ML (非線形オプション) 対数尤度方程式名 ;

OR

ML (非線形オプション) プロシジャー名 パラメータリスト ;

機能 :

ML は一般的な最尤推定プロシジャーです。この方法は、対数尤度関数を TSP 方程式の形 (FRML) や、プロシジャーでの対数尤度の評価 (PROC) に書くことができるならば、識別可能なモデルのパラメータ推定に自由に使うことができます。

使用法:

FRML 法 最も簡単な方法では、LOGL を従属変数とする対数尤度関数を FRML で書きます。この方程式の観測期間は、現在指定されている SMPL によって定められます。期間によって使われる方程式が変わるような場合は、ダミー変数を個々の方程式に掛けて、個々の方程式の和の方程式を書きます。このダミー変数によって、適切な観測期間にあった方程式が選ばれるようにします。この様な場合分けは、従属変数が離散選択変数の場合に使われます。観測値は通常 *i.i.d.* ですが、時間に依存して値を変える一般的なダミー変数を使うことにより、標本期間の分割に応じて尤度関数を変化するように作ることができます。もちろん対数尤度は標本期間に関して加算的で分離可能でないといけません。

次に PARAM 文によって LOGL 方程式に含まれるどの変数が推定されるべきかを定め、希望するなら初期値を与えることもできます。PARAM に続いて標準の NONLINEAR オプションと尤度方程式を定める方程式名を与えます。ML はこの関数を標準的な傾斜 (gradient) 法によってパラメータに関して最大化します。実際に使われるヘッセ行列は、HITER オプションに依存して決められます。特に指定がなければ BHHH 法、スコアリング法、が使われます。しかしその場合も対数尤度方程式の一次導関数の標本共分散行列が、期待値の代わりに使われています。

FRML 法のよい応用例:

1. 両側が切断されている TOBIT モデルなど、CNORM() を含む切断回帰モデル。
2. PROBIT, TOBIT, LOGIT などの非線形方程式。このコマンドは係数間の制約やパラメータを固定する場合にも使われます。
3. 自分で書いた FORTRAN プログラムによって最大化を行う場合は、その二次導関数が正しいかどうか調べます。
4. LOGL = ABS(Y-XB) のようなロバストモデルに使われます。
5. 最小化問題。(最大化問題の符号を逆にした場合です。)
6. TSP が認識する全ての関数を使った一般的な最大尤度問題 (ガンマ, χ^2 , ベータ, t , そして F 密度関数に対して用いられる, SQRT, POS, FACTORIAL, GAMMA の様な新しい関数があります)。大きなデータセットを使う通常フォートランで書かれるような複雑な尤度関数でも、ML によってずっと少ないコストで容易に計算できます。もちろん演算時間は増加するのですが、プログラミングの時間

が激減しますから、全体として計算が簡単になります(そして、CPUの相対価格は通常小さく、また小さくなる傾向にあります)。解析的な導関数を使う最大化の計算は、数値演算による導関数を使った最大化の計算よりずっと高速です。数値微分を使うとプログラミングは簡単にできますが、計算は遅くなります。計算時間の比較については下の表をみてください。

PROC法

時として、対数尤度関数を1つのFRMLで表すことが、(EQSUBを使っても)きわめて難しい場合や不可能な場合があります。このいくつかの例については下のリストを参照して下さい。この形のMLコマンドに対しては、対数尤度を計算するためのPROCを書いて、それを@LOGLに保存します。このPROCの名前をMLコマンドの(任意のオプションを与えた後に)最初の引数として与え、続いて推定するPARAMのリストをおきます。PROCを書くときは、PARAMに関する制約をチェックしてから推定を開始するようにします。もしどれかの制約が満たされない場合は、PROCから出る前に@LOGLに@MISSを入れます。もし残差系列のように途中の結果を形成するのに倍精度を用いたい場合は、OPTIONS DOUBLE;を用いることを推奨します。

