パラメータ推定の実践的テクニック
― 幻の大地 ―

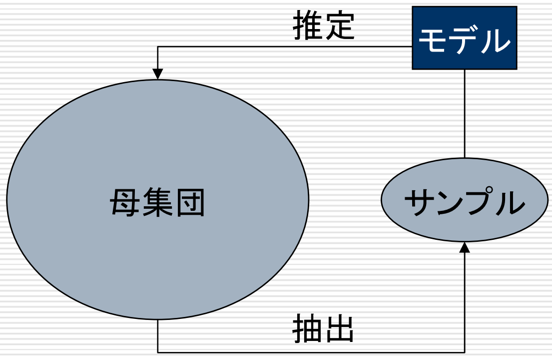
山梨大学
佐々木邦明

BESTGUY
Behavior Study-group in Transport Grounds, University of Yamanashi

推定されるモデルの話

行動モデル？古典的統計モデル

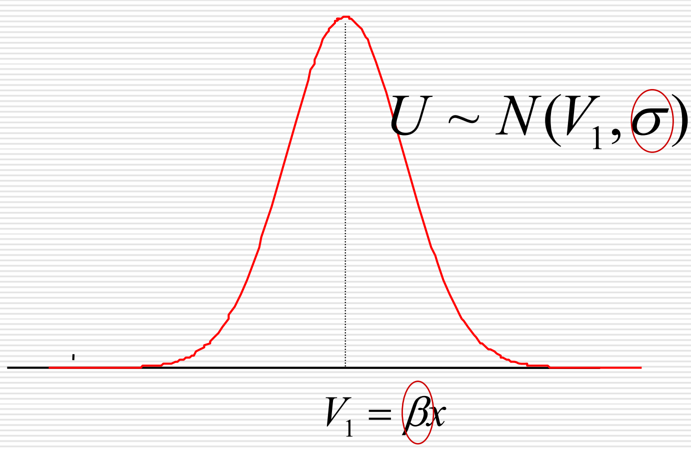
- 未知の分布をもつ母集団から抽出した標本より，母集団の特性を記述する母数を推定すること
- 母集団の構造にあらかじめ何らかの仮定を置くことがモデル
  - 例えば効用が確率分布をして，分布の期待値が線形関数の構造を持つと仮定
- モデルの推定とは，その仮定に従って母集団より標本が得られたという前提での母数の推定



確率モデルにおけるパラメータとは

- 行動モデルの母数
  - ある確率分布を特徴付ける数
- 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

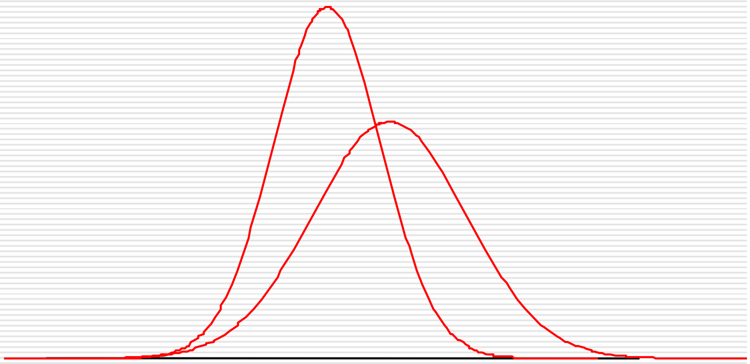
ランダム効用モデルの例

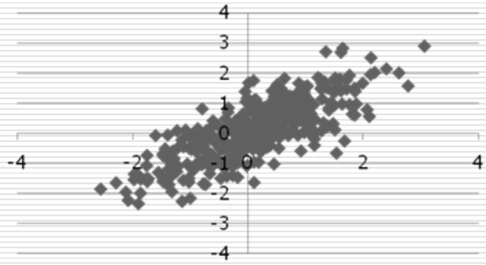


ランダム効用モデル

- 確率分布の構造
  - 分布形は？
    - ガンベル分布，正規分布，混合分布．．．
  - 複数の確率分布の関係はどうか？
    - 相関の有無，分散の大小，分散の同一性，

MNL, NL, CNL, MXL・・・・





期待値の構造(効用関数の構造)

