SAS/ORによる
数理計画最適化

SAS Institute Japan
コンサルティングサービス部
門馬　道也（Ph.D.）

Agenda

- 数理計画とSAS/OR
  - 数理計画概論
  - SAS/ORによる最適化概要
- 応用例
  - 予測モデル構築
  - 途上与信戦略最適化
- まとめ

数理計画最適化

- ある制約条件のもと、ある指標（目的関数）を最適化する
  - 何かコントロールできるパラメータの組み合わせを調整（施策の組み合わせ、その時期の選択）して、最適な結果を得たい場面で活用されます

  例） ビジネス上の制約を満たしつつ、利益を最大化するような施策を策定
- 数理計画最適化の種類
  - 線形計画（LP）、混合整数計画（MILP）、2次計画（QP）、非線形計画(NLP)
- 応用例
  - SCM（生産計画、輸送問題、設備配置、人員配置）
  - マーケティング施策、与信戦略
  - データマイニング
  - ダイエット、ポートフォリオ
- SAS/ORでは
  - 9.1以前： LP, INIPOINT, NLP, QP, NETFLOW
  - 9.1以降： OPTMODEL, OPTLP, OPTMILP, OPTQP

なぜSASで最適化？ 予測解析と意思決定の関係

- 最適化は施策、戦略を決定する場面で有効です
- 施策、戦略の策定は、将来を予見して打つ手を決めることです。
- 予測解析による結果を用いて、戦略を決定することが必要となります。

ＳＡＳで予測モデル構築→ＳＡＳ上で戦略最適化

SAS/OR概要

- 数理計画最適化
- プロジェクト・リソーススケジューリング
- 離散イベントシミュレーション
- 遺伝アルゴリズム
- 制約計画
- BOM（bill of materials）最適化

数理計画概論

- 最適化問題の一般的な形

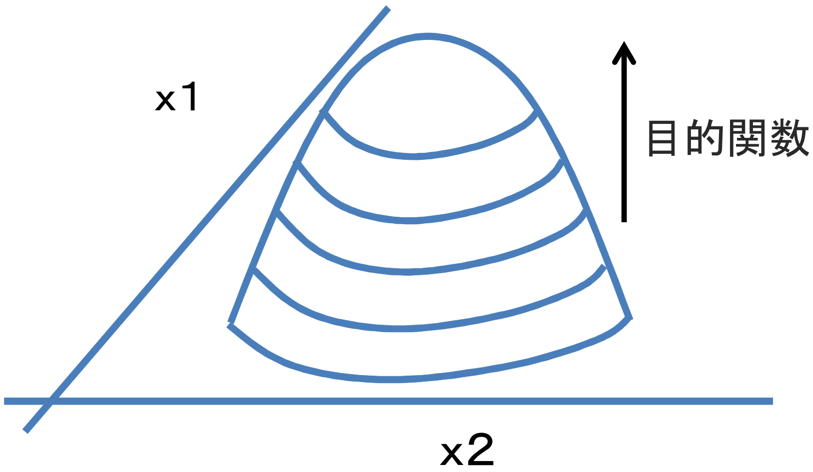
  Min (Max)　f(X)

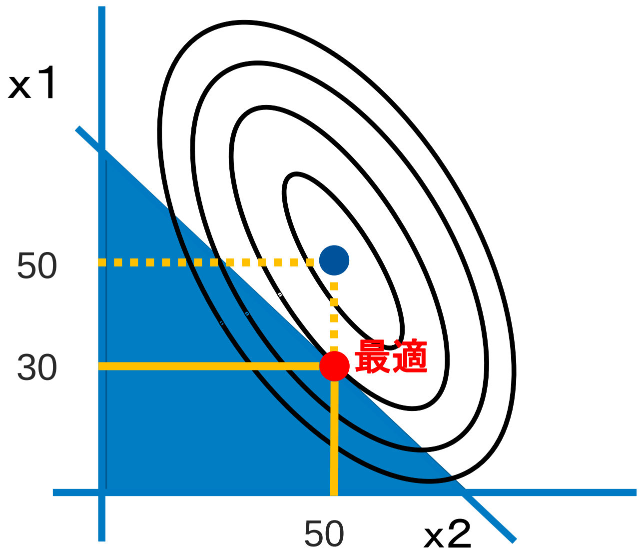
  Subject to　$c_i(X)\ \{\leqq, =, \geqq\}\ b_i$ (i= 1, 2, …, m)

