統計数理(2008)
第 56 巻 第 1 号 3–18

特集「分子進化と統計科学」
[研究詳解]

滑らかでない境界をもつ領域のための マルチスケール・ブートストラップ法の理論と実装

下平 英寿†

(受付 2007 年 8 月 24 日；改訂 2007 年 9 月 12 日)

要　　旨

データ解析から得られる結果の信頼度を計算するためにブートストラップ法が広く利用されている．その信頼度はブートストラップ確率と呼ばれ，仮説検定の p-値として解釈するとバイアスがあることが知られている．そこでバイアス補正をおこなって精度の高い信頼度を計算するためにマルチスケール・ブートストラップ法が考案された．当初は仮説境界が滑らかであることを前提としていたが，現在では境界が滑らかでない場合に一般化された．分子系統樹推定および階層型クラスタリングの実例を通して，このアルゴリズムの理論と実装を解説する．

キーワード：近似的に不偏な検定，ブートストラップ確率，バイアス補正，スケール変換則，系統樹推定．

1. はじめに

データ解析の結果をどれほど信頼できるのか，そこにひそむランダムネスを測定して信頼度を計算することが本稿で説明するマルチスケール・ブートストラップ法の目的である．複雑なデータ解析でも，その中身に立ち入らずにブラックボックスとして扱い，とりあえず信頼度を計算できる．そのデータ解析の手続きの反復実行が要求されて計算量が膨大になるが，並列計算が容易なため将来的に困難はないだろう．

このような手法の代表的なものがブートストラップ法(Efron, 1979)と呼ばれる一種の確率シミュレーションである．ブートストラップ法によって計算される信頼度はブートストラップ確率(Felsenstein, 1985)と呼ばれる．これは広く利用されてきたが，信頼度がバイアスを持つ傾向がある．そこで Shimodaira (2002, 2004, 2008)の一連の研究によって考案されたのが，マルチスケール・ブートストラップ法である(下平, 2002)．データのサンプルサイズを変化させ，そのときのブートストラップ確率の変化から精度の高い信頼度を計算する原理である．Efron and Tibshirani (1998)で議論された「領域の問題」を基礎として，ブートストラップ確率のスケール変換則を導入した(3 節)．

マルチスケール・ブートストラップ法は分子系統樹推定の信頼度計算(2 節)や階層型クラスタリングの信頼度計算(4 節)で利用されている．それぞれ `CONSEL` (Shimodaira and Hasegawa, 2001)および `pvclust` (Suzuki and Shimodaira, 2006)としてソフトウエアに実装されている．こ

† 東京工業大学 数理・計算科学専攻：〒 152–8552 東京都目黒区大岡山 2–12–1

表 1. 6 種の哺乳類について 15 通りの木.

```
((human,(seal,cow)),rabbit,(mouse,opossum)); t1
(human,rabbit,((seal,cow),(mouse,opossum))); t2
(human,((seal,cow),rabbit),(mouse,opossum)); t3
(human,(rabbit,mouse),((seal,cow),opossum)); t4
((human,(seal,cow)),(rabbit,mouse),opossum); t5
(human,((seal,cow),(rabbit,mouse)),opossum); t6
(human,(((seal,cow),rabbit),mouse),opossum); t7
(((human,(seal,cow)),mouse),rabbit,opossum); t8
(((human,mouse),(seal,cow)),rabbit,opossum); t9
(human,rabbit,(((seal,cow),mouse),opossum)); t10
(human,(((seal,cow),mouse),rabbit),opossum); t11
((human,((seal,cow),mouse)),rabbit,opossum); t12
(human,rabbit,(((seal,cow),opossum),mouse)); t13
((human,mouse),rabbit,((seal,cow),opossum)); t14
((human,mouse),((seal,cow),rabbit),opossum); t15
```

れらは Shimodaira (2002) で提案されたマルチスケール・ブートストラップ法の実装であり，仮説領域の境界が滑らかな曲面であることを前提に理論が導出されていた．ところが，上記応用の仮説領域は凸錐などが多く，境界が滑らかでない．この状況で理論を新たに導出したのが Shimodaira (2008) であり，一般化されたアルゴリズムが `scaleboot` (Shimodaira, 2006) に実装されている．これは `R` (R Development Core Team, 2005) で利用するパッケージで，`R` の公式ライブラリに登録されている．本稿ではこの利用法についても説明する．

2. 分子系統樹推定の信頼度

2.1 哺乳類のミトコンドリア DNA データ

DNA データから生物進化の系統樹を推定するために，近隣結合法や最尤法などが用いられる．これらの手法はデータに最も適合する系統樹トポロジー (これを木と呼ぶことにする) をひとつ出力するが，その木がどれほど信頼できるのかを評価するためにマルチスケール・ブートストラップ法が利用できる．木の信頼度は検定の p-値として計算される．ここでは，Shimodaira and Hasegawa (1999) のミトコンドリア DNA データで最尤法をおこなう実例を `scaleboot` パッケージの利用手順とともに示す．

6 種の哺乳類 (human, seal, cow, rabbit, mouse, opossum) の系統樹として，表 1 に示した 15 通りの木 (`t1`, ..., `t15` と書く) を考えた．このうちどれかが真実の木であり，それを推定したいと考える．あらかじめ `PAML` (Yang, 1997) を利用して，各木の対数尤度を計算しておく．その結果を保存したファイルを `mam15.lnf` とする．これを `scaleboot` で読み込める形式に変換するために，`CONSEL` に含まれるコマンド `seqmt --paml mam15.lnf` を実行する．これで各木の site-wise log-likelihood が行列形式のデータとしてファイル `mam15.mt` に保存された．

2.2 マルチスケール・ブートストラップ法の実行

`R` を起動し，次を実行する．

```
> library(scaleboot) # scaleboot ライブラリの読み込み
> data <- read.mt("mam15.mt") # 対数尤度ファイルの読み込み
```

```
> dim(data) # データ行列の次元 = 3414 * 15
[1] 3414   15
```

マルチスケール・ブートストラップ法の計算は次の 1 行で完了する.

```
> result <- relltest(data)
```

パソコンならば 30 分程度である．大規模な問題では実行速度が問題になるが，scaleboot は並列計算の機能があり，CPU 数にほぼ比例して速く計算できる.

