

図5 初回来店の算出スクリプト

図5のスクリプトに示した①から③において2002年7月以降に初回来店した顧客の一覧を作成している．①のxtcutにて必要項目を抜き出した後，②のxtaggにて顧客をキーにして（"-k"），最も小さい（"-c min"）日付が出力される．ここで，「-f 日付：初回来店日」は，入力項目名である「日付」を「初回来店日」に変更していることを意味する．そして③のxtselコマンドは条件指定による行選択のコマンドで，上で求めた初回来店日が2002年7月00日以上の行が選択され，その結果がfirstVisit.xtファイルに出力される．日付は8桁の固定長として表現されているので，数値比較演算である"＞＝"の利用が可能となる．出力結果であるfirstVisit.xtは図6のとおりである．

次に，④から⑦の処理では，もとの購買履歴データ（dat.xt）から以下の処理で必要となる基本データを作成している．④，⑤によって，顧客別日別の金額および粗利金額の合計を求めている．次に⑥のxtjoinによって上記で求めた顧客別の初回来店日を結合している．このコマンドは"-k"で指定された項目をキーにして，"-m"で指定された参照ファイルの"-f"で指定された項目を結合する．なお，暗黙的に参照ファイルのキー項目は-kで指定した項目名とみなされる．参照ファイルの項目名が異なる場合は"-K"で指定すればよい．また出力結果としてはキー項目でマッチングできた行のみが出力される．マッチしなかった行も出力（いわゆるouter join）したければ"-n"もしくは"-N"を指定すればよい．

顧客	初回来店日
A00009	20011006
A00036	20011009
A00049	20010919
A00072	20010708
A00088	20010814

図6 顧客別初回来店日

顧客	日付	金額	粗利金額	初回来店日	日数
A00009	20011006	4607	1300	20011006	0
A00036	20011009	4059	804	20011009	0
A00036	20011018	7155	1817	20011009	9
A00036	20011031	4838	889	20011009	22
A00036	20011223	2300	653	20011009	75

図7 顧客別日別集計データ

最後に⑦のxtcalコマンドによって，各日付が初回来店から何日後かを計算し，「日数」という項目を作成し，base.xtというファイルに出力している．xtcalは項目間演算のためのコマンドで，数値や文字列，日付などの各データ型に関するさまざまな関数が用意されている（詳しくはマニュアルを参照）．

ここまでの処理で作成されたデータ（base.xt）は図7のとおりである．

4・6 結果属性の作成

前節で作成した基礎データbase.xtを使って，結果属性を作成する（図8）．①から④で初回来店から6か月後における3か月間の顧客別来店回数を計算している．①のxtselの条件である，「180≦日数≦270」によって，初回来店から半年後（180日後）の3か月（90日）のデータが選択される．②のxtcutで顧客と日付項目を抜き出した後，③のxtuniqにて顧客と日付の値が重複する行を単一（ユニーク）にしている．ここまでの処理にて，顧客別に来店した日数分の行が出力されていることになる．そこで④のxtcountにより，"-k"で指定された顧客を単位に，その行数をカウントすれば来店回数が求まる．新しく求められた項目は"-a"により指定された「来店回数」という項目名として出力される．出力結果（visitCnt1.xt）は図9のとおりである．なお，図中の「日付」項目は途中の処理でのみ必要とされる項目であ

```
xtsel -c '$日数>=180 && $日数<=270' -i base xt |  ····· ①
xtcut -f 顧客,日付 |  ····· ②
xtuniq -k 顧客,日付 |  ····· ③
xtcount -k 顧客 -a 来店回数 -o visitCnt1 xt  ····· ④

xtsel -c '$日数>=180 && $日数<=270' -i base xt |  ····· ⑤
xtcut -f 顧客,金額,粗利金額 |  ····· ⑥
xtagg -k 顧客 -f 金額,粗利金額 -c sum |  ····· ⑦
xtcal -c '$粗利金額/$金額' -a 粗利率 |  ····· ⑧
xtjoin -k 顧客 -m visitCnt1 xt -f 来店回数 |  ····· ⑨
tee classOrg xt |  ····· ⑩
xtbucket -f 来店回数 来店回数クラス,粗利率 粗利率クラス -n 5 |  ····· ⑪
xtsel -c '($来店回数クラス +$粗利率クラス )>=8' |  ····· ⑫
xtsetchr -v 優良 -a クラス -o class xt  ····· ⑬
```