  $l_j \leqq x_j \leqq u_j$ (j = 1, 2, …, n)
- Xは決定変数の集合：　$X = \{x_1, x_2, \ldots, x_n\}$;
- f(X)は目的関数
- $c_i(X)$は決定変数の関数
- l_jとu_jは決定変数x_jに関する上限、下限
- f、cの関数形によって、最適化問題の種類が決まる
- 整数計画の場合は、Xは整数値を持つ変数の集合

基本アイデア： どうやって最適化を実行するか？

- 目的関数に着目
  - 決定変数の関数として
  - x1、x2の値がある程度まで大きくなれば目的関数値が上がる
- 制約条件に着目
  - 決定変数に関しての制限
  - x1とx2の合計が決まっている
- 最適化理論、最適化パッケージを用いると少ないステップで自動的に最適値を求めることができます。





SAS OPTMODEL Procedure の特長

- モデリング環境
  - モデリング言語が数式に近い形で用意
- モデルパラメータの入力が容易
  - SASデータセットからの読み込み
- ソルバーとの連携
  - 直接ソルバーを呼び出すか、一旦モデルをSASデータセットとして保存、その後SAS/ORのプロシージャで解くことが可能
- ソルバーの種類（OPTMODELでは自動的に選んでくれる）

Problem	Solver
linear programming	LP
mixed integer programming	MILP
quadratic programming (experimental)	QP
nonlinear programming, unconstrained	NLPU
general nonlinear programming	NLPC
general nonlinear programming	SQP
general nonlinear programming (experimental)	IPNLP

PROC OPTMODEL言語概説

PROC OPTMODELの構造と4要素

```
proc optmodel;
    /* ① 集合＆パラメータ宣言 */
    /* ② 決定変数宣言 */
    /* ③ 制約条件宣言 */
    /* ④ 目的関数宣言 */
    solve;
    quit;
```

最適化準備

各種宣言

- 集合＆パラメータ
  - 集合（数値、文字）
    - » set <num> index = /1 2 3/;
    - » set <str> color = /r b g/;
  - パラメータ
    - » num x; str name;
    - » num vec{index};
    - » num matrix{index1,index2};
    - » str day{1..5} = [mon tue wed thu fri];
- 決定変数
  - var bias;
  - var w{index} >= 0;
  - var x binary;

各種宣言

- 制約条件
  - con nonneg: sum{i in index} w[i]*x[i] >= 0;
  - con predfunc{i in index}: f[i] = sum{v in vars} matrix[i,v] * w[v] + bias;

ラベル

- 目的関数
  - min obj = sum {i in index} (f[i] -tag[i])^2;

予測モデル構築への応用

- 課題
  - 予測モデル構築時に、説明変数を回帰係数(決定変数)の値が非負になるように設計。
  - モデル構築した結果のコントロールができないため、変数選択や変数生成に時間を要する。
- 解決策
  - 回帰係数の値のとる範囲を制約条件としてモデルを構築

