Cox比例ハザードモデルの変数選択

2012年7月9日

日本ニューメリカルアルゴリズムズグループ

Web サイト： www.nag.co.uk

www.nag.com

www.nag-j.co.jp

問い合わせ先： sales@nag-j.co.jp

ヘルプデスク： naghelp@nag-j.co.jp

目次

1 概要

Cox 比例ハザードモデルは、死亡や故障といったイベントまでの時間と、共変量として知られている多くの説明変数とを関連づけています。観測値のいくつかは右側打ち切りの場合があります。つまり死亡（故障）までの正確な時間が不明であり、観測された時間より大きいということだけしかわかりません。

t_i（ $i = 1, 2, \ldots, n$ ）が m 個の共変量のベクトル z_i をもつ i 番目の観測値の死亡（故障）時間もしくは打ち切り時間だとします。また打ち切りと死亡（故障）のメカニズムは独立していると仮定します。もし個体が時間 t まで生存した場合、ハザード関数 $\lambda(t,z)$ は共変量 z を持つ個体が時間 t で死亡する確率です。Cox 比例ハザードモデル[1]では $\lambda(t,z)$ は以下の式で表されます：

$$\lambda(t,z) = \lambda_0(t)exp(z^t\beta + \omega)$$

ここで λ_0 はベースラインハザード関数、不特定の時間関数、β は未知のパラメータのベクトル、ω は既知のオフセットです。

Cox 比例ハザードモデルをフィッティングする NAG ルーチンはFortran ライブラリをお使いの場合はG12BAF で、C ライブラリをお使いの場合は g12bac です。この資料ではCox 比例ハザードモデルに含める説明変数を選択する自動変数選択法に関する３つの主な手法を実装するルーチンの使い方をご説明します。３つの手法とは、前進選択法、後進選択法、ステップワイズ選択法です。

これらの分析で使用される NAG ルーチンを説明する際に Fortran ライブラリに重点を置いていますが、サンプルプログラムやコードスニペットは、Fortran ライブラリ と C ライブラリの両方について提供されています。

2 前進選択法

前進選択法の処理は以下のとおりです：

1. ヌルモデル（説明変数をもたないモデル）から開始する。
2. モデルに含まれない各変数のスコアSi を計算する。そのスコアは既にモデルに含まれている全ての変数で調整される。
3. $S_j \geq S_i$ （全ての i で、$i \neq j$ ）となる j を見つける。つまり S が最大となる変数（モデルに含まれない）を見つける。j 番目のスコアに関連する変数 Z_j を選ぶ。
4. S_j に関連する p 値である、p を計算する。
5. $p > p_a$ の場合、ステップ 8 へ進む。

6. モデルに変数 Z_j を追加する。
7. モデルに含まれない変数がまだ残っている場合、ステップ 2 へ戻る。
8. 終了する。

ステップ 3 で最も高いスコアの変数が２つ以上ある場合、そのうちの一つが選択されます。この選択は任意です。

前進選択法の処理を実行するためにはスコアリング統計量 S とカットオフ p_a を選択する必要があります。Cox 比例ハザードモデルで前進選択法を実行する際は、ある著名な統計パッケージでは S に対してScore 検定統計量が使用され、p_aのデフォルト値として 0.05 が使用されます。

2.1 Score 検定統計量

Score 検定統計量 S は以下の式で表されます：

$$\mathrm{S}(\beta_0) = \left(\frac{\partial \ln L}{\partial \beta}(\beta_0)\right)^T \left(\frac{\partial^2 \ln L}{\partial \beta^2}(\beta_0)\right)^{-1} \left(\frac{\partial \ln L}{\partial \beta}(\beta_0)\right)$$

