データ解析 講義資料

下平英寿

http://www.is.titech.ac.jp/~shimo/class/

目次

イントロダクション

[講義「データ解析」について]

- 「R を用いたデータ解析入門」
- 目標
  1. Rを利用して実践的なデータ解析ができるようになること
     ⇒ R に含まれる関数を呼び出してデータ解析を実行する
  2. 背後にある数学，統計学，アルゴリズムを理解すること
     ⇒ 自分自身で関数を記述し，それを用いてデータ解析を行う
- R プログラミング技法に関しては，「習うより慣れろ」とする．文法は C や Java と似ている．
- 講義資料やデータセット等はウェブ http://www.is.titech.ac.jp/~shimo/class/ から各自ダウンロードする．
- 成績評価：レポート提出でおこなう．
- レポート：本年度からレポートのスタイルが変わります．データ解析をおこない出力をメールの添付ファイルで提出する（2 回程度）．たとえば，「ギブスサンプラーによる画像復元」，「スパムメール判別」

など，講義資料で扱っているものを基本とします．これとは別に，講義資料の「課題」もしくはそれに類似の問題をレポート提出する予定です（2 回程度）．レポートの詳細はのちほどお知らせします．

- 出席：本年度から出席はとらないことにします．

[この講義資料について]

- 2007 年度から内容を一新しました．
- 各章の「サブセクション」，「キーワード」のなかによく知らない単語があれば Google 等で検索してみる．
- 「例」のRプログラムを自分でも実行してみること．なお，これらはあくまでサンプルであり，実行速度の工夫はあまりしていない．
- 「課題」のうち重要なものは講義で説明するので，よく理解すること．説明されなかったものについても各自考えてみる．
- ２年生の「確率と統計第一」，「確率と統計第二」で習った事柄の復習も含まれる．
- 講義を進めるうちに講義資料を改訂するかもしれない．ウェブで最新版を確認してください．

[R とは？]

- **データ操作，統計計算，グラフィックスのための統合ソフトウエア環境．**
- **行列操作に優れている，データ解析の一貫したツール群，簡単で効率的なプログラム言語．**
- **R はフリーソフトでソースも公開** http://cran.r-project.org**．**
- **R の開発は１９９０年代後半からネット上で行われている．安定して広く使われるようになったのは２０００年ころから（だと思う）．**
- **R の前身であるは S は C 言語や UNIX と同じ** AT&T(**現** Lucent Technologies) **のベル研究所で１９８４年ころ開発 (ちなみに** C **言語および** UNIX **の開発は１９７１年ころ)．**
- **現在では，膨大なライブラリがユーザによって開発されている．**2007 **年** 3 **月** 5 **日にしらべると** CRAN **に登録されている公式ライブラリだけで** 1008 **個だった...（関数の数でなくて，ライブラリの数！）**
- **そのほかの統計関連ソフトウエア：**SAS, SPSS, Mathematica

[R **の入手や情報源]**

- **本家サイト：** The R Project for Statistical Computing http://cran.r-project.org
- **日本語による** Wiki **情報サイト：**RjpWiki http://www.okada.jp.org/RWiki/

- **インストール：** Windows, Mac, Linux **等でバイナリ配布物があります．**
- R **の日本語マニュアル：当専攻の間瀬茂先生のページにマニュアルの日本語訳があります** http://www.is.titech.ac.jp/~mase/R.html**．**PDF **版は東京学芸大学の森厚さんページ** http://buran.u-gakugei.ac.jp/~mori/LEARN/R/**．特に，**R **の「公式マニュアル日本語版** Introduction to R ver.1.7.0**」を入手して，ざっと目を通すことをお勧めします．**Appendix A **の「入門セッション」をとりあえず実行してみるのも良いでしょう．（ただし，かなり古いバージョンであることに注意．）**

[R **の利用]**

起動 OS **のコマンドラインから** % R [return]

終了 R **のコマンドラインから** > q() [return] **のあとに**
Save workspace image? [y/n/c]:**に対して** y **と打つ．これで作業ディレクトリに**.RData **というファイルが自動的に作られて定義したオブジェクトが保存される．次回** R **を起動したときに自動的に読み込まれる．以降，**R **のコマンドラインからの入力を**>**によって示す．**

代入 > a <- 1:10 **は** (1,2,...,10) **というベクトルを** a **に代入．**

計算 > a^2 **は** a **の要素を２乗して結果を表示**

```
 [1]   1   4   9  16  25  36  49  64  81 100
```

グラフ > plot(a,a^2) **は** a **と** a^2 **の２次元プロット（散布図と言う）．**

関数定義 > foo <- function(x) sum(x^2) **は要素の２乗和を求める関数を定義し** foo **に代入．呼び出しは** foo(a) **とすれば，**[1] 385 **と結果が表示される．**

繰り返し for(i in 1:10) {...} **は** i **を** 1,...,10 **まで変化させて括弧内を実行．**

```
> x <- rep(0,10); for(i in 1:10) x[i] <- i^2
> x
 [1]   1   4   9  16  25  36  49  64  81 100
```

ヘルプ > help(for) **は** for **文（と関連する制御構造）についての解説（もしエラーになったら**> help("for") **を試してください）．**> help(":") **は**:**オペレータの解説．**

ライブラリ > library() **はシステムにインストールされているライブラリパッケージの一覧表示．**
> library(MASS) **は** MASS **ライブラリをロード．**

デモ > demo() **でデモの一覧．たとえば**> demo(graphics) **や**> demo(image) **等で** [return] **を押していけばグラフのデモが見れる．**

emacs **ユーザ は** ESS **という** emacs **パッケージを利用すると便利．**(M-x R **で** R **を起動する．**)

講義で用いるデータファイル等 は講義ホームページ http://www.is.titech.ac.jp/~shimo/class/ **におきます．**

[**参考文献]**

- **機械学習，統計手法の定番教科書：**Trevor Hastie, Robert Tibshirani, Jerome H. Friedman **著**, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2001 **年**.
- R **(もしくは** S) **を用いた統計解析の定番教科書：**W. N. Venables, Brian D. Ripley **著，**Modern Applied Statistics with S **第４版，**Springer-Verlag**，２０００年．**
- R **を用いた統計解析の入門書：間瀬茂 他 著，「工学のためのデータサイエンス入門—フリーな統計環境** R **を用いたデータ解析」，数理工学社，２００４年．**

- ウェブで入手できるもの：青木繁伸 著，「R によるデータ解析」http://aoki2.si.gunma-u.ac.jp/R/Rstat.pdf，２００７年．
- 「データ解析」の講義資料２００６年版 http://www.is.titech.ac.jp/~shimo/class/gakubu200603.html．

[単回帰分析]　表形式のテキストファイル gakureki-shushou.txt

```
"Gakureki" "Shushou"
"Hokkaido" 7.7 1.23
"Aomori" 5.5 1.47
"Iwate" 6.1 1.56
"Miyagi" 9.6 1.39
... 以下略...
```

を読み込み，「学歴」と「出生率」の関係を調べる．

```
> dat <- read.table("gakureki-shushou.txt") # データの読み込み
> dim(dat) # 行列の次元 （実際には matrix 形式でなくて data.frame という形式で格納されている）
[1] 47  2
> dat[1:5,] # 最初の 5 行だけ表示
         Gakureki Shushou
Hokkaido      7.7    1.23
Aomori        5.5    1.47
Iwate         6.1    1.56
Miyagi        9.6    1.39
```



```
Akita         5.6    1.45
> plot(dat) # 散布図
> f <- lm(Shushou ~ Gakureki, dat) # 単回帰分析
> summary(f) # 結果の詳細
Call:
lm(formula = Shushou ~ Gakureki, data = dat)

Residuals:
      Min        1Q    Median        3Q       Max
-0.294968 -0.048132 -0.009319  0.045992  0.326105

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.742483   0.039973   43.592  < 2e-16 ***
Gakureki    -0.028249   0.003946   -7.158 5.94e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09205 on 45 degrees of freedom
Multiple R-Squared: 0.5324,        Adjusted R-squared: 0.522
F-statistic: 51.24 on 1 and 45 DF,  p-value: 5.943e-09
> abline(f,col="red") # 回帰直線
> plot(dat,type="n"); text(dat,rownames(dat)); abline(f,col="red",lty=2) # 再作図
```

- モデル 出生率 $= \beta_0 + \beta_1 \times$ 学歴 $+ \epsilon$
- 回帰係数 $\hat{\beta}_0 = 1.74$, $\hat{\beta}_1 = -0.028$
- 回帰係数の標準誤差 $\hat{\sigma}_0 = 0.04$, $\hat{\sigma}_1 = 0.004$

- 回帰係数の確率値 $p_0 = 2 \times 10^{-16}$, $p_1 = 6 \times 10^{-9}$
- まとめ：高学歴者の多い都道府県ほど出生率が低下する．大卒率が 10 ポイント増えると，出生率が 0.28 下がる．
- 注意：これは都道府県の特徴を議論しているのであり，個人（又は世帯）における学歴と出生率の関係を議論しているのではない．いずれにしても，学歴と出生率の因果関係を示唆するとは限らない．

[ニュース記事は注意して読む]

２００４年９月２４日 社会ニュース

＜肺がん発生率＞幹線道路近くの住人で高く 胃がんも

幹線道路から５０メートル以内に住んでいる人は肺がんや胃がんになるリスクが高いことが、千葉県がんセンター研究局疫学研究部の三上春夫部長らの調査で分かった。男性の肺がんで１．７６倍、男女の胃がんで１．６８倍、それぞれ発生率が高くなっているという。２９日から福岡市で開かれる日本癌（がん）学会で発表する。三上部長らは９０～９４年に同県内のある市で胃、大腸、肝、子宮、乳房のがんと診断された人のうち、１２時間の交通量が５０００台以上の幹線道路から５００メートル以内に住む５２８人について、幹線道路からの距離を精密に計測した。続いて、当時の国勢調査に基づいた人口と実際の患者数から、５００メートル以内に住む人のがん発生率を割り出した。これをもとに５０メートル以内の発生数を予測し、実際の患者数と比べた。この結果、予測発生数と実際の患者数は、男性の肺がんで９．６４人と１７人、男性の胃がんで２２．０１人と３７人、女性の胃がんで１２．５４人と２１人だった。幹線道路から５０メートル

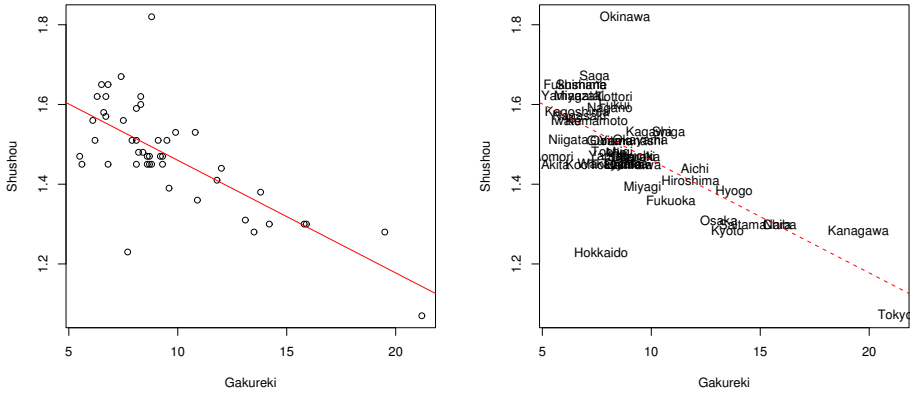


図 1　（左）単回帰分析，（右）県名をいれて再作図したもの

以内に住む人はより遠くの住民よりも、発生率が男性の肺がんで１．７６倍、男女の胃がんで１．６８倍高いことになる。他のがんでは、女性の肺がん２．００倍、男性の大腸がん１．３２倍、女性の大腸がん１．６２倍、男性の肝がん１．４６倍、女性の肝がん１．１９倍、乳がん０．８７倍、子宮がん１．０４倍――との結果だったが、患者数が少ないなどで統計的に意味のある数字にならなかった。三上部長は「５０メートル以内に住むがん患者の年齢は全県平均より若く、交通量の多い幹線道路特有の事情があると考えられる。自動車の排ガスに含まれる有害成分が関与しているとみられるが、胃がんでもリスクが高くなっているので、単純に吸入だけの影響ではないようだ」と話している。【吉川学】毎日新聞ウェブ版 - 9 月 24 日 3 時 5 分更新 より引用

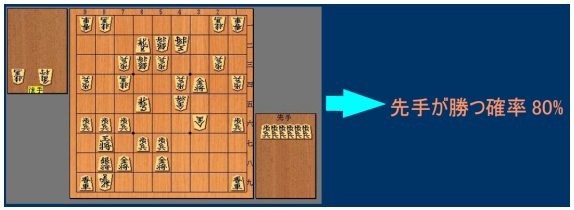
- 幹線道路に近いと空気が悪いので肺がんになりやすい？ (因果関係？)
- 幹線道路に近いと騒音などストレスが多く胃がんになりやすい？
- 幹線道路の近くに住む人はどういう人？ (傾向に関連？)

[卒論紹介 (I)]

2004 年度学士論文

統計的学習を用いたゲーム勝敗予測とコンピュータ将棋への応用

情報科学科 下平研究室 谷口智也



- 勝敗予測関数の構成
  - ロジスティック回帰 (y は先手勝ち=1，負け=0)

$$\log \frac{P(Y=1|x)}{P(Y=0|x)} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

  - 変数選択 (x は盤面評価の特徴量を 369 変数)
- コンピュータ将棋への応用

[主成分分析] `gakureki-rikon-12.txt`: 日本の 47 都道府県について，以下の表にある 12 変量の値．サイズ 47×12 の実数行列．

変数名	コード	意味
Gakureki	E09504	最終学歴が大学・大学院卒の者の割合 (%)
Shushou	A05203	合計特殊出生率
Zouka	A05201	自然増加率 (%)
Ninzu	A06102	一般世帯の平均人員 (人:person)
Kaku	A06202	核家族世帯割合 (%)
Tomo	F01503	共働き世帯割合 (%)
Tandoku	A06205	単独世帯割合 (%)
65Sai	A06301	65 歳以上の親族のいる世帯割合 (%)
Kfufu	A06302	高齢夫婦のみの世帯の割合 (%)
Ktan	A06304	高齢単身世帯の割合 (%)
Konin	A06601	婚姻率（人口千人当たり）
Rikon	A06602	離婚率（人口千人当たり）

これは後述の `X2000data.txt` から 14 変数だけを取り出したもの．「主成分分析」によって次元縮小して，変数間の関連を解釈する．

```
> dat <- read.table("gakureki-rikon-12.txt") # データの読み込み
> dim(dat) # 行列の次元
[1] 47 12
> pairs(dat,pch=".") # ペアごとの散布図
> f <- princomp(dat,cor=T) # 主成分分析
> biplot(f) # バイプロット
```

- 第１主成分 都市型 vs 農村型？

  ＋離婚，＋核家族，＋単独，＋結婚，＋学歴，＋自然増加

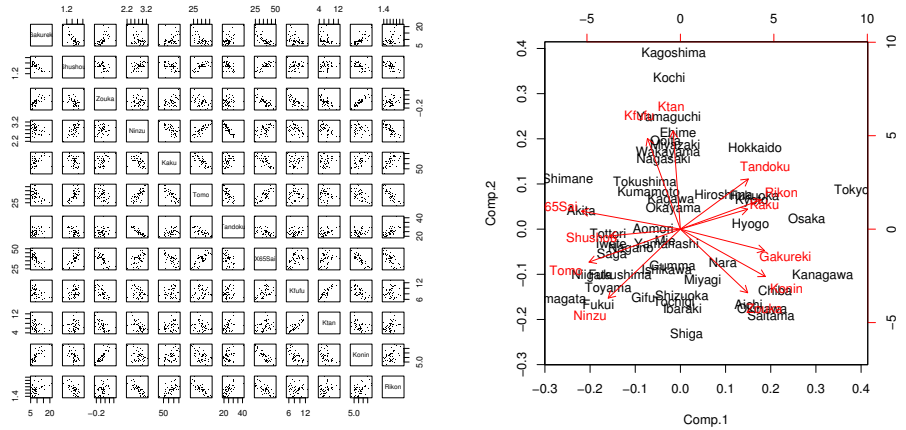


図 2　（左）ペアごとの散布図では全体の関連がつかめない，（右）主成分分析の結果（バイプロット）

− 65 歳以上, −共働き, −人数, −出生

- **第２主成分 高齢化?**

  ＋高齢単身, ＋高齢夫婦, −人数, −自然増加, −結婚

- **第３主成分 核家族?**

  ＋核家族, −単独

- **たとえば上の第 1 主成分をみると，「高齢者世帯，人数の多い世帯，共働き世帯の多い東北地方などでは出生率が高い傾向がある」，「離婚，結婚，核家族，単独世帯，高学歴者の多い東京，大阪，愛知，神奈川などでは出生率が低い傾向がある」ことがわかる．ただし，自然増加の軸を見ると出生率とはむしろ逆向きの傾向があることが分かる．**

```
> ## 単回帰分析をする関数を用意する：変数名を x と y で指定．
> myregplot <- function(x,y,dat) {
+   e <- formula(paste(y,"~",x)) # モデル式を y~x の形式にする
+   plot(e,dat,type="n") # プロットの枠を準備
+   text(dat[,x],dat[,y],rownames(dat)) # ラベルでプロット
+   f <- lm(e,dat) # 回帰分析の実行
+   abline(f,col="red",lty=2) # 回帰直線
+   title(sub=paste(names(f$coef),round(f$coef,4),sep="=",collapse=", "))
+   summary(f)$coef # サマリーの係数部分だけ出力
+ }
> myregplot("Tomo","Shushou",dat)
             Estimate  Std. Error    t value     Pr(>|t|)
(Intercept) 1.03086343 0.091150676 11.309444 9.591410e-15
```



```
Tomo        0.01276565 0.002591909  4.925192 1.179622e-05
> myregplot("Zouka","Shushou",dat[-match(c("Okinawa","Tokyo"),rownames(dat)),])
             Estimate Std. Error    t value     Pr(>|t|)
(Intercept)  1.4947258 0.01599394 93.455770 2.638463e-51
Zouka       -0.3117099 0.09310979 -3.347767 1.701308e-03
```

[ニュース記事は注意して読む２]　女性が働く県ほど出生率高い 調査会が報告書 2006 年 10 月 01 日 00 時 54 分

働く女性の割合が高い県ほど出生率が高い――。政府の調査会の報告書でこんな傾向が裏付けられた。女性が生涯に産む子どもの数を示す合計特殊出生率はどの都道府県も低下傾向にあるが、比較的出生率が高く、下げ幅も小さい自治体では、仕事と子育てが両立しやすい環境が整っていた。内閣府は「両立を支援しないと仕事をする女性も減るし子どもも生まれないことを示している」としている。男女共同参画会議の「少子化と男女共同参画に関する専門調査会」（会長＝佐藤博樹・東大教授）がまとめた。出生率、その減少率、働く女性の割合を示す有業率の三つの数値で４７都道府県を７分類した。出生率が比較的高くて減少率も低いうえ女性有業率が高いグループには、山形県、福井県、熊本県など１６県があてはまった。すべて逆のグループは東京都や大阪府、福岡県など大都市中心の１６都道府県だった。双方を「地域の子育て環境」「雇用機会の均等度」など、両立しやすい環境が整っているかどうかの指標で比べると、明らかな差があることがわかった。中でも「適正な労働時間」、３世代同居などの「家族による世代間支援」、正規雇用の男女の偏りなどの「社会の多様性寛容度」の３項目で特に差が大きかった。もともと地方は大都市より家族や地域の支援を得やすく出生率も高い傾向はあるが、出生率と女性の有業率に正の相関関係があることは国際比較でも確認されている。報告書は（１）家族に代わる地

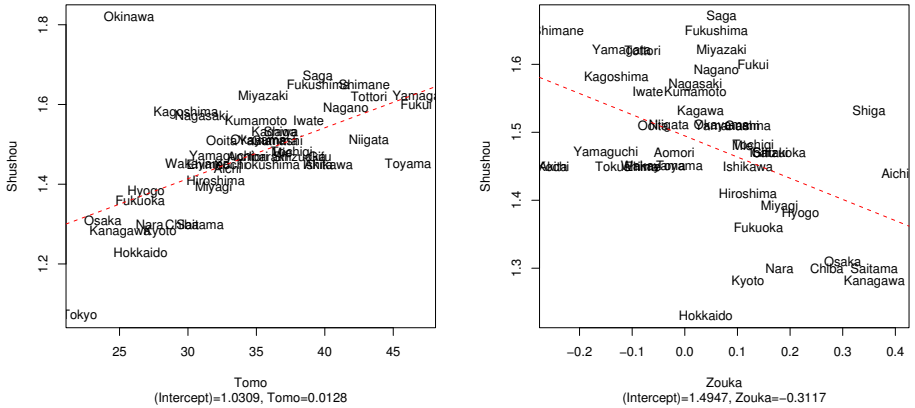


図 3　（左）共働きと出生率の関係，（右）自然増加率と出生率の関係（沖縄と東京を除外）

域の支援体制（２）先進国の中でも際だつ長時間労働（３）非正規化で不安定になっている女性や若者の雇用――への対応が強く求められるとしている。朝日新聞 asahi.com **より引用** http://www.asahi.com/life/update/1001/001.html

[社会・人口統計体系データ]　総務庁統計局統計センターが公開している社会・人口統計体系データ http://www.stat.go.jp/data/ssds/ **では，47 都道府県の様々な調査項目 (2000 年度の 1182 項目) がエクセル形式で公開されているが，これを下平が講義で利用するために** R **で利用できる形式に変換したものが** X2000data.txt**．サイズ** 47×1182 **の実数行列．たとえば学歴と出生率をとりだしてファイルに書き込むには次のようにする．**

```
> X2000.data <- read.table("X2000data.txt") # X2000 データセットのよみこみ
> dim(X2000.data) # 行列の次元
[1]   47 1182
> dat <- X2000.data[,c("E09504","A05203")] # 変数のＩＤ番号
> names(dat) <- c("Gakureki","Shushou") # 分かりやすい名前をつけておく
> write.table(dat,"test.txt") # 表の書き出し
```

なお変数の意味など参考情報は X2000item.txt, X2000code.txt, X2000name.txt **にある．**

[日本語の取り扱い]　ウェブにおいてある txt **ファイルは** Windows **用 (**shift-jis **コード**, CR/LF **改行) としてある．**X2000data.txt **では変数名等すべて半角英数字なので関係ないが，他の** X2000item.txt **等のファイルでは読み込み時に工夫が必要になる場合がある．**Linux **や** Mac **などで使う場合，**nkf **等のコマンドであらかじめファイル形式を変換しておくのはひとつの解決法である．もしくは** R **でファイルを読み込むときに文字コードを直接指定するには，次のようにすればよい．**

```
> ## cp932 はシフト JIS のこと．enconding="shift-jis"でもほとんど同じ
> X2000.item <- read.table(file("X2000item.txt",encoding="cp932"))
> dim(X2000.item) # 表の次元
[1] 1182    4
> X2000.item[1,] # 最初の 1 行
                                  Imi Tani Zenkoku                  Bunrui
A01101 全国総人口に占める人口割合 (%)      100 A. 人口・世帯 1) 人口分布
> X2000.item[c("E09504","A05203"),] # Gakureki と Shushou の行
                                  Imi Tani Zenkoku                  Bunrui
E09504 最終学歴が大学・大学院卒の者の割合 (%)    11.90      E. 教育 7) 教育普及度
A05203                        合計特殊出生率          1.36 A. 人口・世帯 5) 人口動態
```

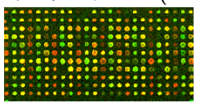
[卒論紹介 (II)]

2003 年度学士論文

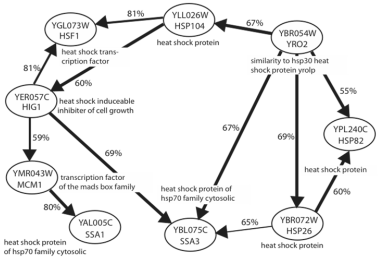
DNA マイクロアレイデータに基づく遺伝子ネットワーク推定

情報科学科 下平研究室 上村健

- **マイクロアレイ** (DNA **チップ**) **で遺伝子の発現レベルを測定する**



- **遺伝子ネットワーク**



- **遺伝子機能の解明，薬剤開発**

[階層的クラスタリング]

- **階層的クラスタリング分析を前節の** 12 **変量データセットに適用し，要素（都道府県）の関係や，変数間の関係を調べる．**
- **要素間の「距離」行列を生成する関数** `dist()`
- **階層的クラスタリングを実行する関数** `hclust()`
- **結果の表示** `plot()`

```
> dat <- read.table("gakureki-rikon-12.txt") # データの読み込み
> x <- scale(dat) # 「標準化」（各変数を平均０，分散１にする）
```



```
> h1 <- hclust(dist(x)) # クラスタ分析
> plot(h1) # プロット
> h2 <- hclust(dist(t(x))) # データ行列を転置してからクラスタ分析
> plot(h2) # プロット
```

[卒論紹介 (III)]

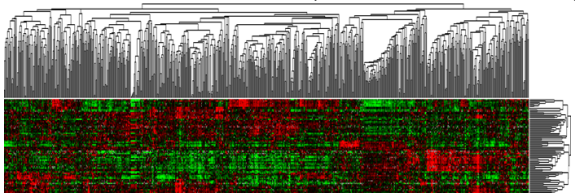
2004 年度修士論文

Assessing the uncertainty in hierarchical cluster analysis via multiscale bootstrap resampling

