動的な対散布図のLisp-Statによる実現

北海道大学* 水田正弘
岡山大学** 垂水共之

(1996年6月 受付)

1. はじめに

データをグラフィックスによって表示することは、基本的な手法であり、古くから広く利用されてきた。しかし、多変量データをグラフィックスによって表現することは、1変量データや2変量データの表現と比較して、グラフ作成の点からもグラフの解釈の点からも難しい。この種の問題は、1970年代前後から多変量グラフ解析法として数多くの研究がなされてきた。

最近、特に注目されているのは「動き」を利用して手法である。「動き」を適切に利用した場合の利点として次の3点が上げられる。

1. データに関するより多くの情報を一度に表現できる。
2. ユーザは、対話的にデータの解析をすることができる。
3. 他に解析結果を印象深く伝えることができる。


この種の手法を開発するのに、BasicやC, C++などのプログラム言語を利用することが多いが、開発のために多くの労力を必要とする。そこで優れた動的グラフィックス機能を有し、オブジェクト指向に基づく統計処理環境であるLisp-Statを利用して「動的な対散布図」を実現することが考えられる。そこで本報告では、「動的な対散布図」およびLisp-Statを紹介するとともに、動的な対散布図をLisp-Statで実現する方法を報告する。

2. 動的な対散布図について

2変量データのグラフとしては散布図が基本的であるとともに最も広く利用されている。散布図を調べることによりデータの広がりや外れ値の有無などを確かめることができる。

3変量以上のデータのグラフとして、散布図の自然な拡張としての対散布図およびローテーションがある。対散布図は多変量データに対して、各変量の全ての組の散布図を作成し、それ
の散布図を行列状に配置したグラフである。ローテーションは、多変量データをいろいろな方向の2次元平面上に射影させる方法である。すなわち、射影する方向を利用者の指示に従って連続的に変化させることで、画面上の点も連続的に動く。

我々は対散布図のローテーションの考え方を導入した動的な対散布図を提案した（水田（1993，1994））。簡単にこの手法を紹介する。以下、$n$ 個の $p$ 変量データを $x_i=(x_{i1},...,x_{ip})$ ($i=1,2,...,n$) とする。

対散布図とは第 $k$ 変量と第 $l$ 変量についての散布図を全ての $k, l$ について作成するグラフである。すなわち、$p$ 個の単位ベクトル $e_j (j=1,2,...,p)$ （ただし，$e_j$ は，第 $j$ 成分のみが1で他の成分は0のベクトル）に対して、$(x; e_k, x; e_l)$ の $n$ 個の2 変量データの散布図を $p(p-1)$, または上三角部分を除いた $p(p-1)/2$ 枚，作成することになる。ここで，$(e_j; j=1,...,p)$ が正規直交系であるとの条件下で連続的に変化させることにより，対散布図に表示される点を同時に動かすことができる。

マウスにより指定された散布図が，2つのベクトル $e_k, e_l$ により射影された平面とする。このとき，マウス等の移動に伴い $e_k, e_l$ を変更する。変更後のベクトルを $e_k^*, e_l^*$ とすると，

$$
\begin{align*}
  e_k^* &= \cos \theta \ e_k + \sin \theta \ e_l \\
  e_l^* &= -\sin \theta \ e_k + \cos \theta \ e_l
\end{align*}
$$

とする。ただし，$\theta$ は，対象とする散布図の中心を原点とした極座標によるマウスの移動量とする。

以上により，任意の固有直交行列（行列式が1の直交行列）によって座標変換した $p$ 変量データに関する対散布図を表示させることができる。すなわち，ユーザがマウスを利用して任意の方向からデータを「同時に見る」ことができる。

3. Lisp-Stat の紹介

Lisp-Stat は Tierney (1990) により開発された統計処理環境である。基本概念はベル研究所で作られた S 言語を参考にしているが，システムは Lisp 言語で開発されている。S 言語よりもオブジェクト指向の方向を明確にしており，プログラムの開発効率が高い。さらに動的なグラフィックスも含めた高度なグラフィックスに関する機能を持っている。現在，Common Lisp のサブセットである XLISP (Betz (1985, 1988)) を利用した XLISP-STAT (バージョン 2.1版) が公表されており，マッキントッシュ，Windows，UNIX で利用できる。本報告においても 2.1 版を前提とする。