結果を見やすくするために，対数尤度の大きい順番に木を並べ替えておく．次の 1 行目は，各木の対数尤度とその最大値との差を取り出す.

```
> x <- attr(result,"stat") # 結果から対数尤度差のとりだし
> result <- result[order(x)] # 結果を対数尤度差の小さい順に並べ替える
> result # 並べ替えた結果の表示（補助的なものだけ）
```

これで表 2 が得られ，並べ替えが正しく行われたことが確認できる．この 2 列(stat と shtest)は対数尤度差と Shimodaira-Hasegawa 検定(SH 検定)の p-値であり，CONSEL の catpv コマンドで表示される 2 列(obs と sh)に相当する．stat の 1 行目が 0 ではなく -2.66 となっているのは仕様なので気にしなくて良い.

最後に，マルチスケール・ブートストラップ法の結果を要約して表示するために,

```
> summary(result) # 結果の表示
```

を実行すると，表 3 が出力される．この各列をどのように解釈して系統樹推定に利用するのかを説明していく.

表 2.　対数尤度差(stat)と SH 検定の p-値(shtest)．p-値は百分率で表示され，カッコ内は標準誤差である．実際にはマルチスケール・ブートストラップ法の実行に関するパラメータなども表示されるが，ここでは省略した.

	stat	shtest
t1	-2.66	94.51 (0.07)
t3	2.66	80.25 (0.13)
t2	7.40	57.85 (0.16)
t5	17.57	17.30 (0.12)
t6	18.93	14.32 (0.11)
t7	20.11	11.49 (0.10)
t4	20.60	10.98 (0.10)
t15	22.22	7.34 (0.08)
t8	25.38	3.31 (0.06)
t14	26.32	3.29 (0.06)
t13	28.86	1.71 (0.04)
t9	31.64	0.61 (0.02)
t11	31.75	0.57 (0.02)
t10	34.74	0.20 (0.01)
t12	36.25	0.12 (0.01)

表 3. マルチスケール・ブートストラップ法の結果．raw はブートストラップ確率，k.1〜k.3 は p_k，$k=1,2,3$ である．いずれも百分率(カッコ内は標準誤差)．model と aic は 3.4 節や 4.3 節を参照.

```
Corrected P-values (percent):
    raw          k.1          k.2          k.3          model  aic
t1  57.58 (0.16) 56.16 (0.04) 74.55 (0.05) 74.55 (0.05) poly.2  964.33
t3  31.86 (0.15) 30.26 (0.05) 46.41 (0.09) 45.33 (0.13) poly.3 1306.50
t2   3.68 (0.06)  3.68 (0.03) 12.97 (0.20) 16.12 (0.45) sing.3   -6.21
t5   1.34 (0.04)  1.33 (0.02)  7.92 (0.25) 10.56 (0.56) sing.3  -14.11
t6   3.18 (0.06)  3.15 (0.02) 13.15 (0.21) 15.86 (0.44) sing.3   -2.49
t7   0.49 (0.02)  0.52 (0.01)  3.66 (0.21)  4.75 (0.42) sing.3  -12.04
t4   1.55 (0.04)  1.53 (0.02) 10.54 (0.27) 14.84 (0.66) sing.3   -7.57
t15  0.08 (0.01)  0.07 (0.00)  1.11 (0.19)  1.85 (0.48) sing.3  -17.08
t8   0.00 (0.00)  0.00 (0.00)  0.04 (0.03)  0.07 (0.07) sing.3  -13.68
t14  0.22 (0.01)  0.23 (0.01)  2.76 (0.26)  4.59 (0.71) sing.3  -10.79
t13  0.02 (0.00)  0.01 (0.00)  0.50 (0.20)  1.30 (0.83) sing.3  -15.14
t9   0.00 (0.00)  0.00 (0.00)  0.23 (0.05)  1.41 (0.29) sing.3  -15.86
t11  0.00 (0.00)  0.00 (0.00)  0.00 (0.00)  0.00 (0.00) poly.3  -19.71
t10  0.00 (0.00)  0.00 (0.00)  0.00 (0.00)  0.00 (0.00) poly.3  -17.27
t12  0.00 (0.00)  0.00 (0.00)  0.00 (0.00)  0.00 (0.00) poly.3  -19.61
```

2.3 系統樹推定のランダムネス

対数尤度を最大にする木は t1 である．しかし t3 は対数尤度が 2.66 小さいだけであり，データに内在するランダムネスの影響で偶然に順位が入れ替わっただけかもしれない．そこで，ブートストラップ法によってデータを何回も(ここでは 10000 回)生成し，各木が対数尤度を最大にした回数を数えて頻度を計算したものがブートストラップ確率である．これが表 3 の raw 列に示されている．t1 は約 58%，t3 は 32% であり，いずれも真実の木である可能性が否定できない．閾値(有意水準)を 5% にとると，そのほかの 13 個は真実の木でないと判断される．

このようにブートストラップ確率は系統樹推定のランダムネスを測る指標となるが，実は偽の発見に結びつきやすいという問題がある．この数値例に関しても，従来は(rabbit，mouse)クレードを含む t4，t5，t6 のどれかが真実と考えられていたが，そのいずれもがブートストラップ確率 5% 未満となり棄却される．もしこの結果が本当ならば新しい発見といえるが，その後の研究によりこの「発見」は誤りらしい．つまり従来の(rabbit，mouse)クレードのほうが本当だったようだ．

そこで，ブートストラップ確率を改良して精度を高めるために考案されたのが，マルチスケール・ブートストラップ法である．とくに，Shimodaira (2008)で提案された近似的に不偏な p-値のクラス p_k，$k=1,2,3,\ldots$ を scaleboot は実装している．表 3 の k.1 は p_1，k.2 は p_2，k.3 は p_3 を示す．一般に k が増えるほど近似の精度が良くなるが，一方で数値的に不安定になるので，$k=2$ または $k=3$ 程度を利用する．p_2 または p_3 が 5% 以上になる木は 6 個あり，さきほどの t4，t5，t6 もそれに含まれるから，もし p_2 か p_3 を使って判断していれば偽りの発見とならなかった．

p_1 はマルチスケール・ブートストラップ法をつかってブートストラップ確率に相当する量を計算したものである．たしかに表 3 をみると raw と k.1 は実質的に同じ数値である(ただし k.1 のほうが標準誤差が小さい)．p_2 は Shimodaira (2002)で提案した近似的に不偏な p-値であ

り，CONSEL が計算する AU 値に相当する．これをさらに改良するのが p_3 である．p_2 に比べると p_3 は SH 検定の p-値に似た性質をもつ．SH 検定は最悪ケース(すべての木の対数尤度がほぼ同じ)を想定しており一般に保守的(p-値が必要以上に大きくなる)である．しかし本当に最悪ケースの場合は逆に p_2 が小さくなりすぎる傾向がある．一般的に p_3 は p_2 と似た数値になるが，最悪ケースに近づくと p_3 は SH 検定に似た数値になる．