- 期待値は何によって変動するのか
  - 変数選択
    - 手段選択の例：費用，所要時間，頻度(待ち時間)アクセス・イグレス時間・・・

- 期待値の関数形は
  - 線形モデル，非線形(非補償型)モデル，

ある意味無限に存在する

何で無限？

観測技術の限界

行動原理の違い

無限からの選択

- 直感的な方法論
  - 例えば，手段選択に費用は関係ある
  - 費用は対数的に影響する
- 技術的な方法論
  - パラメータ推定が容易，意味が分かりやすい

これらはモデルのパラメータを決めること

推定アルゴリズム

最尤推定法

- 右の式をθの関数とみなしたものが尤度関数

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

- 尤度関数を最大化するθの値を最尤推定量とする推定方法を最尤推定法とよぶ.

$$L_n(\hat{\theta}_{ML} \mid x) = \max_{\theta \in \Theta} L_n(\theta \mid x)$$

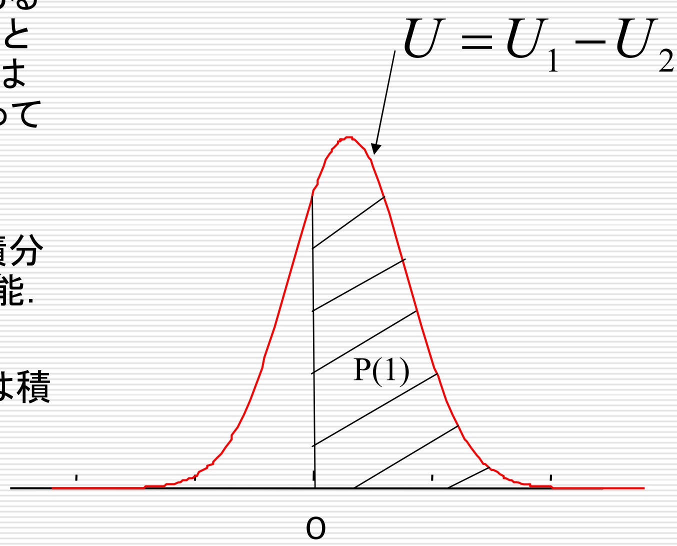
選択モデルの場合, fが選択確率になっている

⇩

最尤推定をするためには個人ごとの選択確率が必要になる.

個人の選択確率

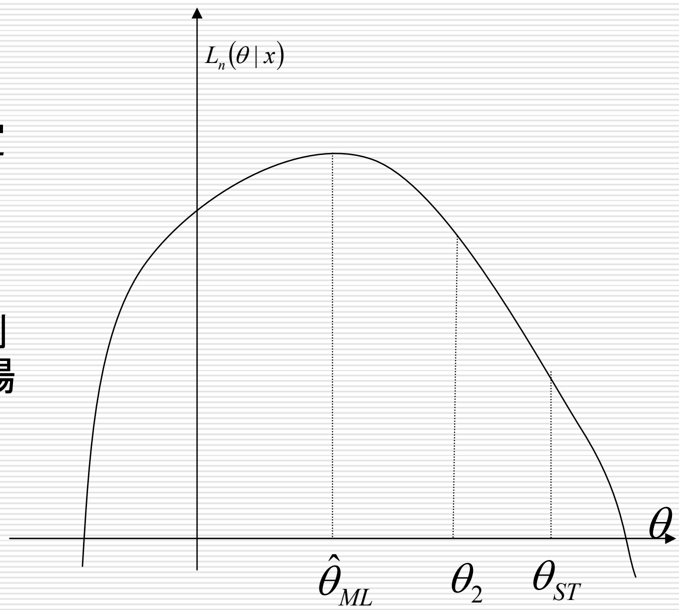
- 確率的選択モデルは効用がある分布に従うと仮定される．そのときある選択肢を選択する確率は効用関数を積分することによって求まることが一般的．
- ロジットモデルはこの積分が積分した結果が閉じた形で表現可能．
- 閉じた形で表現されないものは積分をしなければならない．
  - 例えばMXL，MNP



最尤推定アルゴリズム

- 尤度関数最大化
  1. 初期値を与える
  2. 初期値周りで次の推定値の方向を決める
  3. 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1に戻る