このとき lnL は対数尤度関数です。S は以下の式で表される仮説の検定に使用されます：

$$H_0 : \beta = \beta_0 \text{ 対 } \quad H_1 : \beta \neq \beta_0$$

帰無仮説H_0のもとでは $\mathrm{S}(\beta_0)$は ν が検定される変数の数を表す χ^2_ν 分布をもちます。

2.2 Score 検定の計算

G12BAF には多くの出力パラメータがあります。それらにはパラメータ COV で返される β の推定値と関連する分散共分散行列Σが含まれます。また、パラメータ SC で返されるスコア関数 $\mathrm{U}(\beta)$ の値も含まれます。スコア関数 U と Score 検定統計量 S は名前が共通ですが、スコア関数 U は前進選択処理で使用される Score 検定統計量 S とは異なる点に注意が必要です。G12BAF は主に共分散係数パラメータ β を推定するために設計されていますが、反復数 MAXIT をゼロに設定することによって任意の β の値に対する他の出力パラメータの値を計算するのに使用することができます。この機能は Score 検定統計量を計算するために使用します。

G12BAF で返されるスコア関数は以下の式で表されます。

$$\mathrm{U} = \mathrm{U}(\beta) = \frac{\partial \ln L}{\partial \beta}$$

また共分散行列は以下の式で表されます。

$$\Sigma = I(\beta)^{-1},$$

$$\mathrm{I}(\beta) = -\frac{\partial^2 \ln L}{\partial \beta^2}$$

従ってScore 検定統計量 S は以下の式で表されます。

$$\mathrm{S} = U^T \Sigma U$$

これは次に示すコードスニペットを用いて計算することができます：

Fortran:

```
! F06PEF: calculate COV * SC
Allocate (covsc(ip))
Call dspmv('Upper',ip,1.0_wp,cov,sc,1,0.0_wp,covsc,1)
! F06EAF: calculate transpose(SC) * COV * SC
! which gives the Score test statistic, S
s = ddot(ip,sc,1,covsc,1)
```

C:

```
/* f16pec: calculate cov * sc,
using the default error structure, which will terminate if an error
occurs as we should only ever be supplying valid input arguments, so
routine should not fail */
covsc = NAG_ALLOC(ip, double);
nag_dspmv(Nag_ColMajor,Nag_Upper,ip,1.0,cov,sc,1,0.0,covsc,1,
NAGERR_DEFAULT);
/* calculate transpose(sc) * cov * sc, which gives the
Score test statistic */
for (i = 0, s = 0.0; i < ip; i++) s+= sc[i] * covsc[i];
```

ここで ip はモデルの変数の数です。S に関連するp 値は以下を用いて得ることができます：

Fortran:

```
p = g01ecf('Upper',s,df,ifail)
```

C:

```
p = g01ecc(Nag_UpperTail,s,df,&fail);
```

ここで df は Score 検定統計量に関する自由度 ν です。

モデルがm個のパラメータを含んでいる場合、Score 検定統計量は２種類の仮説を検定するのに使用されると考えられます：

1. $\beta_i = 0$ （$i = 1,2,\ldots,m$） である。モデルの全パラメータがゼロかどうかを同時に検定するため、これはグローバル仮説と呼ばれています。グローバル仮説 $\nu = \mathrm{m}$ を検定します。
2. $\beta_i = \hat{\beta}_i$ となる場合 $\beta_j = 0$ $(\mathrm{i} \neq \mathrm{j})$ である。他のパラメータは値がある場合に一つのパラメ

ータはゼロであり、従って $\nu = 1$ であるという仮説を検定します。これは前進選択法の処理のステップ 4 でp 値を計算する際に使用される仮説です。

3 後進選択法

後進選択法の処理は以下のとおりです：

1. フルモデル（全ての説明変数を含むモデル）で開始する。
2. モデルの各変数のスコア W_i を計算する。このスコアはモデルの他の全ての変数で調整される。
3. $W_k \leq W_i$（全ての i で、$i \neq k$）となる k を見つける。つまり W が最小となる変数（モデルに含まれている）を見つける。k 番目のスコアに関連する変数 z_k を選ぶ。
4. W_i に関連するp 値である、p を計算する。
5. $\mathrm{p} < p_d$ の場合、ステップ 8 へ進む。
6. モデルから変数z_kを除去する。
7. モデルにまだ変数がある場合、ステップ 2 へ戻る。
8. 終了する。

