パラメータ推定の理論と実践

山梨大学
佐々木邦明

BESTGUY

BEhavior Study for Transportation Graduate school, Univ. of Yamanashi

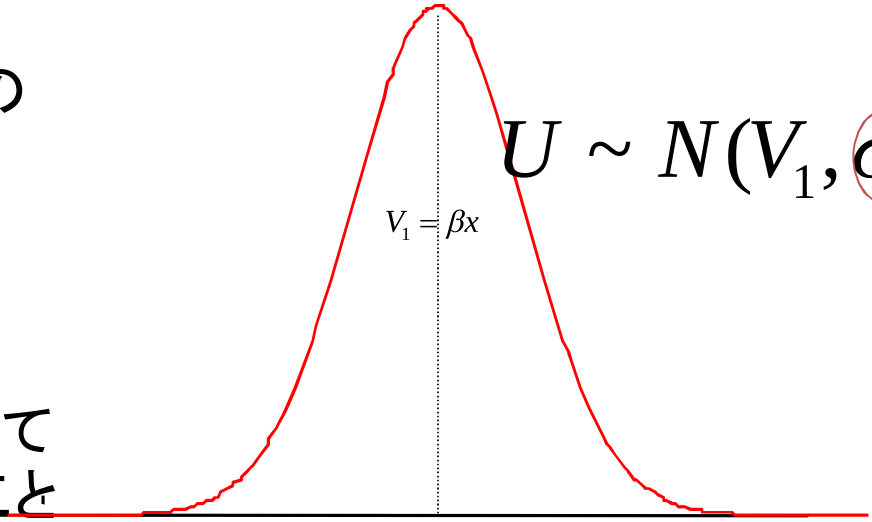
モデルのパラメータ(母数)とは

- 離散型選択モデルの母数
  - 確率分布を特徴付ける数
  - 例えば，期待値，分散等
  - 変動要因を規定する少数の要素に分解する

- パラメータ推定
  - 観察されたデータに基づいて母数を統計的に推定すること

ランダム効用モデル

期待値と分散が母数



$$U \sim N(V_1, \sigma)$$

$$V_1 = \beta x$$

推定アルゴリズム

最尤推定法

- 点推定量を求める最もポピュラーな方法
- 右上の式をθの関数とみなしたものが尤度関数
- 尤度関数を最大化するθの値を最尤推定量とするのが最尤推定法

$$L_n(\theta \mid x) = \prod_{i=1}^{n} f(x_i \mid \theta)$$

データ（a，b）が得られたとき，全体の平均がいくつとするのがよいか
＝平均がいくつだったら（a，b）が得られやすいか？

選択モデルの場合，f が選択確率

個人の選択確率を全員で掛け合わせる

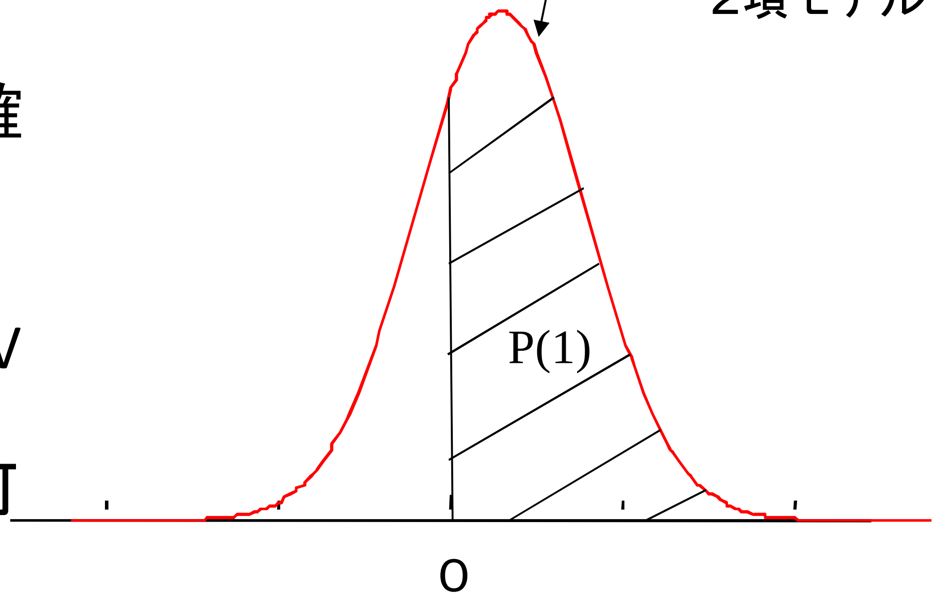
$$MaxLikelihood = \max_{\theta \in \Theta} \prod_{i} P_i(x_i \mid \theta)$$

