C 言語と Mathematica による統計プログラミングの基礎

2009SE046 早川由宏
指導教員：白石高章

1 はじめに

本研究では, 統計解析を行うために有効な C 言語のプログラムの開発を行った. 多重比較統計量の上側 $100\alpha\%$ 点を求めるために二重積分の方程式を解く必要がある. その実現のため, 数式処理ソフトウェアの Mathematica を使う手順を示した.

2 C 言語での計算プログラム

統計学で必要とされる基礎的なプログラムに加え検定を行うプログラムを C 言語で作成した. 基礎的なプログラム例は以下のとおりである. 多くは文献 [3] のアルゴリズムを参考にした.

「標準正規分布の密度関数の値を求める」
「標本平均, 標本分散, 標本標準偏差, 標本歪度, 標本尖度を求める」
「2 次元データを入力し, 各成分の標本平均, 標本分散, 標本相関係数から最良線形単回帰直線, 残差平方和と寄与率を求める」
「n 個の観測値を入力し, 順序統計量を求め, 標本 $5, 10, 25, 50, 75, 90, 95$ パーセント点および標本範囲, 最大値, 最小値を計算する」
「t, χ^2, F, t 分布の上側確率の値を求める」
「Mersenne Twister(MT19937) により発生した擬似一様乱数から標準正規分布, 混合正規分布, 異常値をもつ混合正規分布, 対数正規分布, ロジスティック分布, 両側指数分布, 指数分布に従う乱数を生成する」

MT19937 は $2^{19937}-1$ という周期を持つ擬似乱数生成のソフトウェア (文献 [4]) で上記プログラムはその出力を利用している.

2.1 閉検定手順のプログラム

$X_{ij} \sim N(\mu_i, \sigma^2) \quad (j = 1, \cdots, n_i; i = 1, \cdots, k)$ とし, すべての X_{ij} は互いに独立とする. k を群の個数, n_i は第 i 群のサイズで $n = \sum_{i=1}^{k} n_i$ とする.

帰無仮説 $H_{(i,i')} : \mu_i = \mu_{i'}$ vs. 対立仮説 $H^A_{(i,i')} : \mu_i \neq \mu_{i'} \quad (1 \leq i < i' \leq k)$ に対する文献 [1] で提案された閉検定手順のプログラム (文献 [2]p.122 検出力の高い閉検定手順 I) を C 言語により作成した. そのアルゴリズムを述べる.

$\mathcal{U} \equiv \{(i,i') | 1 \leq i < i' \leq k\}$ とおくと帰無仮説のファミリー $\mathcal{H} \equiv \{H_{\boldsymbol{v}} | \boldsymbol{v} \in \mathcal{U}\}$ となる. さらに $\phi \subsetneq V \subset \mathcal{U}$ を満たす V に対して, $\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}}$ は k 個の母平均に関していくつかが等しいという仮説となる. $I_1, \cdots, I_J \ (I_j \neq \phi, \ j = 1, \cdots, J)$ を添え字 $1, \cdots, k$ の互いに素な部分集合の組とし同じ $I_j \ (j = 1, \cdots, J)$ に含まれる添字をもつ母平均は等しいという帰無仮説を $H(I_1, \cdots, I_J)$ で表す. このとき, $\phi \subsetneq V \subset \mathcal{U}$ を満たす任意の V に対して, ある自然数 J と上記のある $I_1, \cdots, I_J$ が存在して

$$\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}} = H(I_1, \cdots, I_J)$$

が成り立つ. $T(I_j) \equiv \max_{i<i', i, i' \in I_j} |T_{ii'}| \ (j = 1, \cdots, J)$,

$$T_{ii'} \equiv \frac{\bar{X}_{i\cdot} - \bar{X}_{i'\cdot}}{\sqrt{V_E(\frac{1}{n_i} + \frac{1}{n_{i'}})}}, V_E \equiv \frac{1}{n-k} \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_{i\cdot})$$

とおき水準 α の帰無仮説 $\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}}$ に対する検定を考えることができる.