ステップ 3 で、最も低いスコアをもつ２つ以上の変数がある場合、そのうちの一つが選択されます。この選択は任意です。

後進選択法の処理を実行するため、スコアリング統計量 W とカットオフp_dを選択する必要があります。Cox 比例ハザードモデルで後進選択法を実行する際は、ある著名な統計パッケージでは W に対し Wald 検定統計量が使用され、 p_d のデフォルト値として 0.05 が使用されます。

3.1 Wald 検定統計量

Wald 検定統計量 W は以下の式で表されます：

$$\mathrm{W} = (\hat{\beta} - \beta_0)^T \left(\frac{\partial^2 \ln L}{\partial \beta^2} (\hat{\beta}) \right) (\hat{\beta} - \beta_0)$$

この場合$\hat{\beta}$ はモデルパラメータ β の最尤推定値で、$\ln L$ は対数尤度関数です。検定統計量W は以下の式で表される仮説を検定するのに使用することができます：

$$H_0 : \beta = \beta_0 \text{ 対 } H_1 : \beta \neq \beta_0$$

帰無仮説のもとでは、W はνが検証される変数の数を表す χ^2_ν 分布を持ちます。

3.2 Wald 検定統計量の計算

Wald 検定統計量の計算の際に、G12BAF が任意のパラメータベクトルの推定値 $\hat{\beta}$ に対する共分散行列Σを返すという点を利用します。Wald 検定統計量はそのため以下の式で表されます：

$$W = (\hat{\beta} - \beta_0)^T \Sigma^{-1} (\hat{\beta} - \beta_0)$$

共分散行列Σを直接反転させることはせず、$\Sigma = U^T U$ といったコレスキー分解を使用し上三角行列U を得て連立方程式 $Ux = \hat{\beta} - \beta_0$ を解き、最終的に $W = x^T x$ を計算します。
Σ は共分散行列であり多くの場合正定値ですが、この場合は半正定値です。標準のコレスキー分解は正定値行列でのみ作用します。Fortranライブラリには完全ピボット選択でコレスキー分解を実行するDPSTRF ルーチンがあります。このルーチンはまれな半正定値行列の処理をすることができます。
残念ながら、Mark 23 では C ライブラリにはこのようなルーチンが含まれていませんが、Mark 24 では含まれるようになります。従ってWald 検定統計量を計算するためのコードは Fortran と C とで若干異なります。

Fortran のコードでは F07KDF を用いて共分散行列を分解します。DPSTRF が非圧縮形式の行列を必要とするのに対して G12BAF は圧縮格納形式で共分散行列を格納するため、最初に行列を解凍する必要があります。

```
! copy COV from packed format to upper triangular format
Allocate (ccov(ip,ip))
k = 0
Do j = 1, ip
Do i = 1, j
k = k + 1
ccov(i,j) = cov(k)
End Do
End Do
```

以下に示す分解ルーチンの呼び出しの前に行います。

```
! use default tolerance in F07KDF
tol = 0.0_wp
! F07KDF: factorize COV so that COV = transpose(U) * U, where
! CCOV = COV on entry and U on exit
Allocate (work(2*ip),piv(ip))
Call dpstrf('Upper',ip,ccov,ip,piv,rank,tol,work,info)
```

ここでip は現在のモデルの共変量の数です。C のコードでは圧縮格納形式の正定値行列のコレスキー分解を行う nag_dpptrf を使用します。そのためこの場合共分散行列を解凍する必要がありません。ただし、nag_dpptrf は入力行列を上書きするので共分散行列をコピーする必要があります。

```
/* copy cov */
lcov = ip*(ip+1)/2;
ccov = NAG_ALLOC(lcov, double);
for (i = 0; i < lcov; i++) ccov[i] = cov[i];
/* f07gdc: factorize cov so that cov = transpose(U) * U, where ccov = cov
on entry and U on exit */
nag_dpptrf(Nag_ColMajor,Nag_Upper,ip,ccov,NAGERR_DEFAULT);
```

デフォルトの NAG エラー構造体 NAGERR_DEFAULT を使用しているため、nag_dpptrf はΣが半正定値である場合に終了します。分解が実行されたらバックソルバーを使用し、$x = U^{-T}(\hat{\beta} - \beta_0)$ を得ます。