ロジットモデルは全域で単峰性あり



代表的な繰り返し計算法

- 最急降下法
  - 目的関数の勾配ベクトルを用いて決定
  - 各ステップの計算が速い
  - 解の収束が遅い
- Newton-Raphson法
  - テイラー展開の2次近似を用いて次を決める
  - 解の収束が早い
  - ヘッセ行列を計算するので時間がかかる

$$\theta_2 = \theta_1 - H^{-1} g_1$$

H:尤度関数の二階微分
ヘッセ行列
g: 裕度関数の一階微分

- 準Newton法（BFGS法）
  - ヘッセ行列を単位行列から，尤度関数と尤度関数の一階微分の差を用いて逐次近似していく
  - 二階微分をする必要がないので早い

アルゴリズム選択と初期値

- アルゴリズムの選択
  - 基本的に同じ推定値を出す
  - 関数形によって計算速度に差が出る
  - アルゴリズムの選択によっては収束しないことがある
- 初期値をどうやって決める？
  - 適当に与える(例：全部0)
  - 一度得られた解を元に与える
  - 符号については直感的なものを与える

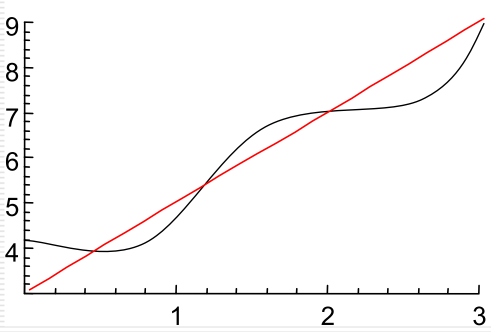
ヘッセ行列がお亡くなりに．．．

- ニュートン法を用いているとヘッセ行列の逆行列が計算できないときが．．．もしくは発散してしまう
  - 準ニュートン法を用いて回避可能な場合有
- それでもt値を求める時にはヘッセ行列の逆行列が必要
  - これが無理な場合→ベイズ推定か
- しかし
  - 推定プログラム等に誤りがあることも→シンプルなモデルで検証
  - データのスケールの統一をしてみる→計算桁数
  - 初期値の場所が悪い→尤度関数の形が見えない

積分の砦

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



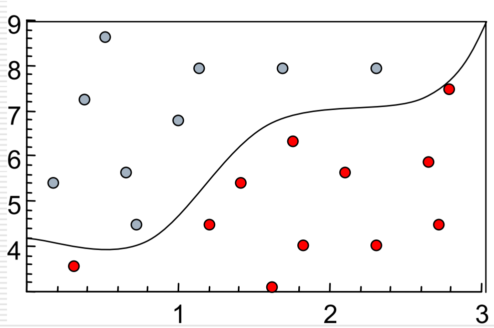
複雑な関数を簡単な1次関数で近似した例.

数値積分の課題

- 数値積分は関数の形状によって精度やロードが変化する.
  - 多項式近似をするので，複雑な形状になると積分が面倒くさくなる（計算機がですが）
  - 複雑なモデルは関数の形がよくわからない
  - 多次元積分は時間がかかる

シミュレーション(乱数)による積分

- 積分というのは面積を求める問題．そこで面積を求める別の方法を考えてみる．
- 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．
- 乱数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．



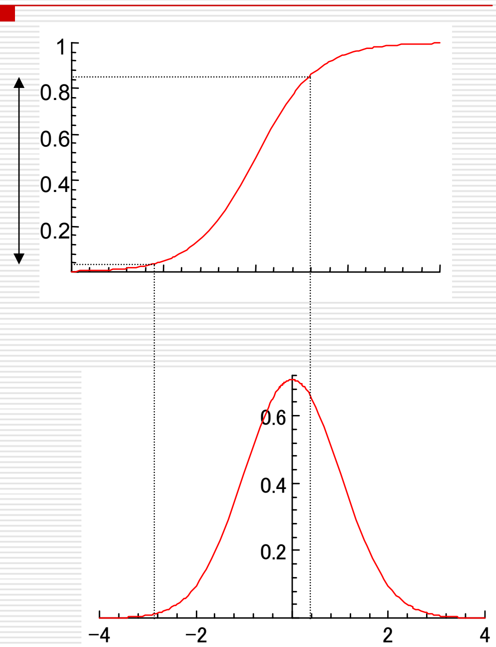
この図だと点の合計は20．そのうち赤丸１１個，青丸9個なのでこれより関数の積分結果は14.85となる

乱数の生成

- 乱数の発生ルーチンはたいていの言語についていますが，必ずしも自分の欲しい乱数を出してくれません．
  - 例えば
    - 制約つきの乱数
    - 多変量正規分布の乱数
- 乱数に近く，同じ値が出ずに，必要な全域をきちんとカバーしてくれるような数値の生成法があるとより効率的な積分が可能になる．

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(これは可能)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

- Choleski変換
  - 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$LL' = \Omega$となる行列．すると η はMVN(b,Ω)になる.