(階層的クラスタリングの信頼性評価をマルチスケール・ブートストラップ法で行う)

数理計算科学専攻 下平研究室 鈴木了太

- **肺腫瘍のマイクロアレイデータ** (p=73 **個体，** n=916 **遺伝子**)



- **７３腫瘍のクラスタリング** (pvclust R **の公式ライブラリに登録済み**)

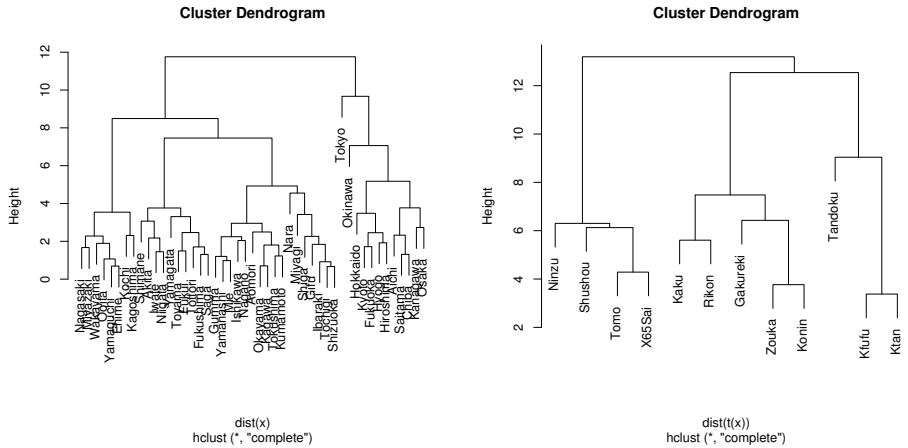
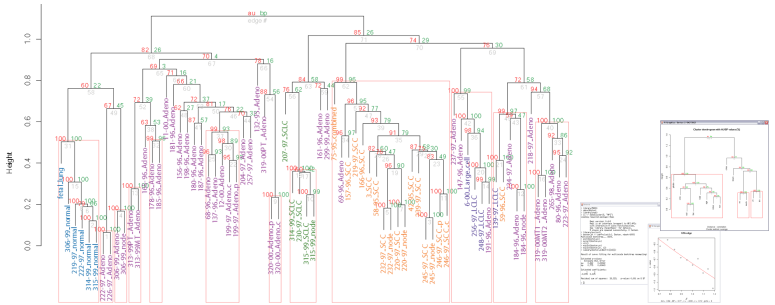


図 4 **（左）県のクラスタリング，（右）変量のクラスタリング**



1　確率論

サブセクション： 期待値，大数の法則，モンテカルロ法，ベイズの定理，積率母関数，中心極限定理

キーワード： 確率変数，分散，ポートフォリオ，分布関数，密度関数，確率収束，経験分布関数，パーセント点，逆関数法（による乱数生成），棄却法，一様分布，正規分布，多変量正規分布，コレスキー分解，マルコフ連鎖，定常分布，マルコフチェイン・モンテカルロ（ＭＣＭＣ），メトロポリス・ヘイスティング，ギブスサンプラー，事前分布，事後分布，画像復元，特性関数，フーリエ変換，法則収束，ポアソン分布，少数の法則

1.1　期待値

実数値の確率変数 X，その実現値 x，確率密度関数 $f(x)$，適当な関数 $g(x)$ とする．

[定義 1.1]　**確率変数 $Y = g(X)$ の期待値（平均ともいう）は**

$$E\left(g(X)\right) = \int_{-\infty}^{\infty} g(x) f(x)\, dx \tag{1.1}$$

[課題 1.1]　**定数 $a, b \in \mathbb{R}$, 関数 $g_1(x), g_2(x)$ に対して以下の線形性を示せ．**

$$E(ag_1(X) + bg_2(X)) = aE(g_1(X)) + bE(g_2(X))$$

[課題 1.2]　**X の期待値 $E(X)$ を (1.1) で表すための $g(x)$ を求めよ．**

[課題 1.3]　**X の分散 $V(X)$ を (1.1) で表すための $g(x)$ を求めよ．**

[課題 1.4]　**区間 (a, b), $a < b$ に X が入る確率 $P(a < X < b)$ を (1.1) で表すための $g(x)$ を求めよ．ヒント：命題 A が真のとき $I(A) = 1$，偽のとき $I(A) = 0$ となる「指示関数」(indicator function) をつかう．**

[課題 1.5]　**分布関数（累積分布関数ともいう）$F(x) = \int_{-\infty}^{x} f(s)\, ds$ を (1.1) で表すために x を定数とみなして $g(s)$ を求めよ．**

[例 1.1]　**区間 $(0, 1)$ の一様分布 $X \sim U(0, 1)$ の密度関数は $f(x) = 1$, $0 < x < 1$, それ以外で $f(x) = 0$ である．X の期待値，分散，分布関数は**

$$E(X) = \int_{-\infty}^{\infty} x f(x)\, dx = \int_{0}^{1} x\, dx = \tfrac{1}{2}$$

$$V(X) = \int_{-\infty}^{\infty} (x - E(X))^2 f(x)\, dx = \int_{0}^{1} (x - \tfrac{1}{2})^2\, dx = \tfrac{1}{12}$$

$$F(x) = \int_{0}^{x} ds = x, 0 < x < 1; \quad F(x) = 0, x \le 0; \quad F(x) = 1, x \ge 1$$

[定義 1.2]　**m 次元ベクトルの確率変数 $\boldsymbol{X}$，その実現値 $\boldsymbol{x}$，密度関数 $f(\boldsymbol{x})$，適当な関数 $g(\boldsymbol{x})$ に対して**

$$E(g(\boldsymbol{X})) = \int_{\mathbb{R}^m} g(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x}$$

と書く．$\boldsymbol{X}$ の期待値は $E(\boldsymbol{X})$，分散（分散共分散行列）は $V(\boldsymbol{X})$ と書く．成分で書けば，

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_m \end{bmatrix}, \quad E(\boldsymbol{X}) = \begin{bmatrix} E(X_1) \\ \vdots \\ E(X_m) \end{bmatrix}, \quad V(\boldsymbol{X}) = \begin{bmatrix} C(X_1, X_1) & \cdots & C(X_1, X_m) \\ \vdots & & \vdots \\ C(X_m, X_1) & \cdots & C(X_m, X_m) \end{bmatrix}$$

ただし共分散 $C(X_i, X_j) = E\left[(X_i - E(X_i))(X_j - E(X_j))\right]$ である．X_i と X_j の相関係数は $\rho(X_i, X_j) = C(X_i, X_j)/\sqrt{C(X_i, X_i)C(X_j, X_j)}$ である．X_i の標準偏差は分散の平方根 $\sqrt{V(X_i)}$ である．

[課題 1.6]　**$\boldsymbol{X}$ は m 次元確率ベクトル，$\boldsymbol{w} \in \mathbb{R}^m$ は定数ベクトルとする．このとき，確率変数 $Y = \boldsymbol{w}'\boldsymbol{X}$ について以下を示せ．ただし，行列 A にたいして A' はその転置行列を表す．**

$$E(Y) = \boldsymbol{w}' E(\boldsymbol{X}), \quad V(Y) = \boldsymbol{w}' V(\boldsymbol{X}) \boldsymbol{w}$$

[例 1.2]　**$\boldsymbol{X}$ は m 次元確率ベクトルで，各成分の平均と分散は $E(X_i) = \mu$，$V(X_i) = \sigma^2$，相関係数はすべ**

て 0 とする．$Y = \sum_{i=1}^{m} X_i/m$ の平均と分散は，$\boldsymbol{w} = (\frac{1}{m}, \ldots, \frac{1}{m})'$ と置くことにより，次式で与えられる．

$$E(Y) = \mu, \quad V(Y) = \frac{\sigma^2}{m}$$

```
> sx <- 0.2 # SD(X)
> m <- 1:10 # m=1,2,...,10
> sy <- sx/sqrt(m) # SD(Y)
> plot(m,sy)
```

[課題 1.7]　$\boldsymbol{X}$ は m 次元確率ベクトル，$\boldsymbol{w} \in \mathbb{R}^m$ は定数ベクトルとする．とくに X_i は i 番目の資産の収益率とすれば，$Y = \boldsymbol{w}'\boldsymbol{X}$ は重み $\boldsymbol{w}$ で構築したポートフォリオの収益率となる．期待収益率 $E(Y)$ があらかじめ定めた値 μ に等しく，重みの和が 1 という条件

$$E(Y) = \mu, \quad \sum_{i=1}^{m} w_i = 1$$

のときに収益率の分散 $V(Y)$ を最小にする重みベクトル（ただし負の成分を許す）が以下で与えられることを示せ．

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}\boldsymbol{A}\left(\boldsymbol{A}'\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\right)^{-1}\begin{bmatrix}\mu \\ 1\end{bmatrix}$$

ただし $\boldsymbol{\Sigma} = V(\boldsymbol{X})$ は正定値行列，$\mathbf{1}_m = (1, \ldots, 1)'$ は成分がすべて 1 で長さ m のベクトル，$\boldsymbol{A} = (E(\boldsymbol{X}), \mathbf{1}_m)$ はランク 2 の $m \times 2$ 行列である．

[例 1.3]　m 個の資産の期待収益率 $E(X_i)$，標準偏差 $\sqrt{V(X_i)}$ を次のように与える．X_i と X_j の相関係数 ρ はどの組み合わせでも同じ値とする．

```
> ex <- c(0.2,0.15,0.1,0.05) # E(X)
> sx <- c(0.2,0.2,0.1,0.05) # SD(X)
> rho <- 0.3
```

このとき $\boldsymbol{\Sigma}$，$\boldsymbol{\Sigma}^{-1}$，$\boldsymbol{A}$，および，$\boldsymbol{C} = \boldsymbol{\Sigma}^{-1}\boldsymbol{A}\left(\boldsymbol{A}'\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\right)^{-1}$ を計算しておく．

```
> m <- length(sx) # ベクトルの長さ = 4
> V <- (sx %o% sx) * (diag(m)*(1-rho) + matrix(rho,m,m)) # Sigma
> B <- solve(V) # Sigma^(-1)
> A <- cbind(ex,1) # A 行列
> C <- B %*% A %*% solve(t(A) %*% B %*% A) # C 行列
```

ポートフォリオの期待収益率が $E(Y) = 0.15$ のときに $V(Y)$ を最小にする重みと，そのときの $\sqrt{V(Y)}$ は

```
> w <- C %*% c(0.15,1) # 最適な重み
> t(w) # 横にして表示
          [,1]      [,2]      [,3]       [,4]
[1,] 0.3850932 0.1705686 0.5035834 -0.0592451
> t(w) %*% ex # 0.15 になるはず
     [,1]
[1,] 0.15
> sqrt(t(w) %*% V %*% w) # SD(Y)
          [,1]
[1,] 0.1195309
```


$E(Y)$ を縦軸，$\sqrt{V(Y)}$ を横軸にしてプロットする．ポートフォリオの「平均-標準偏差ダイアグラム」と呼ばれる．

```
> # まず E(Y) から SD(Y) を計算する関数を準備
> mysy <- function(ey) {
+   w <- C %*% c(ey,1)
+   sqrt(t(w) %*% V %*% w)
+ }
> ey <- seq(-0.1,0.4,length=100) # E(Y) を -0.1 から 0.4 で 100 等分する．
> sy <- sapply(ey,mysy) # SD(Y) の計算
> plot(sy,ey,type="l")
> points(sx,ex,col="red")
```

[課題 1.8]　任意の x で $h(x) \geq 0$ とする．$E(h(X))$ が存在するとき，任意の $a > 0$ に対して以下の性質（マルコフの不等式）を示せ．

$$E(h(X)) \geq aP(h(X) \geq a)$$

この結果を用いて，任意の $\epsilon > 0$ に対して以下の性質（チェビシェフの不等式）を示せ．ただし $E(X) = \mu$ と $V(X) = \sigma^2$ の存在を仮定する．

$$P(|X - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$$

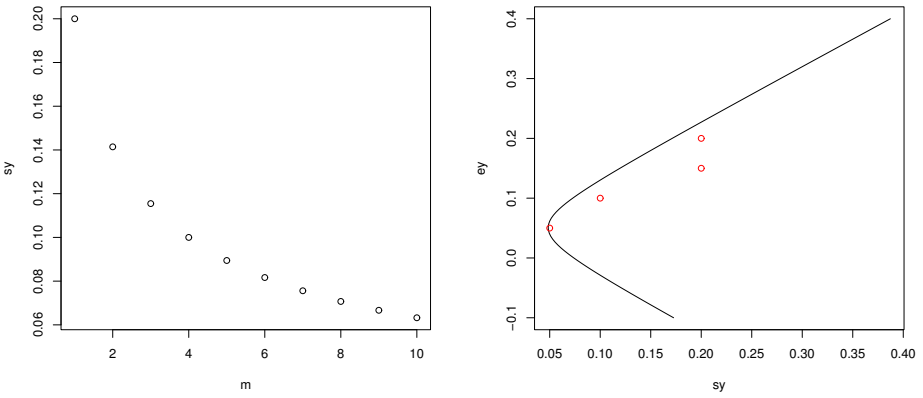


図 5　（左）独立な場合，（右）平均-標準偏差ダイアグラム

[注意]　定義 1.1は連続分布を想定して密度関数 $f(x)$ が存在することを仮定している．X のとりうる値が $s_0, s_1, \ldots$ の離散分布に対しては確率関数 $p(x)$ を使って

$$E\left(g(X)\right) = \sum_{i=0}^{\infty} g(s_i)p(s_i) \tag{1.2}$$

とかける．ディラックのデルタ関数 $\delta(x)$ を使えば，形式的に

$$f(x) = \sum_{i=0}^{\infty} p(s_i)\delta(s_i)$$

とおくことにより (1.1) に帰着する．この場合をふくめて議論するために，一般に f を確率分布（＝確率測度）とする．集合 $A \subset \mathbb{R}$ にたいして $f(A) = P(X \in A)$ と書き，期待値はルベーグ積分を用いて次のように書く．

$$E(g(X)) = \int_{\mathbb{R}} g(x)f(dx)$$

本資料では上式を形式的に (1.1) と書く．

1.2　大数の法則

確率変数の系列 $Y_1, Y_2, \ldots$ を考える．これを $Y_n,\ n = 1, 2, \ldots$ とかく．

[定義 1.3]　確率変数列 Y_n が確率変数 Y に確率収束 (convergence in probability) するとは，任意の $\epsilon > 0$ に対して

$$\lim_{n\to\infty} P\left(|Y_n - Y| > \epsilon\right) = 0 \tag{1.3}$$

となることである． これを $Y_n \xrightarrow{p} Y$ とかく．(Y が定数でもよいことに注意)．$Z_n = Y_n - Y$ とおけば要するに

$$\forall\epsilon > 0;\ \lim_{n\to\infty} P(|Z_n| > \epsilon) = 0$$

[注意]　いろいろな収束の定義があり，それらは互いに意味が異なる．

- 確率収束より強い意味での収束に，概収束（がいしゅうそく, almost surely convergent）がある．

$$P\left(\lim_{n\to\infty} Y_n(\omega) = Y(\omega)\right) = 1$$

- 確率収束より弱い意味での収束に，法則収束または分布収束 (convergence in distribution) がある．

$$\forall y;\ \lim_{n\to\infty} P(Y_n \le y) = P(Y \le y)$$

- 一般に

$$\text{概収束} \Rightarrow \text{確率収束} \Rightarrow \text{法則収束}$$

- Levy (1937) の定理：$X_k,\ k = 1, 2, \ldots$ が互いに独立ならば，$Y_n = \sum_{k=1}^{n} X_k,\ n = 1, 2, \ldots$ の概収束，確率収束，法則収束は同等．

[定理 1.1]　X_n は独立に同一の分布にしたがう確率変数列とし，$E(X_1) = \mu$ の存在を仮定する．最初の n 個の平均値を

$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

とかく．$\bar{X}_n$ も確率変数列であることに注意する．このとき，$n \to \infty$ の極限で

$$\bar{X}_n \xrightarrow{p} \mu \tag{1.4}$$

が成り立つ．これを大数の法則という．より強い性質「大数の強法則」もいえるが，ここでは議論しない．いずれにても，「標本数 (sample size) を増やせば標本平均はいずれ期待値に収束する」と解釈できる．

[証明]　ここでは簡単のために分散 $\sigma^2 = V(X_1)$ の存在を仮定する．$V(\bar{X}_n) = \sigma^2/n$ であるから，チェビシェフの不等式より

$$P\left(|\bar{X}_n - \mu| \ge \epsilon\right) \le \frac{\sigma^2}{n\epsilon^2}$$

となる．$n \to \infty$ で右辺は 0 に収束する．

[課題 1.9]　X_n は独立に同一の分布にしたがう確率変数列とし，$E(g(X_1))$ の存在を仮定する．

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i) \tag{1.5}$$

とおく．以下を示せ．

$$E(\bar{Y}_n) = E(g(X_1)),\quad \bar{Y}_n \xrightarrow{p} E(g(X_1))$$

したがって，n を十分に大きく取ることにより，(1.1) は (1.5) によって近似できる．また，$V(g(X_1))$ の存在を仮定するとき $V(\bar{Y}_n) = V(g(X_1))/n$ であることを示せ．

[例 1.4]　確率変数列 X_n が独立に区間 $(0, 1)$ の一様分布 $X_n \sim U(0, 1)$ に従うことを擬似乱数を使いシミュレーションする．$X_1, \ldots, X_n$ の実現値 (標本) を $x_1, \ldots, x_n$ とする．

```
> ## 一様分布
> x <- runif(10000) # U(0,1) を 10000 個つくる
> x[1:5] # 最初の 5 個
[1] 0.30776611 0.25767250 0.55232243 0.05638315 0.46854928
> ## 平均値の収束を確かめる
> mean(x[1:10]) # 最初の 10 個の平均
[1] 0.4026008
> mean(x[1:100]) # 最初の 100 個の平均
[1] 0.51987
> mean(x[1:1000]) # 最初の 1000 個の平均
[1] 0.5180817
```

```
> mean(x) # 10000 個の標本平均 (0.5)
[1] 0.5002956
> y1 <- cumsum(x)/(1:10000) # 最初の n 個の平均
> plot(y1,log="x"); abline(h=0.5)
> ## 分散も「期待値」だから，やっぱり収束
> mean((x - mean(x))^2) # 標本分散 (1/12 = 0.833)
[1] 0.08273506
> ## x が 0.1 より小さくなる確率も「期待値」だから，やっぱり収束
> mean(x < 0.1) # P(X<0.1)=0.1
[1] 0.0941
> z <- x < 0.1 # 0 か 1 とみなしてよい
> as.numeric(z[1:100]) # 最初の 100 をみてみる
  [1] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [75] 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
> cumsum(z[1:100]) # さいしょから順番に足していく
  [1] 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [75] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 4 4
> y2 <- cumsum(z)/(1:10000) # 最初の n 個の平均
> plot(y2,log="x"); abline(h=0.1)
```

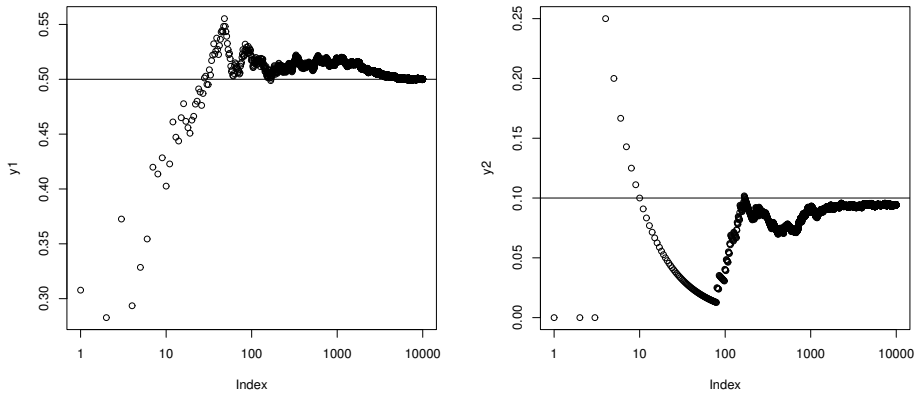



図 6　（左）$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n X_i$，（右）$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^n I(X_i < 0.1)$

[定義 1.4]　標本 $x_1, \ldots, x_n$ の経験（累積）分布関数は

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^n I(x_i \le x) = \frac{\#(x_i \le x)}{n} \tag{1.6}$$

である．対応する確率密度関数は

$$\hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^n \delta(x - x_i) \tag{1.7}$$

である．各 x_i の確率が $1/n$ の離散分布とみなせる．

[課題 1.10]　**x を固定して，$\hat{F}_n(x)$ を $X_1, \ldots, X_n$ の関数とみなす．このとき $\hat{F}_n(x) \xrightarrow{p} F(x)$ を示せ．また，$n\hat{F}_n(x)$ が 2 項分布に従うことを示し，$\hat{F}_n(x)$ の平均，分散を求めよ．**

[注意]　**x の各点で $\hat{F}_n(x) \xrightarrow{p} F(x)$ となるだけでなく，より強く一様収束をのべる以下の結果（Glivenko-Cantelli の定理）がいえる．**

$$P\left\{\lim_{n\to\infty}\sup_{x\in\mathbb{R}}|\hat{F}_n(x) - F(x)| = 0\right\} = 1$$

いずれにしても，「標本数を増やせば経験分布関数は分布関数に収束する」と解釈できる．大数の法則では平均値が期待値への収束することを述べていたが，期待値は分布のひとつの性質に過ぎず，サンプルが持ってる情報は分布そのものを再現できることを示している．（もっとも，大数の法則で $g(X)$ を自由に選べることか

ら分布の収束がいえる）．

[定義 1.5]　分布関数 $p = F(q)$ の逆関数を $q = F^{-1}(p) = \inf_{F(q)\ge p} q$ と書く (inf は F の不連続点を考慮するため)．q は分布 F の（下側）$100p$ パーセント点という．$F^{-1}(1-p)$ は上側 $100p$ パーセント点という．経験分布を用いて $\hat{F}_n^{-1}(p)$ は（下側）$100p$ パーセント標本点という．$p = 0.5$ ときが中央値 (median) である．`help(quantile)` を実行すれば R における標本パーセント点の実装が読める．

[例 1.5]　**例題 1.4で作成した $x_1, \ldots, x_n$ を再利用する．**

```
> hist(x,prob=TRUE) # ヒストグラム（確率密度表示）
> y <- sort(x) # 小さい順に並べ替え
> y[5000] # 経験分布の逆関数 (p=0.5)
[1] 0.4969066
> (y[5000]+y[5001])/2 # 線形補間したもの (p=0.5)
[1] 0.4969936
> quantile(x,p=0.5) # median(x) でも同じ
      50%
0.4969936
> p <- (1:10000)/10000 # 0.0001, 0.0002, ..., 0.9999, 1
> plot(y,p,pch=".") # 経験分布関数
```

[例 1.6]　**平均 0，分散 1 の正規分布（これを標準正規分布という）$X \sim N(0,1)$ の密度関数は $f(x) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})$ である．`rnorm` を利用して $X_1, \ldots, X_{10000} \sim N(0,1)$ を生成し，標本平均，標本分散，標本中央値，標本上側 5% 点，標本上側 2.5% 点を計算する．それらを理論値（$\hat{F}_n$ のかわりに F から得られる値）と比較する．**

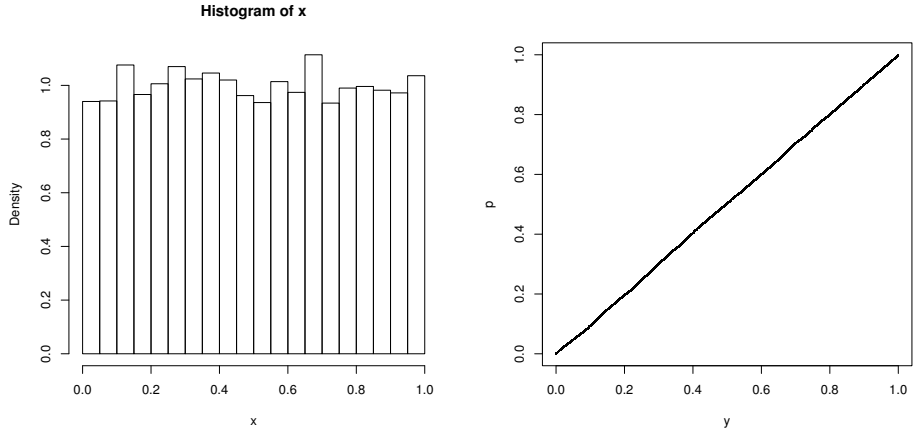


図 7　（左）ヒストグラム，（右）経験分布関数


```
> x <- rnorm(10000) # N(0,1) から 10000 個生成
> mean(x) # 標本平均（理論値は０）
[1] -0.009716994
> mean((x-mean(x))^2) # 標本分散（理論値は１）
[1] 0.9990848
> quantile(x,p=c(0.5,0.95,0.975)) # 中央値，上側 5%，上側 2.5%
        50%         95%       97.5%
-0.00939972  1.62948079  1.95144043
> qnorm(c(0.5,0.95,0.975)) # 理論値
[1] 0.000000 1.644854 1.959964
> ### 以下は参考
> library(MASS) # MASS ライブラリのロード
> truehist(x) # ヒストグラム（確率密度表示）
> x0 <- seq(min(x),max(x),length=10000) # x のレンジを 1000 分割
> lines(x0,dnorm(x0),col=2) # 正規分布の密度関数（赤色）
> p <- (1:10000)/10000 # 0.0001, 0.0002, ..., 0.9999, 1
> plot(sort(x),p,pch=".") # 経験分布関数
> lines(x0,pnorm(x0),col=2) # 正規分布の累積分布関数（赤色）
```