図8　結果属性の作成スクリプト

顧客	日付	来店回数
A00072	20020401	27
A00091	20020421	4
A00109	20020313	2
A00110	20020327	25
A00142	20020213	1

図9　顧客別来店回数

り，ここではすでに意味のない項目となっている．

次に⑤から⑧の処理において，顧客別の粗利率の計算を行っている．上記と同様に初回来店から半年後の3か月のデータを選択（⑤）した後，粗利率の計算に必要な項目を抜き出し，顧客別に金額と粗利金額を合計している（⑥，⑦）．そして⑧のxtcalにおいて「粗利金額/金額」を計算し，粗利率項目を出力している．そして⑨では，①から④で求めた来店回数項目を結合している．ここまでで，優良顧客を定義する基礎データができたことになる．次の⑩のteeはUNIXの標準コマンドで，標準入力で読み込んだ内容を，標準出力とパラメータで指定したファイルの両方に出力する．パイプで連結された途中経過の内容を確認したいときに利用できる．図10にその内容（classOrg.xt）を示す．

4・2節で見てきたように，優良顧客を定義するためには，粗利率と来店回数の二つの次元について，5×5のマトリックスを作成する必要がある．そこで利用するコマンドが⑪のxtbucketである．このコマンドは数値をカテゴライズするためのコマンドである．カテゴライズの方法は，数値データの最大値と最小値の幅を均等に分割する方法と，各カテゴリーに分類されるケース数をできるだけ均等になるように分割する方法の二つがある．デフォルトの動作は後者であり，前

顧客	金額	粗利金額	粗利率	来店回数
A00072	61970	12989	0 2096	27
A00091	20504	3521	0 1717	4
A00109	5785	1144	0 1977	2
A00110	60917	16233	0 2664	25
A00142	293 110	110	0 3754	1

図10　結果属性の定義に使う基礎データ

顧客		粗利率	来店回数	粗利率クラス	来店回数クラス	クラス
A00110	···	0 2664	25	5	5	優良
A00307		0 2257	27	5	3	優良
B00006		0 2244	26	5	3	優良
B00120		0 2556	5	3	5	優良
B00239		0 2354	17	4	4	優良

図11　結果属性データ

者の方法は“-c rng”オプションを指定すればよい．

本ケースでは，4・2節に示したように後者の方法を使う．またカテゴリー数は5で“-n”オプションで指定する．なお，出力されるカテゴリー番号は1からの連番で，本ケースでは1から5の値をとる．最も小さい数値範囲に対応するカテゴリー番号は1で，最も大きい数値範囲に対応するカテゴリー番号は5である．カテゴリー番号の項目は，来店回数クラス，粗利率クラスとして出力される．

優良顧客は図3における右上，すなわち各軸のカテゴリー番号の合計値が8以上のセルとして定義される．そこで⑫のxtselにて，この条件を満たす優良顧客を選択している．

最後に⑬のxtsetchrによって結果属性項目「クラス」を作成している．ここで選択された顧客はすべて優良顧客なので，全行に「優良」という値をセットしている．「非優良」顧客については4・8節で取り上げる．以上の結果（class.xt）は図11に示されている．

4・7　説明属性の作成

説明属性として，初回来店から3か月における来店回数と粗利率を利用する．スクリプトは図12のようになる．

このスクリプトは①と⑤の選択条件を除いて図8のスクリプト①～⑨の処理と同様である．結果（exp.xt）のみ図13に示しておく．

```
xtsel -c '$日数<=90' -i base xt |  ····· ①
xtcut -f 顧客,日付 |  ····· ②
xtuniq -k 顧客,日付 |  ····· ③
xtcount -k 顧客 -a 来店回数 -o visitCnt2  ····· ④

xtsel -c '$ 日数<=90' -i base xt |  ····· ⑤
xtcut -f 顧客,金額,粗利金額 |  ····· ⑥
xtagg -k 顧客 -f 金額,粗利金額 -c sum |  ····· ⑦
xtcal -c '$粗利金額/$金額' -a 粗利率 |  ····· ⑧
xtjoin -k 顧客 -m visitCnt2 -f 来店回数 -o exp xt  ····· ⑨
```