    例えば，2次元の例を示すと

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

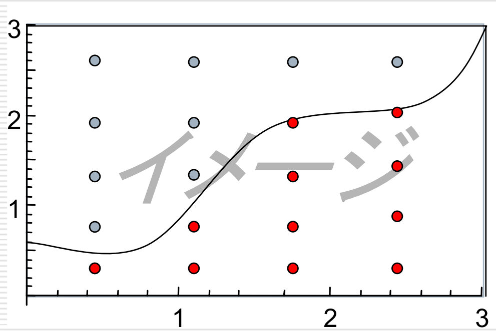
$$\varepsilon_1 = b_1 + l_{11}\eta_1$$

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

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

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・・・

- その他にもSobol，Faureなど様々，ただし，次数，関数型系によってどれが効率が高いかは異なるようだ

- 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

準乱数列の問題と対処

- 乱数じゃないので，乱数特性に基づいた推定量の評価は使えない．
  - 対処：ランダム化したHalton数列
    Halton数列に正規分布を足し合わせる

- 多次元積分をするとでかい素数が必要になる．その上，大きな素数間のHalton数列は相関を持つ
  - 対処：スクランブルをかけたHalton数列
    多次元Halton数列の組み合わせる順序を入れ替える
  - 次元に応じた数列を用いる（各数列は次元や関数に応じて得意不得意があるらしい）

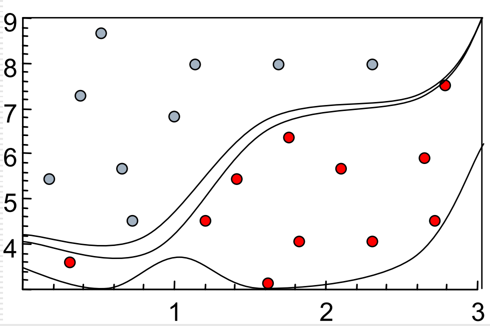
離散判定シミュレーション積分

- 少し前のスライドで示したようなものが離散型シミュレーションによる積分．

- 手順
  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. 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．

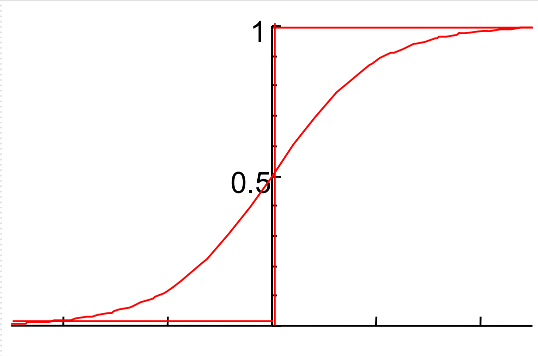
離散判定シミュレーションの問題点

- パラメータの微小変化に対して，確率の変化が起こらない場合がある．つまり変化が離散的である．
- どんなときでも，選択確率が0になってしまう可能性がある．



連続型シミュレーション積分

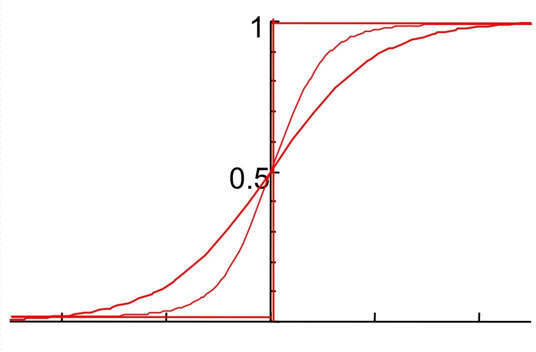
- 離散にして困るなら大小関係を保った連続型に変換すればいい．
- 右の図のようにロジット型に変換することで，いいか悪いかは別として，先の問題二つはいずれも解決可能．