[例 1.7]　`library(MASS)` に含まれる `truehist` 関数では，ヒストグラム区間数を `nbis` で指定する．デフォルトでは Scott (1979) の方法を用いている．

```
> truehist(x) # デフォルトは nbins="Scott"
> nclass.scott # nbins を計算する関数
function (x)
{
    h <- 3.5 * sqrt(stats::var(x)) * length(x)^(-1/3)
    ceiling(diff(range(x))/h)
}
```

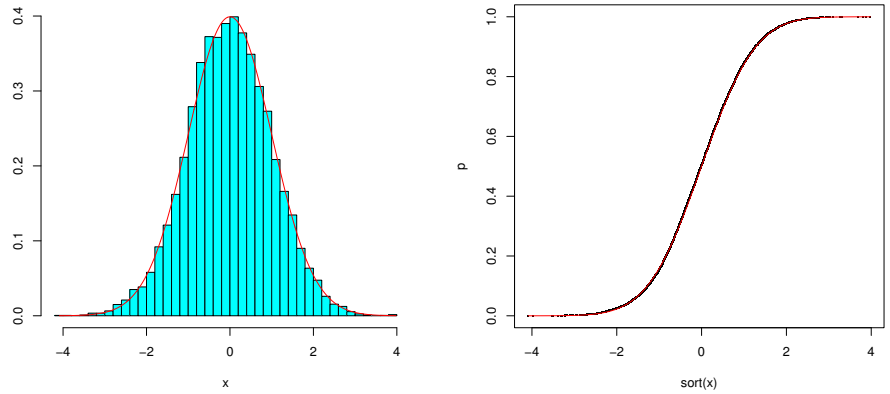



図 8　（左）ヒストグラム，（右）経験分布関数


```
<environment: namespace:grDevices>
> truehist(x,nbins=nclass.scott(x)*2) # デフォルトの 2 倍
> truehist(x,nbins=nclass.scott(x)/2) # デフォルトの半分
```

[課題 1.11]　区間 $(a,b]$, $a<b$ に $X_1,\ldots,X_n$ が入る回数を C で表す．X の累積分布関数 $F(x)$ を用いて $Y=\frac{C}{n(b-a)}$ の平均 μ，分散 σ^2 を求めよ．

[課題 1.12]　密度関数 $f(x)$ が十分になめらかで，とくに 2 回微分可能としておく．ヒストグラムの最適な区間幅 h を定めるために，次の手順に従って考察せよ．(i) $x\in(a,b)$ において $f(x)$ を上記課題の Y で近似するとき，そのバイアスと分散が次式で近似できることを示せ．ただし $b-a-h$ とおく．

$$E(Y)-f(x)=f'(a)\left(\frac{h}{2}+(x-a)\right)+O(h^2),\quad V(Y)=\frac{f(a)}{nh}+O(n^{-1})$$

したがって x における平均 2 乗誤差が

$$\mathrm{MSE}(x)=V(Y)+(E(Y)-f(x))^2=f'(a)^2\left(x-\frac{a+b}{2}\right)^2+\frac{f(a)}{nh}+O\left(h^3+n^{-1}\right)$$

(ii) これを (a,b) で平均すると，

$$\frac{1}{b-a}\int_a^b \mathrm{MSE}(x)\,dx\approx f'(a)^2\frac{h^2}{12}+\frac{f(a)}{nh}$$

(iii) これを全区間で足し合わせるとヒストグラムの平均 2 乗誤差は次式で表される．

$$\int_{-\infty}^{\infty}\mathrm{MSE}(x)\,dx\approx\int_{-\infty}^{\infty}\left[f'(x)^2\frac{h^2}{12}+\frac{f(x)}{nh}\right]dx=\frac{h^2}{12}\int_{-\infty}^{\infty}f'(x)^2\,dx+\frac{1}{nh}$$

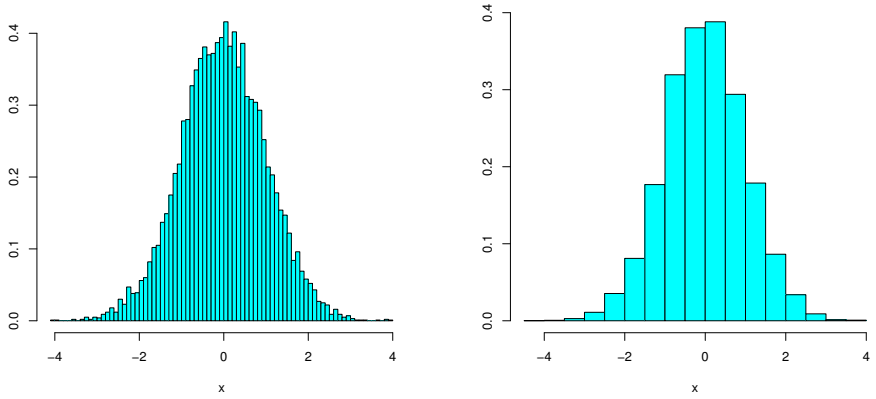


図 9　（左）デフォルトの 2 倍の区間数，（右）半分の区間数

(iv) これを最小にする h は次式で与えられる．

$$h = \left[\frac{n}{6}\int_{-\infty}^{\infty} f'(x)^2\,dx\right]^{-1/3}$$

(v) とくに $f(x)$ として，分散 σ^2 の正規分布を想定すると，

$$\int_{-\infty}^{\infty} f'(x)^2\,dx = \frac{1}{4\sqrt{\pi}\sigma^3}$$

したがって，$h = (24\sqrt{\pi})^{1/3}\sigma n^{-1/3} \approx 3.49\sigma n^{-1/3}$ となる．これが Scott (1979) の方法．

1.3　モンテカルロ法

区間 $(0,1)$ の一様分布に従う独立な確率変数列 $U_1, U_2, \ldots \sim U(0,1)$ が利用できると仮定する．コンピュータではメルセンヌツイスター法などのアルゴリズムにより擬似乱数系列が生成できるから，これを利用すれば実質的に問題ない．このとき，ある指定した分布 $F(x)$（密度関数 $f(x)$）に従う $X_1, X_2, \ldots$ をどうやって得るかを考える．目的は (1.1) を (1.5) によって近似すること．

[課題 1.13]　$X = F^{-1}(U)$ とおけば，X の分布関数が $F(x)$ であることを示せ．これを逆関数法という．

[例 1.8]　指数分布

$$f(x) = e^{-x}, \quad F(x) = 1 - e^{-x}$$

に従う乱数を生成する．$F^{-1}(p) = -\log(1-p)$ である．


```
> ## X ~ exp(-x) を生成する
> ## あくまで方法を説明する例題．本当は rexp を使ったほうがラク．
> u <- runif(10000) # U(0,1) を 10000 個生成
> x <- -log(u) # 各要素に X = F^{-1}(U) を適用
> truehist(x) # ヒストグラム（確率密度表示）
> x0 <- seq(min(x),max(x),length=1000)
> lines(x0,exp(-x0),col=2) # 密度関数（赤）
```

[課題 1.14]　$f(x)$, $g(x)$ を連続分布の密度関数とする．$U \sim U(0,1)$ と $V \sim g(v)$ にしたがう乱数を利用して，$X \sim f(x)$ を生成したい．次のアルゴリズムでこれが実現できることを示せ．

1. すべての x で $f(x) \le cg(x)$ となるような c を選ぶ．
2. $U \sim U(0,1)$ と $V \sim g(v)$ をそれぞれ 1 個生成．
3. $U \le \frac{f(V)}{cg(V)}$ なら $X = V$ として終了．そうでなければ 2 へ戻る．

これを棄却法 (rejection method) という．

[例 1.9]　$X \sim N(0,1)$ を指数分布から生成する．

$$f(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}, \quad g(x) = \frac{1}{2}e^{-|x|}, \quad c = \sqrt{\frac{2e}{\pi}}$$

とおくと $f(x) \le cg(x)$ をみたす．$f(x)/(cg(x)) = e^{-\frac{(|x|-1)^2}{2}}$ に注意すると，次のように計算できる．説明をカンタンにするため，条件をみたす X をまとめて出力する．


```
> ## X ~ N(0,1) を生成する
> ## あくまで方法を説明する例題．本当は rnorm を使ったほうがラク．
> u0 <- 2*runif(10000)-1 # U(-1,1) を 10000 個生成
> u <- abs(u0) # U(0,1)
> s <- sign(u0) # +1,-1 を確率で取る
> v <- -log(runif(10000)) # v ~ exp(-v) 指数分布
> a <- u <= exp(-(v-1)^2/2) # 条件を満たすかどうか？ （a は論理型のベクタ）
> x <- (s*v)[a] # 条件を満たすものだけまとめて取り出す
> length(x) # とりだした個数
[1] 7575
> truehist(x) # ヒストグラム（確率密度表示）
> x0 <- seq(min(x),max(x),length=10000) # x のレンジを 1000 分割
> lines(x0,dnorm(x0),col=2) # 正規分布の密度関数（赤色）
```

[課題 1.15]　多変量正規分布に従う m 次元確率ベクトル $\boldsymbol{X} \sim N_m(\boldsymbol{\mu}, \Sigma)$ の生成を考える．密度関数は

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = (2\pi)^{-m/2}|\Sigma|^{-1/2}\exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big)$$

である．Σ は非負定値対称行列なので，$m \times m$ 行列 $\boldsymbol{A}$ で $\Sigma = \boldsymbol{A}\boldsymbol{A}'$ となるものがある．（ここで $\boldsymbol{A}'$ は行列 $\boldsymbol{A}$ の転置を表す）．$\boldsymbol{A}$ には任意性があるが，たとえばコレスキー分解（ガウス消去法の一種）を用いて下三角行列となる $\boldsymbol{A}$ が計算できる．このとき，m 次元確率ベクトル $Z \sim N_m(\boldsymbol{0}, \boldsymbol{I}_m)$ を使って，$\boldsymbol{X} = \boldsymbol{A}Z + \boldsymbol{\mu}$ とすればよいことを示せ．ヒント：「多変量正規分布に従う確率ベクトルを線形変換したものは多変量正規分布

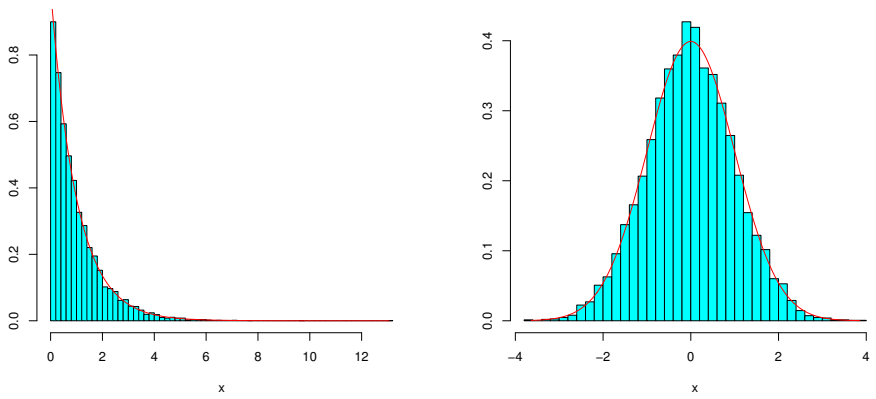


図 10　（左）逆関数法（指数分布），（右）棄却法（正規分布）

に従う」という定理を利用してよい．

[**例** 1.10]　**２次元正規分布** $\boldsymbol{X} \sim N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ **を生成する．パラメタを**

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

ただし $\mu_1 = 5, \mu_2 = 0, \rho = 0.5$ **とおく．**

```
> S <- matrix(c(1,0.5,0.5,1),2,2) # 行列 Sigma の定義
> S # 表示
     [,1] [,2]
[1,]  1.0  0.5
[2,]  0.5  1.0
> A <- t(chol(S)) # コレスキー分解（下三角行列）
> A # 表示
     [,1]      [,2]
[1,]  1.0 0.0000000
[2,]  0.5 0.8660254
> z <- matrix(rnorm(2*1000),2,1000) # 各成分 N(0,1) の 2x1000 行列
> var(t(z)) # z の分散共分散行列
            [,1]        [,2]
[1,]  0.9203834 -0.0512616
[2,] -0.0512616  0.9602724
> x <- A %*% z + c(5,0) # 「z の各列に線形変換を適用」を 1000 回まとめて実行
> var(t(x)) # x の分散共分散行列
          [,1]      [,2]
[1,] 0.9203834 0.4157979
[2,] 0.4157979 0.9059063
> plot(z[1,],z[2,]) # z のプロット
```



```
> plot(x[1,],x[2,]) # x のプロット
```

[**課題** 1.16]　**系列** $X_1, X_2, \ldots$ **を次のアルゴリズムで生成する．このマルコフ連鎖の推移確率を決める条件付確率分布** $p(x_{t+1}|x_t)$ **を求めよ．その定常分布が** $f(x)$ **となることを示せ．ヒント：詳細釣合** $p(x_{t+1}|x)f(x) = p(x|x_{t+1})f(x_{t+1})$ **を示せばよい．そうすればもし** $X_t \sim f(x_t)$ **なら** X_{t+1} **の従う分布は**

$$\int_{-\infty}^{\infty} p(x_{t+1}|x_t)f(x_t)\,dx_t = \int_{-\infty}^{\infty} p(x_t|x_{t+1})f(x_{t+1})\,dx_t = f(x_{t+1})$$

となって $f(x)$ **が定常分布であることが分かる．**

1. **条件付密度関数** $q(v|x)$ **を決める．初期値** X_1 **を選び** $t = 1$ **とする．**
2. $U \sim U(0,1)$ **と** $V \sim q(v|X_t)$ **をそれぞれ１個生成．**
3. **次の数を計算．**
$$\alpha(V, X_t) = \min\left\{1, \frac{q(X_t|V)f(V)}{q(V|X_t)f(X_t)}\right\}$$
4. $U \leq \alpha(V, X_t)$ **なら** $X_{t+1} = V$ **とする．そうでないなら** $X_{t+1} = X_t$ **とする．**
5. t **の値をひとつ増やして** 2**へもどる．**

[**注意**]　**ここでは実数** x **で記述したが一般にベクトル** x **でも同様．この方法は** Metrololis-Hastings **アルゴリズムと呼ばれる．**$f(x)$ **の直接計算が困難であるような複雑なアプリケーションで近年多用される（そのよう**

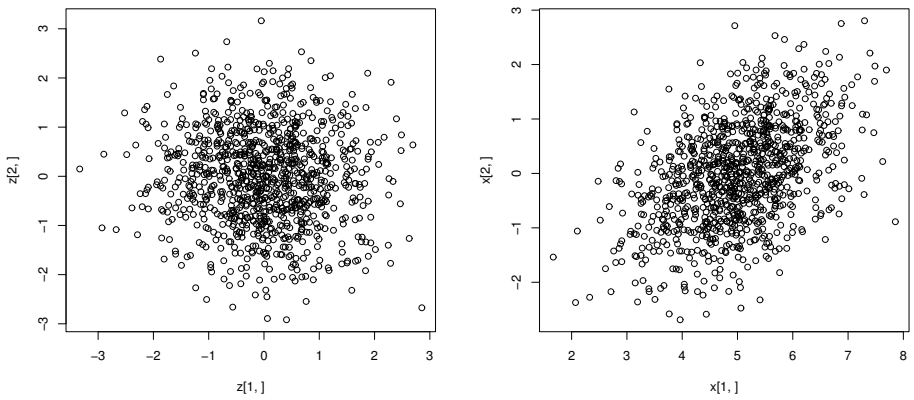


図 11　（左）1000 個の z**，（右）1000 個の** x

な場合でも比 $f(\boldsymbol{v})/f(\boldsymbol{x})$ の計算が容易なことが多い）．マルコフチェイン・モンテカルロ法 (Markov Chain Monte Carlo，略して MCMC) の代表的なアルゴリズムであり，他の方法（メトロポリス法，ギブスサンプラー）を一般化したもの．$q(v|x)$ は提案分布 (proposal distribution) と呼ばれ，ユーザーが工夫して自由に設定する．アルゴリズムの出力する系列が既約なマルコフ過程になれば，(1.1) を (1.5) によって近似できる．適当な初期値 X_0 から初めて十分に大きな N（たとえば $N = 100,000$）回反復して $X_1, X_2, \ldots, X_N$ を生成する．最初のうちは定常分布からはずれたところを推移するので，$M < N$（たとえば $M = 10,000$）個をすてて残りの $X_{M+1}, X_{M+2}, \ldots, X_N$ を使って $\sum_{t=M+1}^{N} X_t/(N-M)$ によって (1.1) を近似する．最初の M 回の反復は “burn-in” (三省堂リーダースプラス：【電算】 バーンイン 新しいコンピューターを出荷する前に一定時間連続稼動させてメモリーチップなどに欠陥のないことを確認すること) と呼ばれる．

[例 1.11]　例題 1.10の２次元正規分布を MCMC で（わざと）生成してみる．あくまでも MCMC を説明するための例題であり，実用的には課題 1.15の方法が優れている．MCMC の実装には $q(\boldsymbol{v}|\boldsymbol{x})$ を指定する必要がある．ここでは以下のもので試してみる（あまり良いものではない）．

$$q(\boldsymbol{v}|\boldsymbol{x}) = \begin{cases} (2d)^{-2} & |\boldsymbol{v}[1] - \boldsymbol{x}[1]| \le d,\ |\boldsymbol{v}[2] - \boldsymbol{x}[2]| \le d \\ 0 & \text{それ以外} \end{cases}$$

つまり各座標成分の増分を $(-d, d)$ の一様分布とする．$q(\boldsymbol{v}|\boldsymbol{x}) = q(\boldsymbol{x}|\boldsymbol{v})$ だから，

$$\alpha(\boldsymbol{v}, \boldsymbol{x}) = \min\left\{1, \frac{f(\boldsymbol{v})}{f(\boldsymbol{x})}\right\}$$

である．この例のように提案分布が v と x に対して対称なものがメトロポリス法．

```
> N <- 1000 # 反復回数
> rho <- 0.5 # 相関係数
> S <- matrix(c(1,rho,rho,1),2,2) # 行列 Sigma の定義
> mu <- c(5,0) # ベクトル mu の定義
> Sinv <- solve(S) # S の逆行列
> myf <- function(x) exp(-0.5*(x-mu)%*%Sinv%*%(x-mu)) # 定数項を無視した f(x)
> d <- 0.5 # 移動幅
> x <- c(0,0) # 初期値
> xs <- matrix(0,2,N) # 系列の保存場所
> fs <- rep(0,N) # f(x) の保存場所（定数項は無視）
> cnt <- 0 # 移動回数の初期化
> for(t in 1:N) {
+   v <- x + (2*runif(2)-1)*d # v ~ q(v|x) を生成
+   a <- myf(v)/myf(x) # alpha=min(1,a) とすべきだが，以下の計算では不要
+   u <- runif(1) # u ~ U(0,1)
+   if(u <= a) {
+     x <- v # v へ移動
+     cnt <- cnt + 1 # 移動回数
+   }
+   xs[,t] <- x # x の保存
+   fs[t] <- myf(x) # f(x) の保存
+ }
> cnt/N # 移動頻度
[1] 0.819
> plot(xs[1,],xs[2,]) # 保存した x のプロット
> segments(c(0,xs[1,-N]),c(0,xs[2,-N]),xs[1,],xs[2,],col="pink") # 系列の軌跡
> plot(log(fs),type="l") # log(f(x)) のプロット（定数項は無視）
```

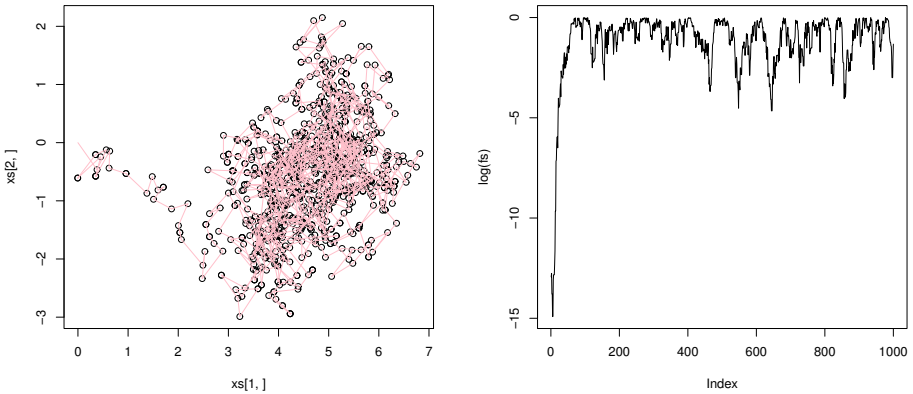



図 12　（左）例題 1.11の $\boldsymbol{X}_t$，（右）$\log(f(\boldsymbol{X}_t))$

[課題 1.17]　マルコフ連鎖の推移確率を定める条件付確率分布を K 種類用意して，$p_k(x_{t+1}|x_t),\ k = 1, \ldots, K$ と書く．それらの定常分布はすべて $f(x)$ とする．系列 $X_1, X_2, \ldots$ を生成する反復の各ステップで，確率 π_k でランダムに $p_k(x_{t+1}|x_t)$ を選ぶものとする（ただし $\sum_{k=1}^{K} \pi_k = 1$）．このとき，このマルコフ連鎖の推移確率を求め，定常分布が $f(x)$ になることを示せ．

[注意]　課題 1.17において，かならずしもランダムに順序を決める必要はない．たとえば，$k = 1, 2, \ldots, K, 1, 2, \ldots$ のように順番に繰り返し p_k を適用しても，定常分布は $f(x)$ である．

[課題 1.18]　m 次元ベクトルの系列 $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ を次のアルゴリズムで生成すると，その定常分布が $\boldsymbol{X} \sim f(\boldsymbol{x})$ であることを示せ．ただし，以下の記法を用いる．m 次元ベクトル $\boldsymbol{x}$ の i 番目の要素を $\boldsymbol{x}[i]$ と書く．$\boldsymbol{x}$ から $\boldsymbol{x}[i]$ を取り除いた $m-1$ 次元ベクトルを $\boldsymbol{x}[-i]$ と書く．$\boldsymbol{x}[i]$ の周辺確率分布を

$$h_i(\boldsymbol{x}[-i]) = \int_{-\infty}^{\infty} f(\boldsymbol{x})\, d\boldsymbol{x}[i]$$

と書き，$\boldsymbol{x}[-i]$ を与えたときの $\boldsymbol{x}[i]$ の条件付確率分布を

$$f_i(\boldsymbol{x}[i] \mid \boldsymbol{x}[-i]) = \frac{f(\boldsymbol{x})}{h_i(\boldsymbol{x}[-i])}$$

と書く．

1. 初期値 $\boldsymbol{X}_1$ を選び $t = 1$ とする．

2. **添え字 i をランダムに選ぶ．**
3. **$V \sim f_i(V|\boldsymbol{X}_t[-i])$ を１個生成．**
4. **$\boldsymbol{X}_{t+1}[i] = V$，$\boldsymbol{X}_{t+1}[-i] = \boldsymbol{X}[-i]$ と代入する．**
5. **t の値をひとつ増やして 2へ戻る．**

[注意]　この方法はギブスサンプラー（Gibbs sampler）と呼ばれる．マルコフチェイン・モンテカルロ法の特別な場合であり，$\alpha = 1$ となっている．

[課題 1.19]　m 次元正規分布 $\boldsymbol{X} \sim N_m(\boldsymbol{\mu}, \Sigma)$ と１次元の Y を考える．$\boldsymbol{X}$ と Y の同時分布が $m+1$ 次元多変量正規分布で，平均と分散を

$$E\left(\begin{bmatrix}\boldsymbol{X}\\ Y\end{bmatrix}\right) = \begin{bmatrix}\boldsymbol{\mu}\\ \lambda\end{bmatrix}, \quad V\left(\begin{bmatrix}\boldsymbol{X}\\ Y\end{bmatrix}\right) = \begin{bmatrix}\Sigma & \boldsymbol{b}\\ \boldsymbol{b}' & c\end{bmatrix}$$

で与える．$\boldsymbol{x}$ を与えたときの Y の条件付分布が

$$Y|\boldsymbol{x} \sim N\left(\lambda + \boldsymbol{b}'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}), c - \boldsymbol{b}'\Sigma^{-1}\boldsymbol{b}\right)$$

で与えられることを示せ．ただし Σ は正定値対称行列とする．ヒント：$e = (c - \boldsymbol{b}'\Sigma^{-1}\boldsymbol{b})^{-1}$, $\boldsymbol{d} = -e\Sigma^{-1}\boldsymbol{b}$, $\boldsymbol{A} = e\Sigma^{-1}\boldsymbol{b}\boldsymbol{b}'\Sigma^{-1}$ とおくとき次の恒等式が成り立つことを利用する．

$$\begin{bmatrix}\Sigma & \boldsymbol{b}\\ \boldsymbol{b}' & c\end{bmatrix}^{-1} = \begin{bmatrix}\Sigma^{-1} + \boldsymbol{A} & \boldsymbol{d}\\ \boldsymbol{d}' & e\end{bmatrix}$$