サンプルデータと変数作成

- サンプルデータ
  - Housing data (UC Irvine Machine Learning Library)
  - ボストン近郊の住宅価格の予測モデル
  - 属性
    - 犯罪率、宅地率、小売り以外のビジネスの割合、部屋数、年齢、地形、固定資産税率等
    - 全13属性のうち、離散変数は1つ。残りは連続変数。
  - オブザベーション数：506
  - 回帰問題。ターゲット変数を2値にビン化して分類問題にも対応
- 変数作成
  - 連続変数を等範囲で最大10にビン化（バケット）
  - 各ビン値に、ターゲット変数の平均値を代入
    - 作成した変数は、**ターゲットに対して非負の相関**
    - モデル診断に活用：負の値がついたものを発見し、妥当性調査

① 線形回帰モデル

```
proc optmodel;
set<num> indx;
/*説明変数を宣言*/
set<str> vars = /m_age m_b m_crim m_distance m_indus m_lstat m_nox
m_pt m_rooms m_tax m_zn m_chas m_radial/;
/*予測ターゲットを宣言*/
num tag{indx};
/*データ行列を宣言*/
num matrix{indx, vars};
/*データを入力*/
read data smpdat.hdata_m into indx=[_N_] tag=mvalue
{v in vars} <matrix[_N_,v] = col(v)>;
/*回帰係数および定数項を定義*/
var w{vars} >= 0;
```

非負制約

```
var bias;
/*予測関数を定義*/
var f{indx};
/*制約条件*/
con predfunc{i in indx}: f[i] = sum{v in vars} matrix[i,v] * w[v] + bias;
/*目的関数：2乗損失*/
min LSLoss = sum{i in indx} (f[i] - tag[i])**2;
solve;
quit;
```

① 制約付加の効果

- 回帰係数の非負制約を入れる前は、いくつかの変数に負の係数値が最適となり、変数作成時に期待していたものと差異が発生
- 非負制約を付加することにより、簡単に変数作成の意図を反映したモデルを算出

[1]	w
m_age	-0.143705
m_b	0.150927
m_chas	0.365758
m_crim	0.309169
m_distance	-0.277824
m_indus	-0.175746
m_lstat	0.572532
m_nox	0.411050
m_pt	0.241154
m_radial	-0.115471
m_rooms	0.498877
m_tax	0.067463
m_zn	-0.041839

bias
-19.431

最適値
8034.2

[1]	w
m_age	0.00000
m_b	0.12925
m_chas	0.51227
m_crim	0.18421
m_distance	0.00000
m_indus	0.00000
m_lstat	0.47491
m_nox	0.24528
m_pt	0.17970
m_radial	0.00000
m_rooms	0.52741
m_tax	0.00000
m_zn	0.00000

bias
-28.234

最適値
8737.4

② ロジスティック回帰モデルとその結果

- 基本的に①の目的関数を変更することにより、ロジスティック回帰モデルについても対応可能
- 変更例は以下のとおり

```
con predfunc{i in indx}: f[i] = sum{v in vars} matrix[i,v] * w[v] + bias;
min NBLoss = sum{i in indx} log(1 + exp(-2*tag2[i]* f[i]));
solve;
```

- モデル係数についても同様に、非負制約を活用して変数の意図を反映

[1]	w
mc_age	-0.523325
mc_b	-0.450916
mc_chas	2.414835
mc_crim	0.069630
mc_distance	-1.826536
mc_indus	-0.379430
mc_lstat	2.097426
mc_nox	2.076549
mc_pt	1.174648
mc_radial	1.099153
mc_rooms	2.344740
mc_tax	1.257118
mc_zn	0.092902

bias
-3.4889

最適値
102.24

[1]	w
mc_age	0.00000
mc_b	0.00000
mc_chas	3.17845
mc_crim	0.00000
mc_distance	0.00000
mc_indus	0.00000
mc_lstat	1.81510
mc_nox	1.60454
mc_pt	1.08589
mc_radial	0.95528
mc_rooms	2.36313
mc_tax	0.81124
mc_zn	0.00000

bias
-4.1094

最適値
104.36

途上与信戦略への応用

- 課題
  - 途上与信では戦略的に与信限度枠をコントロールすることにより、デフォルトを抑え、利益を最大化します。
  - 様々なビジネス的な制約を満たしつつ最適な戦略(誰に、(いつ)いくら増枠するのか？)を決定する必要があります。
    - » 増枠の総額(予算制約)
    - » 貸し倒れ損失総額
- 解決策
  - 予測モデル＋最適化
    - » データから作用効果モデル(増枠に対してどのぐらい残高や貸し倒れ損失が増えるか)を構築
    - » 各ユーザー(セグメント)に対してどのぐらい増枠すれば利益が最大化されるかを最適化を利用して算出