図12　説明属性の作成スクリプト

顧客	金額	粗利金額	粗利率	来店回数
A00009	4607	1300	0 2821	1
A00036	18352	4163	0 2268	4
A00049	20448	5166	0 2526	4
A00072	57239	12817	0 2239	25
A00088	7693	1584	0 2059	2

図13　説明属性データ

4・8 データセットの作成

ここまでの処理で，結果属性（class.xt）および説明属性（exp.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14）．

①により，説明属性ファイル（exp.xt）から必要項目を抜き出し，②のxtjoinで結果属性（「クラス」項目）を結合する．その際，“-n”オプションが指定されているが，この指定により，①の出力結果に含まれており class.xt に含まれていない顧客も出力される（Outer Join）．マッチングされなかった行の「クラス」項目にはNULL値がセットされる．ここでNULL値がセットされた顧客は，説明属性ファイルには含まれるが結果属性ファイルには含まれない，すなわち「非優良」の顧客である．そこで③のxtnultoにより，「クラス」項目のNULL値を「非優良」に置き換える．以上の処理により，モデル構築で必要とされるデータセットができ上がった．その結果（dataset.xt）を図15に示す．

さらに，④のxtselrandによりデータセットをモデル構築に利用するトレーニングデータと精度の検証のために用いるテストセットに分割しておく．xtselrandはランダムにデータ行を選択するコマンドで，-pに選択するデータの割合をパーセントで指定する．-kクラスを指定することによりクラスの値（優良，非優良）別に層化（stratification）して選択できる．選択された20％のデータは-oで指定されたtest.xtに出力され，それ以外のデータはtrain.xtに出力される．

```
xtcut -f 顧客,来店回数,粗利率 -i exp xt |                    ……①
xtjoin -n -k 顧客 -m class xt -f クラス |                    ……②
xtnulto -f クラス -v 非優良 -o dataset xt                    ……③

xtselrand -k クラス -p 20 -S 1 -i dat xt -o test xt -u train xt  ……④
```

図14 データセットの作成スクリプト

顧客	粗利率	来店回数	クラス
A00009	0 2821	1	非優良
A00036	0 2268	4	非優良
A00049	0 2526	4	非優良
A00110	0 2304	33	優良

図15 データセット

4・9 モデルの構築

ここまでに作成したデータセットを用い，決定木による予測モデルを構築する．図16にそのスクリプトを示す．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[Breiman 84]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．枝刈りについてはC4.5[Quinlan 93]で採用されているError Based Pruningを用いている．また，説明属性として数値属性とカテゴリー属性以外にも文字列パターン属性を扱うことができ，九州大学で開発されたBONSAI[Shimozono 94]の機能が組み込まれている．

```
xtclassify -n 来店回数,粗利率 -c クラス -i train xt -I test xt -C cost xml –o model txt –P model pmml
```

図16 予測モデルの構築スクリプト

```
<?xml version="1 0" encoding="euc-jp"?>
<missClassificationCost>
  <cost class="優良" predict="非優良" value="9"/>
</missClassificationCost>
```

図17 コストファイル

```
[Decision Tree]
if($来店回数<=10)
  then if($来店回数<=6 5)
    then $クラス ="非優良" (hit/sup)=(89/89)
    else if($粗利率<=0 2601160575)
      then $クラス ="非優良" (hit/sup)=(15/15)
      else if($粗利率<=0 2703059945)
        then $クラス ="優良" (hit/sup)=(1/1)
        else $クラス ="非優良" (hit/sup)=(4/4)
  else $クラス ="優良" (hit/sup)=(17/28)

[Confusion Matrix]
## TRAINING DATA ##
            Predicted As
            優良      非優良      Total
優良        18        0           18
非優良      11        108         119
Total       29        108         137

accuracy=0 9197080292

## TEST DATA ##
            Predicted As
            優良      非優良      Total
優良        4         0           4
非優良      6         23          29
Total       10        23          33

accuracy=0 8181818182
```

図18 予測モデル（抜粋）

xtclassifyでは，数値属性である来店回数と粗利率を-nオプションで指定し，結果属性である「クラス」項目は-cで指定する．トレーニングデータとテストデータはそれぞれ“-i”と“-I”で指定する．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，優良顧客と非優良顧客の比が1：9と極端な分布であるため，コストファイルの指定なしには，有益なモデルの構築が望めない．