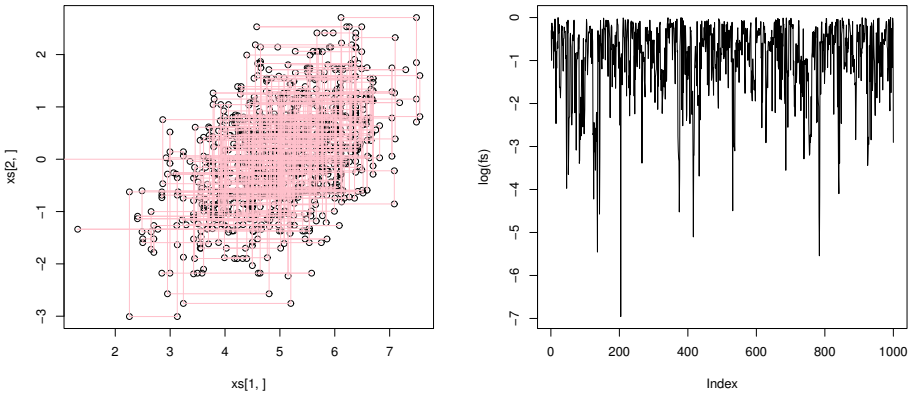


図 13　（左）例題 1.12の $\boldsymbol{X}_t$，（右）$\log(f(\boldsymbol{X}_t))$

[例 1.12]　例題 1.10，例題 1.11の２次元正規分布をギブスサンプラーで生成してみる．

$$\boldsymbol{X}[2]|\boldsymbol{x}[1] \sim N\left(\mu_2 + \rho(\boldsymbol{x}[1] - \mu_1), 1 - \rho^2\right)$$

である．

```
> N <- 1000 # 反復回数
> rho <- 0.5 # 相関係数
> S <- matrix(c(1,rho,rho,1),2,2) # 行列 Sigma の定義
> mu <- c(5,0) # ベクトル mu の定義
> Sinv <- solve(S) # S の逆行列
> myf <- function(x) exp(-0.5*(x-mu)%*%Sinv%*%(x-mu)) # 定数項を無視した f(x)
> x <- c(0,0) # 初期値
> xs <- matrix(0,2,N) # 系列の保存場所
> fs <- rep(0,N) # f(x) の保存場所（定数項は無視）
> for(t in 1:N) {
+   i <- floor(runif(1)*2)+1 # 変化させる成分を決める
+   v <- mu[i]+rho*(x[-i]-mu[-i]) + rnorm(1)*sqrt(1-rho^2) # X[i]|x[-i] の生成
+   x[i] <- v # 移動
+   xs[,t] <- x # x の保存
+   fs[t] <- myf(x) # f(x) の保存
+ }
> plot(xs[1,],xs[2,]) # 保存した x のプロット
> segments(c(0,xs[1,-N]),c(0,xs[2,-N]),xs[1,],xs[2,],col="pink") # 系列の軌跡
> plot(log(fs),type="l") # log(f(x)) のプロット（定数項は無視）
```


[例 1.13]　２次元配列 $\boldsymbol{x}[i]$, $i = (1,1), \ldots, (m,m)$ の各要素が $\{+1, -1\}$ のどちらかの値を取る．「エネルギー」を $H(\boldsymbol{x}) = -\gamma \sum_{(i,j)} \boldsymbol{x}[i]\boldsymbol{x}[j]$ で定義する物理系を「イジングモデル (Ising model)」という．ただし $\gamma > 0$ は定数，和の添え字は２次元格子のすべての隣同士である．これは強磁性体の性質を分析するために単純化したモデルである．$f(\boldsymbol{x}) \propto \exp(-H(\boldsymbol{x}))$ である．これをギブスサンプラで実現する．$h[i] = \gamma \sum_{j:(i,j)} \boldsymbol{x}[j]$（ただし和の添え字は i に隣接する j に関してとる）とおくと，$f_i(v|\boldsymbol{x}[-i]) \propto \exp(vh[i])$ であるから，

$$f_i(v|\boldsymbol{x}[-i]) = \frac{\exp(vh[i])}{\exp(h[i]) + \exp(-h[i])}$$

とすればよい．要するに，ランダムに画素 i を選び，その４近傍画素の $+1, -1$ の個数に応じて画素 i を $+1$ または-1 にする確率を決める．周辺の $+1$ の個数が多いほど画素 i も $+1$ になる確率が増える．一種のセルオートマトン（cf. 自発磁化，自己組織化，脳の情報処理）．

```
> N <- 100000 # 反復回数
> m <- 50 # ２次元配列を m*m とする．
> x0 <- matrix(runif(m*m)>0.5,m,m)*2-1 # ランダムに +1,-1 を設定
> gamma <- 1.0 # gamma > 1/2.269 で自発磁化
> x <- x0 # 初期値を x に代入
> xi <- function(i) x[(i[1]-1)%%m+1,(i[2]-1)%%m+1] # x[(i[1],i[2])] のとりだし
> d1 <- c(1,0); d2 <- c(-1,0); d3 <- c(0,1); d4 <- c(0,-1) # 「隣接」の定義
> for(t in 1:N) { # 反復の開始
+   i <- trunc(runif(2)*m)+1 # ランダムにセルを選んで i とする
+   a <- exp((xi(i+d1)+xi(i+d2)+xi(i+d3)+xi(i+d4))*gamma) # a=exp(h[i])
+   p <- a/(a+(1/a)) # p=f_{i}(+1 | x[-i])
+   if(runif(1)<=p) v <- 1 else v <- -1 # 確率 p で v=+1, 確率 (1-p) で v=-1
```

```
+   x[i[1],i[2]] <- v # x[i] := v
+ }
> bw <- rev(gray((0:64)/64)) # 64 階調のグレースケール
> image(x0,axes=F,col=bw) # 初期値
> image(x,axes=F,col=bw) # N 回の反復後
```

1.4 ベイズの定理

[課題 1.20] 全事象を Ω，その分割を $\sum_{i=1}^{\infty} D_i = \Omega$ とする．ある事象 A が起きたという条件の下で事象 D_i のおきる条件付確率は，

$$P(D_i|A) = \frac{P(A|D_i)P(D_i)}{\sum_{j=1}^{\infty} P(A|D_j)P(D_j)}$$

であることを示せ．ただし $P(A) > 0$ とする．これをベイズの定理（ベイズの公式）という．

[課題 1.21] 実数値の確率変数 $X,\ Y$ について，同時確率密度関数を $f(x,y)$ と書く．カンタンのため同じ f という記号を用いて，周辺密度は $f(x) = \int_{-\infty}^{\infty} f(x,y)\,dy,\ f(y) = \int_{-\infty}^{\infty} f(x,y)\,dx$ と書く．周辺密度も $f(x|y)$，$f(y|x)$ などと書く．このとき，ベイズの定理

$$f(x|y) = \frac{f(y|x)f(x)}{\int_{-\infty}^{\infty} f(y|x)f(x)\,dx}$$

を示せ．ただし $f(y) > 0$ とする．

図 14 （左）初期値，（右）100,000 回の反復後

[注意] 一般に a 次元確率変数 $\boldsymbol{X}$ と b 次元確率変数 $\boldsymbol{Y}$ について，同時確率密度関数を $f(\boldsymbol{x},\boldsymbol{y})$ と書けば，$f(\boldsymbol{y}) > 0$ に対して

$$f(\boldsymbol{x}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})}{\int_{\mathbb{R}^a} f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})\,d\boldsymbol{x}}$$

である．$f(\boldsymbol{x})$ を事前分布，$f(\boldsymbol{x}|\boldsymbol{y})$ を事後分布と呼ぶ．

[課題 1.22] 確率変数 $X,\ Y$ が，次の分布に従っているとする（$Y|x$ は x を与えたときの Y の条件付分布）．

$$X \sim N(0,a^2), \quad Y|x \sim N(x,b^2)$$

このとき

$$X|y \sim N\left(cy, cb^2\right), \quad c = \frac{1}{1+\frac{b^2}{a^2}}$$

であることを示せ．上式の c を (i) $a = 3, b = 1$ と (ii) $a = 1, b = 3$ の場合に計算せよ．

[課題 1.23] 確率変数 $\mu \sim N(0,\tau^2)$ の実現値を μ とする（同じ記号を使ってしまっていることに注意）．サンプルサイズ n のデータが平均 μ，分散 σ^2 の正規分布にしたがうものする．つまり $x_1,\ldots,x_n \sim N(\mu,\sigma^2)$ である．データの平均値 $\bar{x} = (x_1+\cdots+x_n)/n$ をつかって，μ の事後分布を求めよ．ただし τ^2, σ^2 は既知とする．

[課題 1.24] 2 次元配列 $\boldsymbol{x}[i]$ と $\boldsymbol{y}[i],\ i = (1,1),\ldots,(m,m)$ の各要素が $\{+1,-1\}$ のどちらかの値を取る．$\boldsymbol{x}$ を「真の画像」，$\boldsymbol{y}$ を「ノイズで汚された画像データ」とする．$\boldsymbol{x}$ の各要素に確率 ϵ $(0 < \epsilon < 1)$ で独立にノイ

ズが入り反転するモデル (つまり $+1 \to -1$ または $-1 \to +1$) を考える．

$$f(\boldsymbol{y}[i]|\boldsymbol{x}[i]) = \begin{cases} 1-\epsilon & \boldsymbol{y}[i] = \boldsymbol{x}[i] \\ \epsilon & \boldsymbol{y}[i] \neq \boldsymbol{x}[i] \end{cases}$$

(i) このとき，$\boldsymbol{x}$ を与えたときの $\boldsymbol{y}$ の条件付分布が

$$f(\boldsymbol{y}|\boldsymbol{x}) \propto \exp\left(\lambda \sum_i \boldsymbol{x}[i]\boldsymbol{y}[i]\right)$$

とかけることを示せ．ただし $\lambda = \frac{1}{2}\log(\frac{1-\epsilon}{\epsilon})$，$\sum_i$ はすべての画像要素に関する和，$\propto$ は $\boldsymbol{x},\boldsymbol{y}$ に関して比例（それ以外は定数とみなす）である．

(ii) $\boldsymbol{x}$ の事前分布を

$$f(\boldsymbol{x}) \propto \exp\left(\gamma \sum_{(i,j)} \boldsymbol{x}[i]\boldsymbol{x}[j]\right)$$

で与える．ただし $\gamma > 0$ は定数，$\sum_{(i,j)}$ はすべての「隣接」に関する和（画像要素の 4 近傍を「隣接」と定義），$\propto$ は $\boldsymbol{x}$ に関して比例とする．このとき，$\boldsymbol{x}$ の事後分布，すなわち $\boldsymbol{y}$ を与えたときの $\boldsymbol{x}$ の条件付分布が

$$f(\boldsymbol{x}|\boldsymbol{y}) \propto \exp\left(\lambda \sum_i \boldsymbol{x}[i]\boldsymbol{y}[i] + \gamma \sum_{(i,j)} \boldsymbol{x}[i]\boldsymbol{x}[j]\right) \tag{1.8}$$

とかけることを示せ．ただし $\propto$ は $\boldsymbol{x}$ に関して比例（$\boldsymbol{y}$ は定数とみなす）である．

[例 1.14] 課題 1.24の (ii) で与えられた事後分布 $f(\boldsymbol{x}|\boldsymbol{y})$ に従う系列 $X_1, X_2, \ldots$ を生成すれば，ノイズ

で汚された画像データ y から真の画像 x をベイズ復元できる．例題 1.13の「エネルギー」を次のように変更する．$H(\boldsymbol{x}) = -\gamma \sum_{(i,j)} \boldsymbol{x}[i]\boldsymbol{x}[j] - \lambda \sum_i \boldsymbol{x}[i]\boldsymbol{y}[i]$．ただし y は固定して考える．物理的には画像要素 i に $\lambda \boldsymbol{y}[i]$ の外部磁場をかけたと解釈できる．$f(\boldsymbol{x}|\boldsymbol{y}) \propto \exp(-H(\boldsymbol{x}))$ をギブスサンプラで実現するには，$h[i] = \gamma \sum_{j:(i,j)} \boldsymbol{x}[j] + \lambda \boldsymbol{y}[i]$ と変更すればよい．定常分布からサンプル x を取り出すか，burn-in 後に「平均画像」$E(\boldsymbol{X}|\boldsymbol{y})$ を推定するなど，いろいろな計算が可能である．例えば平均画像の要素が正なら $+1$，負なら-1 にして 2 値化する（これは後ほど述べる MAP 推定量）．まずはパラメタを定義するなど下準備をする．

```
> ## パラメタ
> N <- 300000 # 反復回数
> M <- 100000 # burn-in
> ns <- 2500 # 誤差を計算する間隔
> m <- 50 # ２次元配列を m*m とする．
> gamma <- 1.0 # gamma > 1/2.269 で自発磁化
> eps <- 0.35 # ノイズの確率 35% (50%未満にしておく)
> lambda <- 0.5*log((1-eps)/eps) # 外部磁場の強さ
> d1 <- c(1,0); d2 <- c(-1,0); d3 <- c(0,1); d4 <- c(0,-1) # 「隣接」の定義
> xi <- function(i) if(i[1]>=1 && i[2]>=1 && i[1]<=m && i[2]<=m)
+   x[i[1],i[2]] else -1 # x[(i[1],i[2])]のとりだしを再定義
```

次に「真の画像」を作成し，それにわざとノイズを入れて汚し，「画像データ」を作成する．

```
> ## 真の２値画像の作成（任意画像をファイルから読み込ませても良い）
> y0 <- matrix(-1,m,m) # 背景は y[i]=-1
> for(i1 in 1:m) for(i2 in 1:m)
+   if(!(abs(i1-0.5*m)<0.15*m && abs(i2-0.5*m)<0.15*m) &&
+      sqrt((i1-0.5*m)^2+(i2-0.5*m)^2) < 0.4*m ) y0[i1,i2] <- +1
> ## ノイズで画像を汚す
```



```
> y <- y0 # 真の画像をコピー
> i <- runif(m*m) < eps # ノイズが入る添え字ベクトル（１次元配列として扱っている）
> y[i] <- - y[i] # 反転
> ## 誤差評価の関数を用意しておく
> myabserr <- function(x) mean(abs(x-y0)/2) # 真の画像からの誤差（y0 を知らないと計算できない）
> mylogf <- function(x) { # log(f(x|y)) の定数項を無視したもの（y0 をしらなくても計算できる）
+   s <- 0
+   for(i1 in 1:m) for(i2 in 1:m) {
+     i <- c(i1,i2)
+     s <- s + xi(i)*(xi(i+d1)+xi(i+d2)+xi(i+d3)+xi(i+d4))
+   }
+   lambda*sum(x*y)+gamma*s/2
+ }
```

画像復元の反復は次から始まる．

```
> ## 反復のための準備
> x <- y # 初期値を x に代入
> myabserr(x) # 誤差
[1] 0.35
> abserrs <- rep(0,1+N/ns); logfs <- rep(0,1+N/ns) # abserr と logf を保存する場所
> abserrs[1] <- myabserr(x); logfs[1] <- mylogf(x) # 初期値
> xp <- matrix(0,m,m) # 平均画像を保存する配列の初期化
> ## 反復計算
> for(t in 1:N) {
+   i <- trunc(runif(2)*m)+1 # ランダムにセルを選んで i とする
+   a <- exp((xi(i+d1)+xi(i+d2)+xi(i+d3)+xi(i+d4))*gamma
+            +y[i[1],i[2]]*lambda) # a=exp(h[i])
+   p <- a/(a+(1/a)) # p=f_{i}(+1 | x[-i],y)
+   if(runif(1)<=p) v <- 1 else v <- -1 # 確率 p で v=+1, 確率 (1-p) で v=-1
```



```
+   x[i[1],i[2]] <- v # x[i] := v
+   if(t>M) xp <- xp + x # 後半の x(t) を足しこんでいく
+   if((t %% ns) == 0) { # 誤差を保存
+     j <- 1 + t %/% ns
+     abserrs[j] <- myabserr(x); logfs[j] <- mylogf(x) # 誤差
+   }
+ }
> xN <- x # 最後の値
> myabserr(xN) # 誤差
[1] 0.0764
> xp <- xp / (N-M) # 平均画像
> myabserr(xp) # 誤差
[1] 0.0775733
> xp2 <- sign(xp) # 平均画像を 2 値化したもの
> myabserr(xp2) # 誤差
[1] 0.0696
> bw <- rev(gray((0:64)/64)) # 64 階調のグレースケール
> image(y0,axes=F,col=bw) # 真の画像
> image(y,axes=F,col=bw) # ノイズで汚された画像データ
> image(xp,axes=F,col=bw) # 後半 N-M 回の平均画像
> image(xp2,axes=F,col=bw) # 平均画像を 2 値化したもの
> plot(0:(N/ns),abserrs,xlab=paste("t/",ns,sep="")) # 誤差
> abline(v=M/ns,lty=2) # この点線より左側が burn-in
> plot(0:(N/ns),logfs,xlab=paste("t/",ns,sep="")) # log(f(y|x))
> abline(v=M/ns,lty=2) # この点線より左側が burn-in
```

[課題 1.25]　例題 1.14の画像復元を現実の応用として考えたとき，本来知らないはずの情報を使ってしまっている点が問題である．とくに $\epsilon = 0.35$ を利用してしまっている．そこで，データからこの値を推定する方

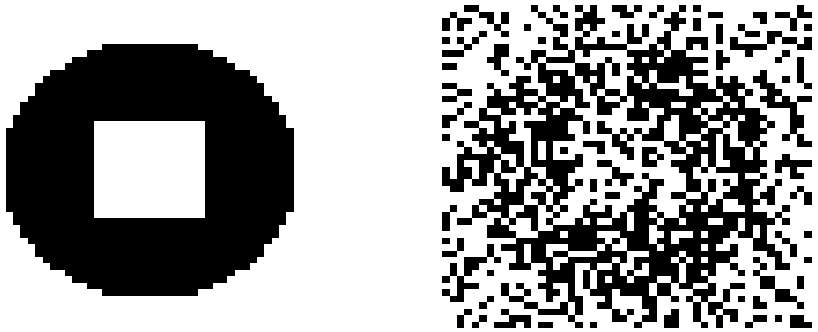


図 15　（左）真の画像，（右）ノイズで汚された画像データ

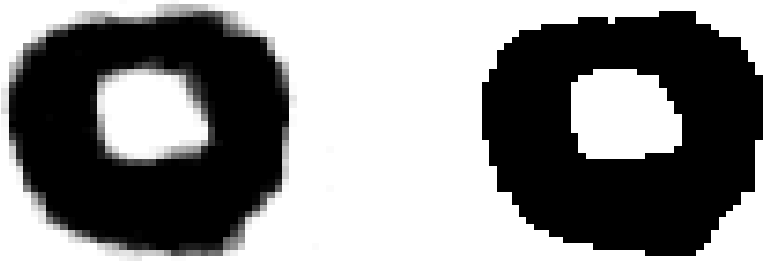


図 16　（左）burn-in 後の平均画像，（右）それを 2 値化したもの

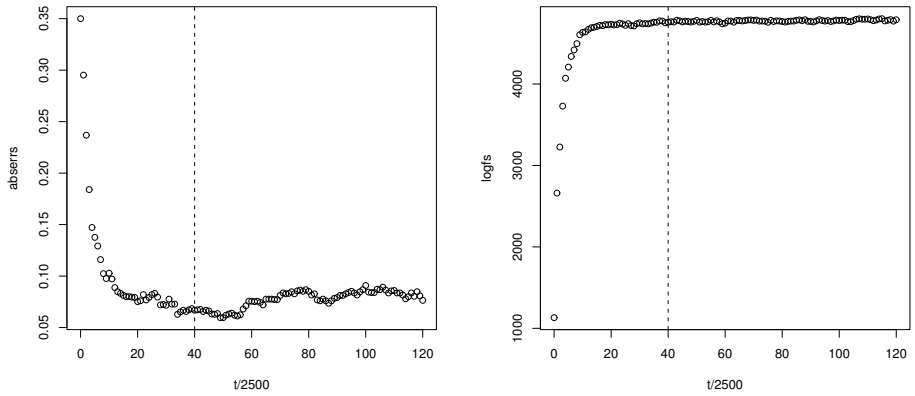


図 17　（左）真の画像からの誤差 $\sum_i |\boldsymbol{X}_t[i] - \boldsymbol{x}[i]|/(2m^2)$ の値，（右）定数項を無視した $\log f(\boldsymbol{X}_t|\boldsymbol{y})$ の値，つまり式 (1.8) の右辺にある $\exp$ 関数の引数の値

法を検討せよ．ヒント：

```
> mean(xp2 != y)
[1] 0.3468
```

[課題 1.26]　**例題 1.14を参考にして，類似の例題を作成せよ．**(i) **「真の画像」を他の形や実画像に変える，**(ii) **「隣接」の定義を変える，**(iii) γ **や** λ **のパラメタ値を変える，などの変更を試みよ．その際，** N **と** M **が十分に大きいかを確認しておくこと．**

[例 1.15]　**画像処理の定番である「メディアンフィルタ」でもかなり良い結果が得られる．各画素の近傍** 3×3 **の領域の画素値を調べ，その中央値（メディアン）をフィルタの値として用いる．要するに，**9 **個の画素の** $+1$ **に個数をしらべ，それが** 5 **以上ならば中心を** $+1$ **にする．このフィルタを数回適用する．**

```
> nf <- 10 # 適用回数
> d0 <- c(0,0) # 中心セル
> d1 <- c(1,0); d2 <- c(-1,0); d3 <- c(0,1); d4 <- c(0,-1) # 「隣接」の定義
> d5 <- c(1,1); d6 <- c(-1,1); d7 <- c(-1,-1); d8 <- c(1,-1) # つづき
> xi9 <- function(i) c(xi(i+d0),xi(i+d1),xi(i+d2),xi(i+d3),xi(i+d4),
+       xi(i+d5),xi(i+d6),xi(i+d7),xi(i+d8)) # 9近傍のベクタを返す
> xf <- y # 画像データを初期値として代入
> abserrs2 <- rep(0,nf+1); abserrs2[1] <- myabserr(y) # 誤差
> ## メディアンフィルタを繰り返し適用する
> for(t in 1:nf) {
+   x <- xf
+   for(i1 in 1:m) for(i2 in 1:m) xf[i1,i2] <- median(xi9(c(i1,i2)))
+   abserrs2[t+1] <- myabserr(xf)
+ }
```



```
> abserrs2[nf+1] # 繰り返し後の誤差
[1] 0.0884
> plot(0:nf,abserrs2,xlab="iteration") # 誤差
> image(xf,axes=F,col=bw) # フィルタ画像
```

1.5　積率母関数，中心極限定理

[定義 1.6]　$t \in \mathbb{R}$ **に対して，確率変数** X **の積率母関数** (moment generating function) **を** $M_X(t) = E(e^{tX})$ **とかく．**

[課題 1.27]　**原点のまわりの** k **次モーメントが以下のように** $M_X(t)$ **を使ってかけることを示せ．**

$$E(X^k) = \frac{d^m M_X(t)}{dt^k}\Big|_{t=0}$$

[課題 1.28]　$X \sim N(0,1)$ **の累積母関数** $M_X(t)$ **を求めよ．この結果を用いて** $E(X^k)$ **の一般式を求め，その値を** $k = 2, 4, 6, 8$ **で計算せよ．**

[課題 1.29]　X **と** Y **は独立な確率変数とする．** $M_{X+Y}(t) = M_X(t)M_Y(t)$ **を示せ．**

[課題 1.30]　**独立な確率変数列** $X_1, \ldots, X_n \sim N(0,1)$ **に対して，** X_1^2 **の積率母関数を求め，それを利用して**

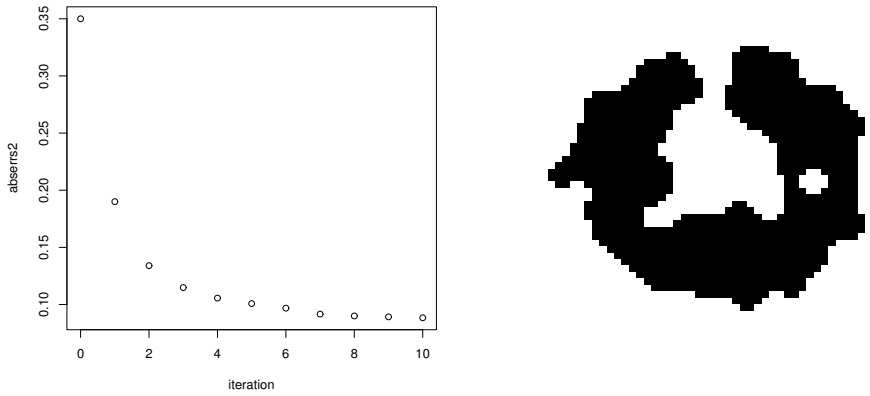


図 18　（左）真の画像からの誤差，（右）メディアンフィルタを１０回適用して得られた画像

$Y_n = X_1^2 + \cdots + X_n^2$ **の積率母関数** $M_{Y_n}(t)$ **を求めよ．自由度** n **のカイ二乗分布** $Y \sim \chi_n^2$ **の密度関数は**

$$f(y) = \frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} e^{-\frac{y}{2}} y^{\frac{n}{2}-1}, \quad y \geq 0$$