確率モデルによる個人の選択記述

- 確率的選択モデルは効用がある分布に従うと仮定
- ある選択肢の選択確率は確率分布を積分する
- ロジットモデルやNL等のGEVモデルはこの積分した結果が積分のない形式で表現可能

$$U = U_1 - U_2$$

２項モデル



$$MaxLikelihood = \max_{\theta \in \Theta} \prod_i \int f(x_i \mid \theta)$$

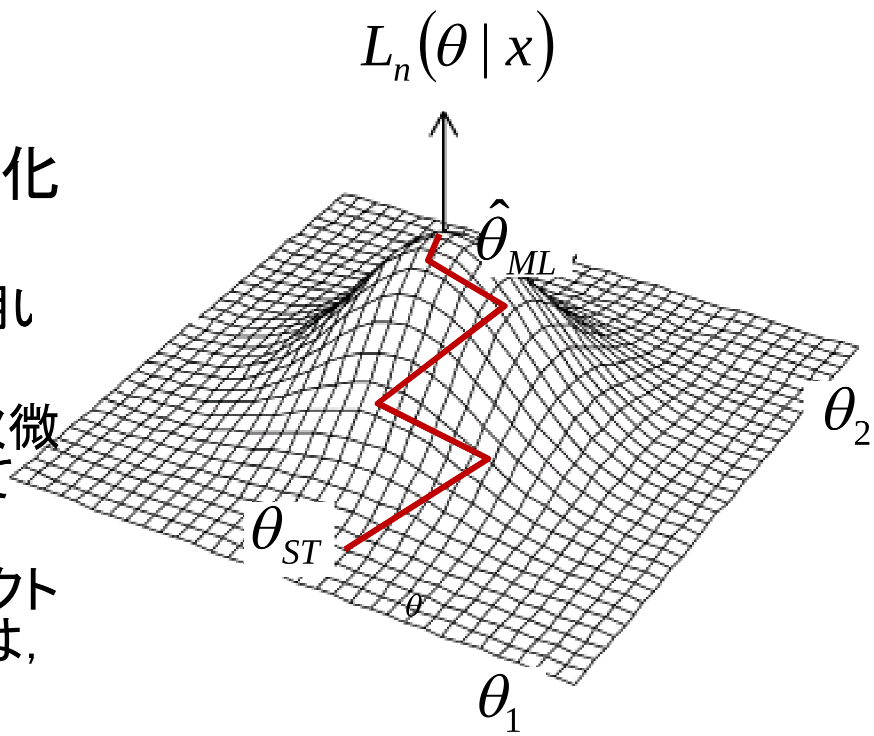
$\prod_i \int f(x_i \mid \theta)$ をどうやって最大化するか？

最尤推定のアルゴリズムの考え方

- 最大値をいきなり求めるのは無理な場合

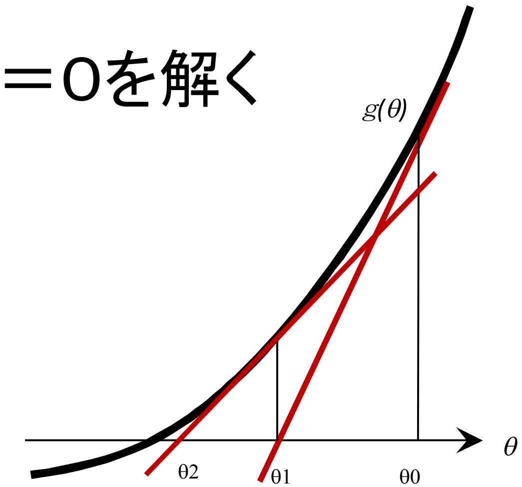
- 対数尤度関数の段階的な最大化
  1. 初期値を与える
  2. 初期値周りで勾配(1次微分）等を用いて次の推定値の方向を決める
  3. 初期値付近のステップサイズを1次微分，2次微分を用いて適切に決めて次の推定値を決める
  4. 収束基準(尤度関数の一時微分ベクトル)を判定し，収束していない場合は，現在の値を初期値として2に進む