3. マルチスケール・ブートストラップ法の理論

3.1 ブートストラップ確率

ここでは，マルチスケール・ブートストラップ法のアルゴリズムとその理論を説明する．まずその基礎となるブートストラップ法を簡単な数値例を使って説明する．

サイズ $n \times m$ のデータ行列を $X=(x_{ij})$，その i 行目を $x_i=(x_{i1},\ldots,x_{im})$ と書く．$x_1,\ldots,x_n$ は独立に同分布にしたがうサンプルサイズ n のデータとみなす．0 または 1 の値をとる関数 $f(X)$ を考える．たとえば系統樹推定で t4 が対数尤度を最大にするとき $f(X)=1$，他の木が対数尤度を最大にするとき $f(X)=0$ とする．先ほどの数値例では $f(X)=0$ であるが，これがどれほど信頼できるのかを評価したい．

ブートストラップ法で生成するブートストラップ標本はサイズ $n' \times m$ のデータ行列 $X^*=(x^*_{ij})$ である．その i 行目を $x^*_i=(x^*_{i1},\ldots,x^*_{im})$ と書くと，X^* はサンプルサイズ n' のデータ $x^*_1,\ldots,x^*_{n'}$ である．ブートストラップ法では各 x^*_i を $x_1,\ldots,x_n$ からランダムに重複を許して選ぶ．この手続きを繰り返し実行して X^* を多数生成しそのバラツキを調べるというのが，Efron (1979)の提案したブートストラップ法のアイデアである．たった一つのデータ X から多数の標本を生成することに由来して命名されている．データ自身から再びサンプリングすることを一般にリサンプリングというが，ブートストラップ法は同じ要素が重複して選ばれることを許すリサンプリング(復元抽出)である．

X^* を生成する手続きを B 回繰り返して $X^{*1},\ldots,X^{*B}$ を生成する．各 X^* で $f(X^*)$ を計算して $f(X^{*1}),\ldots,f(X^{*B})$ のバラツキを調べれば $f(X)$ の値をどれほど信頼できるかが分かるだろう．そこで Felsenstein (1985)によって考案された信頼度がブートストラップ確率

$$\alpha_{\sigma^2}=\frac{1}{B}\sum_{b=1}^{B} f(X^{*b})$$

である．この値が大きいほど $f(X)=1$ の可能性が高いと判断する．

X^* のバラツキの程度をあらわすスケールを $\sigma=\sqrt{n/n'}$ としておき，n' のかわりに σ^2 を用いてスケール依存性を表記する．ブートストラップ確率は σ^2 の関数である．通常は $\sigma^2=1$ を用いるが，マルチスケール・ブートストラップ法ではいくつかの σ^2 の値で α_{σ^2} を計算する．なお，リサンプリングによって $B\alpha_{\sigma^2}$ は 2 項分布にしたがうから，標準誤差は $\sqrt{\alpha_{\sigma^2}(1-\alpha_{\sigma^2})/B}$ と見積もれる．これを小さくするために B は十分大きくしておく(本稿では $B=10000$)．

各 $f(X^*)$ の計算コストが大きい場合，ブートストラップ法を上記の定義どおりに実行するとパソコン程度の計算能力では不十分なことがある．並列計算はひとつの解決策であるが，今のところ誰もが使えるわけではない．そこで Kishino et al. (1990)は RELL (resampling of estimated log likelihoods)法と呼ばれる線形近似計算を考案した．RELL 法は分子系統樹推定で普及し，CONSEL や scaleboot の relltest 関数にも実装されている．本来のデータである DNA 配列をリサンプリングせず，あらかじめ最尤法を実行して得られた site-wise log-likelihood を直接リサンプリングして $f(X^*)$ を近似計算することにより，大幅な高速化が実現されている．

3.2 簡単な数値例

R を用いた数値例として $n=100$, $m=10$ のデータ行列 X を用意した．その最初の 3 行，すなわち x_1, x_2, x_3 は

```
> round(X[1:3, ], 2) # X の最初の 3 行を有効数字 2 桁で表示
     [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8]   [,9] [,10]
[1,] -6.56 0.93  0.31 10.09  2.89  3.52 -4.53 7.59  -6.71 -2.01
[2,] -0.64 0.57 -0.91  8.63  2.96  0.04 -2.60 5.55  -7.60 24.69
[3,] -5.92 7.25  3.72  8.96 -6.42 -4.05 -5.91 2.75 -10.04  4.06
```

である．$f(X)$ の例として，次のようなものを考えた．10 次元ベクトル $y=\frac{1}{n}\sum_{i=1}^{n}x_i$ とおいて，$y_1 \geq y_j$, $j=2,\ldots,10$ のとき $f(X)=1$，そうでないとき $f(X)=0$ とする．y を計算すると

```
> y <- apply(X, 2, mean) # 各列の平均値を計算
> round(y, 15) # 有効数字 15 桁で表示
 [1] 0 1 1 1 1 1 1 1 1 1
```

すなわち $y_1=0$, $y_2=\cdots=y_{10}=1$ である(こうなるように X を準備した)から，$f(X)=0$ である．$f(X)$ に相当する R 関数は

```
> mc1 <- function(y) all(y[1] >= y[-1]) # y[1] が最大値？
> mc1(y)
[1] FALSE
```

である(0, 1 のかわりに FALSE, TRUE を返す)．

scaleboot パッケージを使って α_{σ^2} を計算するには，図 1 を実行する．σ^2 などのパラメータを指定して scaleboot 関数を呼び出す．σ^2 の値は 1/9 から 9 までの区間を対数目盛りで 13 個選んだ．result1 の Scales の行を見ると，$\sigma^2=0.1111, 0.1603, 0.2309, \ldots, 9.091$ とある．指定したはずの σ^2 と多少ずれるのは n' を整数にするためである．ブートストラップ確率は $\alpha_{\sigma^2}=0.0000, 0.0001, 0.0005, \ldots, 0.0461$ であった．各スケールにおいて $B=10000$ であったから，たとえば $\sigma^2=9.091$ $(n'=11)$ のとき，$f(X^{*b})=1$ となった回数が 461 回である．

scaleboot 関数の結果をプロットすれば，α_{σ^2} の変化がわかる．たとえば

```
> plot(result1, xval = "sigma", log = "x", yval = "pvalue", legend = "topleft")
```