である．これから Y **の積率母関数** $M_Y(t)$ **をもとめ，** $M_{Y_n}(t)$ **と比較せよ．**

[**定義** 1.7]　$t \in \mathbb{R}$ **に対して，確率変数** X **の特性関数** (characteristic function) **を** $\varphi_X(t) = E(e^{itX})$ **とかく．**

[**注意**]　**形式的には** $\varphi_X(t) = M_X(it)$ **（ただし** $i = \sqrt{-1}$**）．積率母関数だと期待値が発散してしまうことがあるが，特性関数はいつも存在するし，理論的な目的に便利．確率分布と特性関数は一対一対応する．** $\varphi_X(t)$ **は要するに密度関数** f **のフーリエ変換にすぎないことを考えれば，** $f(x)$ **の関数としての情報を** $\varphi_X(t)$ **が持っていることを理解できるだろう．フーリエ変換とその逆変換は**

$$\tilde{f}(t) = \mathcal{F}f(t) = \int_{-\infty}^{\infty} e^{-itx} f(x)\, dx, \quad f(x) = \mathcal{F}^{-1}\tilde{f}(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-itx}\tilde{f}(t)\, dt$$

したがって $\varphi_X(t) = \tilde{f}(-t)$**．もし** $M_X(t)$ **が存在すれば** $\varphi_X(t)$ **の代わりに用いて分布と対応づけてよい．多変量確率分布の場合も同様である．**

[**定義** 1.8]　**非負整数値の確率変数** X **を考える．** $-1 \leq z \leq 1$ **に対して，** X **の確率母関数を** $G_X(z) = E(z^X)$ **とかく．**

[**注意**]　**形式的には** $G_X(e^t) = M_X(t)$ **である．非負整数値の確率変数** X **に対して** $G_X(z)$ **のほうが扱いやすい場合もあるが，** $M_X(t)$ **を使って議論しても問題ない．**

[**課題** 1.31]　**確率変数** X **は試行回数** $n > 0$**，成功確率** $0 < p < 1$ **の２項分布に従い，確率変数** Y **は期待値パラメータ** $\lambda > 0$ **のポアソン分布に従う．それぞれの確率関数は** $0 \leq x \leq n, y \geq 0$ **の整数に対し**

$$p_X(x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}, \quad p_Y(y) = \frac{e^{-\lambda}\lambda^y}{y!}$$

である．積率母関数が

$$M_X(t) = \Big(pe^t + (1-p)\Big)^n, \quad M_Y(t) = \exp\Big(\lambda(e^t - 1)\Big)$$

であることを示せ．

[**定義** 1.9]　m **次元ベクトルの確率変数** $\boldsymbol{X} = (X_1, \ldots, X_m)'$**，** m **次元ベクトル** $\boldsymbol{t} = (t_1, \ldots, t_m)'$ **に対して積率母関数は** $M_{\boldsymbol{X}}(\boldsymbol{t}) = E(e^{\boldsymbol{t}'\boldsymbol{X}})$ **である．ただし「′」は行列やベクトルの転置を表す．**

[**課題** 1.32]　m **次元確率変数** $\boldsymbol{X}$ **の平均と分散が** $E(\boldsymbol{X}) = \boldsymbol{\mu}$**，** $V(\boldsymbol{X}) = \boldsymbol{\Sigma}$ **のとき，これを線形変換した** $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b}$ **の平均と分散を求めよ．ただし** $\boldsymbol{A}$ **は** $k \times m$ **行列，** $\boldsymbol{b}$ **は** k **次元ベクトルとする．**

[**課題** 1.33]　**上の課題で** $\boldsymbol{X}$ **が多変量正規分布** $\boldsymbol{X} \sim N_m(\boldsymbol{\mu}, \Sigma)$ **に従うとする．このとき，** $\boldsymbol{Y}$ **の従う分布を求めよ．**

[**定義** 1.10]　**確率変数列** Y_n**，確率変数** Y**，任意の有界な連続関数** g **に対して**

$$\lim_{n\to\infty} E(g(Y_n)) = E(g(Y)) \tag{1.9}$$

となるとき， Y_n **は** Y **に法則収束（または分布収束）するという．これを** $Y_n \xrightarrow{d} Y$ **とかく．** Y **の分布が** F **なら** $Y_n \xrightarrow{d} F$ **などと書いてもよい．**

[**注意**]　**確率変数列** Y_n **の確率分布を** $f_n(y)$**，確率変数** Y **の確率分布を** $f(y)$ **とする．** $Y_n \xrightarrow{d} Y$ **は要するに** $f_n \to f$ **と理解すればよい．つまり** $g(x) = I(x \in A)$ **とすれば** (1.9) **は** $\lim_{n\to\infty} f_n(A) = f(A)$**．特に** $A = (-\infty, y)$ **なら累積分布関数に関して** $\lim_{n\to\infty} F_n(y) = F(y)$**．じつは** $F(y)$ **の任意の連続点** y **でこれがい**

えれば $Y_n \xrightarrow{d} Y$ である．

[定理 1.2]　X_n は独立に同一の分布にしたがう確率変数列とし，$E(X_1)=0$ と $V(X_1)=1$ とする．最初の n 個の平均値を $\sqrt{n}$ 倍したものを

$$Y_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i$$

とかく．定理 1.1の記号でいえば，$Y_n = \sqrt{n}\bar{X}_n$ である．このとき，$n \to \infty$ の極限で

$$Y_n \xrightarrow{d} N(0,1) \tag{1.10}$$

が成り立つ．これを中心極限定理という．

[注意]　もし $E(X_1)=\mu$ と $V(X_1)=\sigma^2$ なら $Z_n = (X_n-\mu)/\sigma$ と置き換えて Z_n に中心極限定理を適用すれば，$\sqrt{n}\bar{Z}_n \xrightarrow{d} N(0,1)$．$n$ が十分に大きければ $\sqrt{n}\bar{Z}_n$ の分布が $N(0,1)$ で近似される．これを漸近正規性 (asymptotic normality) といい，$\sqrt{n}\bar{Z}_n \overset{a}{\sim} N(0,1)$ とかく．次のように書いてもよい．$\bar{Z}_n \overset{a}{\sim} N(0,1/n)$，$\bar{X}_n \overset{a}{\sim} N(\mu,\sigma^2/n)$．ちなみに $E(\bar{X}_n)=\mu$，$V(\bar{X}_n)=\sigma^2/n$ であるから，$\bar{X}_n$ のしたがう近似分布を求めるには，その平均と分散だけ計算して正規分布とすればよい．

[証明]　カンタンのため $E(X_1^k), k=1,2,\ldots$ が存在すると仮定する．X_1 の特性関数は $\varphi_{X_1}(t) = E(e^{itX_1}) =$

$\sum_{k=0}^{\infty} \frac{(it)^k}{k!} E(X_1^k) = 1 - \frac{t^2}{2} - \frac{it^3}{6}E(X_1^3) + \frac{t^4}{24}E(X_1^4) + \cdots$ と展開できる．したがって，

$$\varphi_{Y_n}(t) = \left[\varphi_{X_1}(t/\sqrt{n})\right]^n = \left[1 - \frac{t^2}{2n} - \frac{it^3}{6n^{3/2}}E(X_1^3) + \frac{t^4}{24n^2}E(X_1^4) + \cdots\right]^n = \left[1 - \frac{t^2}{2n}(1+o(1))\right]^n$$

極限をとれば $\lim_{n\to\infty} \varphi_{Y_n}(t) = e^{-t^2/2}$．

[例 1.16]

```
> ## myclt: n 個の一様分布の標本平均を b 個生成する
> ## 入力: n=サンプルサイズ，b=シミュレーションの繰り返し回数
> ## 出力: b 個の標本平均のヒストグラム
> myclt <- function(n,b) {
+   X <- matrix(runif(n*b),n,b) # n*b 行列に U(0,1) を代入
+   a <- rep(1/n,n) # 1/n を n 個ならべたベクトル．
+   y <- a %*% X # b 次元横ベクトル
+   x0 <- seq(min(y),max(y),length=1000) # y のレンジを 1000 分割
+   truehist(y) # ヒストグラム（密度関数表示）
+   lines(x0,dnorm(x0,mean=0.5,sd=sqrt(1/120)),col=2) # 正規分布（赤色）
+ }
> myclt(5,10000)
> myclt(10,10000)
```

結果は図 19．

[例 1.17]

```
> ## myclt2: n 回のベルヌーイ試行（確率 p で 1，確率 1-p で 0）の和を b 個生成する
> ## 入力: n=サンプルサイズ，p=+1 の確率，b=シミュレーションの繰り返し回数
> ## 出力: b 個の標本和のヒストグラム
```

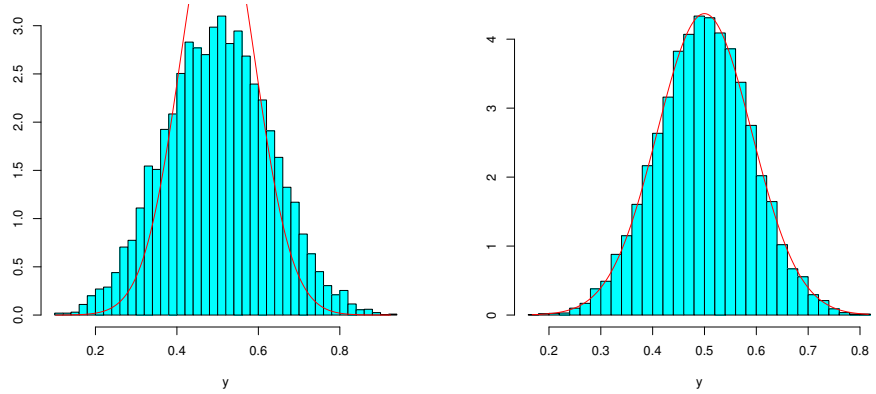



図 19　n 個の一様分布の平均．（左）$n=5$ では正規近似は良くない，（右）$n=10$ では正規近似が良い．


```
> myclt2 <- function(n,p,b) {
+   X <- matrix(as.numeric(runif(n*b)<p),n,b) # n*b 行列に 1 または 0 を代入
+   a <- rep(1,n) # 1 を n 個ならべたベクトル．
+   y <- a %*% X # b 次元横ベクトル
+   x0 <- seq(from=min(y)-0.5,to=max(y)+0.5) # レンジを 0.5 ずらして 1 刻み
+   truehist(y,breaks=x0) # 確率をプロット
+   x0 <- seq(from=min(y),to=max(y)) # レンジを 1 刻み
+   points(x0,dbinom(x0,size=n, prob=p)) # 2 項分布（黒）
+   points(x0,dpois(x0,lambda=n*p),col=3) # ポアソン分布（緑）
+   x0 <- seq(from=min(y)-0.5,to=max(y)+0.5,length=10000) # レンジを 1000 分割
+   lines(x0,dnorm(x0,mean=n*p,sd=sqrt(n*p*(1-p))),col=2) # 正規分布（赤色）
+ }
> myclt2(10,0.05,10000)
> myclt2(10,0.5,10000)
```

結果は図 20．

[課題 1.34]　ベルヌーイ試行（n 回，成功確率 p）の成功回数 X は 2 項分布に従う．(i)　定数 $\lambda > 0$ を与えて，$p = \lambda/n$ としたとき，$n \to \infty$ の極限で X は期待値 λ のポアソン分布に従うことを示せ．これを「少数の法則」と呼ぶ．ヒント： 課題 1.31の結果を参考にして，$M_X(t) \to M_Y(t)$ を示せばよい．(ii) $0 < p < 1$ を固定して $n \to \infty$ としたとき，$(X-np)/\sqrt{n}$ が平均 0，分散 $p(1-p)$ の正規分布に従うことを，中心極限定理を使わないで直接確かめよ（積率母関数の極限を求める）．

[課題 1.35]　$\boldsymbol{X}_n$ は独立に同一の分布に従う m 次元確率変数列とし，$E(\boldsymbol{X}) = \boldsymbol{0}$，$V(\boldsymbol{X}) = \Sigma = (\sigma_{ij})$ とす

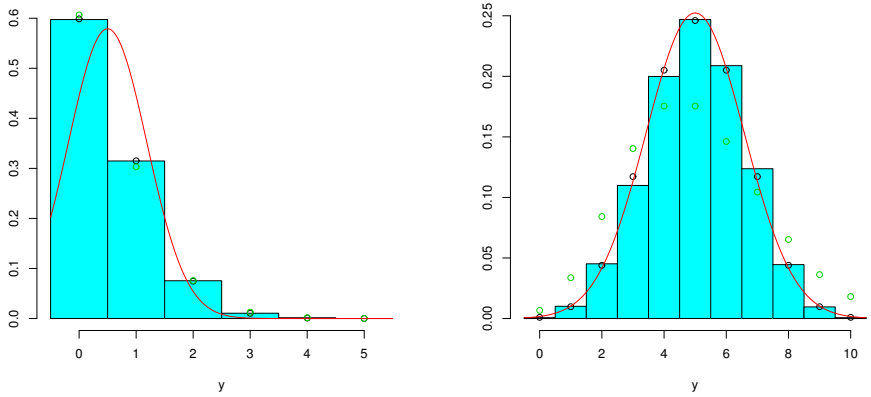


図 20　n 回のベルヌーイ試行の成功回数．（左）$n = 10, p = 0.05$ では正規近似は良くない，（右）$n = 10, p = 0.5$ では正規近似が良い．

る．標本平均ベクトルを $\bar{\boldsymbol{X}}_n = \sum_{i=1}^n \boldsymbol{X}_i/n$ と書く．中心極限定理

$$\sqrt{n}\bar{\boldsymbol{X}}_n \xrightarrow{d} N_m(\boldsymbol{0}, \boldsymbol{\Sigma}) \tag{1.11}$$

を示せ．ただし N_m は多変量正規分布を表し，高次モーメント $E(X_1^{k_1} \cdots X_m^{k_m})$, $k_1, \ldots, k_m \geq 0$ がすべて存在すると仮定する．ヒント：特性関数 $\varphi_{\sqrt{n}\bar{\boldsymbol{X}}}(\boldsymbol{t}) = E(e^{i\sqrt{n}\boldsymbol{t}'\bar{\boldsymbol{X}}})$ の収束を言えばよい．

2　推定論

サブセクション：確率モデル，判別問題，パラメタ推定，EM アルゴリズム，最尤推定量の性質

キーワード：混合分布，ベイジアンネット，ベイズ決定，最尤推定（ＭＬ），ＭＡＰ推定，学習用データ，テスト用データ，クラメール・ラオの不等式，Fisher 情報行列，信頼区間

2.1　確率モデル

m 次元確率変数 $\boldsymbol{X}$ の従う分布を $q(\boldsymbol{x})$ とし，これを $\boldsymbol{X} \sim q(\boldsymbol{x})$ と書く．本来 $q(\boldsymbol{x})$ は密度関数の記号であるし，抽象的に分布を表すならむしろ $\boldsymbol{X} \sim q$ などと書いたほうが適切であろうが，このような記法が便利なことがあるので採用する．n 個の m 次元確率変数 $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ が独立に同じ分布 $q(\boldsymbol{x})$ に従う (i.i.d.=independently identically distributed) **とし，これを $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim q(\boldsymbol{x})$** (i.i.d.) **と書く．$n$ をサンプルサイズと呼ぶ．以下では，観測したデータを $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ と書く．**

[定義 2.1]　密度関数 $q(\boldsymbol{x})$ を近似するために，p 次元実ベクトル $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^p$ をパラメタとする密度関数 $f(\boldsymbol{x}; \boldsymbol{\theta})$ を考える．これをパラメトリック確率モデル（または単に確率モデル）と呼ぶ．

[定義 2.2]　「モデル $f(\boldsymbol{x}; \boldsymbol{\theta})$ が正しい」とは，ある $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}$ が存在して，$q = f(\cdot; \boldsymbol{\theta}_0)$ となるときである（密度関数が等しいという意味）．つまり，$\boldsymbol{x}$ が取りうるすべての値に対して，$q(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}_0)$．

[注意]　現実には一般にモデルは正しくなく，あくまで近似である．モデルとは，データから情報を抽出するための「手段」である．

[例 2.1]　モデルが正規分布 $X \sim N(\mu, \sigma^2)$ ならば $\boldsymbol{\theta} = (\mu, \sigma^2)$, $p = 2$．m 次元の多変量正規分布 $\boldsymbol{X} \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ ならば $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $p = m(m+3)/2$ である．

[例 2.2]　k 個の情報源があり，i 番目の情報源から出る信号を観測すると $X \sim f_i(x; \boldsymbol{\theta}_i)$ である．実際に観測する信号は k 個の信号のどれかひとつが毎回ランダムに選ばれる．情報源 i を観測する確率（事前確率）を $p(i) = \pi_i > 0$ とする（$\sum_{i=1}^k \pi_i = 1$）．このとき，観測信号 X と信号源 i の同時分布をあらわす確率モデルは

$$f(x, i; \boldsymbol{\theta}) = f_i(x; \boldsymbol{\theta}_i)\pi_i$$

である．ただし $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{k-1}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$ ．上式は $f(x, i; \boldsymbol{\theta}) = f(x|i; \boldsymbol{\theta})p(i; \boldsymbol{\theta})$ と書いても良いだろう．例えば信号源 i が $X \sim N(\mu_i, \sigma_i^2)$ とすれば，$\boldsymbol{\theta}_i = (\mu_i, \sigma_i^2)$，$f_i(x; \boldsymbol{\theta}_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2})$．

[例 2.3]　例 2.2において，X がどの信号源から得られたかが分からないとする．つまり i は観測できず X だ

けが観測される場合を考える．このとき観測信号 X の確率モデルは，周辺分布になる．

$$f(x;\boldsymbol{\theta}) = \sum_{i=1}^{k} f_i(x;\boldsymbol{\theta}_i)\pi_i$$

[定義 2.3]　例 2.3のように，分布に重みをつけて平均することを「混合する」といい，そのようにして得られた分布を「混合分布 (mixture distribution)」という．とくに各成分が正規分布の場合，得られたモデルを正規混合モデル (normal mixture model) という．

[注意]　確率モデルをグラフ（有向グラフ）で表現すると便利なことが多い．変数 i と x をそれぞれノード，i から x へ向かう枝を考える．この枝は i から x への関係，すなわち条件付分布 $f(x|i)$ を表す．ノード i に向かう枝がないが，この場合は条件なしの分布 $p(i)$ を与える．もしたとえば，i, j の二つの変数があって，それらの条件付分布として $f(x|i,j)$ が定義されていれば，i と j の二つのノードから x へ向かう２本の枝を書く．このような表現はグラフィカルモデルまたはベイジアンネットと呼ばれる．ノードや枝にあたえた分布や条件付分布をすべて掛け合わせれば全体の同時確率分布が得られる．そして注目した確率変数 (ここでは X) に関する周辺分布を求めることによってモデルが得られる．連続音声認識等で用いられる大規模なHMM（隠れマルコフモデル）も，ノードや枝の数は多いけれど，この一例である．アプリケーションでは回路図設計のような工学的センスが重要．

[例 2.4]　正規混合分布（例題 2.3）からデータを生成する．ここでは $n_1 = 100$, $k = 3$,

$$\pi_1 = 0.5, \pi_2 = 0.3, \pi_3 = 0.2, \mu_1 = 0, \mu_2 = 4, \mu_3 = -3, \sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 1.$$

```
> ### 正規混合分布のデータの作成
> ## サンプルサイズ: n1
> ## 成分数: k
> ## 確率: pr[1],...,pr[k-1]
> ## 平均: mu[1],...,mu[k]
> ## 分散: ss[1],...,ss[k]
> ## 生成したデータは，xx1 に保存
> n1 <- 100; k <- 3
> pr <- c(0.5,0.3); mu <- c(0,4,-3); ss <- c(1,2,1)^2
> pr[k] <- 1 - sum(pr)
> ## 密度関数を描く関数（後で使う）
> ## x0=プロットする x 軸，(k,pr,mu,ss)=確率モデルのパラメタ，col=色
> drawnormmix <- function(x0,k,pr,mu,ss,col) {
+   ## 各 i で f(x0[t]|i)*p(i),t=1,2,.. の計算
+   fi <- matrix(0,length(x0),k) # fi(xt), i=1,...,k, t=1,...,n
+   for(i in 1:k) fi[,i] <- pr[i]*dnorm(x0,mean=mu[i],sd=sqrt(ss[i]))
+   ## f(x0[t]),t=1,2,... の計算
+   f <- apply(fi,1,sum) # f(xt), t=1,...,n
+   ## 描画
+   if(col != FALSE) {
+     matlines(x0,fi,col=col,lty=1) # 各成分を描く
+     lines(x0,f,col=col,lty=2,lwd=2) # 和を描く
```



```
+   }
+   invisible(list(f=f,fi=fi)) # 値を返す（表示は抑制）
+ }
> ## ここからデータ作成
> ii1 <- sample(k,n1,replace=TRUE,prob=pr) # 信号源をランダムに選ぶ
> ii1[1:30] # 選んだ信号源（最初の３０個）
 [1] 1 1 2 1 1 1 3 1 2 1 2 3 1 1 2 2 1 1 1 2 2 2 2 2 1 1 2 3 2 1
> zz1 <- rnorm(n1) # N(0,1) を生成
> xx1 <- zz1*sqrt(ss)[ii1] + mu[ii1] # データの作成
> x0 <- seq(min(xx1),max(xx1),length=400) # 後で密度関数を描くためにデータのレンジを 400 等分
> hist(xx1,nclas=10,prob=T) # ヒストグラム
> rug(xx1) # データ点を横軸に線分で描く
> f0 <- drawnormmix(x0,k,pr,mu,ss,"darkgreen")$f # 真の密度関数（f0 をあとで再利用する）
```

次にモデルのパラメタを推定する．まず (x,i) が観測できる場合を考える．$i = 1,\ldots,k$ で場合わけして，単純に平均と分散を求める．

```
> ## 推定結果は，pr1,mu1,ss1 に保存
> pr1 <- mu1 <- ss1 <- rep(0,k) # 初期化
> for(i in 1:k) {
+   pr1[i] <- sum(ii1==i)/length(ii1) # 信号源 i の確率（データにおける頻度で推定する）
+   x <- xx1[ii1==i] # 信号源 i のものだけ取り出す
+   mu1[i] <- mean(x) # 標本平均
+   ss1[i] <- mean((x-mu1[i])^2) # 標本分散
+ }
> pr1 # 推定した pr
[1] 0.50 0.32 0.18
> mu1 # 推定した mu
[1] -0.07201132  3.86638323 -3.16861588
> sqrt(ss1) # 推定した sqrt(ss)
```



```
[1] 1.0881641 1.8268178 0.8969115
> drawnormmix(x0,k,pr1,mu1,ss1,"red") # 信号源ごとに推定
```

この推定法は，次節で述べる最尤法の一例である．ここで用いたサンプルサイズ n_1 のデータを $\mathcal{X}_1 = \{(x_t, i_t), t = 1,\ldots,n_1\}$ と書く．最尤法によるパラメタ推定では，対数尤度関数

$$\log L(\boldsymbol{\theta}|\mathcal{X}_1) = \sum_{t=1}^{n_1} \log\left(f_{i_t}(x_t;\boldsymbol{\theta}_{i_t})\pi_{i_t}\right)$$

を最大にするパラメタ値を求める．この例では最尤法が解析的に得られて，上記の方法に帰着する．

2.2　判別問題（分類，識別）

[例 2.5]　例 2.3の混合分布において，x を与えたときの i の事後確率 $p(i|x)$ は

$$\pi_i(x;\boldsymbol{\theta}) = \frac{f_i(x;\boldsymbol{\theta}_i)\pi_i}{f(x;\boldsymbol{\theta})}$$

である．事後確率最大になる i をつかえば，信号源（一般にクラスという）の判別ができる．つまり

$$\hat{i}(x;\boldsymbol{\theta}) = \arg\max_{i=1,\ldots,k} \pi_i(x;\boldsymbol{\theta})$$

によって，x がクラス $\hat{i}(x;\theta)$ に属するものと判断する．この方法をベイズ決定，ベイズ識別などともという．なお θ が未知であれば，データから推定したパラメタ値を $\hat{\theta}$ を使って事後確率を計算する．以下では単純な

例（一変量の正規分布）を扱うが，ここで説明する考え方は，現実の複雑なアプリケーションにも適用可能であり，パターン認識の基本的な手法である．たとえば音声認識では x が音声データ，i が単語である．(i.i.d. ではなくて，隠れマルコフモデルという時系列モデリングを行うが，考え方は同じ)．