Fortran:

```
! pivot B into PB
Allocate (pb(ip))
Do i = 1, ip
pb(i) = b(piv(i))
End Do
! F06YJF: solve CCOV x = PB for x, putting the result in PB
Call dtrsm('Left','Upper','Transpose','NonUnit',rank,1,1.0_wp,ccov,ip, &
pb,ip)
```

C:

```
/* copy b */
cb = NAG_ALLOC(ip, double);
for (i = 0; i < ip; i++) cb[i] = b[i];
/* f16plc: solve ccov * x = cb for x, putting the result in cb */
nag_dtpsv(Nag_ColMajor,Nag_Upper,Nag_Trans,Nag_NonUnitDiag,ip,1.0,ccov,cb,
1,NAGERR_DEFAULT);
```

上記のようにFortran コードでは、パラメータ推定値 b をコピーする際に DPSTRF により実行されるピボット選択を考慮する必要があります。バックソルバーDTRSM と nag_dtpsv は、b がプログラム開始時に $(\hat{\beta} - \beta_0)$ の値をもち終了時に x の値をもつようb を上書きするため、Fortran と C の両方の場合でコピーが必要となります。

最終的に W は以下のように計算されます：

Fortran:

```
w = ddot(rank,pb,1,pb,1)
```

C:

```
for (i = 0, w = 0.0; i < ip; i++) w+= cb[i] * cb[i];
```

C ライブラリは２つのベクトルの内積を実行するドキュメント化された DDOT と同等のルーチンがありません。従って C のコードスニペットではforループを使用する必要があります。W に関連する p-値 は以下を用いて得ることができます：

Fortran:

```
p = g01ecf('Upper',s,df,ifail)
```

C:

```
p = g01ecc(Nag_UpperTail,s,df,&fail);
```

ここで df は Wald 検定統計量に関連する自由度 ν です。

モデルがm 個のパラメータを含んでいる場合、Wald検定統計量は２種類の仮説の検定に使用されると考えられます：

1. $\beta_i = 0$ （$i = 1,2,\ldots,m$）である。モデルの全パラメータがゼロかどうかを同時に検定するため、これはグローバル仮説と呼ばれています。グローバル仮説 $\nu = \mathrm{m}$ を検定します。
2. $\beta_i = \hat{\beta}_i$ となる場合に $\beta_j = 0$ $(\mathrm{i} \neq \mathrm{j})$ である。他のパラメータは値がある場合に一つのパラメータはゼロであり、従って $\nu = 1$ であるという仮説を検定します。これは後進選択法の処理のステップ 4 で p 値を計算する際に使用される仮説です。この場合、Wald 検定統計量は以下のように簡略化されます。

$$\mathrm{W} = \frac{\hat{\beta}_i^{\,2}}{\sigma_{ii}}$$

ここでσ_{ii} はΣの (i, i)番目の要素です。

4 ステップワイズ選択法

私たちが考慮する最後の変数選択法はステップワイズ選択法です。ステップワイズ選択法は後進消去に続いて前進選択法が実行されるため前進選択法と後進選択法の混合と見なすことができます。ステップワイズ選択法の処理は以下のように要約されます：

1. ヌルモデル（説明変数のないモデル）から開始する。
2. モデルの各変数のスコア Si を計算する。このスコアはモデルの他の全ての変数で調整される。
3. $S_j \geq S_i$ （全ての i で、$i \neq j$）となる j を見つける。つまりS が最大となる変数（モデルに

含まれない）を見つける。j 番目のスコアに関連する変数 Z_j を選ぶ。

4. S_j に関係する p 値である、p を計算する。
5. $\mathrm{p} > p_a$ の場合、ステップ 8 へ進む。
6. モデルに変数 Z_j を追加する。
7. モデルの各変数のスコア W_i を計算する。このスコアはモデルの他の全ての変数で調整される。
8. $W_k \leq W_i$（全ての i で、$i \neq k$）となる k を見つける。つまり W が最小となる変数（モデルに含まれる）を見つける。k 番目のスコアに関連する変数 z_k を選ぶ。
9. W_i に関連するp 値である、p を計算する。
10. $\mathrm{p} \geq p_d$ の場合、
    a. モデルから変数 z_k を除去する。
    b. $z_k = z_j$ ならば、つまり今回の反復で追加された変数が除去された場合、ステップ 12 へ進む。
11. モデルに含まれていない変数がまだある場合、ステップ 2 へ戻る。
12. 終了。