とすれば図 2 (a) を得る．軸の指定をしない場合，

```
> plot(result1, legend = "topleft")
```

とすれば図 2 (b) を得る．このとき縦軸は σz_{σ^2} であり，これを正規化ブートストラップ z-値 (normalized bootstrap z-value)，略して正規化 z-値と呼ぶことにする．ただし，ブートストラップ z-値を

$$z_{\sigma^2}=-\Phi^{-1}(\alpha_{\sigma^2})$$

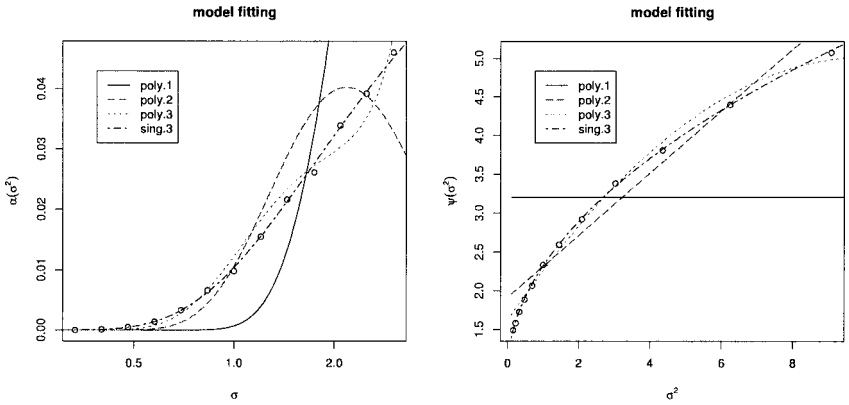
で定義しておく．$\Phi^{-1}(\cdot)$ は標準正規分布の分布関数 $\Phi(\cdot)$ の逆関数である．ブートストラップ確率の変化をモデル化する際に，この正規化 z-値が重要な役割を果たす．

3.3 正規モデル

さきほどの数値例では $y=\frac{1}{n}\sum_{i=1}^{n}x_i$ を通して $f(X)$ を定義していた．より一般的に，なんらかの非線形変換 $y=g(X)$ を通して $f(X)$ を定義する．そして，適当な K 次元空間で y は多変

```
> ## まず重みベクトル w から f(X^*) を計算する関数を用意しておく
> ## w[i] はリサンプリングにおいて i 番目の要素が選ばれた回数
> countw <- function(x, w, fn) {
+     y <- apply(w * x, 2, sum)/sum(w) # x[i] の重み w[i] から y を計算
+     fn(y) # f(X^*)
+ }
> ## パラメータの指定
> sa <- 9^seq(-1, 1, length = 13) # sigma^2 のベクタ
> nb <- 10000 # ブートストラップ反復数 B
> ## マルチスケール・ブートストラップ法の実行と結果の表示
> result1 <- scaleboot(X, nb, sa, countw, mc1) # これに数分かかる
> result1 # 結果の表示（補助的なものだけ）
Multiscale Bootstrap Probabilities (percent):
1    2    3    4    5    6    7    8    9    10   11   12   13
0.00 0.01 0.05 0.14 0.33 0.66 0.98 1.55 2.16 2.61 3.39 3.92 4.61

Numbers of Bootstrap Replicates:
1     2     3     4     5     6     7     8     9     10    11    12    13
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000

Scales (Sigma Squared):
1      2      3      4      5      6      7 8     9     10   11    12   13
0.1111 0.1603 0.2309 0.3333 0.4808 0.6944 1 1.449 2.083 3.03 4.348 6.25 9.091

Coefficients:
       beta0           beta1           beta2
poly.1 3.2017 (0.0182)
poly.2 1.9077 (0.0217) 0.4004 (0.0069)
poly.3 1.6103 (0.0279) 0.6756 (0.0209) -0.0335 (0.0024)
sing.3 0.9280 (0.0287) 1.3860 (0.0203)  1.0000 (0.0000)

Model Fitting:
       rss     df pfit   aic
poly.1 3690.22 12 0.0000 3666.22
poly.2  220.16 11 0.0000  198.16
poly.3   37.23 10 0.0001   17.23
sing.3    1.95 10 0.9967  -18.05

Best Model:  sing.3
```

図 1.　マルチスケール・ブートストラップの実行．ブートストラップ確率は 3.2 節，モデル当てはめは 3.4 節を参照．



(a) (b)

図 2. ブートストラップ確率の変化．各スケールで計算されたブートストラップ確率が○でプロットされ，それに 4 種のモデルをあてはめた結果が曲線として示されている．(a) 横軸=σ (対数目盛り)，縦軸 $=\alpha_{\sigma^2}$．(b) 横軸=σ^2，縦軸=σz_{σ^2}．`scaleboot` においてプロットの縦軸はそれぞれ $\alpha(\sigma^2)$ および $\psi(\sigma^2)$ と表記される．

量正規分布

$$y \sim N_K(\mu, I) \tag{3.1}$$

にしたがうと仮定する (Efron and Tibshirani, 1998)．期待値ベクトル μ は未知，分散共分散行列は単位行列 I である．前節の簡単な数値例では各 x_i を多変量正規分布から生成していて，実は (3.1) が成り立っている．

$f(X)=1$ となる y の集合を $\mathcal{H}$ と表記する．さきほどの数値例では，$\mathcal{H}$ は不等式 $y_1 \geq y_j$, $j=2,\ldots,10$ によって作られる凸多面体錐である．ブートストラップ標本 X^* は $y^*=g(X^*)$ と変換され，これがしたがう分布を

$$y^* \sim N_K(y, \sigma^2 I) \tag{3.2}$$

と近似する．

この正規モデルが厳密に成り立つように g がとれればよいし，もしそうでなくても近似的に正規モデルが成り立つとして議論を進める．マルチスケール・ブートストラップ法を応用するときは，このような g の存在だけを仮定すればよく，実際に g の関数形や K の値を求める必要はない．

データから $y \in \mathcal{H}$ かどうかはすぐわかるが，(3.1) にしたがって y をサンプリングするたびにその値は変化する．このランダムネスの影響を取り除いたとき $y \in \mathcal{H}$ がどうなるかに興味がある．もし n を増やせるなら (3.1) の分散共分散行列は n^{-1} に比例して小さくなり，$n \to \infty$ の極限では $y \to \mu$ である．そこで，

$$\mu \in \mathcal{H} \tag{3.3}$$

を帰無仮説とした検定を行い，その不偏な p-値を計算することを目的とする．

これ以降，(3.1)，(3.3)，(3.2) を前提として議論を進める．正規モデルを前提とせず，より一般の指数型分布族における議論は，Shimodaira (2004) を参照されたい．