Lisp-Stat を利用する際の問題点をあえて上げるとしたら Lisp に対する抵抗感の存在であろう。多くの統計の専門家および統計のユーザにとって，Lisp は学習の機会が少なく，Fortran や C ほど慣れていない。しかし，Lisp-Stat を通常のデータ処理に利用する際には，Lisp の知識はそれほど必要ない。さらに，多少の努力で Lisp の基本事項を理解した後には，オブジェクト指向に従ったプログラムの作成や高度なグラフィックス機能の利用が効率的にできるようになる。特に，柔軟なデータ形式の記述や容易にユーザ独自の解析法を作成できるところは，Lisp を基にしたシステムの長所である。

データ解析のユーザにとって Lisp-Stat の最も大きな特徴の一つは，フリーソフトであることが上げられる。Lisp-Stat は，anonymous ftp により最新版を入手することができる。また，既に入手している別の人がコピーをもらうことも許可されている。フリーソフトであり，さらにマッキントッシュ，Windows，UNIX など代表的なコンピュータで利用できるので，非常に手軽に使用することができる（図 1）。
図1. Lisp-Statの入手例（一部メッセージを省略）。

以下に、簡単な実行例を示す。これらは、利用するコンピュータの種類に依存せず、ほぼ同様な結果が得られる。$5 + 6 + (2 \times 3)$ は、前置記法を使って入力し,

```plaintext
> (+ 5 6 (* 2 3))
17
```

のように計算する。12個の1次元データを

```plaintext
> (def dat (list 3.14 5.67 2.4 7.1 4.5 8.9 5.6 3.4 5.2 4.7 6.5 8.2))
```

DAT によって dat と名前をつけ,

```plaintext
> (mean dat)
5.4425
> (median dat)
5.4
> (standard-deviation dat)
1.99383
```

によって、平均 mean, 中央値 median, 標準偏差 standard-deviation などを容易に求めることができる。さらに、

```plaintext
> (histogram dat)
> (boxplot dat)
```

によって、ヒストグラム histogram, ボックスプロット boxplot を表示できる（図2(a), (b))。この場合に、S 言語などのように表示するウィンドウ等を設定するためのコマンドは必要ない。Tierney (1990) の本にある例題のデータもファイルで提供されており、

```plaintext
> (load "xlispsstat/book")
> (def mm (copy-list (list LOSS AIR TEMP CONC)))
```

によって、21個の4変量データ mm を用意する。このデータに対して
図 2. Lisp-Stat による基本的なグラフ。

＞(spin-plot mm)
＞(scatterplot-matrix mm)

により、回転プロットおよび対散布図を表示する（図 2(c), (d))。

3.1 オブジェクト指向

Lisp-Stat の特徴の一つにオブジェクト指向がある。Lisp-Stat では、各グラフなどを、オブジェクトとして扱う。オブジェクトには、情報を格納する場所がいくつかあり、それらをスロットと呼ぶ。オブジェクトに対して、(1) ユーザ、(2) ウィンドウの変化に伴うエクスプローラー、(3) マウス、(4) 他のオブジェクト等、からメッセージが送られる。メッセージの名前（セレクタ）により、オブジェクトに設定されているメソッドが選択され、起動される。すなわち、別のオブジェクトに対して同じメッセージを送った場合でも同じ作用をおよぼすとは限らない。これにより、各オブジェクトごとに最も適切なメソッドを設定できる（図 3）。

例えば、先に紹介した例と同様に、mm を 4 変量の 21 個のデータとする。このとき、
オブジェクト

図3. オブジェクト、スロット、メッセージ、メソッド。

\begin{verbatim}
> (def w (scatterplot-matrix mm))
W
により、対散布図であるオブジェクト w ができる。このオブジェクト w に、メッセージ:help
を送ると

> (send w :help)
SCATMAT-PROTO
Scatterplot matrix
Help is available on the following:

: ALL-POINTS-SHOWING-P : ALL-POINTS-UNMASKED-P : ALLOCATE
: ANY-POINTS-SELECTED-P : APPLY-TRANSFORMATION : BACK-COLOR
: BRUSH : BUFFER-TO-SCREEN : CANVAS-HEIGHT : CANVAS-TO-REAL
: CANVAS-TO-SCALED : CANVAS-WIDTH : CENTER : CHOOSE-MOUSE-MODE
: COPY-TO-CLIP : CURRENT-VARIABLES : CURSOR : CUT-TO-CLIP
: DELETE-DOCUMENTATION : DELETE-METHOD : DELETE-MOUSE-MODE
...以下、省略...

と表示される。さらに、オプションであるトピックス:brush をつけて、:help メッセージを w
に送ると、

> (send w :help :brush)
:BRUSH
Method args: (x y width height)
\end{verbatim}
Sets or retrieves current brush. Brush is specified in terms of the lower lefthand corner and its width and height.