ロジットモデルのみ全域で単峰性あり



代表的な繰り返し計算法

尤度関数を最大化:尤度関数の一階微分＝0を解く



- Newton-Raphson法
  - テイラー展開の1次近似を利用して進める
  - 解の収束が早い(ステップ数が少ない)
  - ヘッセ行列を計算するので時間がかかる

$$\theta_{n+1} = \theta_n - H^{-1} g_1$$

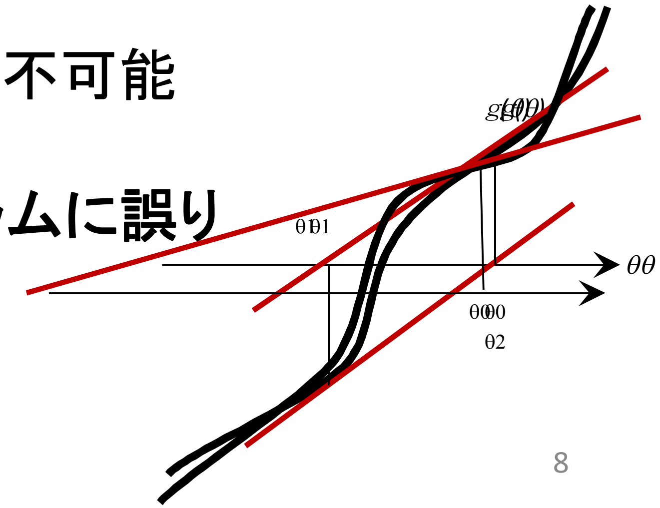
- 準Newton法（BFGS法）
  - ヘッセ行列を逐次近似する.
  - 二階微分をする必要がないので計算が早い

H:尤度関数の二階微分
ヘッセ行列
g: 尤度関数の一階微分

パラメータ推定がうまくいかない

- 収束するとはϑ_{n+1}とϑ_nが同じになる
  - g_1が0になる
- 収束しない
  - 無限に繰り返す
  - ϑ_2が計算不能
- 局所最適解
  - 見かけ上の最大化

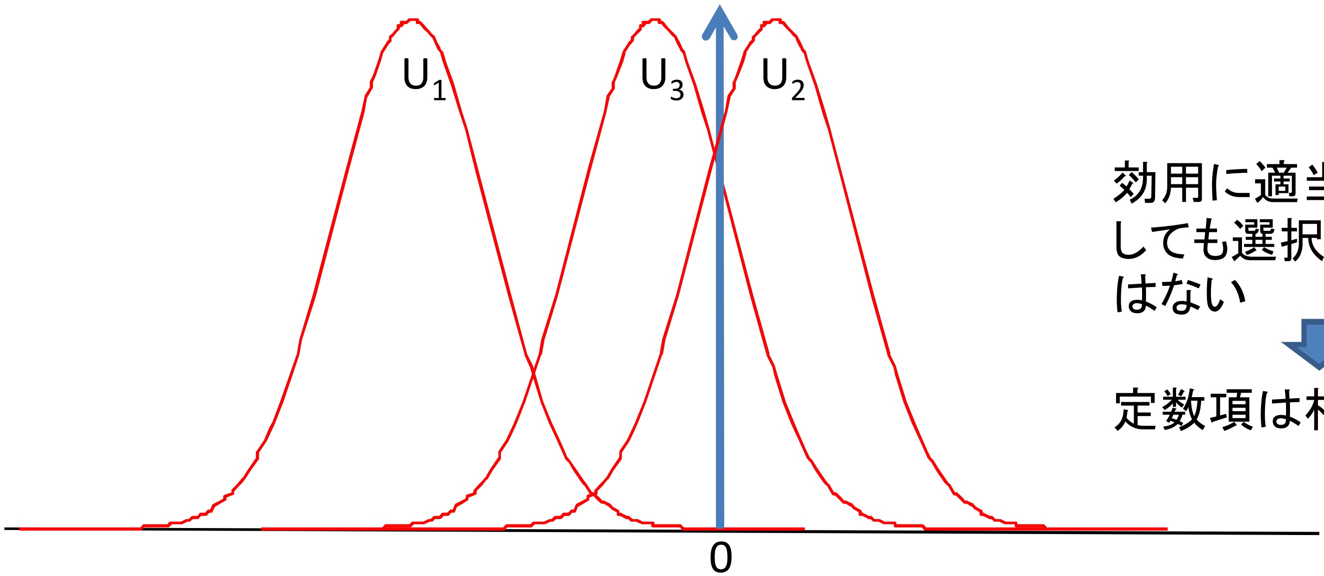
- H^{-1}が存在しない(計算できない)
  - 変数が完全相関
  - 変数が効用関数に影響していない
- 関数の近似状況
  - 2次関数近似
  - 初期値の問題
- そもそも推定不可能
- **推定プログラムに誤り**