3.4　ブートストラップ確率のスケール変換則

正規化 z-値のスケール依存性をあらわすモデルを一般に $\psi(\sigma^2|\beta)$ と表記する．すなわち，モデルが正しく指定されていれば $\sigma z_{\sigma^2}=\psi(\sigma^2|\beta)$，$\sigma>0$ である．β はパラメータベクトルである．図 2 (a) では 13 個のブートストラップ確率に 4 種のモデルをあてはめ，パラメータの最尤推定値 $\hat\beta$ を得ている．どのモデルでもおなじ β という記号を用いるが，その成分の意味はモデルごとに再定義する．図 2 (a) および図 2 (b) では，それぞれ $\Phi(-\psi(\sigma^2|\hat\beta)/\sigma)$ および $\psi(\sigma^2|\hat\beta)$ の曲線を示してある．

このモデルを議論するために，Shimodaira (2008) は一種の漸近論を考察した．まず適当に座標変換して仮説を

$$\mathcal{H}=\{(u,v):v\le -h(u),u\in\mathbb{R}^{K-1}\} \tag{3.4}$$

と表現しておく．ただし $y=(u,v)\in\mathbb{R}^K$，$u\in\mathbb{R}^{K-1}$，$v\in\mathbb{R}$ であり，$h(u)$ は連続関数とする．そして仮説領域の境界 $\partial\mathcal{H}$ は十分に平坦と仮定する．つまり h の変動のオーダを $O(\tau)$ とおいて，$\tau\to 0$ を仮定する．τ には特別な意味はなく，この漸近論のために導入した．通常の漸近論とあえて対応付ければ，$\tau=n^{-1/2}$ に相当する．

もし h が滑らかならば

$$\sigma z_{\sigma^2}=\sum_{j=0}^{\infty}\beta_j\sigma^{2j}+O(\tau^2)$$

と書けて，β_j は h の $2j$ 階微分で表現される．そこで，最初の k 項を使い σ^2 の多項式モデル

$$\psi(\sigma^2|\beta)=\sum_{j=0}^{k-1}\beta_j\sigma^{2j},\quad k\ge 1.$$

を定義し，`poly.k` と表記する．ところが図 2 (a) をみると，`poly.1`，`poly.2`，`poly.3` の当てはまりは良くない．

もし $\mathcal{H}$ が錐ならば，その頂点付近で

$$\sigma z_{\sigma^2}=\beta_0+\beta_1\sigma+\sum_{j=1}^{\infty}\beta_{j+1}\sigma^{-j}+O(\tau^2),\quad \beta_j=O\big((\text{頂点からの距離})^j\big),\ j\ge 2$$

と書ける．最初の 2 項をつかった $\beta_0+\beta_1\sigma$ を先ほどの `poly.(k-1)` に取り込んで

$$\psi(\sigma^2|\beta)=\beta_0+\sum_{j=1}^{k-2}\frac{\beta_j\sigma^{2j}}{1+\beta_{k-1}(\sigma-1)},\quad 0\le\beta_{k-1}\le 1,\quad k\ge 3$$

と定義し，`sing.k` と表記する．図 2 (a) をみると，`sing.3` の当てはまりはとても良い．そのときパラメータ推定値は $\hat\beta_2=1$ であったから，`sing.3` は最も簡単な錐モデル $\psi(\sigma^2|\beta)=\beta_0+\beta_1\sigma$ だったことになる．

どのモデルの当てはまりが最も良いかを AIC をつかって評価する．α_{σ^2} を計算した σ^2 の値を σ_i^2，$i=1,\dots,S$ (ここでは $S=13$)，各スケールで $f(X^*)=1$ を観測した回数を $C_i=B\alpha_{\sigma_i^2}$ とすると，各 C_i が 2 項分布にしたがうことから対数尤度関数は

$$\ell(\beta)=\sum_{i=1}^{S}\Big\{C_i\log\Phi(-\psi(\sigma_i^2|\beta)/\sigma_i)+(B-C_i)\log\Phi(\psi(\sigma_i^2|\beta)/\sigma_i)\Big\}$$

である．この $\ell(\beta)$ を数値的に最大化して $\hat\beta$ を求めている．AIC 値は制約のないモデルとの差をとって

$$\mathrm{AIC}=(-2\ell(\hat\beta)+2k)-\left\{-2\sum_{i=1}^{S}(C_i\log(C_i/B)+(B-C_i)\log(1-C_i/B))+2S\right\}$$

によって計算している．図 1 をみると，sing.3 の AIC 値が最小であり，最も当てはまりの良いモデルとして選択されている．

3.5 近似的に不偏な確率値

検定の p-値を一般に p または $p(y)$ と書く．不偏検定の p-値は，もしそれが存在するならば，$\mu \in \partial\mathcal{H}$ のとき一様分布 $p(y) \sim U(0,1)$ にしたがう．そして Shimodaira (2008) の Theorem 1 によれば

$$p = \Phi(-\psi(-1|\beta)) + O(\tau^3) \tag{3.5}$$

である．正規化 z-値を $\sigma^2 = -1$ に外挿したものを q とおけば $p = \Phi(-q)$ であることを意味する．いわばブートストラップ確率を $n' = -n$ に外挿すれば不偏な p-値が得られると解釈してもよい．

現実には，当てはまりの最も良いモデルを用いて $\psi(\sigma^2|\hat{\beta})$ を $\sigma^2 = -1$ へ外挿する．ここでは $\sigma_0^2 = 1$ の周りでテーラー展開してその k 項を用いた外挿

$$p_k = \Phi\left\{ -\sum_{j=0}^{k-1} \frac{(-1-\sigma_0^2)^j}{j!} \frac{\partial^j \psi(\sigma^2|\hat{\beta})}{\partial(\sigma^2)^j}\Big|_{\sigma_0^2} \right\} \tag{3.6}$$

を計算し，p-値のクラス p_k，$k = 1,2,3,\ldots$ を定義する．さきほどの数値例で $p_1,\ldots,p_4$ を実際に計算すると，

```
> summary(result1,k=1:4) # k のデフォルト値は k=1:3
Raw Bootstrap Probability:  0.98(0.10)
Corrected P-values(percent):
           k.1          k.2          k.3          k.4         aic
  poly.1  0.07(0.00)   0.07(0.00)   0.07(0.00)   0.07(0.00)  3666.22
  poly.2  1.05(0.05)   6.59(0.35)   6.59(0.35)   6.59(0.35)   198.16
  poly.3  1.21(0.05)  15.03(0.95)  18.37(1.30)  18.37(1.30)    17.23
  sing.3  1.03(0.04)  17.67(0.74)  40.71(1.47)  67.65(1.71)   -18.05
Best Model:  sing.3
```

