非線形最適化を行なうSASの新プロシジャNLPの紹介

岸本　淳司

1. 非線形最適化とＮＬＰの概要

非線形な関数$f(x_1, x_2, \ldots, x_n)$について，その最小値（または最大値）と，そのときのパラメタの値を求める技法を非線形最適化と呼ぶ．一般にパラメタ間の関係に等式または不等式による制約条件を含むこともある．いわゆる数理計画法のうち線形計画でもないものの総称と考えることができる．また統計学の応用で，非線形最小二乗法または最尤推定法の解法に利用される（このときは制約は考えないことが多い）．非線形最適化には，問題の性質に応じてさまざまな解法のバリエーションが存在する．共通したアルゴリズムとして，なんらかの方法で定められた初期解からより良い解を反復して推定する逐次近似法がとられる．

NLP (Non Linear Programming) は，リリース6.07 (UNIX は6.09) のSAS/OR に追加された非線形最適化のための新しいプロシジャである．あたかも最適化手法のカタログのように数多くの手法群を網羅している．現在のバージョンでは．線形の等式または不等式による制約を与えることができる．将来のバージョンでは非線形の制約も可能になる予定である．

ＮＬＰで採用されている最適化手法群では，１つの例外を除いて，目的関数の１階の微係数（勾配）を必要とする．さらに２階の微係数（ヘッセ行列）を必要とする手法群もある．ＮＬＰでは，微係数の計算に次の３通りの方法がある．

- 内部関数コンパイラにより，解析的な導関数を自動的に計算する．
- 有限差分近似により微係数を計算する．
- 導関数の解析的な式を明示的に指定する．

前２者では，計算の速度は遅くなるが，導関数を指定する手間は省ける．

きしもと　じゅんじ　㈱ＳＡＳインスティチュートジャパン
〒104 中央区明石町６－４

ＮＬＰは，現実の非線形最適化問題の形式を考慮して，目的関数や制約式の入力や結果の出力が柔軟にできるようになっている．

2. ＮＬＰで実行できる最適化手法群

ＮＬＰプロシジャで実行できる最適化手法群には次のようなものがある．

- 小規模問題用（パラメタ数40まで）
  トラストリージョン法
  Newton-Raphson 法
  直線探索つき Newton-Raphson 法
- 中規模問題用（パラメタ数200まで）
  準 Newton 法 (BFGS, DFP)
  双準 Newton 法 (DBFGS, DDFP)
  ダブルドッグレッグ法
- 大規模問題用（パラメタ数200以上）
  共役勾配法
- 微係数が計算しにくいとき
  Nelder-Mead シンプレックス法
- 非線形最小二乗問題専用
  Levenberg-Marquadt 法
  ハイブリッド準 Newton 法
  Gauss-Newton 法

最初のグループは２階微係数（ヘッセ行列）を用いる手法群である．収束の効率はよいが，ヘッセ行列の計算には時間がかかるので，パラメタ数40程度までの小規模な問題に適している．Newton-Raphson 法はパラメタ数40までのときのデフォルトである．トラストリージョン法はGay (1983) とMore and Sorensen (1983) の提案に近いアルゴリズムであり，Newton-Raphson 法より遅いことが多いが，数値的にはより安定している．最適点でヘッセ行列が特異になる場合には，直線探索つきの Newton-Raphson 法もよい方法である．

２番目のグループは，２階微分を計算しないで１階微分のみを用いてヘッセ行列の近似を更新していく手法群

である．問題のパラメタが多くなると，目的関数と勾配の計算に比べて相対的にヘッセ行列の計算に時間がかかるようになる．したがって，比較的大規模な問題に関しては，最初の手法群に比べて収束に必要な反復数は多くなるものの，総合的な効率はよくなる．ただし，近似ヘッセ行列の保存のために Newton-Raphson法と同程度のメモリを必要とするため，パラメタ数200程度が上限となる．準 Newton 法と双準 Newton 法での更新公式は BFGS (Broydon, Fletcher, Goldfarb, Shanno) の方法とDFP (Davidon, Fletcher, Powell) の方法から選択する．直線探索の手法も12種用意されている．ダブルドッグレッグ法は，準 Newton 法と似ているが，ヘッセ行列のコレスキー因子の近似を勾配から計算する手法で，Dennis and Mei (1979) の提案による．

パラメタ数が200を越えるような大規模な問題の場合，パラメタ数の2乗のオーダーで増加するヘッセ行列（の近似）をメモリに収めておくことができなくなる．そのような場合は，収束は一般に遅いが，共役勾配法が唯一の解法となる．共役勾配法で利用可能な更新公式には，Powell-Beale, Fletcher-Reeves, Polak-Ribiere, 共役降下法の4種がある．直線探索の手法も準Newton法と同様12種類使える．

これまで紹介してきた手法はすべて微分の計算を必要としていた．もし導関数が非連続あるいはきわめて計算しにくいときには，微分計算を要しない Nelder-Mead シンプレックス法を使うことになる．ただし，この方法はパラメタの数が多いとき効率が悪く，パラメタ数が30程度までの小規模な問題に対して適用すべきである．さらに，この方法では線形式による一般的な制約をつけることができない．