そもそも推定不能

- 最大値において唯一解が求まらない可能性がある（最大値となるパラメータベクトルが無数にある）→Identification Problem

例



効用に適当な数を足しても選択確率に変化はない

定数項は相対的数値

ちょっと考える

- 例：3カ所の買物目的地選択

A:駐車場有料，B：駐車場有料，C:駐車場無料

このとき，駐車場無料は大きな選択要因

ならば駐車場無料ダミー変数を選択肢に導入

$$U_{1i} = \alpha_1 \times 1 + \beta_1 \times 0 + \ldots + \varepsilon_{1i}$$

$$U_{2i} = \alpha_2 \times 1 + \beta_1 \times 0 + \ldots + \varepsilon_{2i}$$

$$U_{3i} = \alpha_3 \times 0 + \beta_1 \times 1 + \ldots + \varepsilon_{3i}$$

これは区別できない

駐車場の台数や料金無料は効果が計測不能？

- 例えば自動車保有ダミーd_dを使ってみる

$$U_{1i} = \left(\alpha_1 + d_{di} \times \beta_1 \times 100\right) \times 1 + \ldots + \varepsilon_{1i}$$

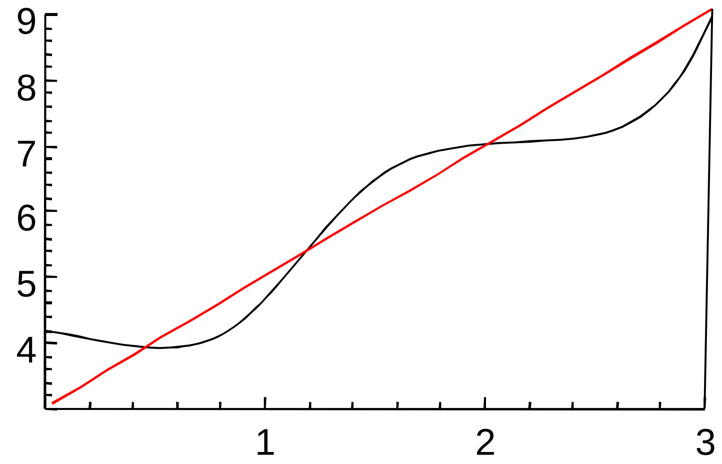
$$U_{2i} = \left(\alpha_2 + d_{di} \times \beta_1 \times 200\right) \times 1 + \ldots + \varepsilon_{2i}$$

$$U_{3i} = \left(d_{di} \times \beta_1 \times 1000\right) \times 1 + \ldots + \varepsilon_{3i}$$

積分（INTEGRATION）

数値積分

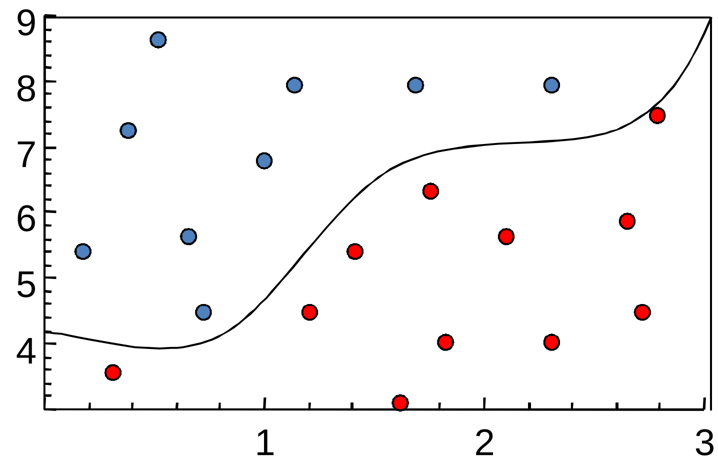
- 個人ごとの確率を計算するためには定積分することが必要だが，閉じた形で表現できないものは数値積分を行う必要がある．
- 数値積分：関数を多項式近似などを用いて積分可能な形にして，ある間隔の関数の和に直して積分を行う方法．
  - Gaussの求積法：近似誤差を最も小さくなるように区間と幅を定める方法．
  - Gauss-Legendre法：多項式近似ができる関数においては，多項式の次元にあわせた分割を行うと非常に精度よく積分を行える方法