ステップワイズ選択法の処理を実行するためには、２つのスコアリング統計量 S と W と２つのカットオフ p_a と p_d を選択する必要があります。スコアリング統計量は同じでも構いませんが、ある著名な統計パッケージではS に対してScore統計量が使用され、W に対してWald 検定統計量が使用されます。また p_a と p_d に対して同じデフォルト値 0.05 が使用されます。Score 検定統計量とWald 検定統計量の両方の計算方法を既にわかっていますので、以前に開発したコードを使用してステップワイズ選択法を実行することができます。

5 追記

Cox 比例ハザードルーチン G12BAF と g12bac は両者ともデザイン行列が提供される必要があります。 全ての共変量が２値かあるいは連続（各共変量の自由度が1である）の場合、デザイン行列はデータを持つ行列と同じです。
共変量のいくつかがカテゴリカル（離散値の一つをもつ）であり２値でない場合（離散値が２つ以上の値をもつ）、ダミー変数を得るために事前処理が行われる必要があります。ダミー変数とそれらの生成のしかたの説明については G04EAF のドキュメントのセクション３をご参照下さい。

このホワイトペーパーで提供されているサンプルコードは全ての共変量の自由度が１であると仮定しています。そうでない場合、何らかの再コーディングが必要となります。再コーディングにはダミー変数の追加やモデルからの除去が可能である必要があります。例えば、共変量が k 個の可能値のうちの一つをとる場合、k － 1 個のダミー変数で表すことができます。通常ダミー変

数の一部を含めることは意味がないため、これらのk － 1 個のダミー変数はまとめてモデルに追加されるか、もしくはモデルから除去される必要があります。さらにp 値の計算の際に使用された自由度については自由度の追加を検討する必要があります。

全ての共変量の自由度が1である（つまり後進選択法のステップ 4 で$\nu = 1$）と仮定しているサンプルコードには副次的な効果があり、Wald 検定統計量の簡易バージョンだけが必要となります（つまり$W = \hat{\beta}_i^{\,2} / \sigma_{ii}$）。しかし、簡素化が可能ではないより複雑なケースでの例を示すため、Wald 統計量を使用してグローバルな帰無仮説を検定するサンプルを含めています。

利用した NAG ルーチン

Fortran ライブラリ

F06EAF	２つの実ベクトルの内積
DDOT	２つの実ベクトルの内積
F06PEF	行列ベクトル積，実対称圧縮行列
DSPMV	行列ベクトル積，実対称圧縮行列
F06YJF	多重右辺をもつ連立方程式の解，実三角係数行列
DTRSM	多重右辺をもつ連立方程式の解，実三角係数行列
F07KDF	実対称半正定値行列のコレスキー分解
DPSTRF	実対称半正定値行列のコレスキー分解
G01ECF	カイ二乗分布に対する確率の計算
G12BAF	コックスの比例ハザード・モデルのフィット

C ライブラリ

f07gdc	実対称正定値行列のコレスキー分解，圧縮型格納形式
nag_dpptrf	実対称正定値行列のコレスキー分解，圧縮型格納形式
f16pec	行列ベクトル積，実対称圧縮行列
nag_dspmv	行列ベクトル積，実対称圧縮行列
f16plc	連立方程式，実三角圧縮行列
nag_dtpsv	連立方程式，実三角圧縮行列
g01ecc	カイ二乗分布に対する確率の計算
g12bac	コックスの比例ハザード・モデルのフィット

参考文献

[1] D R Cox. Regression models in life tables (with discussion). *J. Roy. Statist. Soc. Ser. B*, 34:187-220, 1972.