となる．単純なブートストラップ確率は $\alpha_1 = 0.0098$ である．これを補正した p_k が各モデル毎に計算されている．当てはまりの悪い poly.k モデルは無視して sing.3 モデルの行をみると，$p_1 = 0.0103$，$p_2 = 0.1767$，$p_3 = 0.4071$，$p_4 = 0.6765$ である．この外挿の様子をプロットするには

```
> plot(summary(result1), legend = "topleft")
```

とすれば図 3 が得られる．

この数値例の帰無仮説は多重比較法で従来から扱われてきたものである．多重比較法（系統樹推定でいえば SH 検定）の p-値は凸錐 $\mathcal{H}$ の頂点に μ があると仮定して計算する．実際にやってみると $p = 0.6591$ であるから，p_3 または p_4 が結果的に近い数値を与えている．

Shimodaira (2008) の Theorem 2 によれば，$\mu \in \partial\mathcal{H}$ のとき $k \to \infty$ で p_k のしたがう確率分布は一様分布に収束する．もし h が $2k-1$ 次以下の多項式ならば p_k は不偏な p を与える．これと同じ性質を k 回反復ブートストラップ（$k = 2$ ならいわゆるダブル・ブートストラップ）が持つことも示されるが，その計算量は $O(B^k)$ であり現実的でない．マルチスケール・ブートストラップ法はずっと少ない計算量 $O(B)$ で同様の性能である．

しかし凸錐の頂点付近のように滑らかでない場合にはそもそも不偏な p が存在しないことが知られている．それでも $k \to \infty$ で $P(p_k(y) < \alpha) = \alpha$ に収束する意味で検定のバイアスを小さ

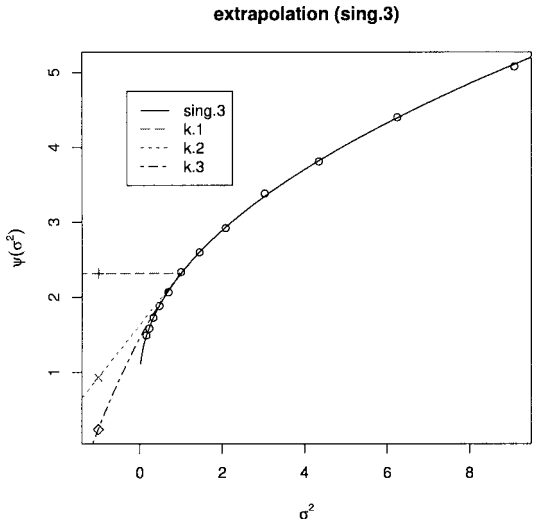


図 3.　正規化 z-値の $\sigma^2=-1$ への外挿．当てはまりの良かった sing.3 モデルの $\psi(\sigma^2|\hat{\beta})$ が実線で示され，$\sigma_0^2=1$ における k 項のテーラー展開による外挿が $k=1,2,3$ まで破線等で示されている．

くするのだが，$p_k(y)$ が y の微小変化で大きく振動して発散してしまう．したがって，k を大きくすると p_k が数値的に不安定になる．不偏検定を目指す，という設定自体に問題があることを示唆しているが，いまのところ他に妥当な設定が考えられていない．

4.　階層型クラスタリングの信頼度

4.1　肺腫瘍の遺伝子発現データ

DNA マイクロアレイ技術によって遺伝子の発現レベルを測定することが広く行われるようになっている．腫瘍を数種類のタイプに分類するために階層型クラスタリングが用いられる．そこで得られたクラスタ (分類群) の信頼度を計算するためにマルチスケール・ブートストラップ法が利用できる．ここでは，Garber et al. (2001) の肺腫瘍データを分析する実例を scaleboot パッケージの利用手順とともに示す．とくに 4.3 節ではモデル診断まで説明する．これは 2 節では行わなかったが，同様の手順を系統樹推定にも適用可能である．

階層型クラスタリングにマルチスケール・ブートストラップ法を実行するには，まず pvclust パッケージを R で利用する．この数値例として肺腫瘍データ lung が用意されている．916 遺伝子の発現レベルを 73 個の肺腫瘍サンプルで測定したものである．

```
> library(pvclust) # pvclust ライブラリの読み込み
> data(lung) # lung データの読み込み
> dim(lung) # データ行列の次元 = 916 * 73
[1] 916  73
> round(lung[1:3,1:5],2) # 最初の 3 行 5 列を有効数字 2 桁で表示
              fetal_lung  232-97_SCC  232-97_node  68-96_Adeno  11-00_Adeno
IMAGE:196992       -0.40        4.28         3.68        -1.35        -1.74
IMAGE:587847       -2.22        5.21         4.75        -0.91        -0.33
```

```
IMAGE:1049185       -1.35       -0.84       -2.88        3.35        3.02
```

4.2 マルチスケール・ブートストラップ法の実行

まず 13 個の σ^2 と反復数 $B=10000$ を指定して，pvclust 関数を実行する．

```
> sa <- 9^seq(-1, 1, length = 13) # sigma^2 のベクタ
> nb <- 10000 # ブートストラップ反復数 B
> lung.pvclust <- pvclust(lung, r = 1/sa, nboot = nb) # ブートストラップ法の計算
```

これをパソコンで実行すると丸一日程度かかってしまうが，pvclust は並列計算の機能がありCPU 数にほぼ比例して速く計算できる．pvclust には p_2 までしか実装されていないので，その出力を scaleboot パッケージの sbfit 関数に入力して p_k，$k \geq 3$ を計算する．

```
> library(scaleboot) # scaleboot ライブラリの読み込み
> lung.sb <- sbfit(lung.pvclust) # ブートストラップ確率へモデル当てはめ
> lung.k3 <- sbpvclust(lung.pvclust, lung.sb, k=3) # 結果 (p3) の書きもどし
```

得られた結果(lung.k3)は pvclust で扱える形式になっていて，pvclust の出力(lung.pvclust)の p_2 が p_3 で置き換えられたものである．図 4 を得るには次を実行する．