複雑な関数を簡単な１次関数で近似した例．

シミュレーション(乱数)による積分

- 面積を求める別の方法を考える
- 2次元乱数を[３×９]の範囲に散らばらせて，それが積分したい関数より上か(青丸)下か(赤丸) を判定する.
- ドットを**うまく**大量に発生させれば精度よく積分を近似できる.



この図だと点の合計は20．そのうち赤丸１１個，青丸9個なのでこれより関数の積分結果は14.85となる

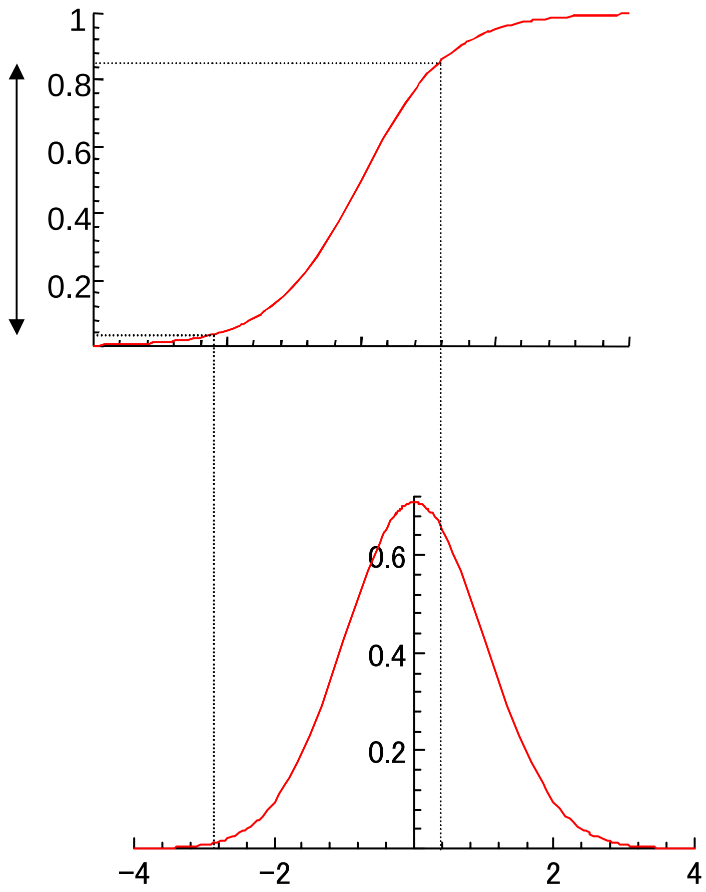
乱数の生成

- 何を積分したいのか？
  - 乱数の発生ルーチンを条件に合わせて改良
    - 制約つきの乱数
    - 多変量正規分布の乱数

- 乱数と同等の性質を持ち，同じ値が出ず，必要な全域がカバーされる数値の生成法があるとより効率的な積分が可能

制約つきの乱数

- 例えば平均0で分散1の正規分布の-3から1までの間の乱数.
- 対応する分布関数の間の乱数を発生させてそれを密度に対応させる
- この変換によって，密度の高いところはたくさん，密度の薄いところは少しという密度関数に応じた乱数になる.



多変量正規分布の発生

- 例えば標準正規分布μを平均b標準偏差sに変換するためには $\varepsilon = b + s\mu$ と線形変換をすればいい
- 同じ事を多変量正規分布に応用する．標準正規分布の乱数ベクトルηを $\varepsilon = b + L\eta$ と変換する．ただしLはLL'=Ωとなる行列．すると はMVN(b,Ω)になる．

例えば，2次元の例を示すと

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} l_{11} & 0 \\ l_{12} & l_{22} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$$