(1) $J \geq 2$ のとき $l = l_1, \cdots, l_J$ に対して $\alpha(M, l) \equiv 1 - (1-\alpha)^{l/M}$ で $\alpha(M, l)$ を定義する. $1 \leq j \leq J$ となるある整数 j が存在して $ta(l_j, m; \alpha(M, l_j)) < T(I_j)$ ならば帰無仮説 $\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}}$ を棄却する.

(2) $J = 1$ のとき $ta(M, m; \alpha) < T(I_1)$ ならば帰無仮説 $\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}}$ を棄却する.

(1),(2) の方法で, $(i,i') \in V \subset \mathcal{U}$ を満たす任意の V に対して, $\bigwedge_{\boldsymbol{v}\in V} H_{\boldsymbol{v}}$ が棄却されるとき, 多重比較検定として $H_{(i,i')}$ を棄却する. 本研究で作成したプログラムは $k = 5$ 以下, 第 i 群の標本数は 100 以下でのみ動作する. 今回検定するデータは $n = 75, k = 5$ となるので以降はそのデータを解析するための方法を述べる.

3 Mathematica による計算

検定を行うために $ta(l_j, m; \alpha(M, l_j))$ を求める.

3.1 分布関数の定義

関数 $TA(t|l_j)$ を

$$TA(t|l_j) \equiv l_j \int_0^\infty A(ts|l_j) g(s) ds,$$

$$A(ts|l_j) \equiv \int_{-\infty}^{\infty} \left\{ \Phi(x) - \Phi(x - \sqrt{2} \cdot ts) \right\}^{l_j - 1} d\Phi(x)$$

と定義し, $ta(l_j, m; \alpha(M, l_j))$ は方程式

$$TA(t|l_j) \equiv 1 - \alpha(M, l_j)$$

を満たす解とする. ただし $m \equiv n - k$, $\Phi(x)$ は $N(0,1)$ の分布関数とし,

$$g(s) \equiv \frac{m^{m/2}}{\Gamma(m/2) 2^{m/2-1}} s^{m-1} e^{-ms^2/2}$$

とする.

3.2 Mathematica での定義

Mathematica で $TA(t|l_j)$ は文献 [5] を参考に「:=」や「^」,「*」などの記号を使って定義をする．式は数値積分を用いて表現すると

```
f[x_] = Exp[-x^2/2]/Sqrt[2 Pi]
F[x_] = Integrate[f[z], {z, -Infinity, x}]
A[t_] := NIntegrate[f[x] (F[x] - F[x -
Sqrt[2]*t])^(l - 1), {x, -5, 5}]
G[a_] := Integrate[x^(a - 1) Exp[-x],
{x, 0, Infinity}]
g[s_] := m^(m/2)/(G[m/2] 2^(m/2 - 1))*
s^(m - 1)*Exp[-m*s^2/2]
TA[t_] := l*NIntegrate[A[t*s]*g[s], {s, 0, 5},
Method -> DoubleExponential]
Off[NIntegrate::inumr]
h[x_] := TA[x] - 1 + al
al := 1 - (1 - a)^(1/M)
```

となる．

「Method -> DoubleExponential」は二重指数求積法を使う指定コマンドで無指定時と比べ計算が高速化する．「Off[NIntegrate::inumr]」は警告を非表示にするコマンドで計算を継続させる．これらの式は複雑で方程式を解く専用のコマンド Solve が使えないので二分法を用いる．