途上与信戦略最適化

増枠効果予測

増額値	１０万円		２０万円		５０万円	
ユーザーセグメント	利益	損失	利益	損失	利益	損失
優良顧客	5	1	15	2	45	3
準優良顧客	5	2	10	3	20	5
高リスク顧客	2	3	8	10	5	20
低利用顧客	2	0	5	1	8	2

各セグメントの人数は1,000人
利益、損失の数字は1ユーザーあたりの金額で1万円単位
セグメント合計額としてみれば1千万円単位

- 目的関数：　利益合計額
- 制約条件
  - 一つのセグメントにつき、最大で一つの増額を付与
  - 予算制約（増額合計値が15億円）
  - 損失制約（損失合計値が1．5億円）

① 問題パラメータ設定

```
proc optmodel;

/*与信枠増加額を10,20,50万円に設定*/
set Clinc = /c10 c20 c50/;
num Cl{Clinc} = [10 20 50];

/*セグメントは、優良、準優良、高リスク、低利用*/
set Segment = /excel good hrisk lowu/;

/*セグメントx増額値での利益予測値*/
num Profit{Segment, Clinc} = [5 15 45
                              5 10 20
                              2 8 5
                              2 5 8];

/*セグメントx増額値での損失予測値*/
num Loss{Segment, Clinc} = [1 2 3
                            2 3 5
                            3 10 20
                            0 1 2];
```

② 最適化問題設定

```
/*決定変数はバイナリ*/
var x{Segment, Clinc} binary;

/*目的関数：利益を最大化*/
max TotProfit = sum{s in Segment, c in Clinc} Profit[s,c] * x[s,c];

/*各セグメントに対して最大1つの増額を付与*/
con Assign{s in Segment}:
        sum{c in Clinc} x[s,c] <= 1;

/*予算制約：増額値合計は15億以下*/
con Budget:
        sum{s in Segment, c in Clinc} Cl[c] * x[s, c] <= 150;

/*損失制約：損失値合計は1.5億以下*/

con TotLoss:
        sum{s in Segment, c in Clinc} Loss[s,c] * x[s,c] <= 15;

/*最適化計算*/
solve;
```

結果表示例

The OPTMODEL Procedure

Solution Summary	
Solver	MILP
Objective Function	TotProfit
Solution Status	Optimal
Objective Value	73
Iterations	12
Best Bound	.
Nodes	1
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	3.330669E-16
Integer Infeasibility	3.330669E-16

最適化計算のステータス：最適値は見つかったか？その質は？

x			
	c10	c20	c50
excel	0	0	1
good	0	0	1
hrisk	0	0	0
lowu	0	0	1

最適解：決定変数の値

TotProfit
73

最適値：目的関数値

まとめ

- 数理計画問題の概要とSAS/ORの機能概要の解説
- SAS/ORによる数理問題最適化例
  - SAS/ORを活用して、予測モデル構築時に分析者の意図を反映
  - SAS/ORを活用して、途上与信戦略最適化

ＳＡＳでは統計解析とともに、近年ＯＲの機能高度化にも力を入れています。
最適化を使えば、制約付き予測モデル構築等、分析者の立場からみて大変便利なこともできます。
ＳＣＭのような大規模なものから、身近な問題まで、適用場面は様々です。
どんどん活用していきましょう！

www.sas.com