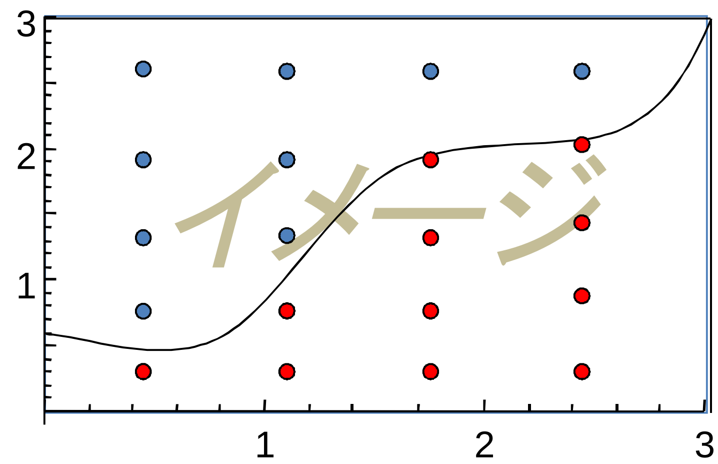
$$\varepsilon_1 = b_1 + l_{11}\eta_1$$

$$\varepsilon_2 = b_2 + l_{12}\eta_1 + l_{22}\eta_2$$

準乱数

- 高次積分において乱数による積分の精度を上げるためには，計算速度が多大に犠牲になる

- 対処方法として系統的な数列（準乱数）がある



準乱数の例

- 代表的例としてHalton数列がある．その計算方法は素数pに対して

$$s_{t+1} = \left\{ s_t, s_t + 1/p^t, s_t + 2/p^t \cdots, s_t + (p-1)/p^t \right\}$$

例えばp=3ならば，初期値0として1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, 8/9・・・

- 多次元化
  - 数列の異なる素数pを決めて，それぞれに応じて数列を作り多次元化する．
- 正規分布化
  - 数列を制約つきの乱数発生と同様の変換をして正規分布化

シミュレーションによる尤度計算

- 手順
  1. 誤差項の密度関数から選択肢の数の次元の(準)乱数を発生させる
  2. この乱数を誤差の値として，各代替案の効用値を計算（積分）する
  3. 代替案iの効用値とその他の代替案の効用との値を比較し，それらの大小関係を1-0の変数Gで記述する．
  4. 1～3のステップを繰り返す．その反復回数をRとする．
  5. シミュレーションされた確率は$P_i = \frac{1}{R}\sum_{r=1}^{R} G^r$となり，この値は不偏推定量である．

効用を確定値にする

確定的に選択を決定

比率を確率に置き換える

これを尤度として最大になるようにパラメータをアップデートする

連続型シミュレーション

1. 誤差項の密度関数から選択肢の数の次元の乱数を発生させる
2. この乱数を誤差の値として，各代替案の効用値を計算する
3. この効用をロジット変換を行い，連続的な0-1変数に変換する．この変換後の値をSとする
4. 1～3のステップを繰り返す．その反復回数をRとする．
5. シミュレーションされた確率は $P_i = \frac{1}{R}\sum_{r=1}^{R} S^r$ となり，この値は不偏推定量である．

シミュレーションベースのパラメータ推定法

- シミュレーション尤度最大化（MSL）
  - シミュレーションによって計算された確率を尤度として，最大化を行う.
- 特性
  - サンプル数と乱数発生回数に依存する.
  - 乱数発生回数が十分大きいと一致性や漸近的有効性を持ち解析積分と一緒の特性を持つ.
  - 乱数発生回数がサンプル数に対して小さく固定されると一致性もない.

まとめ

統計的な視点

- 推定尤度に基づくモデル構造の選択
  - 投入する情報量(データ)が一緒なら尤度関数の大小で判定
  - 情報量の異なるモデル間ではAIC等の統計量による比較

- 推定パラメータに基づく変数の選択
  - t値による取捨選択
  - 推定値の符号による判定
    - 費用と所要時間のパラメータは負

モデル構造はどうやって選ぶ？

- 行動に関するアプリオリな知識による（仮説）
  - 手段選択には費用は影響する
    - 線形効用関数に費用を導入する．でもタクシーは選択肢に入らない
    - 費用は非補償型のモデル構造とする

    ⇒ 検証
  - バスとパークアンドライドの選択において，誤差項に相関があるだろう
    - NLやCNLの採用を考える

- 推定結果からの後天的な知識による（仮説の修正）
  - モデルの推定結果が仮説を支持しないときに仮説が間違っている可能性を考える
    - 線形効用関数で費用のパラメータが負にならない→キャプティブな選択があるのでは？

    ⇒ 検証
    - ネスティッドロジットモデルのログサムの係数が0～1にならない→ネスト構造が違っている？