3.3 二分法のアルゴリズム

この研究では, 方程式の近似解を求める方法として, 二分法を使用する．ただし, 分布関数は連続で単調増加である性質を使う．$h(x) = 0$ を解く際, 区間 $[p, q]$ 内に解がある時 $h(p)h(q) \leq 0$ が常に成り立つのでこの区間を狭くすれば良い．$h((p+q)/2) < 0 \Rightarrow$ 区間を $[(p+q)/2, q]$, $h((p+q)/2) \geq 0 \Rightarrow$ 区間を $[p, (p+q)/2]$ と新しく定め $q - p \leq 10^{-5}$ 程度まで繰り返す．Mathematica の繰り返しコマンド For を用いて最初の区間の決定は

```
For[p = 1, ! (h[p] < 0 && 0 < h[p + 1]), p++]
```

で区間を縮めるには

```
For[q = p + 1, Abs[p - q] > 10^-5,
If[h[(p + q)/2] < 0, p = (p + q)/2, q = (p + q)/2]]
```

である．

3.4 解を求める

Mathematica で 1 つだけ解を求めるには関数を定義後単純に二分法を行う．複数の解を求めさせるには For 文をうまく使えば良い．解 $ta(l_j, m, \alpha(M, l_j))$ は l, m, α, M の 4 元で表されている．そこで α, m を固定し M, l を動かすことで複数解を得る．コマンドは

```
a = 0.01; m = 70
For[M = 5, M != 1,
For[l = 2, l != 6, If[l == M - 1, l++];
For[For[p = 1, !(h[p] < 0 && 0 < h[p + 1]),p++];
s = p; t = p + 1, Abs[s - t] > 10^-5,
If[h[(s + t)/2] < 0,
s = (s + t)/2, t = (s + t)/2]];
Print["M=", M, ",l=", l "のとき t=", N[s, 4]];
l++; If[M < l, Break[]]]; M--]
```

とすれば $2 \leq M \leq 5$, $2 \leq l \leq 5$ で必要な解 7 個を得られる．

3.5 計算結果

Mathematica で求めた $ta(l, m; \alpha(M, l))$ の一例を以下に示す．

表 1 $\alpha = 0.01$ のときの $ta(l, 70; \alpha(M, l))$ の値

$M \backslash l$	2	3	4	5
5	2.976	3.187	*	3.384
4	2.898	*	3.229	
3	*	3.011		
2	2.648			

4 C 言語プログラムによる閉検定手順のアルゴリズム

プログラムは main 関数の中に検定をする関数がある．main 関数は変数宣言で始まり Mathematica で得た解をメモリーに格納する．標本がメモリーに格納されると統計量 $T_{ii'}$ などを計算し, ファミリーを 4 次元の配列に格納する．ファミリーの作成は検定の関数と独立しており k の値によって場合分けして出力する．今回使う $k = 5$ は $k = 2, 3, 4$ と同様のアルゴリズムで作成後足りない帰無仮説を追加する．検定する関数は得られたファミリーに含まれる i, i' について $|T_{ii'}|$ の最大値を求め添字を控える．最後に $T_{ii'}$ と $ta(l_j, m; \alpha(M, l_j))$ を比較し H_v が棄却されない場合に検定の関数を終了する．この検定の関数を二重の繰り返しで検定を繰り返し行う．

5 おわりに

C 言語では 2 重積分など複雑な計算は簡単にできない．C 言語の文法, 数学の近似法などの知識に加えオーバーフローなどコンピュータ特有のエラーにも対処する必要があり, 精度保証は難しいが情報の整理がしやすい．一方 Mathematica は C 言語のように複雑な手続きはさせにくいが精度の良い近似値が容易に得られる．それぞれメリット, デメリットを考え適切なソフトを使うことが望ましいことがわかった．

参考文献

[1] 白石高章：『多群モデルにおけるすべての平均相違に関する閉検定手順』．計量生物学，2011.

[2] 白石高章：『多群連続モデルにおける多重比較法』. 共立出版，東京，2011.

[3] 白石高章：『FORTRAN による統計プログラミング』. 2011/6/30, 講義ノート.

[4] Mersenne Twister の Web Page http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/mt.html

[5] Wolfram Mathematica のドキュメントセンター http://reference.wolfram.com/mathematica/tutorial/VirtualBookOverview.html