最後のグループは最小二乗問題専用の手法群である．ヤコビ行列の積和を計算してヘッセ行列を近似するためその計算が容易にできるパラメタ数60程度までの小規模な問題に適している．Levenberg-Marquardt 法は安定性でハイブリッド準ニュートン法に優る．大きな残差があるときはハイブリッド準ニュートン法の方が速い場合がある．ハイブリッド準ニュートン法の3つのバージョン（Flecher and Xu 1987）はすべてサポートされている．直線探索法もオプションで選択できる．Gauss-Newton 法は一般に安定性に劣るので推薦できない．

3. ＮＬＰの簡単な実行例

ＮＬＰの簡単な実行例を示そう．例題は，最適化問題のテストに頻用される Rosenbrock 関数である．

$$f(x)=100(x_2-x_1^2)^2+(1-x_1)^2$$

初期値 $x^0=(-1.2,\ 1.0)$ として，$x^*=(1,\ 1)$ の点で最小値 $f(x^*)=0$ が得られる．この問題を素直にコーディングすると，次のようになる．

```
proc nlp;
   min f;
   parms x1=-1.2, x2=1.0;
   f=100*(x2-x1*x1)**2+(1-x1)**2;
run;
```

まず，PROC ステートメントで NLP プロシジャを呼び出す．デフォルトの最適化法として，Newton-Raphson 法が選ばれる（パラメタ数から決定される）．最適化法を明示的に示すには，TECH=NEWRAP とオプション指定を行なう．微係数の計算は，デフォルトにより解析的に行なわれる．収束条件や表示情報などの設定はすべてデフォルト設定による．MIN ステートメントでは，最小化したい変数を指定する．最大化するためには，代わりに MAX ステートメントを指定することになる．MIN または MAX ステートメントでは，複数の変数を指定することもできるが，その意味は後述する．PARMS ステートメントでは，目的関数のパラメタを指定する．ここでは同時に初期値も指定している．初期値を複数指定すると，反復過程に入る前にグリッドサーチを行なう．初期値を指定しなかった場合は，乱数により初期値が設定される．最後に目的関数を通常のプログラム言語と同様な方法で指定する．ここでは1行で目的関数を書いたが，必要なら配列やＤＯループなどを使ってロジックを記述することもできる．

上に示したプログラムをＳＡＳの環境で実行すると，解が OUTPUT ウィンドウに現われる．その中には，初期値とそのときの勾配，反復過程の履歴などが含まれる．最終的なパラメタ値と目的関数の値は次のように表示される．

Optimization Results
Parameter Estimates

	Parameter	Estimate	Gradient
1	$x1$	1.000000	0.000000404
2	$x2$	1.000000	−0.000000216

Value of Objective Function=3.120954E-16

4. 複数の関数の統合（最小二乗推定と最尤推定）

非線形最適化問題では，複数の関数 $f_1, \ldots, f_m$ の二乗和

$$f(x)=\frac{1}{2}\sum_{i=1}^{m} f_i^2(x)$$

または複数の関数の和

$$f(x)=\sum_{i=1}^{m} f_i(x)$$

を扱うことが多い．ＮＬＰでＭＩＮまたはＭＡＸステートメント中に変数を複数指定した場合，それらの値を二乗和か総和の形式で統合した目的関数を最適化する．具体的には，ＬＳＱオプションをつけると各変数の二乗和を，FSUM オプションをつけると各変数の和を目的関数として評価する．

たとえば Rosenbrock 関数の場合，次のようにプログラムを書くことができる．

```
proc nlp lsq;
  min f1 f2;
  parms x1=-1.2, x2=1.0;
  f1=10*(x2-x1*x1);
  f2=1-x1;
run;
proc nlp fsum;
  min f1 f2;
  parms x1=-1.2, x2=1.0;
  f1=100*(x2-x1*x1)**2;
  f2=(1-x1)**2;
run;
```

最小二乗推定では，実測値と予測値との差（残差）の二乗和を最小化する．たとえば，$y=(5\,3\,4\,1\,1)'$，$x=(1\,2\,3\,4\,5)'$ というデータに対して $y=a*\exp(b*x)$ という非線形なモデルを当てはめることを考えよう．このときは，次の目的関数を最小化するのが問題となる．

$$r=\sum_{i=1}^{n}\{y_i-a*\exp(b*x_i)\}^2$$

ところが，これを上の形式で書くとすると，データの件数だけプログラムステートメントが必要になり，またデータをいちいちプログラム中に書き込まなければいけないので不便である．残差の二乗和のように同一形式の関数を統合した目的関数を最適化するには，あらかじめ作成されたデータセットから係数を読み込むという方法を用いる．

```
data testl;
    input y x @@; cards;
5 1  3 2  4 3  1 4  1 5
;
proc nlp data=testl lsq;
  min r;
  parms a b;
  r=y-a*exp(b*x);
run;
```