[例 2.6]　例 2.4に $n_2 = 300$ のテスト用データを追加する．テスト用データの i をみないで x だけから $\hat{i}(x; \hat{\boldsymbol{\theta}})$ を計算して，判別の正解率を確かめる．パラメタ値は例 2.4の学習用データ (x_t, i_t) から推定したものを用いる．(学習用データの i は利用している点に注意)．

```
> ## サンプルサイズ: n2
> ## 生成したデータは，xx2 に保存
> n2 <- 300
> ii2 <- sample(k,n2,replace=TRUE,prob=pr) # 信号源をランダムに選ぶ
> ii2[1:30] # 選んだ信号源（最初の３０個）
 [1] 3 2 3 2 1 3 1 2 2 3 1 3 1 1 1 2 2 2 2 2 1 2 2 3 1 3 2 2 2 3
> zz2 <- rnorm(n2) # N(0,1) を生成
> xx2 <- zz2*sqrt(ss)[ii2] + mu[ii2] # データの作成
> hist(xx2,nclass=15,prob=T) # ヒストグラム
> rug(xx2) # データ点を横軸に線分で描く
> drawnormmix(x0,k,pr,mu,ss,"darkgreen") # 真の密度関数
> ## i の事後確率を計算．ii2 を見ないで，xx2 だけから計算する （パラメタ推定に xx1 と ii1 を利用している）
> a <- drawnormmix(xx2,k,pr1,mu1,ss1,FALSE) # n1 個のサンプルから推定したパラメタ値を利用
> round(t(a$fi[1:10,]),3) # f(xx2[t]|i)*p(i) 最初の 10 個 （有効数字３桁，転置してから表示）
      [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
[1,] 0.000 0.032 0.00 0.007 0.168 0.040 0.180 0.000 0.177 0.016
[2,] 0.000 0.041 0.00 0.057 0.011 0.000 0.009 0.062 0.010 0.000
[3,] 0.014 0.000 0.05 0.000 0.000 0.033 0.000 0.000 0.000 0.060
> round(a$f[1:10],3) # f(xx2[t]) 最初の 10 個
```

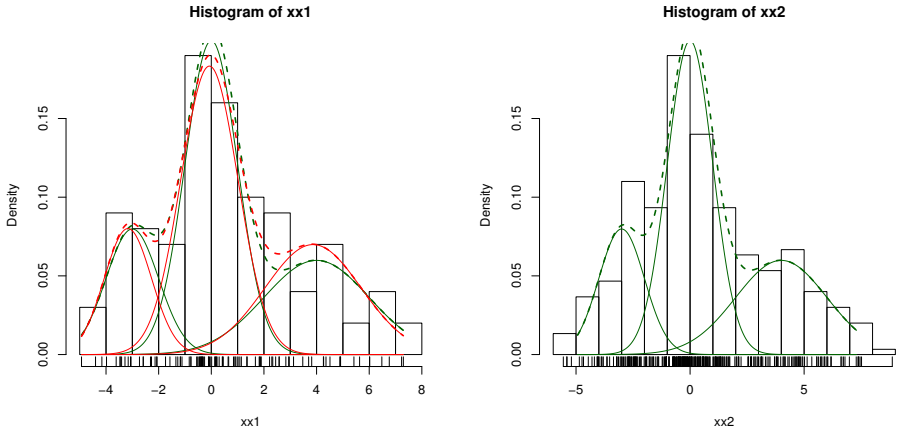



図 21　正規混合分布から生成したデータ．真の密度関数（緑），(x_t, i_t), $t = 1, \ldots, n_1$ から推定した密度関数（赤）．


```
 [1] 0.014 0.073 0.050 0.064 0.179 0.073 0.189 0.062 0.186 0.076
> pr2x1 <- a$fi / a$f # p(i|x) の計算
> round(t(pr2x1[1:10,]),3) # 最初の 10 個（有効数字３桁，転置してから表示）
      [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8]  [,9] [,10]
[1,] 0.001 0.44 0.005 0.112 0.937 0.548 0.954    0 0.948 0.210
[2,] 0.000 0.56 0.000 0.888 0.063 0.006 0.046    1 0.051 0.002
[3,] 0.999 0.00 0.995 0.000 0.000 0.446 0.000    0 0.000 0.788
> ## クラス（信号源）のベイズ決定: ii2x1 （pr2x1 の代わりに a$fi を使っても同じ）
> ii2x1 <- apply(pr2x1,1,function(p) order(-p)[1])
> ii2x1[1:30] # 推定された信号源（最初の３０個）
 [1] 3 2 3 2 1 1 1 2 1 3 1 3 1 1 1 2 2 2 2 2 1 2 2 3 1 1 2 2 2 3
> sum(ii2x1 == ii2)/length(ii2) # 正解率
[1] 0.9033333
```

[例 2.7]　スパムメールの判別を考える．混合分布で $k = 2$ とする．$i = 0$ をスパムで無いメール (ham)，$i = 1$ をスパムメール (spam) とする（添え字の範囲をひとつずらした)．あらかじめ選定した m 種類の単語について，単語 $j = 1, \ldots, m$ がメールに含まれるとき $\boldsymbol{x}[j] = 1$，含まれないとき $\boldsymbol{x}[j] = 0$ とする．i を与えたときの $\boldsymbol{x}$ の条件付確率 $p(\boldsymbol{x}|i)$ が，次式で与えられると仮定する．

$$f_i(\boldsymbol{x}|i; \boldsymbol{\theta}_i) = p(\boldsymbol{x}[1]|i)p(\boldsymbol{x}[2]|i) \cdots p(\boldsymbol{x}[m]|i)$$

つまり i の条件付きで，各単語の出現が独立と仮定する．ただし $\boldsymbol{x}[j] = 1$ なら $p(\boldsymbol{x}[j]|i) = \boldsymbol{\theta}_i[j]$，$\boldsymbol{x}[j] = 0$ なら $p(\boldsymbol{x}[j]|i) = 1 - \boldsymbol{\theta}_i[j]$ とする．確率モデルのパラメータは，$\boldsymbol{\theta} = (\pi_1, \boldsymbol{\theta}_0[1], \ldots, \boldsymbol{\theta}_0[m], \boldsymbol{\theta}_1[1], \ldots, \boldsymbol{\theta}_1[m])$ である．このモデルは現実を近似するに過ぎないが，とりあえずこれを仮定してスパムメール判別を行う．こ

の条件付独立を仮定したベイズ事後確率の計算は，Naive Bayes（簡単ベイズ？）と呼ばれる．以下で用いるのは，Spambase データセット（UCI Repository of machine learning databases http://www.ics.uci.edu/~mlearn/MLRepository.html から入手可能）の一部を取り出したものである．本来は単語頻度の情報があるが，それを単語の有無に変換してある．

```
> ### データの読み込み
> load("spam1.rda") # dat1.train,spam.train,dat1.test,spam.test
> dim(dat1.train) # 学習用データ n=3601, m=54
[1] 3601   54
> t(dat1.train[1:20,1:10]) # 最初の 20 個のメール，10 個の変数だけ表示
                   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Iword_freq_make    0 0 0 0 0 0 1 0 0  0  0  0  0  0  0  1  0  0  0  0
Iword_freq_address 0 1 0 0 0 0 1 0 0  0  0  0  0  0  0  0  1  0  0  0
Iword_freq_all     0 1 1 0 1 1 1 0 1  0  0  1  0  0  0  1  1  1  1  0
Iword_freq_3d      0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
Iword_freq_our     0 1 1 0 0 0 1 0 1  0  0  1  0  1  0  1  1  1  1  0
Iword_freq_over    0 1 0 0 0 0 1 0 0  0  1  1  0  0  0  0  0  1  1  0
Iword_freq_remove  0 1 0 0 1 0 0 0 0  0  0  1  0  0  0  1  1  1  0  0
Iword_freq_internet 0 1 1 0 0 1 1 0 0  0  0  0  0  0  0  0  0  1  0  0
Iword_freq_order   0 0 0 0 0 0 1 0 0  0  0  1  0  1  0  0  0  0  0  0
Iword_freq_mail    0 1 1 1 0 0 0 1 0  0  0  1  0  0  0  0  1  0  1  0
> spam.train[1:20] # spam=1, ham=0
 [1] 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 1 1 0 0
> dim(dat1.test) # テスト用データ
[1] 1000   54
> ### パラメタ theta の推定
> ## x : 0,1 の行列 （例：dat）
> ## y : 0,1 のベクタ （例：spam）
```

```
> myMLE <- function(x,y) {
+   py1 <- mean(y) # p(y=1)
+   px0 <- apply(x[y==0,],2,mean) # p(x[j]=1|y=0)
+   px1 <- apply(x[y==1,],2,mean) # p(x[j]=1|y=1)
+   list(py1=py1,px0=px0,px1=px1) # パラメタ theta を返す
+ }
> ### ベイズ事後確率 p(y=1|x) の計算
> ## th : list(py1,px0,px1)
> ## x : 0,1 の行列 (例：dat)
> myPP <- function(th,x) {
+   x <- as.matrix(x) # x がベクタのときも，形式的に行列（横ベクトル）にしておく
+   x0 <- x1 <- x
+   for(j in seq(ncol(x))) { # 単語 j=1,...,m のループ
+     a <- x[,j] + 1 # すべてのメール t=1,...,n について，単語 j がある=2, ない=1 のベクタ
+     x0[,j] <- c(1-th$px0[j],th$px0[j])[a] # p(x[j]|y=0)
+     x1[,j] <- c(1-th$px1[j],th$px1[j])[a] # p(x[j]|y=1)
+   }
+   p0 <- apply(x0,1,prod) # p(x|y=0)
+   p1 <- apply(x1,1,prod) # p(x|y=1)
+   th$py1*p1/(th$py1*p1 + (1-th$py1)*p0) # p(y=1|x)
+ }
> ### 判別結果のチェック
> myPPplot <- function(spam,pp,pth) {
+   ## プロットを２段にするおまじない．
+   def.par <- par(no.readonly = TRUE); on.exit(par(def.par))
+   layout(matrix(1:2,2,1))
+   ## spam/ham の論理型ベクタを準備して，判別確率を計算しておく
+   sp <- spam==1 # spam=TRUE, ham=FALSE のベクタ
+   p0 <- mean(pp[!sp]>pth) # ham メールを spam と判別する確率
```



```
+   p1 <- mean(pp[sp]>pth) # spam メールを spam と判別する確率
+   ## 上段プロットは ham メールという条件付での事後確率の分布
+   hist(pp[!sp],col="blue",nclass=50,prob=T,main="ham mails",
+        sub=paste("P(say spam | ham mail)=",round(p0,5)))
+   abline(v=pth,col="green")
+   ## 下段プロットは spam メールという条件付での事後確率の分布
+   hist(pp[sp],col="red",nclass=50,prob=T,main="spam mails",
+        sub=paste("P(say spam | spam mail)=",round(p1,5)))
+   abline(v=pth,col="green")
+   ## グラフに記入した判別確率を値として返す
+   ret <- c(pth,p0,p1)
+   names(ret) <- c("pth","p0","p1")
+   ret
+ }
> ### ここからデータ解析の開始．
> ## まず学習用データからパラメータの推定
> th <- myMLE(dat1.train,spam.train) # パラメタを th に保存
> names(th) # リスト成分の名前
[1] "py1" "px0" "px1"
> th$py1 # p(y=1)
[1] 0.4004443
> round(th$px0[1:10],3) # p(x[j]=1|y=0) 最初の１０個の単語だけ表示
    Iword_freq_make  Iword_freq_address      Iword_freq_all       Iword_freq_3d
              0.149               0.096               0.277               0.002
     Iword_freq_our     Iword_freq_over   Iword_freq_remove Iword_freq_internet
              0.224               0.117               0.014               0.077
   Iword_freq_order     Iword_freq_mail
              0.076               0.171
> round(th$px1[1:10],3) # p(x[j]=1|y=1)
```



```
    Iword_freq_make  Iword_freq_address      Iword_freq_all       Iword_freq_3d
              0.354               0.342               0.621               0.022
     Iword_freq_our     Iword_freq_over   Iword_freq_remove Iword_freq_internet
              0.618               0.383               0.426               0.342
   Iword_freq_order     Iword_freq_mail
              0.311               0.457
> ## 一応，学習用データでどのくらいうまく判別できるかチェック
> pp.train <- myPP(th,dat1.train) # 事後確率
> round(pp.train[1:20],3) # p(y=1|x) 最初の２０個だけ表示
    1     2     3     4     5     6     7     8     9    10    11    12    13
0.966 1.000 0.999 0.000 0.250 1.000 1.000 0.000 0.002 0.002 0.751 1.000 0.000
   14    15    16    17    18    19    20
0.667 0.000 1.000 1.000 1.000 0.000 0.001
> myPPplot(spam.train,pp.train,0.5) # 閾値 0.5 で判別（図は省略）
      pth        p0        p1
0.5000000 0.0676239 0.8196949
> ## 次にテスト用データでスパム判別と結果のチェック
> pp.test <- myPP(th,dat1.test) # 事後確率
> round(pp.test[1:20],3) # p(y=1|x) 最初の２０個だけ表示
    1     2     3     4     5     6     7     8     9    10    11    12    13
0.114 0.064 0.000 1.000 0.000 0.000 0.000 1.000 0.002 0.000 0.000 0.601 1.000
   14    15    16    17    18    19    20
0.000 0.000 0.000 1.000 1.000 1.000 0.000
> spam.test[1:20] # いままで隠しておいた「答え」
 [1] 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0
> myPPplot(spam.test,pp.test,0.5) # 閾値 0.5 で判別（図は省略）
       pth         p0         p1
0.50000000 0.06677266 0.81401617
> ## 学習用データで ham を spam と判別する確率を 0.01 になるように閾値を決める
```



```
> pth <- quantile(pp.train[spam.train==0],p=0.99)
> pth # もし p(y=1|x)>pth なら，spam と判定する．
      99%
0.9992213
> myPPplot(spam.train,pp.train,pth) # 閾値 pth で判別（左図）
       pth         p0         p1
0.99922133 0.01018990 0.59778086
> myPPplot(spam.test,pp.test,pth) # 閾値 pth で判別（右図）
      pth         p0         p1
0.9992213 0.0190779 0.5714286
```

手法として特別な工夫をしていないが，まあまあの判別結果．ここではあらかじめ選定した 54 単語しか使っていないし，各メールで単語の有無の情報しか使っていない．アプリケーションでは，どのような特徴量をつかうか，ということが判別の性能に強く影響する．

[課題 2.1] 例 2.5において $\boldsymbol{x}$ は m 次元ベクトル，各クラスのモデル $f_i(\boldsymbol{x};\boldsymbol{\theta}_i)$ が多変量正規分布 $\boldsymbol{x} \sim N_m(\boldsymbol{\mu}_i, \Sigma_i)$ とする．ただし $\boldsymbol{\theta}_i = (\boldsymbol{\mu}_i, \Sigma_i)$ である．クラス i とクラス j を比べると $\pi_i f_i(\boldsymbol{x};\boldsymbol{\theta}_i) > \pi_j f_i(\boldsymbol{x};\boldsymbol{\theta}_j)$ なら i のほうが j より良いが，この判別境界が $\boldsymbol{x}$ の２次式になることを示せ．ヒント： $S_i(\boldsymbol{x}) = \log(\pi_i f_i(\boldsymbol{x};\boldsymbol{\theta}_i))$ を計算する．

[課題 2.2] 課題 2.1において，もし i によらず $\Sigma_i = \Sigma$ ならば，判別境界が $\boldsymbol{x}$ の１次式になることを示せ．

[課題 2.3] 例 2.5において，パラメタ $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{k-1}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$ の値が既知とする．あらゆる判別ルールの中でベイズ法（事後確率最大の i を選ぶ）が正解率を最大にすることを示せ．ここで判別ルールとは，x

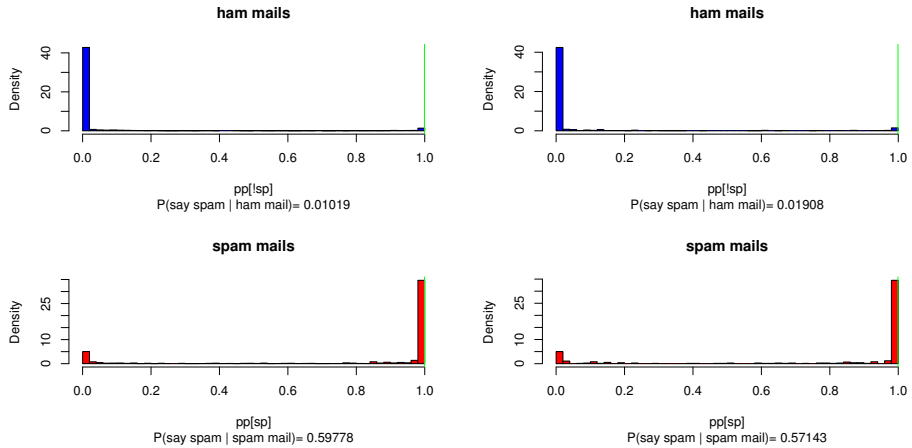


図 22　（左）学習用データにおける事後確率，（右）テスト用データにおける事後確率

がとりうる値の集合の分割 $R_1 \cup R_2 \cup \cdots \cup R_k$ として表現できて，$x \in R_i$ ならクラス i に属すると判断する．判別ルールを $\mathcal{R} = \{R_1, \ldots, R_k\}$ と標記すると，この正解率は

$$P(\mathcal{R}) = \sum_{i=1}^{k} \int_{R_i} \pi_i f_i(x; \boldsymbol{\theta}_i)\, dx$$

である．事後確率最大の判別ルールを $\mathcal{R}^* = \{R_1^*, \ldots, R_k^*\}$ と標記すると

$$x \in R_i^* \Rightarrow \pi_i f_i(x; \boldsymbol{\theta}_i) \geq \pi_j f_j(x; \boldsymbol{\theta}_j), j = 1, \ldots, k$$

である．このとき，$P(\mathcal{R}^*) \geq P(\mathcal{R})$ を示せばよい．

2.3　パラメタ推定

データ $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ からモデル $f(\boldsymbol{x}; \theta)$ のパラメタ θ を推定する．その推定値にはハットをつけて $\hat{\theta}$ と書く．$\hat{\theta}$ の計算のしかたに注目しているときは，推定量ということが多い．以下では，データ $\mathcal{X}$ とモデル $f(\boldsymbol{x}; \theta)$ を与えたとき，どのようにして $\hat{\theta}$ を計算するか，その方法の一般論を議論する．

[定義 2.4]　$(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ の同時密度関数を θ の関数とみなしたものを尤度（ゆうど）関数とよび次のように書く．

$$L(\boldsymbol{\theta}|\mathcal{X}) = f(\boldsymbol{x}_1; \boldsymbol{\theta}) \cdots f(\boldsymbol{x}_n; \boldsymbol{\theta})$$

[定義 2.5]　「θ は確率変数である」という前提で，なんらかの事前分布 $\pi(\theta)$ を与える．ベイズの定理より事後分布は $\pi(\boldsymbol{\theta}|\mathcal{X}) \propto L(\boldsymbol{\theta}|\mathcal{X})\pi(\boldsymbol{\theta})$ である．$\boldsymbol{\theta} \in \boldsymbol{\Theta}$ のうち $\pi(\boldsymbol{\theta}|\mathcal{X})$ を最大にする値を MAP **推定量（**Maximum A Posteriori estimator**）または最大事後確率推定量という．これを**

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \pi(\boldsymbol{\theta}|\mathcal{X})$$

と書く．

[課題 2.4]　$X_1, \ldots, X_n | \mu \sim N(\mu, 1)$ (i.i.d.), $\mu \sim N(0, \tau^2)$ とする．（μ は確率変数とその実現値を同じ文字で書いてしまっている．この記法を許すことにする）．このとき，以下を示せ．(i) $L(\mu|\mathcal{X}) \propto \exp(-\frac{n}{2}(\mu - \bar{x})^2)$**．ただし $\bar{x} = \sum_{t=1}^{n} x_t/n$．**(ii) μ の** MAP **推定量は $\hat{\mu}_{\mathrm{MAP}} = \bar{x} - \frac{\bar{x}}{1+n\tau^2}$．**

[定義 2.6]　$\boldsymbol{\theta} \in \boldsymbol{\Theta}$ のうち $L(\boldsymbol{\theta}|\mathcal{X})$ を最大にする値を最尤（さいゆう）推定量（Maximum Likelihood estimator**）という．これを**

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} L(\boldsymbol{\theta}|\mathcal{X})$$

と書く．MAP **推定量において形式的に $\pi(\theta) =$ 定数 と置いたものである．**

[課題 2.5]　$X_1, \ldots, X_n \sim N(\mu, \sigma^2)$ (i.i.d.) とする．このとき，$\boldsymbol{\theta} = (\mu, \sigma^2)$ の ML **推定量は $\hat{\mu}_{\mathrm{ML}} = \bar{x}$, $\hat{\sigma}^2_{\mathrm{ML}} = \sum_{t=1}^{n}(x_t - \bar{x})^2/n$ であることを示せ．**

[課題 2.6]　$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (i.i.d.) とする．このとき，$\boldsymbol{\theta} = (\boldsymbol{\theta}, \boldsymbol{\Sigma})$ の最尤推定量は $\hat{\boldsymbol{\mu}}_{\mathrm{ML}} = \bar{\boldsymbol{x}}$, $\hat{\boldsymbol{\Sigma}}_{\mathrm{ML}} = \sum_{t=1}^{n}(\boldsymbol{x}_t - \bar{\boldsymbol{x}})(\boldsymbol{x}_t - \bar{\boldsymbol{x}})'/n$ であることを示せ．

[課題 2.7]　k は 2 以上の整数する．$X \in \{1, 2, \ldots, k\}$ の離散分布を考え，パラメータを $\theta_i = P(X = i)$, $i = 1, \ldots, k-1$，$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{k-1})$ とする．ただし $P(X = k) = 1 - \sum_{i=1}^{k-1} \theta_i$．また $x_1, \ldots, x_n$ が i になった回数を z_i と書く．このとき最尤推定は $\hat{\theta}_i = z_i/n$ であることを示せ．ヒント：$(Z_1, \ldots, Z_k)$ は多項分布に従う．

[課題 2.8]　例 2.2において，データを (x_t, i_t), $t = 1, \ldots, n$ とする．つまり信号 X だけでなく信号源 i も観測できる場合を考える．このとき θ の最尤推定における各 θ_i 成分は $i = 1, \ldots, k$ で場合わけして最尤推定を行えばよいこと，および，π_i 成分は i の頻度でよいことを示せ．すなわち，

$$\hat{\boldsymbol{\theta}}_i = \arg\max_{\boldsymbol{\theta}_i} \sum_{t=1}^{n} I(i_t = i) \log f_i(x_t; \boldsymbol{\theta}_i), \quad \hat{\pi}_i = \frac{z_i}{n}$$

ただし $z_i = \sum_{t=1}^{n} I(i_t = i)$ とおく．とくに，各成分が正規分布の場合は，$\hat{\mu}_i = \sum_{t=1}^{n} I(i_t = i)x_t/z_i$，$\hat{\sigma}_i^2 = \sum_{t=1}^{n} I(i_t = i)(x_t - \hat{\mu}_i)^2/z_i$ となることを示せ．ヒント：ようするに，信号源の頻度は課題 2.7，各信

号源ごとの X は課題 2.5を適用するだけである．

[例 2.8]　今度は，i は観測できず，X だけが観測出来る場合を考える．例 2.3の混合モデル $f(x;\theta)$ に最尤法を適用する．考え方はシンプルだが，解析的に答えを求められず，数値的に計算する．例 2.6で生成した n_2 個の x_t だけから（i_t は見ないで）θ を推定する．つまりデータは $\mathcal{X}_2 = \{x_t, t = n_1+1, \ldots, n_1+n_2\}$ であり，対数尤度関数は

$$\log L(\boldsymbol{\theta}|\mathcal{X}_2) = \sum_{t=n_1+1}^{n_1+n_2} \log f(x_t;\boldsymbol{\theta}) = \sum_{t=n_1+1}^{n_1+n_2} \log \left(\sum_{i=1}^{k} f_i(x_t;\boldsymbol{\theta}_i)\pi_i \right)$$

である．これを最大化する．

```
> ## 正規混合分布の最尤推定
> ## データ: xx2
> ## 成分数: k
> ## 推定結果は，pr2,mu2,ss2 に保存
> ## まず対数尤度関数の定義
> mylik2 <- function(theta) {
+   pr <- theta[1:(k-1)]; mu <- theta[k:(2*k-1)]; ss <- theta[(2*k):(3*k-1)]
+   pr[k] <- 1 - sum(pr)
+   f <- drawnormmix(xx2,k,pr,mu,ss,FALSE)$f # データは xx2
+   -sum(log(f)) # 対数尤度関数*(-1)
+ }
> ## ここから最尤法の計算（optim による数値的最適化）
> th0 <- c(1/3,1/3,0,2,-2,1,1,1) # 初期値
> opt2 <- optim(th0,mylik2,method="BFGS",control=list(trace=1,reltol=1e-14),hessian=TRUE)
initial  value 1011.863187
iter  10 value 759.738582
```



```
iter  20 value 733.979179
iter  30 value 730.918755
iter  40 value 730.913498
final  value 730.913498
converged
> th2 <- opt2$par
> pr2 <- th2[1:(k-1)]; mu2 <- th2[k:(2*k-1)]; ss2 <- th2[(2*k):(3*k-1)]
> pr2[k] <- 1 - sum(pr2)
> a <- order(-pr2) # pr2 の要素の大きい順番
> pr2 <- pr2[a]; mu2 <- mu2[a]; ss2 <- ss2[a] # 並べ替え
> pr2 # 推定した pr
[1] 0.3853882 0.3283223 0.2862895
> mu2 # 推定した mu
[1]  0.06464457  3.91822918 -2.65935238
> sqrt(ss2) # 推定した sqrt(ss)
[1] 0.8274128 2.0351055 1.2760710
> hist(xx2,nclass=15,prob=T) # ヒストグラム
> rug(xx2) # データ点を横軸に線分で描く
> drawnormmix(x0,k,pr2,mu2,ss2,"blue") # 推定した密度関数
> lines(x0,f0,col="darkgreen",lty=2,lwd=2)
```