NIL

:brushに関するヘルプができる。その他、この対数散布図のオブジェクトに、オブジェクトが有するストロットの名前を得るメッセージ: slot-names、ストロットの値を管理するメッセージ: slot-valueなど多くのメッセージに対応させるメソッドが用意されていることがわかる。さらに、新しいメソッドを追加することもできる。これについては、次節で紹介する。

3.2 動的なグラフィックス機能

特に、Tierney (1990) による本のタイトルにもあるように、Lisp-Statは、動的なグラフィックスの能力が充実している。以下に、Lisp-Statの特徴などを列挙する。

1. Common Lisp のサブセットである XLISP で記述されている。
2. オブジェクト指向である。プロット（グラフ）に対してメッセージを送ることにより図の追加・変更などができる。
3. グラフパッキングにより滑らかなアニメーションが実現できる。
4. マウスによる入力をイベントとしてサポートしている。
5. メニュー、スライドなどが用意されている。
6. Tierney (1990) では、ローテーション、グランドツアーなどのプログラム例が示されていいる。
7. ワークステーションでは、Lisp-Statから Cの関数や FORTRAN のサブルーチンをダイナミックローディングにより呼び出すことができる。

特に、オブジェクト指向に基づくことより、これらの特徴を生かしたプログラムの作成は容易である。

4. 動的な対数散布図の実現

前節で紹介した機能のいくつかを利用して、Lisp-Statで動的な対数散布図の実現を試みた。

始めに「動的な対数散布図」を開発する際の方針を検討した。もちろん、Lisp-Statの基本的な機能を利用して、いかプログラムを作成することも可能である。しかし、資源の有効利用と開発効率、メンテナンスの容易さから既存の関数を利用することにした。

基になる関数としては、回転プロット spin-plot と対数散布図 scatterplot-matrix が候補に上がった。前述の :helpメッセージ等を利用してこれらの関数により作成されたオブジェクトが持つメソッドを調査した。その結果、回転プロットを拡張して散布図を増やすより、対数散布図に回転の機能を入れる方が少ない開発手順で動的な散布図を実現できると判断した。特に、回転の機能は対数散布図のオブジェクトに既に準備されている :rotate-2を使って実現することができる。

> (send w :help :rotate-2)
:ROTA TE-2
Method args: (var1 var2 angle &key (draw t))
Rotates int the plane of variables with indices VAR1 and VAR2 by ANGLE, in radians.
sends the :REDRAW-CONTENT message if DRWA is true.
NIL
動的な対数ブロックの Lisp-Stat による実現

対数ブロックのオブジェクトにメッセージ: ad-slot を送ることにより、回転の角度を表す新しいスロット th を準備する。さらに、defmeth により新しいメソッド do-dsm を定義する。また、メッセージ: add-mouse-mode により、対数ブロックが持っている 2 つのマウスマウス: Selecting Mode と Brushing Mode に、新しいマウスマウス Dynamic-Scatterplot-Matrix を加えた。その時のマウスカーネルは、hand を利用した。以上の方針より作成したプログラムを表 1 に示す（このリストにおいて、以下記述はコメントである。UNIX 用の Lisp-Stat などいくつかのシステムでは、コメントとして日本語を利用できる場合が多いが、システムや漢字コードによってはエラーがでる可能性もある。そのときは、日本語のコメント部分を削除する）。

表 1. ある対数ブロックのプログラム。

(defun standardize (x) (/ (- x (mean x)) (standard-deviation x)))

(defun dynamic-scatterplot-matrix (mm)
  (dotimes (i (length mm)) (setf (select mm i) (standardize (select mm i))))
  ; 各変数の標準化
  (setf w (scatterplot-matrix mm)); 対数ブロックの作成
  (send w :add-slot 'th 0); スロットの追加
  (dotimes (i (length mm)) (send w :range i -5 5))
)