同様に，最尤推定では対数尤度を最大化することが問題となる．たとえば，$y=(1\,2\,3\,4\,5)'$ というデータが正規分布 $N(\mu, \sigma^2)$ から発生したとして，パラメタの μ と σ^2 を推定するには，次の目的関数（対数尤度の−2倍）を最小化することになる．

$$-2\log L=\sum_{i=1}^{n}\{\log(2\pi\sigma^2)+\frac{(y_i-\mu)^2}{\sigma^2}\}$$

ＮＬＰでは次のようにコーディングできる．

```
data test2;
    input y @@; cards;
1 2 3 4 5
;
proc nlp data=test2 f sum;
    min f;
    parms m s2;
    f=log(2*3.14159*s2)+(y-m)**2/s2;
run;
```

5. 制約条件のつけかた

一般に数理計画法の一部として非線形最適化を考えた場合，パラメタ間の関係に等式または不等式の制約をつけることが多い．ＮＬＰには，各変数の許容範囲の境界を指定する方法と，任意の線形関数について等式または不等式の制約をつける機能がある．将来のバージョンでは非線形の制約式も可能になる予定である．

パラメータの境界制約は，BOUNDSというステートメントで指定する．変数と定数との間に＜＝，＞＝，＝のいずれか（意味は自明）の関係演算子を指定する．複数の制約をつけるときには，式をカンマで区切って指定する．複数の変数について共通の制約を与えるときや，上限と下限とを同時に指定するときには，省略記法を用いることができる．次の３つのステートメントは，同じ制約の別表現である．

```
bounds    0 <=x1, 0<=x2, 0<=x3,
          x1<=10, x2<=10, x3<=10;
bounds    0 <=x1 x2 x3, x1 x2 x3 <=10;
bounds    0 <=x1-x3 <=10;
```

より一般的に，線形式による等式または不等式の制約与えたいときには，LINCON ステートメントを指定する．このステートメントでは，線形式と定数との間に関係演算子を指定する．複数の制約を与えるときには，カンマを区切って指定するのも同様である．BOUNDSステートメントで指定する境界制約は，LINCON ステートメントでも指定できるが，2種類の表現方法があるのは便利なものである．

制約を含む非線形最適化の例として，Betts 関数をとりあげよう．

$$f(x)=0.1x_i^2+x_2^2-100$$

制約：$10x_1-x_2 \geqq 10$,

$$2 \leqq x_1 \leqq 50,\quad -50 \leqq x_2 \leqq 50$$

この関数は，$x^*=(2,\ 0)$ の点で最小値 $f^*=f(x^*)=-99.96$をとる．初期値には $x^0=(-1,-1)$ を指定する(これは非許容である)．この問題は，ＮＬＰでは次のように記述できる．

```
proc nlp;
   min f;
   parms x1 x2=-1;
   bounds    2 <=x1 <=50, -50<=x2<=50;
   lincon    10*x1-x2>=10;
   f=0.1*x1*x1+x2*x2-100;
run;
```

大規模な問題になると，制約式をいちいち記述するのは不便なので，制約式の係数をデータセットから読み込む機能を利用することになる．ＮＬＰではパラメタの初期値，境界制約，線形制約の情報を INEST＝オプションで指定したデータセットから読み込むことができる．このデータセット中には，目的関数に含まれる変数のほか，_TYPE_という文字型変数が必要である．また，一般線形制約をつけるときには _RHS_ という変数も必要になる．

INEST＝のデータセット入力を使って Betts 関数を最適化する例を示す．

```
data betts (type=est);
   input    _type_ $ x1 x2 _rhs_; cards;
PARMS       -1   -1   .
LOWERBD      2   -50  .
UPPERBD     50    50  .
GE          10   -1   10
;
proc nlp inest=betts;
   min f;
   parms x1 x2;
   f=0.1*x1*x1+x2*x2-100;
run;
```

ここでは，制約の係数を含むデータセットを，DATA ステップで作成している．_TYPE_変数の値がPARMSの行は，パラメタの初期値を与えている．LOWERBD の行は各パラメタの下限を，UPPERBD の行は各パラメタの上限を，それぞれ BOUNDS ステートメントに対応して示している．最後の GE の行は，LINCON ステートメントで指定する一般的な線形制約に対応している．GE というのは Greater than or Equal の略であり，与えられた係数と変数との線形結合が _RHS_(Right Hand Side)の値より大きいかあるいは等しいということを表わしている．小さいかまたは等しいという関係にはLE(Less than or Equal)が，等しいという関係にはEQ(EQual)というキーワードを用いることになる．

5. おわりに

ＮＬＰプロシジャは，作者 Wolfgang Hartmann のノウハウを凝集した非線形最適化の至宝ともいうべき技法集である．本稿では，その使い方の一部を紹介したのみで，スピード，問題サイズなどの性能評価は行なわなかった．それは，いままで非線形計画法の包括的なプログラムはなかったと思われるので，比較するものがないからである．ＮＬＰは，任意に設定した尤度関数を手軽に最適化できることから，新しい統計手法の開発に利用されることも期待したい．

参考文献

今野　浩，山下　浩 (1978) 非線形計画法　日科技連

Hartmann, M. W., (1992) The NLP Procedure Extended User's Guide, SAS Institute.