数値的最適化によって $\hat{\boldsymbol{\theta}} = (\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}, \hat{\mu}_1, \ldots, \hat{\mu}_k, \hat{\sigma}_1^2, \ldots, \hat{\sigma}_k^2)$ を得た．このパラメタ値を利用して，i の事後確率を計算する．このように，モデルのパラメタを最尤推定してからベイズ法を適用することを，「経験ベイズ法」という．

```
> ## i の事後確率を計算．ii2 を見ないで，xx2 だけから計算する
> a <- drawnormmix(xx2,k,pr2,mu2,ss2,FALSE) # n2 個の x だけから推定したパラメタ値を利用
> pr2x2 <- a$fi / a$f
> round(t(pr2x2[1:10,]),3) # 最初の 10 個（有効数字３桁，転置してから表示）
```



```
     [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8]  [,9] [,10]
[1,]    0 0.247 0.000 0.021 0.899 0.104 0.904    0 0.904 0.018
[2,]    0 0.750 0.001 0.978 0.074 0.011 0.056    1 0.061 0.005
[3,]    1 0.002 0.999 0.000 0.027 0.885 0.040    0 0.035 0.977
> ## クラス（信号源）の MAP 推定: ii2x2
> ii2x2 <- apply(pr2x2,1,function(p) order(-p)[1])
> ii2x2[1:30] # 推定された信号源（最初の３０個）
 [1] 3 2 3 2 1 3 1 2 1 3 1 3 1 1 1 2 2 2 2 2 1 2 2 3 1 1 2 2 2 3
> sum(ii2x2 == ii2)/length(ii2) # 正解率
[1] 0.8933333
```

あらかじめ教えたのは信号源の個数 k とデータ $x_1, \ldots, x_n$ だけである．それから信号源のパラメタ，および，各 x_t がどの信号源に属するかを自動的に推定できた．これは「教師無し学習」によるクラスタリングの一例である．もし各 x_t が属する信号源 i を教えてモデルのパラメタ推定をするならば「教師あり学習」と呼ばれる．

[課題 2.9]　さらに今度は，サンプルサイズ n_1 のデータ $\mathcal{X}_1 = \{(x_t, i_t), t = 1, \ldots, n_1\}$，およびサンプルサイズ n_2 のデータ $\mathcal{X}_2 = \{x_t, t = n_1+1, \ldots, n_1+n_2\}$ の両方をつかって最尤推定を行う．このとき対数尤度関数が次式で表されることを示せ．

$$\log L(\boldsymbol{\theta}|\mathcal{X}_1, \mathcal{X}_2) = \log L(\boldsymbol{\theta}|\mathcal{X}_1) + \log L(\boldsymbol{\theta}|\mathcal{X}_2) = \sum_{t=1}^{n_1} \log \left(f_{i_t}(x_t;\boldsymbol{\theta}_{i_t})\pi_{i_t} \right) + \sum_{t=n_1+1}^{n_1+n_2} \log \left(\sum_{i=1}^{k} f_i(x_t;\boldsymbol{\theta}_i)\pi_i \right)$$

[例 2.9]　課題 2.9の対数尤度関数を数値的に最大化する．$\mathcal{X}_1$ と $\mathcal{X}_2$ の両方をつかうので，それらを個別につかってパラメタ推定するよりも情報量が大きい．推定の性能（推定量 $\hat{\theta}$ の分散の大きさ）が理論上は向上する．もちろん，個別の事例において誤差を計算したとき，それが小さくなるとは限らない．シミュレーションで何回もデータを発生させて２乗誤差の期待値を計算したとき，それが小さくなるという意味である．

```
> ## 正規混合分布の最尤推定
> ## データ: xx1, ii1, xx2
> ## 成分数: k
> ## 推定結果は，pr3,mu3,ss3 に保存
> ## まず対数尤度関数の定義
> mylik3 <- function(theta) {
+   pr <- theta[1:(k-1)]; mu <- theta[k:(2*k-1)]; ss <- theta[(2*k):(3*k-1)]
+   pr[k] <- 1 - sum(pr)
+   ## まず xx1,ii1 の部分
+   fi <- drawnormmix(xx1,k,pr,mu,ss,FALSE)$fi
+   f1 <- fi[seq(n1) + (ii1-1)*n1]
+   ## 次に xx2 の部分
+   f2 <- drawnormmix(xx2,k,pr,mu,ss,FALSE)$f # データは xx2
+   ## 合計
+   -sum(log(c(f1,f2))) # 対数尤度関数*(-1)
+ }
> ## ここから最尤法の計算
> th0 <- c(1/3,1/3,0,2,-2,1,1,1) # 初期値
> opt3 <- optim(th0,mylik3,method="BFGS",control=list(trace=1,reltol=1e-14)) # 数値的最適化
initial  value 1372.011849
```

```
iter  10 value 1009.523874
iter  20 value 998.727891
iter  30 value 998.726049
iter  30 value 998.726049
iter  30 value 998.726049
final  value 998.726049
converged
> th3 <- opt3$par
> pr3 <- th3[1:(k-1)]; mu3 <- th3[k:(2*k-1)]; ss3 <- th3[(2*k):(3*k-1)]
> pr3[k] <- 1 - sum(pr3)
> pr3 # 推定した pr
[1] 0.4870557 0.3082942 0.2046501
> mu3 # 推定した mu
[1] -0.08596124  4.05636269 -3.11973790
> sqrt(ss3) # 推定した sqrt(ss)
[1] 1.060265 1.917332 1.032354
> hist(xx2,nclass=15,prob=T) # ヒストグラム
> rug(xx2) # データ点を横軸に線分で描く
> drawnormmix(x0,k,pr3,mu3,ss3,"orange") # 推定した密度関数
> lines(x0,f0,col="darkgreen",lty=2,lwd=2)
> ## i の事後確率を計算．ii2 を見ないで，xx2 だけから計算する
> a <- drawnormmix(xx2,k,pr3,mu3,ss3,FALSE) # n1 個の (x,i) と n2 個の x から推定
> pr2x3 <- a$fi / a$f
> round(t(pr2x3[1:10,]),3) # 最初の 10 個（有効数字３桁，転置してから表示）
     [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7] [,8]  [,9] [,10]
[1,]    0 0.445 0.003 0.105 0.941 0.469 0.955    0 0.951 0.180
[2,]    0 0.555 0.000 0.895 0.058 0.006 0.042    1 0.047 0.002
[3,]    1 0.000 0.997 0.000 0.001 0.525 0.003    0 0.002 0.818
> ## クラス（信号源）の MAP 推定: ii2x3
```

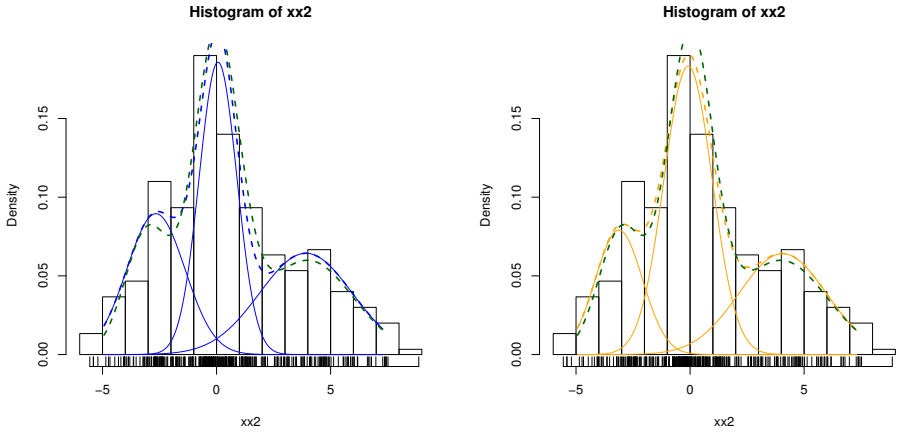



図 23　n_2 個の x_t だけから推定した密度関数（青）．n_1 個の (x_t, i_t) と n_2 個の x_t から推定した密度関数（オレンジ）．


```
> ii2x3 <- apply(pr2x3,1,function(p) order(-p)[1])
> ii2x3[1:30] # 推定された信号源（最初の３０個）
 [1] 3 2 3 2 1 3 1 2 1 3 1 3 1 1 1 2 2 2 2 2 1 2 2 3 1 1 2 2 2 3
> sum(ii2x3 == ii2)/length(ii2) # 正解率
[1] 0.9033333
> ### 誤差の比較（小さいほど良い）
> sum((c(pr1,mu1,ss1)-c(pr,mu,ss))^2) # n1 個の (x,i) から推定
[1] 0.5636234
> sum((c(pr2,mu2,ss2)-c(pr,mu,ss))^2) # n2 個の x から推定
[1] 0.6626585
> sum((c(pr3,mu3,ss3)-c(pr,mu,ss))^2) # n1 個の (x,i) と n2 個の x から推定
[1] 0.1497713
```

[注意]　例 2.8と例 2.9では，対数尤度関数を数値的に最大化した．この数値的最適化には R の組み込み関数 optim を用いた．これに実装されているアルゴリズムは汎用で十分性能の高いものであるけれど，パラメタの次元が大きくなってくると不安定になりやすい．実際，例 2.8や例 2.9でも，データや初期値によっては不適解が得られる．混合分布のモデルの最尤推定では，もっと実装が簡単で安定な方法が知られている．これは EM アルゴリズムと呼ばれる．

2.4　EM アルゴリズム

[例 2.10]　一般論はあとで説明することにして，例 2.9の計算をこのアルゴリズムでやりなおしてみる．反復の添え字を $r = 0, 1, 2, \ldots$ とする．$r = 0$ を初期値として，現在のパラメタ値 $\theta^{(r)}$ から次のパラメタ値 $\theta^{(r+1)}$

への更新ルールを次で定める．以下，$n = n_1 + n_2$ とおく．

- **E ステップ (Expectation step)**．とりあえず $\theta^{(r)}$ を信用して，$t = 1, \ldots, n_1 + n_2$ について i の事後確率 $p(i_t|x_t) = \pi_i(x_t; \theta^{(r)})$ を計算する．

$$\pi_i(x_t; \theta^{(r)}) = I(i = i_t), \quad t = 1, \ldots, n_1 \quad (\mathcal{X}_1 \text{ では } i_t \text{ を観測している})$$

$$\pi_i(x_t; \theta^{(r)}) = \frac{f_i(x_t; \theta_i^{(r)})\pi_i^{(r)}}{f(x_t; \theta^{(r)})}, \quad t = n_1 + 1, \ldots, n$$

- **M ステップ (Maximization step)**．$\pi_1, \ldots, \pi_k$ を事後確率の平均値で推定する (ただし $\sum_{i=1} \pi_i = 1$)．

$$\pi_i^{(r+1)} = \frac{1}{n} \sum_{t=1}^{n} \pi_i(x_t; \theta^{(r)})$$

各サンプル x_t の「重み」を次式で計算する．

$$w_t = \frac{\pi_i(x_t; \theta^{(r)})}{\sum_{t'=1}^{n} \pi_i(x_{t'}; \theta^{(r)})}, \quad t = 1, \ldots, n$$

そして，各成分 $i = 1, \ldots, k$ で場合分けして重みつきの最尤推定を行う．正規混合分布では次式で推定する．

$$\mu_i^{(r+1)} = \sum_{t=1}^{n} w_t x_t, \quad t = 1, \ldots, n$$

$$\sigma_i^{2\,(r+1)} = \sum_{t=1}^{n} w_t (x_t - \mu_i^{(r+1)})^2, \quad t = 1, \ldots, n$$

上記の E ステップと M ステップを反復するだけである．計算はきわめて単純であるが，以下の実行結果を見ると，optim によって得られた結果と等価なものが得られている．また，反復のたびに対数尤度がかならず増加している．なお，反復計算そのものには，対数尤度の $\log L(\boldsymbol{\theta}^{(r)}|\mathcal{X}_1, \mathcal{X}_2)$ を計算する必要さえない．

```
> ## EM アルゴリズムによる正規混合分布の最尤推定
> ## データ: xx1, ii1, xx2
> ## 成分数: k
> ## 推定結果は，pr4,mu4,ss4 に保存
> pr4 <- c(1/3,1/3,1/3); mu4 <- c(0,2,-2); ss4 <- c(1,1,1)  # 初期値
> nr <- 30 # とりあえず反復回数をきめておく
> mystat <- function(pr,mu,ss) { # 反復の途中経過を表示する関数を準備しておく
+   lik <- mylik3(c(pr[-k],mu,ss)) # 目的関数
+   cat(format(lik,digits=10),round(c(pr,mu,ss),3),"\n") # 一行でサマリを表示
+   c(lik,pr,mu,ss)
+ }
> stat3 <- mystat(pr3,mu3,ss3) # 参考のため，optim で mylik3 を最適化した結果を表示しておく
998.7260493 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
> stat4 <- matrix(0,1+nr,length(stat3)) # 途中結果を保存する場所を確保
> stat4[1,] <- mystat(pr4,mu4,ss4) # 初期値における値
```



```
1372.011849 0.333 0.333 0.333 0 2 -2 1 1 1
> xx <- c(xx1,xx2) # 推定に利用するデータを結合しておく
> pp1 <- matrix(0,n1,k); pp1[seq(n1)+(ii1-1)*n1] <- 1 # xx1 における「事後確率」
> t(pp1[1:10,]) # 最初の１０個のメール
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    0    1    1    1    0    1    0     1
[2,]    0    0    1    0    0    0    0    0    1     0
[3,]    0    0    0    0    0    0    1    0    0     0
> ## EM アルゴリズムの反復計算はここから
> for(r in 1:nr) { # 反復計算:本当は収束判定して break するべき．
+   a <- drawnormmix(xx2,k,pr4,mu4,ss4,FALSE); pp2 <- a$fi/a$f # xx2 における事後確率
+   pp <- rbind(pp1,pp2) # xx1 と xx2 の事後確率を結合しておく
+   pr4 <- apply(pp,2,sum)/(n1+n2) # pr の推定
+ #  wt <- pp/rep(apply(pp,2,sum),rep(n1+n2,k)) #「重み」を事後確率で定義
+   wt <- sweep(pp,2,apply(pp,2,sum),"/") # 上記と同じだけど sweep 関数を使ってみた
+   mu4 <- apply(xx*wt,2,sum) # mu の推定
+ #  ss4 <- apply((xx-rep(mu4,rep(n1+n2,k)))^2*wt,2,sum) # ss の推定
+   ss4 <- apply(sweep(matrix(xx,n1+n2,k),2,mu4,"-")^2*wt,2,sum) # 上記と同じ
+   cat(r,": ")
+   stat4[r+1,] <- mystat(pr4,mu4,ss4) # 目的関数を保存しておく（反復計算には必要ない）
+ }
1 : 1009.451812 0.352 0.364 0.283 -0.056 3.607 -2.558 0.883 4.279 1.776
2 : 1004.582255 0.399 0.343 0.257 -0.034 3.733 -2.706 0.916 4.368 1.691
3 : 1002.053705 0.425 0.332 0.243 -0.034 3.825 -2.819 0.955 4.215 1.51
4 : 1000.541512 0.442 0.325 0.233 -0.04 3.892 -2.905 0.985 4.062 1.363
5 : 999.6855876 0.454 0.32 0.226 -0.047 3.94 -2.968 1.009 3.945 1.26
6 : 999.2271227 0.463 0.317 0.22 -0.055 3.975 -3.013 1.03 3.861 1.191
7 : 998.9887642 0.47 0.314 0.216 -0.062 3.999 -3.044 1.048 3.803 1.147
8 : 998.865351 0.474 0.313 0.213 -0.068 4.015 -3.066 1.064 3.763 1.119
```



```
9 : 998.8007836 0.478 0.311 0.211 -0.072 4.027 -3.081 1.077 3.737 1.101
10 : 998.766521 0.48 0.311 0.209 -0.076 4.035 -3.092 1.088 3.718 1.09
11 : 998.748113 0.482 0.31 0.208 -0.078 4.041 -3.1 1.097 3.706 1.083
12 : 998.738132 0.483 0.31 0.207 -0.08 4.045 -3.105 1.103 3.697 1.078
13 : 998.7326863 0.484 0.309 0.207 -0.082 4.048 -3.109 1.109 3.691 1.074
14 : 998.7297025 0.485 0.309 0.206 -0.083 4.05 -3.112 1.112 3.687 1.072
15 : 998.728063 0.486 0.309 0.206 -0.084 4.052 -3.114 1.115 3.684 1.07
16 : 998.7271603 0.486 0.309 0.205 -0.084 4.053 -3.115 1.118 3.682 1.069
17 : 998.7266627 0.486 0.309 0.205 -0.085 4.054 -3.116 1.119 3.68 1.068
18 : 998.7263881 0.486 0.308 0.205 -0.085 4.055 -3.117 1.121 3.679 1.068
19 : 998.7262365 0.487 0.308 0.205 -0.085 4.055 -3.118 1.121 3.678 1.067
20 : 998.7261528 0.487 0.308 0.205 -0.085 4.055 -3.118 1.122 3.678 1.067
21 : 998.7261065 0.487 0.308 0.205 -0.086 4.056 -3.119 1.123 3.677 1.066
22 : 998.726081 0.487 0.308 0.205 -0.086 4.056 -3.119 1.123 3.677 1.066
23 : 998.7260668 0.487 0.308 0.205 -0.086 4.056 -3.119 1.123 3.677 1.066
24 : 998.726059 0.487 0.308 0.205 -0.086 4.056 -3.119 1.124 3.677 1.066
25 : 998.7260546 0.487 0.308 0.205 -0.086 4.056 -3.119 1.124 3.677 1.066
26 : 998.7260522 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
27 : 998.726051 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
28 : 998.7260502 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
29 : 998.7260498 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
30 : 998.7260496 0.487 0.308 0.205 -0.086 4.056 -3.12 1.124 3.676 1.066
> pr4 # 推定した pr
[1] 0.4870391 0.3082993 0.2046617
> mu4 # 推定した mu
[1] -0.08593303  4.05631674 -3.11967172
> sqrt(ss4) # 推定した sqrt(ss)
[1] 1.060217 1.917354 1.032377
```



```
> plot(0:nr,stat4[,1],type="b",xlab="iteration",ylab="lik") # 目的関数のグラフ
> abline(h=stat3[1],lty=2,col="pink") # optim の結果を赤線で表示
> matplot(0:nr,stat4[,-1],type="b",xlab="iteration",ylab="parameters") # パラメタ推定値の変化
> abline(h=stat3[-1],lty=2,col="pink") # optim の結果を赤線で表示
```

[定義 2.7]　観測したデータを $\mathcal{X}$，未観測データを $\mathcal{Y}$ で表す（対応する確率変数も同じ文字で書く）．課題 2.9では，$\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2\} = \{x_1, \ldots, x_{n_1+n_2}, i_1, \ldots, n_{n_1}\}$，$\mathcal{Y} = \{i_{n_1+1}, \ldots, i_{n_1+n_2}\}$ である．$(\mathcal{X}, \mathcal{Y})$ の同時分布を表す尤度を $L(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})$，$\mathcal{X}$ の周辺分布を表す尤度を $L(\boldsymbol{\theta}|\mathcal{X}) = \int L(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})\, d\mathcal{Y}$ で表す．ただし積分は $\mathcal{Y}$ の取りうるすべての値について取る（課題 2.9では離散分布なので積分は和になる）．$L(\boldsymbol{\theta}|\mathcal{X})$ を最大化するための EM アルゴリズムでは，$r = 0$ を初期値として，現在のパラメタ値 $\boldsymbol{\theta}^{(r)}$ から次のパラメタ値 $\boldsymbol{\theta}^{(r+1)}$ への更新ルールを次で定める．

- E ステップ (Expectation step)．とりあえず $\boldsymbol{\theta}^{(r)}$ を信用して，$\mathcal{X}$ を与えたときの $\mathcal{Y}$ の条件付分布 $f(\mathcal{Y}|\mathcal{X}; \boldsymbol{\theta}^{(r)})$ を計算する．

$$f(\mathcal{Y}|\mathcal{X}; \boldsymbol{\theta}^{(r)}) = \frac{L(\boldsymbol{\theta}^{(r)}|\mathcal{X}, \mathcal{Y})}{L(\boldsymbol{\theta}^{(r)}|\mathcal{X})}$$

この条件付分布（つまり $\mathcal{Y}$ の事後確率）に関して，$\log(L(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y}))$ の期待値を定義する．

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}) = \int \log(L(\boldsymbol{\theta}|\mathcal{X}, \mathcal{Y})) f(\mathcal{Y}|\mathcal{X}; \boldsymbol{\theta}^{(r)})\, d\mathcal{Y}$$

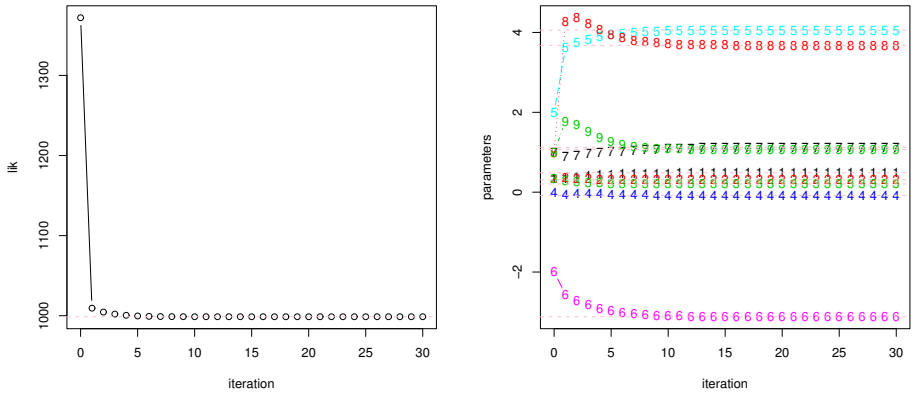


図 24　（左）$-\log L(\boldsymbol{\theta}^{(r)}|\mathcal{X}_1,\mathcal{X}_2)$, $r = 0,\ldots,30$ のプロット．実用的には最初の 10 回程度の反復で収束した．optim による値はピンクの線で示した．（右）パラメタ $\boldsymbol{\theta}^{(r)} = (\pi_1^{(r)},\pi_2^{(r)},\mu_1^{(r)},\mu_2^{(r)},\mu_3^{(r)},\sigma_1^{2\,(r)},\sigma_2^{2\,(r)},\sigma_3^{2\,(r)})$ のプロット

- **M ステップ (Maximization step)．$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)})$ を $\boldsymbol{\theta}$ の関数とみなし最大化する．**

$$\boldsymbol{\theta}^{(r+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)})$$

これを反復すると，$L(\boldsymbol{\theta}^{(r+1)}|\mathcal{X}) \geq L(\boldsymbol{\theta}^{(r)}|\mathcal{X})$ である．

[証明]　$L(\boldsymbol{\theta}|\mathcal{X}) = L(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y})/f(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta})$ **の対数をとると**

$$\log L(\boldsymbol{\theta}|\mathcal{X}) = \log L(\boldsymbol{\theta}|\mathcal{X},\mathcal{Y}) - \log f(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta})$$

この両辺を条件付分布 $f(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta}^{(r)})$ に関して期待値を計算すると左辺はそのままなので，

$$\log L(\boldsymbol{\theta}|\mathcal{X}) = Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)}) - H(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)})$$

ただし

$$H(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)}) = \int \log(f(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta}))f(\mathcal{Y}|\mathcal{X};\boldsymbol{\theta}^{(r)})\,d\mathcal{Y}$$

とおく．一般に任意の $\boldsymbol{\theta}$ で $H(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)}) \leq H(\boldsymbol{\theta}^{(r)},\boldsymbol{\theta}^{(r)})$ であることに注意すると，$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(r)}) \geq Q(\boldsymbol{\theta}^{(r)},\boldsymbol{\theta}^{(r)})$ となるように $\boldsymbol{\theta}$ を選びさえすれば，$\log L(\boldsymbol{\theta}|\mathcal{X}) \geq \log L(\boldsymbol{\theta}^{(r)}|\mathcal{X})$ であることが分かる．

[注意]　(i) **EM アルゴリズムは $\log L(\boldsymbol{\theta}|\mathcal{X})$ を増加させるが，その最大値に収束するとは言えない．適当な条件下で極大値に収束することが示されている．この問題を回避する現実的な方法は，いくつかの初期値から EM アルゴリズムを実行して，そのなかで尤度を最大にするものが選ぶ．**(ii) **正規混合モデルでは，もし μ_1**

をデータのどれかの点 x_t に一致させると $\log L(\boldsymbol{\theta}|\mathcal{X})$ は無限大になる．つまり最大化は意味をなさず，極大解の中で尤度を最大にするものを求めることが目的になる．

[課題 2.10]　**密度関数 $f(x)$ と $g(x)$ が $-\infty < x < \infty$ で $f(x) > 0$, $g(x) > 0$ とする．このとき**

$$\int_{-\infty}^{\infty} \log(g(x))f(x)\,dx \leq \int_{-\infty}^{\infty} \log(f(x))f(x)\,dx$$