------------------------------------------------------------------------

(send w :add-mouse-mode 'Dynamic-Scatterplot-Matrix
 :title "Dynamic-Scatterplot-Matrix"
 :cursor 'hand
 :click :do-dsm); マウスマウスの追加
(defmeth w :do-dsm (x y m1 m2); メソッドの定義
 (setf p (send self :num-variables))
   (setf v (send self :content-variables))
   (setf cent (/ (send self :size)) 2)
   (setf org (- cent (* haba p 0.5)))
   (setf dx (- (list x y) (* org (* haba v) (/ haba 2)) )
   (setf th
     (if (/= (select dx 0) 0) (atan (/ (- (select dx 1)) (select dx 0)))
       (/ pi 2)))
   (setf th
     (if (< (select dx 0) 0) (+ th pi)
       th))
   (send self :slot-value 'th th)
(flet ((dsm (x y)); 回転角の計算と回転をさせる関数
   (setf v (send self :content-variables))
   (setf (select v 1) (- p 1 (select v 1)))
   (setf dx (- (list x y) (* org (* haba v) (/ haba 2)) )
   (setf th
     (if (/= (select dx 0) 0) (atan (/ (- (select dx 1)) (select dx 0)))
       (/ pi 2)))
   (setf th
     (if (< (select dx 0) 0) (+ th pi) th))
   (send self :rotate-2 (select v 0) (select v 1) (- th (send self :slot-value 'th)))
   (send self :while-button-down #'dsm)))
)
このプログラムを入力した後、

\[(\text{dynamic-scatterplot-matrix mm})\]

するとウィンドウが現れ、データ mm の動的な対散布図が表示される。そのウィンドウの上部にある Menu をプルダウンさせ、Mouse Mode ... を選択すると、マウスモードを選ぶダイアログと呼ばれるウィンドウが表示される。ここで、Dynamic-Scatterplot-Matrix をクリックしてから OK をクリックすると動的な対散布図のマウスカーソルが手形になり各散布図を回転させることができる。ブラッシングやデータのリンクなど基になる対散布図が持っていた機能もそのまま利用できる。

図 4 に Lisp-Stat により実現した動的な対散布図を使ったデータ表示の例を示す。データは、図 2(d) と同様に Tierney (1990) の本にある 21 個の 4 変量データを利用した。画面の中央に、通常の「対散布図」を表示し、その右側には、各変量ごとの「ヒストグラム」、下に同様な「ボックスプロット」を配置している。この「対散布図」は、図 2(d) よりもデータを大きく表示している。画面の左側に、「動的な対散布図」があり、右側のグラフを参照しながら、さまざまな方向からデータを同時に見ることができる。「動的な対散布図」の下に配置されているのは、マウスモードを選ぶダイアログであり、通常の対散布図に用意されている Selecting Mode, Brushing Mode 以外に Dynamic-Scatterplot-Matrix がある。二つ以上の散布図と、四つのヒストグラムはリンクされているために、どれかのグラフに指定したデータは、他のグラフでも強調されている。

5. おわりに

本報告では、Lisp-Stat を紹介するとともに、「動的な対散布図」を Lisp-Stat で実現する方法
を示した。Lisp-Stat はオブジェクト指向であり、本来、動的な機能を持たない対散布図をもとに、必要なスロット、メソッドを追加、変更するだけで動的な対散布図を構成することができた。C 言語を利用して作成した動的な対散布図のプログラムのサイズが基本的な部分だけでも 1000 行以上になることと比較すると、開発の労力は 1 倍以上改善されている。さらに、データの「リンク」により、同時に表示した他の統計グラフとお互いにブラッシング機能を共有できる。

グラフィックス機能の優れ、オブジェクト指向である Lisp-Stat を利用することにより、新しいグラフィカルな手法を効率的に開発することが可能になることを示すことができた。

参考文献

水田正弘 (1993). Dynamic pair-wise scatterplot について、第 61 回日本統計学会講演報告集、136-137。
水田正弘 (1994). 動的な対散布図の提案とその実現について、計算機統計学、7(2)，155-162。
柳 貴久男，片岡 武，脇本和昌 (1994)。連続回転グラフを用いたクラスタリング法，計算機統計学，7(1)，21-28。
Dynamic Scatterplot Matrix and Its Implementation on Lisp-Stat

Masahiro Mizuta
(Division of Information and Data Analysis, Faculty of Engineering, Hokkaido University)

Tomoyuki Tarumi
(Division of Environmental Statistics, Faculty of Environmental Science
and Technology, Okayama University)

The aim of this paper is to present a method to implement Dynamic Scatterplot Matrix
on Lisp-Stat.

Lisp-Stat is a statistical environment based on the Lisp language and includes support
for an object-oriented programming system and dynamic graphics.

We have developed a graphical method: Dynamic Scatterplot Matrix. The method is
an extension of the scatterplot matrix with rotation. We survey Dynamic scatterplot
matrix and Lisp-Stat and then show the program list for Dynamic Scatterplot Matrix on
Lisp-Stat.

Key words: Dynamic graphics, rotation graph, statistical package.