FRMLの代わりにPROCを用いる場合の主な欠点は、解析的な微分を利用できないことです。しかし、数値微分(HITER=FとGRAD=C2が既定値)でしばしば十分と思われます。あるとすれば、わずかな欠点は、コマンド行で明示的に推定すべきPARAMをリストしなければならない点です。これは、すべてのPARAMを一か所に集めた単一のFRMLは存在しないので必ず必要になります。

PROC法のよい応用例は、次のような場合でしょう。

1. ARMAやGARCHのような時系列モデルで、方程式が逐次的である(以前の期の残差や分散に依存している)場合のモデル。KALMANコマンドによって評価されたモデルもこのカテゴリーに当てはまります(MLは、従ってハイパー・パラメータの推定を含みます)。
2. FIMLのような多元方程式モデル。これらはヤコービアン、逆行列、行列式の計算を含み、明示的に対数尤度関数の中に入れられます(ヤコービアンが疎でない限り、大体4本以上の方程式では非常に難しい)。
3. シミュレーションによる多変量正規積分や、TSPに組み込まれていない関数のような評価するのにいくつかのコマンドを必要とするモデル。このクラスの他の例としては集約対数尤度関数があります。
4. 複雑な制約のあるモデル。典型的な例はARCHモデルであり、この場合条件付き分散は各観測値に対して正でなくてははいけません。

よくない応用は次の分野です。

1. TSPにすでに入っている線形モデル(PROBIT, TOBIT, LOGIT, SAMPSEL)。通常のTSPコマンドはずっと効率的で、数値的問題に対してずっと抵抗力があり、多くの場合もっとよい推定値を得、モデル特有の統計量を与えます。以下の使用時間の例を参照して下さい。

この便利さのコストをCPU時間で計ると、385観測値、8変数のProbit分析をVAX 11/780で計算した例は、次の通りになります。

CPU時間(秒) プロシジャー

05.24	PROBIT コマンド
65.65	ML(HITER=B,HCOV=N)
75.97	ML(HITER=N,HCOV=N)

教訓として、Fortran で書かれた推定プログラムがある場合には ML は用いるべきではないが、そのようなプログラムを開発する時間が惜しいなら有用であるということでしょう。さらに、この場合スコア法はニュートン法よりはいくらか早いですが、ニュートン法の方がより強力です(イタレーションが少ない)。

助言:

1. $\text{Log}(x \leq 0)$ あるいは $\text{Exp}(x > 88)$ のようなことを避けるために、注意して方程式を書きます。(初期値を用いた) 最初の関数計算でこれが生じると致命的な誤りになります。(プログラムが自動的にもっと小さなステップサイズを用いる) イタレーション中なら致命的には成りませんが、効率はよくありません。しばしば、このような問題は尤度関数をパラメトライズし直す事によって避けることができます。この標準的な例は、 σ^2 の代わりに σ (あるいは σ^{-1}) を計算することです。
もし数値エラーが生じて、尤度関数を書き直すことができないなら、問題の観測値を除くために SELECT を用いてためしてください。収束したら、収束した値を初期値として用いて全標本を用いて再推定します。
2. 注意して初期値を選択する(前項参照)。
3. 方程式を書き直す手間を少なくし、もっと効率的なコードを得るために EQSUB を用いる。
4. “Working space =” のメッセージは、微分コードの長さの指標になる。
5. もし 2 階微分行列が特異なら、対数尤度関数に符号の誤りがあるかもしれない(逆行列のルーティンは 2 次微分行列が不定符号であることを仮定しています)。
6. 導関数を用いる場合には、用いる関数が微分可能かどうか確かめます。論理演算子はあらゆるところで微分可能ではありません。特定の点以外のすべての点で微分可能です。TSP はできる場合にはベストを尽くしますが、キंक(コーナー)で終わる場合は不安定になります。

例:

FRML 法 例えば PROBIT では、尤度方程式は $Y \leq 0$ については $\text{CNORM}(XB)$, $Y > 0$ については $(1 - \text{CNORM}(XB))$ です。つまり

```
GENR Y0 = Y <= 0 ;
GENR Y1 = Y > 0 ;
FRML EQ1 LOGL = Y0*LOG(CNORM(XB)) + Y1*LOG(1 - CNORM(XB)) ;
```

です。XB は後で指定することができます。あるいは EQSUB コマンドで指定することもできます。もちろん次の例で分かるように、非線形方程式も使えます。

```
FRML NLXB XB = B0 + B1*X1 + (B2/B1)*X2 ;
EQSUB EQ1 NLXB ;
ML EQ1 ;
```

PROC 法 次は単純な集約対数尤度関数で、タイム・トレンドの平均を推定し、関数の非線形性を減らすために分散パラメータを集約する例です。

```
OPTION DOUBLE; ? 残差が倍精度で保存されるようにする
SMPL 1,9;
TREND T; ? MSD T; あるいは OLSQ T C; と同じ結果を得る
PARAM MT,2;
ML NRMLC MT; ? ML コマンドの PROC 型
```

```

PROC NRMLC;
  E = T - MT;      ? 残差
  MAT SIG2=(E'E)/@NOB; ? sigma-squared(分散)
  SET PI=4*ATAN(1); ? tan(pi/4)=1
  SET @LOGL=-(@NOB/2)*(LOG(SIG2)+1+LOG(2*PI));
ENDPROC ;

```

オプション:

NONLINEARのセクションで説明されている、標準の非線形のオプションが使えます。指定がなければ、FRML法では HITER=B, HCOV=B が、PROC法では HITER=F, HCOV=F, GRAD=C2が使われます。

初期値は PARAM か SET で定められます。CONST コマンドを使えば、計算中パラメータの値を固定することができます。

アウトプット:

名前	タイプ	長さ	内容
@RNMS	リスト	パラメータ数	右辺の RHS 変数名
@LOGL	スカラー	1	尤度関数の対数
@IFCONV	スカラー	1	収束状態。(1は成功)
@NCOEF	スカラー	1	推定されるパラメータ数
@NCID	スカラー	1	識別されるパラメータの数
@COEF	ベクトル	パラメータ数	係数推定値
@GRAD	ベクトル	パラメータ数	log L の傾斜
@SES	ベクトル	パラメータ数	標準誤差
@T	ベクトル	パラメータ数	t 統計値
@VCOV	行列	パラメータ数×パラメータ数	推定された係数の分散-共分散

HCOV オプションを用いたときの別の名前については NONLINEAR の項目を見て下さい。

方法:

計算方法は、標準の傾斜法で、ユーザース・マニュアルの第6章に説明されています。簡潔に言えば、反復毎に尤度関数の傾斜によって定められる方向に新しいパラメータ値は動いていきます。傾斜は、その点における二次微分行列の推定値によって重みづけられています。計算の収束は、パラメータの変化が非常に小さいときに達成されたと判断されます。またこの変化の小ささは、TOL = のオプションによって変えることができます。

ML プロシジャーは、(FRML法では)通常解析的な一次(と二次)微分を使います。HITER=D, HCOV=D のオプションを指定すれば、数値的にも最大化を達成することができます。オプションについてのより詳しい説明は、NONLINEAR のセクションを参照して下さい(HCOV=N を指定すれば、解析的な二次微分を使った標準誤差が計算されます)。

参考文献:

Berndt, E. K., B. H. Hall, R. E. Hall, and J. A. Hausman, "Estimation and Inference in Nonlinear Structural Models," *Annals of Economic and Social Measurement*, October 1974, pp. 653-665.

Gill, Phillip E., Walter Murray, and Margaret H. Wright, *Practical Optimization*, Academic Press, New York, 1981.