コストファイルの内容は図17に示されるようにXMLにて記述する．

ここでは，優良顧客を非優良顧客と予測した場合のコストを9倍に設定している．

実行結果は，テキストとしてmodel.txtに，そしてPMMLとしてmodel.pmmlに出力される．PMMLとは，データマイニングで利用される予測モデルのXMLによる

```
<?xml version="1 0" encoding="euc-jp"?>
<PMML version="2 0">

<DataDictionary numberOfFields="3">
  <DataField name="来店回数" optype="continuous">
   <Value value="*" property="missing"/>
  </DataField>
  <DataField name="粗利率" optype="continuous">
   <Value value="*" property="missing"/>
  </DataField>
  <DataField name="クラス" optype="categorical"/>
 </DataDictionary>
 <TreeModel functionName="classification" splitCharacteristic="binarySplit">

  <Node score="非優良" recordCount="137">
   <True/>
   <Node score="非優良" recordCount="109">
    <SimplePredicate field="来店回数" operator="lessOrEqual" value="10"/>
    <Node score="非優良" recordCount="89">
     <SimplePredicate field="来店回数" operator="lessOrEqual" value="6 5"/>
    </Node>

   </Node>
   <Node score="優良" recordCount="28">
    <SimplePredicate field="来店回数" operator="greaterThan" value="10"/>
   </Node>
  </Node>
 </TreeModel>
</PMML>
```

図19　PMMLによる予測モデル

標準フォーマットであり，Data Mining Group（DMG）により策定が進められている[DMG 04].

それぞれの結果が図18，図19に示されている.

4・10　PMMLを用いた予測

最後に，前節で構築したPMMLによる予測モデル（model.pmml）を利用して，未知データの予測を行う．ここでは未知データとしてテストデータ（test.xt）を用いることにする.

pmmldtclsコマンドによりPMMLで記述されたモデルに従って未知データの予測を行うことができる．-pでPMMLのモデルを指定し，-iに未知データを指定する．この際，PMMLデータのデータディクショナリで定義された項目すべてが未知データにも存在する必要がある．また予測値は，データディクショナリ上のクラス項目名を利用する（ここでは「クラス」）．test.xtにはすでに「クラス」項目が存在するので，そのような場合には-aで新しい項目名を指定すればよい．図21に予測結果を示す.

```
pmmldtcls -p model pmml -a 予測値 -i test xt -o predict xt
```

図20　未知データの予測スクリプト

顧客	粗利率	来店回数	クラス	予測値
A00307	0 1983	22	優良	優良
B00006	0 2544	25	優良	優良
A00036	0 2268	4	非優良	優良
A00072	0 2239	25	非優良	非優良

図21　未知データの予測結果

```
xt2arff -n 粗利率,来店回数 -d クラス -i dataset xt -o dataset arff
```

図22　ARFFフォーマットへの変換スクリプト

```
@RELATION       "顧客購買履歴データ"

@ATTRIBUTE      "粗利率"  numeric
@ATTRIBUTE      "来店回数"          numeric
@ATTRIBUTE      "クラス"   {"優良","非優良"}

@DATA
0 2821792924,1,"非優良"
0 2268417611,4,"非優良"
0 2526408451,4,"非優良"
0 2239207533,25,"非優良"
0 2059014689,2,"非優良"
```

図23　ARFFデータ

4・11　WEKAとの連係

MUSASHIには，本特集の第1回に紹介されたフリーのデータマイニングツールWEKA用のARFFデータフォーマットへの変換コマンドxt2arffも実装されており，MUSASHIで作成したデータセットをWEKAの多様なマイニングコマンドで分析するといった利用が可能である．4・8節で作成したdataset.xtデータをARFFに変換するスクリプトを図22に示す.

xtarffコマンドでは，-n，-d，-s，-Dのパラメータに数値属性，カテゴリー属性，文字列属性，日付属性をそれぞれ指定する．変換結果は図23に示されている.

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，ブロック遊びをする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かかわらず，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，今後も精力的に改変を進めていく計画である．最後に今後の課題についてまとめておく.