ロジット変換

- ロジット変換の一般式は右のようになる.
- ロジット型のスケールパラメータは分析者があらかじめ定める必要がある.
- μ の値はロジット変換の効用スケールの問題であり, これに関しては, 適当な値を比較して, その結果を比較する必要がある.

$$S_i = \frac{e^{U_i/\mu}}{\sum_j e^{U_j/\mu}}$$



スケールの違いによる変換の違い

連続型シミュレーション積分の手順

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.
- シミュレーションモーメント法（MSM）
  - シミュレートした確率と選択結果の差を変数でウェイト付けして和をとる方法.
- シミュレーションスコア法（MSS）
  - シミュレーティッドスコアに対数尤度の微分ではなく，計算される確率と，対数尤度の積を定義する.
- このような方法間には違いがある. なぜならば，各推定量は解析的推定に対して，シミュレーションバイアスとシミュレーションノイズが関係してくるからである.

推定方法間の比較

- MSL
  - 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.
- MSM
  - 適切なウェイトが用いられた場合には固定された乱数発生回数に対して一致性を持つ
  - 適切なウェイトが用いられないと一致性はない
  - 乱数発生回数がサンプル数に対して十分大きいと漸近的有効になる
- MSS
  - 固定された乱数発生回数に対して一致性を持つ
  - 乱数発生回数が増加すると漸近的有効.
  - ただし途中の計算にもう一度シミュレーションが必要で，そのシミュレーションに対して離散型シミュレーションの問題が発生する.

よいところ

- シミュレーションを使うと，積分の入るめんどうくさそうなモデルも，複雑なプログラムを書かずとも，簡単なルーチンの組み合わせでかなり確実に計算結果を出してくれると思われる

そんなに詳しくないが

ベイズ界

ベイズ推定とは？

- ベイズの定理を用いたパラメータ推定

$$P(\theta|x)=\frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}$$

- 事前分布がわかっていないと仮定できるなら尤度を最大化するθは事後確率を最大化するθである

$$P(\theta|x)=\frac{p(x|\theta)}{\int p(x|\theta)d\theta} \Leftrightarrow L(x|\theta)*P(x|\theta)$$

つまりパラメータ推定がベイズアプローチでも可能

ベイズ推定法の特質

- パラメータの分布はデータが決める
- 利点
  - パラメータが一部識別不可でもその他を推定することが可能
  - 通常の最尤推定が局所解になるのを避けることができる場合がある

MCMCを用いたベイズ推定法

- Markov Chain Monte Carlo法
  - 事前分布に基づいて，次の値（事後確率）を求めることを繰り返す.
  - モンテカルロ法は単純に乱数だけれども，マルコフ性を考える（前の回の事後分布を事前分布とする）
  - 簡単に言えば，パラメータを適当に動かして，尤度を計算して，事後分布を元に再度パラメータをちょっと動かす，を繰り返して，パラメータの頻度分布をとる
  - たくさんパラメータがあるときに一個一個動かせばいいから計算が簡単

詳細なアルゴリズムは各自お勉強を

モデル選択Stage

統計的な視点(普通の話)

- 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
  - 手段選択には費用は欠かせない
    - 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
  - バスとパークアンドライドの選択において，誤差項（非観測要因）に同じものが出るだろう
    - 単純なロジットモデルの適用は難しい→NL，CNL

- 推定結果からの後天的な知識による
  - モデルの推定結果が芳しくないときにモデルの構造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？
    - 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？

変数選択の考え方の案

- 政策分析に必要な変数は排除しない
  - 費用のパラメータが無いときには費用の変動による需要の変動の分析はできない
  - 費用が有意な負にならないときは？
    - データの再考（費用がほとんど一緒）
    - テクニカルな変数設定の再考
  - 変数間の相関の排除
    - データの追加的収集，
- 影響しない変数と判断することも必要
  - 費用が影響していないのに無理ある推定は百害あって三利位しかない
  - 初期の目的を忘れずに

モデル構造の再考について

- 誤差項の構造の仮定＝選択肢の属性における非観測要因の仮定
- 誤差構造を複雑にするより，観測変数をきちんと取る方がいいのと思われる
- 尤度が高いと符号とかなんでもいいのか？
  - 良いらしい（噂）

おわりに

- パラメータを推定することが究極の目的ではありません．交通行動の理解を求めることが目的です．
- またそれを交通計画へ適用するということが最終目的であることが多いことも忘れずに