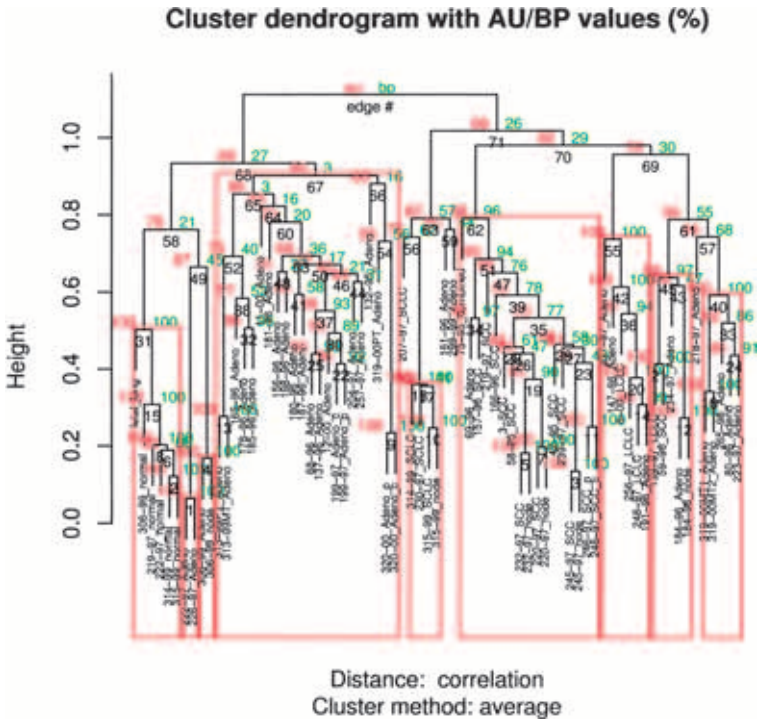


図 4. lung データの樹形図．各クラスタにつけられた数値は左側が p_3，右側が α_1，下側がクラスタ番号である．$p_3 > 95\%$ となる 9 個のクラスタが赤い枠で囲まれている．

```
> plot(lung.k3, cex = 0.5, cex.pv = 0.7) # 樹形図の表示
> pvrect(lung.k3) # p3 > 95%のクラスタを赤い枠で囲む
```

赤い枠で囲まれた 9 個のクラスタは十分に信頼できると判断する．そのクラスタが存在しないという帰無仮説の p-値は $1-p_3$ で与えられるので，$p_3 > 0.95$ ならばクラスタの存在が示唆される．

4.3　モデル診断

各クラスタの p_3 と α_1 は図 4 から読み取れるが，α_1，p_1，p_2，p_3 を直接表示するには sbfit 関数の結果(lung.sb)を利用すればよい．たとえば，クラスタ番号 62，67，69，71 で表示するには次を実行する．

```
> summary(lung.sb[c(62, 67, 69, 71)]) # p-値の表示
Corrected P-values(percent):
        raw          k.1          k.2          k.3       model     aic
62   95.68(0.20)  95.92(0.10)  98.64(0.10)  98.61(0.12)  poly.3  -12.01
67    3.63(0.19)   3.31(0.07)  77.02(0.47)  95.10(0.17)  sing.3   25.11
69   29.49(0.46)  29.65(0.17)  75.37(0.22)  75.83(0.34)  poly.3  -14.09
71   25.20(0.43)  25.95(0.17)  84.44(0.18)  85.91(0.27)  poly.3   11.49
```

正規化 z-値のモデル $\psi(\sigma^2|\beta)$ は 3.4 節で説明した poly.k，sing.k が実装されている．sbfit 関数はデフォルトで poly.1，poly.2，poly.3，sing.3 のモデル当てはめを実行する．各クラ

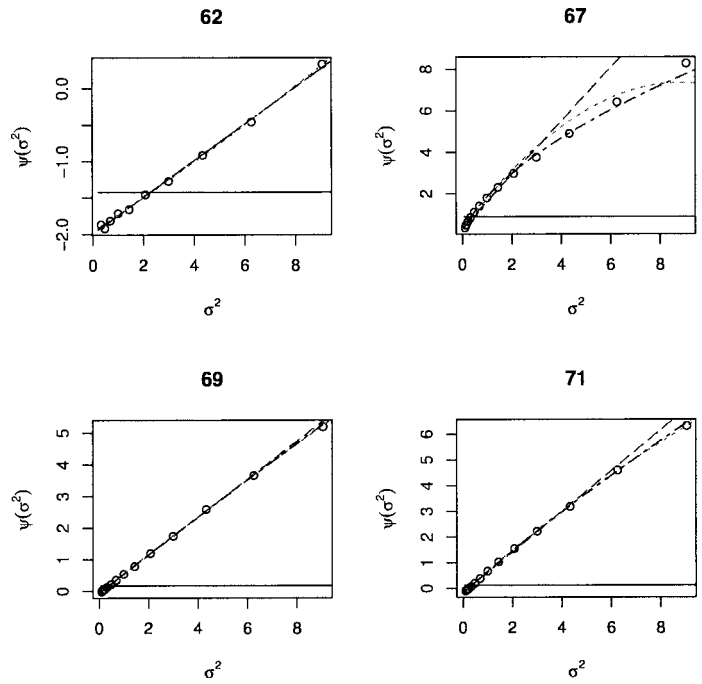


図 5.　4 個のクラスタ (番号=62，67，69，71) のモデル当てはめ．横軸=σ^2，縦軸=σz_{σ^2}．
図 2 (b) と同じ 4 種のモデルを当てはめて曲線を示した．

スタでどのモデルが選択されたかは上記 summary 表示の model 列からわかる.

このモデル当てはめの良さを診断するために,

```
> plot(lung.sb[c(62, 67, 69, 71)]) # モデル当てはめのプロット
```

によって正規化 z-値をプロットして図 5 に示す. クラスタ番号 62, 69, 71 では poly.3 モデルが選択されていたが, 正規化 z-値はほぼ直線で近似されるので poly.2 モデル $\psi(\sigma^2|\beta)=\beta_0+\beta_1\sigma^2$ も十分に当てはまりがよい. 仮説境界 $\partial\mathcal{H}$ が 2 次曲面で近似できる程度しか曲がっておらず, 不偏な p-値に十分な妥当性がある. p_2 と p_3 の差もほとんどなく, $k=2$ でほぼ収束している.