- マイニングコマンドの充実

  MUSASHIに実装されているマイニングコマンドはまだまだ少ない．MUSASHIプロジェクトはAPIもオープンにしており，マイニングアルゴリズムの研究者が独自に開発されたマイニングツール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くことを期待している．興味のある方はぜひとも連絡を

いただきたい．また，本特集の次の記事に紹介されているフリーの統計パッケージである「R」を利用した統計コマンドの開発も考えていきたい．

●XMLtableによる処理履歴の記述

現在XMLtableは項目名の管理が主な目的であるが，将来的には，コマンドの処理履歴をXMLtable内に記述できるように拡張する計画である．このことにより，結果データを見るだけで，そのデータがどのようにして作成されたかを確認でき，またそのコマンド処理履歴を再利用して同様の処理をほかのデータソースで簡単に実現することも可能となる．

●業務系システムへの対応

現在のところMUSASHIはシェルスクリプトとして記述している．近年のPHPを中心としたWebアプリケーションの発展をにらみ，MUSASHIのコマンドをPHPの拡張関数として実装していくことも考えている．このことによりWebアプリケーションとしての業務系システムの構築が容易となる．さらに業務系で必須となる更新関連のコマンドも精力的に実装していく予定である．

このように多くの課題が山積しているが，これらの課題を一つ一つこなすことによって，MUSASHIを日本発のAll in Oneのオープンソース知識発見システムとして発展させていきたい．しかしながら，一方で開発者の不足は深刻で，MUSASHIの開発に興味をお持ちの方はぜひとも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 ◇

[Breiman 84] Breiman, L. Friedman, J. H, Olshen, R A and Stone, C J. *Classification and Regression Trees*, Chapman & Hall (1984)

[DMG 04] http://www.dmg.org/

[Gancarz 96] Gancarz, M *The Unix Philosophy*, Butterworth-Heinemann (1996)

[Hamuro 03] Hamuro, Y., Katoh, N. and Yada, K.: MUSASHI Flexible and Efficient Data Preprocessing Tool for KDD based on XML, *Proc. First International Workshop on Data Cleaning and Preprocessing*, pp 38-49 (2003)

[Hamuro 04] 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, pp. 95-108, 技術評論社 (2004)

[Quinlan 93] Quinlan, J R. *C4 5 Programs for Machine Learning*, Morgan Kaufmann Publishers (1993)

[Shimozono 94] 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)

[Washio 03] Washio, T and Motoda, H · State of the Art of Graph-based Data Mining, *ACM, SIGKDD Explorations*, Vol.5, Issue 1, pp. 59-68 (2003)

[Yada 04] 矢田勝俊：データマイニングと組織能力，多賀出版 (2004)

2004年10月2日 受理

著 者 紹 介

羽室 行信

1991年神戸商科大学大学院経営学研究科修士課程修了．1994年神戸商科大学大学院経営学研究科博士後期課程単位取得満期退学．現在，大阪産業大学経営学部経営学科助教授．専門は，経営情報論．特に，データマイニング，データベースの分野を中心に，企業の情報化に関する研究に従事．

加藤 直樹

1973年京都大学工学部数理工学科卒業 1975年京都大学工学研究科数理工学科修了．1981年神戸商科大学商経学部管理科学科講師，1982年同助教授，1990年同教授を経て1997年より京都大学大学院工学研究科建築学専攻教授 組合せ最適化，計算幾何学，建築システム最適化，最適資源配分，データマイニングの研究に従事．工学博士．

矢田 勝俊（正会員）

1994年神戸商科大学大学院経営学研究科修士課程修了．1997年神戸商科大学大学院経営学研究科博士課程修了．博士（経営学）．1997年大阪産業大学経営学部専任講師を経て，現在，関西大学商学部助教授（経営情報論，データマイニング論）．経営戦略，情報システム戦略に関する研究を経て，現在はヒジネス分野におけるデータマイニングの理論と実践に関する研究に従事

鷲尾 隆（正会員）

1983年東北大学工学部原子核工学科卒業．1988年東北大学大学院原子核工学専攻博士課程修了．工学博士．1988年から1990年にかけてマセチューセッツ工科大学原子炉研究所客員研究員．1990年（株）三菱総合研究所入社 1996年退社．現在，大阪大学産業科学研究所助教授（知能システム科学研究部門）．原子力システムの異常診断手法に関する研究，定性推論に関する研究を経て，現在は人工知能の基礎研究，特に科学的知識発見，データマイニングなどの研究に従事．AAAI，計測自動制御学会，情報処理学会，日本ファジィ学会各会員．