一方, クラスタ番号 67 は sing.3 モデルが選択されており, 仮説境界の形状が滑らかな曲面で近似できず錐となる部分が含まれることが分かる. 詳細を確認するため, 次を実行して図 6 (a), (b) を得る.

```
> plot(lung.sb[[67]], legend = "topleft") # モデル当てはめのプロット
> plot(summary(lung.sb[[67]]), legend = "topleft") # 外挿のプロット
> summary(lung.sb[[67]]) # p-値の詳細表示
Raw Bootstrap Probability:  3.63(0.19)
Corrected P-values(percent):
             k.1          k.2          k.3          aic
  poly.1 18.41(0.10)  18.41(0.10)  18.41(0.10)  52878.29
  poly.2  5.46(0.09)  83.51(0.32)  83.51(0.32)   1356.63
  poly.3  3.95(0.08)  86.05(0.29)  92.56(0.28)    464.71
  sing.3  3.31(0.07)  77.02(0.47)  95.10(0.17)     25.11
Best Model:  sing.3
```

図 6 (a) や上記 AIC をみると, poly.3 でも当てはまりが悪く sing.3 が必要であることが確認できる. この場合 p_k は k の増加でかならずしも収束せず数値的に不安定になるので, k をあまり大きくせずに $k=3$ 程度の利用が望ましい.

ここで調べた 4 個のクラスタでは与えたモデルで十分な当てはまりが得られているが, もし

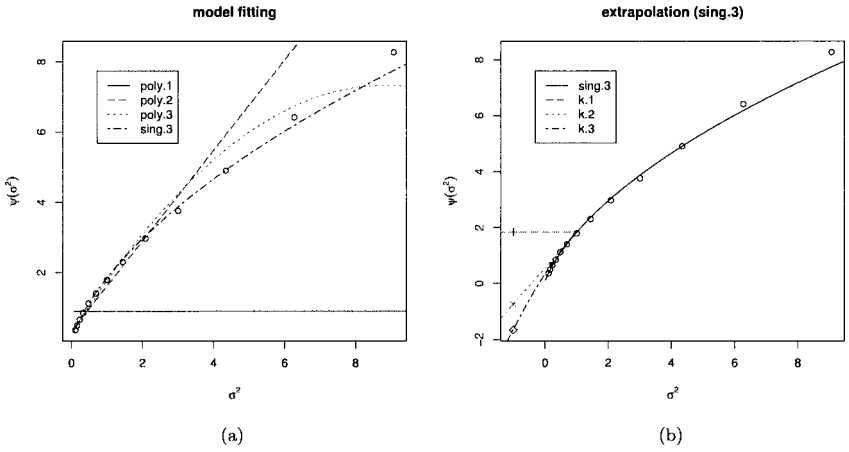


図 6. クラスタ番号 67 (図 4 の左から 4 番目の赤い枠) のモデル当てはめと $\sigma^2=-1$ への外挿.

それで不十分な場合は `poly.k` や `sing.k` の $k \geq 4$ を利用するか，より適切なモデルを開発する必要がある．

5. おわりに

マルチスケール・ブートストラップ法の理論と実装を説明した．Shimodaira (2008)のアルゴリズムが計算する信頼度 p_k は，仮説境界が滑らかな場合には $k \to \infty$ とするとき不偏な p-値に収束する．仮説境界が滑らかでない場合には一般に不偏な p-値が存在しないが，それでもバイアスを減少させる．しかしながら，不偏な p-値を求めること自体に十分な合理性があるとはいえず，これに代わる設定をさがすべきかもしれない．単にベイズにすれば解決するというような問題ではないが，Efron and Tibshirani (1998)では不偏な p-値とベイズ事後確率の接点が議論されていて，今後のひとつの方向性になるかもしれない．

参 考 文 献

Efron, B. (1979). Bootstrap methods: Another look at the jackknife, *Annals of Statistics*, **7**, 1–26.

Efron, B. and Tibshirani, R. (1998). The problem of regions, *Annals of Statistics*, **26**, 1687–1718.

Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap, *Evolution*, **39**, 783–791.

Garber, M. E., Troyanskaya, O. G., Schluens, K., Petersen, S., Thaesler, Z., Pacyna-Gengelbach, M., van de Rijn, M., Rosen, G. D., Perou, C. M., Whyte, R. I., Altman, R. B., Brown, P. O., Botstein, D. and Petersen, I. (2001). Diversity of gene expression in adenocarcinoma of the lung, *Proceedings of the National Academy of Sciences of the United States of America*, **98**, 13784–13789.

Kishino, H., Miyata, T. and Hasegawa, M. (1990). Maximum likelihood inference of protein phylogeny and the origin of chloroplasts, *Journal of Molecular Evolution*, **30**, 151–160.

R Development Core Team (2005). *R*: *A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, `http://www.R-project.org`.

下平英寿(2002). ブートストラップ法によるクラスタ分析のバラツキ評価，統計数理，**50**, 33–44.

Shimodaira, H. (2002). An approximately unbiased test of phylogenetic tree selection, *Systematic Biology*, **51**, 492–508.

Shimodaira, H. (2004). Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling, *Annals of Statistics*, **32**, 2616–2641.

Shimodaira, H. (2006). Scaleboot: Approximately unbiased p-values via multiscale bootstrap, `http://www.is.titech.ac.jp/~shimo/`.

Shimodaira, H. (2008). Testing regions with nonsmooth boundaries via multiscale bootstrap, *Journal of Statistical Planning and Inference*, **138**, 1227–1241, `http://dx.doi.org/10.1016/j.jspi.2007.04.001`.

Shimodaira, H. and Hasegawa, M. (1999). Multiple comparisons of log-likelihoods with applications to phylogenetic inference, *Molecular Biology and Evolution*, **16**, 1114–1116.

Shimodaira, H. and Hasegawa, M. (2001). CONSEL: For assessing the confidence of phylogenetic tree selection, *Bioinformatics*, **17**, 1246–1247.

Suzuki, R. and Shimodaira, H. (2006). Pvclust: An R package for assessing the uncertainty in hierarchical clustering, *Bioinformatics*, **22**, 1540–1542.

Yang, Z. (1997). PAML: A program package for phylogenetic analysis by maximum likelihood, *Computer Applications in the Biosciences*, **13**, 555–556.

 Proceedings of the Institute of Statistical Mathematics Vol. 56, No. 1, 3–18 (2008)

Theory and Implementation of Multiscale Bootstrap for Regions with Nonsmooth Boundaries

Hidetoshi Shimodaira

Department of Mathematical and Computing Sciences, Tokyo Institute of Technology

Bootstrap method has been used widely for computing confidence levels of the output of data analysis. This confidence level is called bootstrap probability, and it is known to be biased as a p-value of hypothesis testing. The multiscale bootstrap method has been invented to compute confidence levels with high accuracy by correcting the bias. Previously the boundary of hypothesis was assumed to be smooth, but the method is generalized by now for non-smooth cases. The theory and implementation of the algorithm will be explained using real examples of molecular phylogenetic inference and hierarchical clustering.

Key words: Approximately unbiased tests, bootstrap probability, bias correction, scaling-law, phylogenetic inference.