を示せ．

[課題 2.11]　**定義 2.7の EM アルゴリズムを正規混合モデルに適用すると例 2.10のアルゴリズムが得られることを示せ．**

2.5　最尤推定量の性質

ここでの議論は次を仮定する．データ $\mathcal{X} = \{x_1,x_2,\ldots,x_n\}$ は

$$X_1,X_2,\ldots,X_n \sim g(x) \quad \text{(i.i.d.)}$$

の実現値．真の分布 $g(x)$ を近似するモデルは $f(x;\boldsymbol{\theta})$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$．パラメタは m 次元とし，$\boldsymbol{\theta} = (\theta_1,\ldots,\theta_m)'$ と書く（列ベクトル）．モデル $f(x;\boldsymbol{\theta})$ が正しい，つまり $g(x) \equiv f(x;\boldsymbol{\theta})$ と仮定して議論を進める．$X_1,\ldots,X_n$ または X に関する期待値を $E_{\boldsymbol{\theta}}(\cdot)$ で表す．対応する分散や分散共分散行列は $V_{\boldsymbol{\theta}}(\cdot)$，共分散は $C_{\boldsymbol{\theta}}(\cdot)$ と書く．尤度関数は $L(\boldsymbol{\theta}|\mathcal{X}) = f(x_1,\ldots,x_n;\boldsymbol{\theta}) = f(x_1;\boldsymbol{\theta})f(x_2;\boldsymbol{\theta})\cdots f(x_n;\boldsymbol{\theta})$ である．これを最大にする最尤推定

量を $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ と書く．データの関数であることを明示するときは $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}(\mathcal{X})$ または $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}(x_1,\ldots,x_n)$ と書く．最尤推定量に限らず，任意の推定量を $\hat{\theta}$ によって表す．

[定義 2.8]　**モデル $f(x;\boldsymbol{\theta})$ が正しいとする．推定量 $\hat{\boldsymbol{\theta}}$ が不偏 (unbiased) であるとは**

$$E_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}(X_1,\ldots,X_n)) = \boldsymbol{\theta}$$

を満たすことである．

[定理 2.1]

不偏推定量 $\hat{\boldsymbol{\theta}}$ の分散共分散行列は次式を満たす．

$$V_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}(X_1,\ldots,X_n)) \geq \tfrac{1}{n}G(\boldsymbol{\theta})^{-1} \tag{2.1}$$

これは推定量の性能限界を表しており，クラメール・ラオの不等式という．ただし，行列 $G(\boldsymbol{\theta})$ が退化していないことを仮定している．対称行列 A,B について $A \geq B$ とは，$A-B$ が非負正定値 (non-negative definite) のことであり，$m \times m$ 行列 $G(\boldsymbol{\theta})$ の成分は次式で定義する．

$$G_{ij}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\left\{\frac{\partial \log f(X;\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial \log f(X;\boldsymbol{\theta})}{\partial\theta_j}\right\}$$

この $G(\boldsymbol{\theta})$ は Fisher 情報行列と呼ばれる．

[注意]　$G(\boldsymbol{\theta})$ はサンプル X_t １個あたりの情報量を表す．サンプルサイズ n のデータ全体の情報量は $nG(\boldsymbol{\theta})$ であり，これも Fisher 情報行列と呼ぶ．課題 2.14で示すように次式で定義してもよい．

$$nG(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\left(-\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right)$$

[証明]　$E_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}(X_1,\ldots,X_n)) = \boldsymbol{\theta}$ を成分で書くと

$$\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\hat{\theta}_i(x_1,\ldots,x_n)f(x_1,\ldots,x_n;\boldsymbol{\theta})\,dx_1\cdots dx_n = \theta_i,\quad i=1,\ldots,m$$

両辺を θ_j で微分すると

$$\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\hat{\theta}_i(x_1,\ldots,x_n)\frac{\partial \log f(x_1,\ldots,x_n;\boldsymbol{\theta})}{\partial\theta_j}f(x_1,\ldots,x_n;\boldsymbol{\theta})\,dx_1\cdots dx_n = \frac{\partial\theta_i}{\partial\theta_j}$$

である．m 次元の列ベクトル $\boldsymbol{S}(x_1,\ldots,x_n;\boldsymbol{\theta})$ の成分を

$$S_j(x_1,\ldots,x_n;\boldsymbol{\theta}) = \frac{\partial \log f(x_1,\ldots,x_n;\boldsymbol{\theta})}{\partial\theta_j},\quad j=1,\ldots,m$$

で定義すれば，$\mathcal{X} = (X_1,\ldots,X_n)$ と書くと

$$E_{\boldsymbol{\theta}}\left\{\hat{\boldsymbol{\theta}}(\mathcal{X})\boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})'\right\} = I_m \tag{2.2}$$

である．一方，上式の導出で形式的に $\hat{\theta}_i = \theta_i = 1$ とおけば分かるように

$$E_{\boldsymbol{\theta}}\left\{\boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})\right\} = \boldsymbol{0} \tag{2.3}$$

が常に成り立つ．(2.2) と (2.3) をまとめると，

$$C_{\boldsymbol{\theta}}\left\{\hat{\boldsymbol{\theta}}(\mathcal{X}),\boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})\right\} = I_m \tag{2.4}$$

と書いても良い．したがって，

$$V_{\boldsymbol{\theta}}\left\{\begin{bmatrix}\hat{\boldsymbol{\theta}}(\mathcal{X})\\ \boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})\end{bmatrix}\right\} = \begin{bmatrix}V_{\boldsymbol{\theta}}\{\hat{\boldsymbol{\theta}}(\mathcal{X})\} & I_m\\ I_m & V_{\boldsymbol{\theta}}\{\boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})\}\end{bmatrix}$$

以下，$A = V_{\boldsymbol{\theta}}\{\hat{\boldsymbol{\theta}}(\mathcal{X})\}$, $B = V_{\boldsymbol{\theta}}\{\boldsymbol{S}(\mathcal{X};\boldsymbol{\theta})\}$ と書く．分散共分散行列は一般に非負正定値であるから，上式の両辺の２次形式を計算すると常に非負となる．つまり任意の m 次元ベクトル $\boldsymbol{a}$, $\boldsymbol{b}$ をつかって

$$\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}\end{bmatrix}'\begin{bmatrix}A & I_m\\ I_m & B\end{bmatrix}\begin{bmatrix}\boldsymbol{a}\\ \boldsymbol{b}\end{bmatrix} = \boldsymbol{a}'A\boldsymbol{a} + 2\boldsymbol{a}'\boldsymbol{b} + \boldsymbol{b}'B\boldsymbol{b} \geq 0$$

である．とくに，$\boldsymbol{b} = -B^{-1}\boldsymbol{a}$ とおけば，

$$\boldsymbol{a}'A\boldsymbol{a} - 2\boldsymbol{a}'B^{-1}\boldsymbol{a} + \boldsymbol{a}'B^{-1}\boldsymbol{a} = \boldsymbol{a}'(A-B^{-1})\boldsymbol{a} \geq 0$$

であるから，$A \geq B^{-1}$ が示せた．データが i.i.d. であることより，$B = nG$ であるから，(2.1) が示せたことになる．

[課題 2.12]　次式を示せ．

$$E_{\boldsymbol{\theta}}\left\{\frac{\partial \log f(X;\boldsymbol{\theta})}{\partial\theta_i}\right\} = 0,\quad i=1,\ldots,m$$

[課題 2.13]　分散共分散行列が一般に非負正定値であることを示せ．

[定理 2.2]　十分に n が大きいとき，最尤推定量 $\boldsymbol{\theta}_{\mathrm{ML}}$ は近似的に，平均 $\boldsymbol{\theta}$，分散共分散行列 $\frac{1}{n}G(\boldsymbol{\theta})^{-1}$ の正規分布に従う．すなわち，

$$\sqrt{n}(\hat{\boldsymbol{\theta}}(X_1,\ldots,X_n) - \boldsymbol{\theta}) \xrightarrow{d} N(0, G(\boldsymbol{\theta})^{-1}) \tag{2.5}$$

[注意]　つまり，データのサンプルサイズが十分に大きければ，最尤推定量はクラメール・ラオの不等式で示されている性能限界を近似的に達成していること意味する．なお，この定理が成立するためには，いろいろ細かい条件が必要であるが，それについては議論しない．以下では形式的な証明を与える．

[証明]　最尤推定量は対数尤度を最大化するので，$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ が Θ の内点であると仮定すれば次式を満たす

$$\left.\frac{\partial \log f(x_1,\ldots,x_n;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \boldsymbol{0}$$

これを $\boldsymbol{\theta}$ の周りでテーラー展開すると，

$$\frac{\partial \log f(x_1,\ldots,x_n;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} + \frac{\partial^2 \log f(x_1,\ldots,x_n;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}(\hat{\boldsymbol{\theta}}_{\mathrm{ML}} - \boldsymbol{\theta}) + O(\|\hat{\boldsymbol{\theta}}_{\mathrm{ML}} - \boldsymbol{\theta}\|^2) = \boldsymbol{0}$$

両辺を $\sqrt{n}$ で割ると，データが i.i.d. であることから，次のように書き換えられる．

$$\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\frac{\partial \log f(x_t;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} - \hat{G}(\boldsymbol{\theta})\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{ML}} - \boldsymbol{\theta}) \xrightarrow{p} \boldsymbol{0}$$

ただし

$$\hat{G}(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{t=1}^{n}\frac{\partial^2 \log f(x_t;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} \xrightarrow{p} E_{\boldsymbol{\theta}}\left\{-\frac{\partial^2 \log f(X;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right\} = G(\boldsymbol{\theta})$$

とおく（課題 2.14参照）．上記をまとめると，

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{\mathrm{ML}} - \boldsymbol{\theta}) \xrightarrow{p} G(\boldsymbol{\theta})^{-1}\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\frac{\partial \log f(x_t;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$$

ところで中心極限定理より

$$\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\frac{\partial \log f(x_t;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \xrightarrow{d} N(\boldsymbol{0}, G(\boldsymbol{\theta}))$$

であるから $G(\boldsymbol{\theta})^{-1}G(\boldsymbol{\theta})G(\boldsymbol{\theta})^{-1} = G(\boldsymbol{\theta})^{-1}$ より (2.5) が示せた．

[課題 2.14]　次式を示せ．

$$E_{\boldsymbol{\theta}}\left\{-\frac{\partial^2 \log f(X;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right\} = E_{\boldsymbol{\theta}}\left\{\frac{\partial \log f(X;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\frac{\partial \log f(X;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}'}\right\}$$

したがって，最尤推定量 $\hat{\theta}$ の分散共分散行列は次式で推定できる．

$$\hat{V}(\hat{\boldsymbol{\theta}}) = \frac{1}{n}\left[E_{\hat{\boldsymbol{\theta}}}\left\{-\frac{\partial^2 \log f(X;\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right\}\right]^{-1} \approx \left[-\frac{\partial^2 \log L(\boldsymbol{\theta}|\mathcal{X})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\Big|_{\hat{\boldsymbol{\theta}}}\right]^{-1} \tag{2.6}$$

[課題 2.15]　X_t は 0 か 1 の 2 値をとり，$P(X_t = 1) = \pi$，$P(X_t = 0) = 1 - \pi$ (i.i.d.) とする（ベルヌーイ試行）．最尤推定量が $\hat{\pi} = \bar{x}$ であることを示せ．フィッシャー情報量が次式であることを示せ．

$$E\left(-\frac{d^2 \log L}{d\pi^2}\right) = \frac{n}{\pi(1-\pi)}$$

[課題 2.16]　$X_t \sim N(\mu, \sigma^2)$ (i.i.d.) とする．最尤推定量が $\hat{\mu} = \bar{x}$，$\hat{\sigma}^2 = \sum_{t=1}^{n}(x_t - \bar{x})^2/n$ であることを示せ．$\boldsymbol{\theta} = (\mu, \sigma^2)$ としたとき，フィッシャー情報行列が次式であることを示せ．

$$E\left(-\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right) = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{bmatrix}$$

[注意]　上記 2 例（ベルヌーイ試行と正規分布）では，(2.6) の二つの分散推定量は等価になっている．

$$E\left(-\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right)\Big|_{\hat{\boldsymbol{\theta}}} = -\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\Big|_{\hat{\boldsymbol{\theta}}}$$

[例 2.11]　例 2.8の最尤推定の結果は opt2 に保存してある．

```
> opt2$par # 最尤推定 theta=(pi1,pi2,mu1,mu2,mu3,ss1,ss2,ss3)
[1]  0.38538824  0.32832228  0.06464457  3.91822918 -2.65935238  0.68461192
[7]  4.14165445  1.62835724
> round(opt2$hessian,4) # 目的関数の 2 階微分
          [,1]      [,2]     [,3]     [,4]     [,5]     [,6]    [,7]    [,8]
[1,] 1269.3400  858.4087  65.2869 -19.2837  86.3077 -86.1234  3.1647 31.1418
[2,]  858.4087 1657.0568 107.3133  33.9383  53.6566  -3.7313 -9.7071 19.9189
[3,]   65.2869  107.3133  92.8453  -8.0421 -12.9578   1.5661  0.1017 -0.1137
[4,]  -19.2837   33.9383  -8.0421  15.5233  -0.8769  -4.2959  2.5846 -0.3016
[5,]   86.3077   53.6566 -12.9578  -0.8769  33.8974   2.9087  0.3394 -9.1353
[6,] -86.1234   -3.7313   1.5661  -4.2959   2.9087  43.9240 -0.3447 -3.3270
[7,]   3.1647   -9.7071   0.1017   2.5846   0.3394  -0.3447  1.7760  0.0395
[8,]  31.1418   19.9189  -0.1137  -0.3016  -9.1353  -3.3270  0.0395  9.8720
> round(solve(opt2$hessian),4) # 分散共分散行列の推定
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]
[1,]  0.0062 -0.0032 -0.0011  0.0263 -0.0176  0.0134 -0.0601 -0.0240
[2,] -0.0032  0.0032 -0.0032 -0.0238  0.0017 -0.0077  0.0560  0.0015
[3,] -0.0011 -0.0032  0.0219  0.0305  0.0277  0.0002 -0.0673  0.0371
[4,]  0.0263 -0.0238  0.0305  0.2781 -0.0144  0.0712 -0.5669 -0.0131
[5,] -0.0176  0.0017  0.0277 -0.0144  0.1273 -0.0330  0.0257  0.1587
[6,]  0.0134 -0.0077  0.0002  0.0712 -0.0330  0.0536 -0.1521 -0.0366
[7,] -0.0601  0.0560 -0.0673 -0.5669  0.0257 -0.1521  1.7700  0.0239
[8,] -0.0240  0.0015  0.0371 -0.0131  0.1587 -0.0366  0.0239  0.3085
> sqrt(diag(solve(opt2$hessian))) # 最尤推定の標準誤差
[1] 0.07893068 0.05632762 0.14791642 0.52736033 0.35678083 0.23160625 1.33043134
[8] 0.55540885
```


2.6　検定と信頼区間

前節の結果をふまえ，ここでは近似的に次式がなりたつと仮定して話を進める．

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \boldsymbol{V}) \tag{2.7}$$

ここで $\boldsymbol{V} = (nG(\boldsymbol{\theta}))^{-1}$ をデータから推定したものを $\hat{\boldsymbol{V}}$ と書く．たとえば，

$$\hat{\boldsymbol{V}} = \left[-\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\Big|_{\hat{\boldsymbol{\theta}}}\right]^{-1}$$

とくに，$\boldsymbol{\theta}$ の第 i 成分に関しては，$\hat{\theta}_i \sim N(\theta_i, v_{ii})$ となり，v_{ii} の推定は $\hat{v}_{ii}$ である．

[課題 2.17]　$\hat{\theta} \sim N(\theta, v)$ として，$v > 0$ は既知と仮定する．帰無仮説: $\theta = 0$ を対立仮説: $\theta \neq 0$ に対して，有意水準 α で検定する．棄却域を $|\hat{\theta}| > c$ の形であたえるとき，

$$c = \sqrt{v}\Phi^{-1}(1 - \alpha/2)$$

となることを示せ．ヒント: $P(|\hat{\theta}| > c|\theta = 0) = \alpha$ となる c を求める．参考: $\Phi^{-1}(0.975) = 1.96 \approx 2$．

[課題 2.18]　課題 2.17で得られた検定について，$\hat{\theta}$ を観測したときの p-値が，

$$p = 2 \times \left[1 - \Phi\left(\frac{|\hat{\theta}|}{\sqrt{v}}\right)\right] \tag{2.8}$$

で与えられることを示せ．ヒント： $\hat{\theta} = \theta \pm c$ となるときの α の値が p-値（α を調節してこの等式が成り立つようにする）．

[課題 2.19]　定数 b をひとつ決める．課題 2.17で帰無仮説を $\theta = b$，対立仮説を $\theta \neq b$ と変更すると，p-値が

$$p(b) = 2 \times \left[1 - \Phi\left(\frac{|\hat{\theta} - b|}{\sqrt{v}}\right)\right]$$

で与えられることを示せ．これを利用して，信頼度 $1 - \alpha$ で θ の信頼区間が，

$$[\hat{\theta} - c, \hat{\theta} + c]$$

で与えられることを示せ．ヒント: $p(b) \geq \alpha$ となる b の集合を求める．

[課題 2.20]　課題 2.19で求めた信頼区間の被覆確率が $1 - \alpha$ であることを示せ．すなわち，

$$P(\theta \in [\hat{\theta} - c, \hat{\theta} + c]) = 1 - \alpha$$

を示せ．

[課題 2.21]　$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \boldsymbol{V})$ として，$\boldsymbol{V}$ が既知とする．$\boldsymbol{\theta}$ の信頼領域（すべての成分に関する同時信頼領域）が，

$$C(\hat{\boldsymbol{\theta}}) = \{\boldsymbol{\theta} \mid (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'\boldsymbol{V}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \leq G_m^{-1}(1 - \alpha)\} \tag{2.9}$$

で与えられることを示せ．ただし G_m は自由度 m のカイ二乗分布の累積分布関数とする．(θ の次元を m としている)．ヒント： $P(\boldsymbol{\theta} \in C(\hat{\boldsymbol{\theta}})) = 1 - \alpha$ を示せばよい．

[課題 2.22]　$\hat{\theta} \sim N(\theta, v)$ として，$v > 0$ は未知であるが，$\hat{v}$ によって推定されるとする．$\hat{v}$ は $\hat{\theta}$ とは独立な確率変数で，$k\hat{v}/v \sim \chi^2_k$（自由度 k のカイ二乗分布）とする．$(\hat{\theta} - \theta)/\sqrt{\hat{v}}$ は自由度 k の t 分布に従うことが知られている（この累積分布関数を F_k と書くことにする）．帰無仮説: $\theta = 0$ を対立仮説: $\theta \neq 0$ に対して，有意水準 α で検定するとき，p-値が次式で与えられることを示せ．

$$p = 2 \times \left[1 - F_k\left(\frac{|\hat{\theta}|}{\sqrt{v}}\right)\right] \tag{2.10}$$


```
> x <- runif(n,min=-1,max=1) # U(-1,1)
> e <- rnorm(n,mean=0,sd=sd) # N(0,sd^2)
> y <- beta0 + beta1*x + e
> plot(x,y) # データのプロット
> title(sub=paste("beta0=",beta0,", beta1=",beta1,sep=""))
> abline(a=beta0,b=beta1,col="darkgreen") # モデル式を表す直線
```

データから回帰係数を推定するには，次式を用いればよい．

$$\hat{\beta}_1 = \frac{\sum_{t=1}^n (x_t - \bar{x})(y_t - \bar{y})}{\sum_{t=1}^n (x_t - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{3.1}$$

ただし，$\bar{x} = \sum_{t=1}^n x_t/n$, $\bar{y} = \sum_{t=1}^n y_t/n$ は標本平均．

```
> ## 回帰係数の推定
> xc <- x - mean(x); yc <- y - mean(y) # 中心化
> b1 <- sum(xc*yc)/sum(xc^2) # beta1 の推定
> b0 <- mean(y) - b1*mean(x) # beta0 の推定
> plot(x,y) # データのプロット
> title(sub=paste("b0=",round(b0,5),", b1=",round(b1,5),sep=""))
> abline(a=b0,b=b1,col="red") # 回帰直線
```

[課題 3.1]　誤差の２乗和

$$\sum_{t=1}^n [y_t - (\beta_0 + \beta_1 x_t)]^2$$

3　多変量解析

サブセクション： 線形回帰分析，ロジスティック回帰分析，主成分分析

キーワード： 説明変数，目的変数，最小２乗法，重回帰モデル，多項式回帰，ニュートン法，射影，固有値，固有ベクトル，因子分析

3.1　線形回帰分析（重回帰分析）

データの要素が x と y のペア (x, y) とする．つまりデータは $\mathcal{X} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ とする．

[例 3.1]　単回帰モデルでは，x と y に次の関係があると考える．

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t, \quad t = 1, \ldots, n$$

ここで，β_0, β_1 は回帰係数，ϵ_t は誤差である．x は説明変数（独立変数，予測変数），y は目的変数（従属変数，応答変数）などと呼ばれる．例に使うために，乱数でデータを生成する．モデルは $\beta_0 = 2$, $\beta_1 = 0.5$, $x_t \sim U(-1, 1)$ (i.i.d.), $\epsilon_t \sim N(0, 0.2^2)$ (i.i.d.) とする．サンプルサイズを $n = 30$ とする．

```
> ## 単回帰分析の例題用データを生成する
> n <- 30 # サンプルサイズ
> beta0 <- 2; beta1 <- 0.5 # 回帰係数
> sd <- 0.2 # 誤差の標準偏差
```

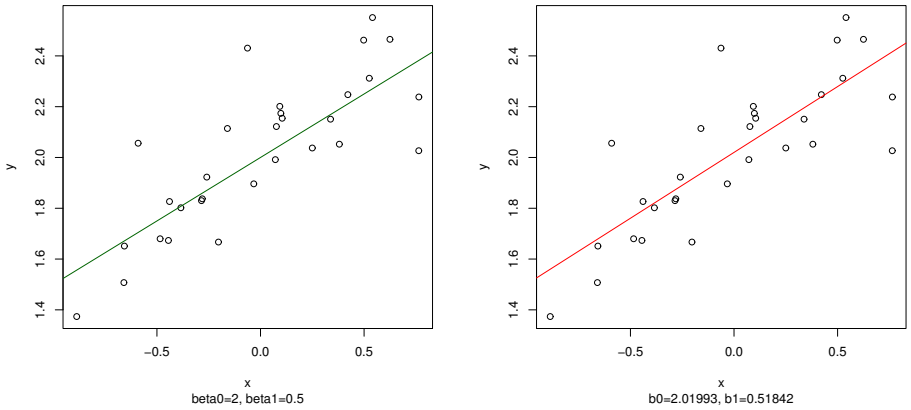



図 25　（左）データの生成，(右）推定した回帰直線

を最小にする β_0, β_1（すなわち，最小２乗法の解）が (3.1) で与えられることを示せ．

[課題 3.2]　$\boldsymbol{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tm})$, $t = 1, \ldots, n$ が m 次元ベクトルとする．重回帰モデル (multiple regression model) では，$\boldsymbol{x}$ と y に次の関係があると考える．

$$y_t = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \cdots + \beta_m x_{tm} + \epsilon_t, \quad t = 1, \ldots, n$$

このとき誤差の２乗和

$$\sum_{t=1}^{n} [y_t - (\beta_0 + \beta_1 x_{t1} + \cdots + \beta_m x_{tm})]^2$$

を最小にする回帰係数（すなわち最小２乗法の解）が

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{3.2}$$

で与えられることを示せ．ただし，

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} x_{10} & x_{11} & \ldots & x_{1m} \\ \vdots & & & \vdots \\ x_{n0} & x_{n1} & \ldots & x_{nm} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

と書いて，$\boldsymbol{X}$ のランクが m とする．なお $x_{t0} = 1$ と形式的におく．

[ヒント]　重回帰モデルを行列表示すると

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.3}$$

であり，最小二乗法は $\|\boldsymbol{\epsilon}\|^2 \to \min$ である．$\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ と書く．

- 目的関数 $\|\boldsymbol{\epsilon}\|^2$ を β で微分して 0 とおけば，正規方程式 $\boldsymbol{X}'\boldsymbol{X}\beta = \boldsymbol{X}'\boldsymbol{y}$ を得ることはできる．（これで極値であることは分かる．厳密には，最小性まで示さないといけない）．
- 点 $\boldsymbol{y}$ から $\mathrm{sp}(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ への射影が $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ である，と幾何的に解釈すれば，線形代数の知識からただちに理解できる．
- 要するに，$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 + \|\hat{\boldsymbol{y}} - \boldsymbol{X}\boldsymbol{\beta}\|^2$ を示せばよい．

[例 3.2]　ボストン市の住宅価格データ (bostondata.txt) の重回帰分析を行う．出典: D. Harrison and D. L. Rubinfeld (1978). 入力ミスの修正済みデータ “boston_corrected” が StatLib Datasets Archive (http://lib.stat.cmu.edu/datasets/) にある．いくつかの変数に二乗や対数変換を施したものを以下で用いる．サンプルサイズ $n = 506$（ボストン市の各ブロックに対応）．説明変数の数 $m = 13$．変数の説明：x_1 = 犯罪率 (Crim)，x_2 = 宅地割合 (Zn)，x_3 = 非商用地割合 (Indus)，x_4 = チャールス川沿いか（ダミー変数）(Chas)，x_5 = 窒素酸化物濃度の二乗 (Nox2)，x_6 = 平均部屋数の二乗 (Rm2)，x_7 =1940 年より古い住宅の割合 (Age)，x_8 = ビジネス街への距離 (Dis)，x_9 = ハイウェイへのアクセス (Rad)，x_{10} = 固定資産税 (Tax)，x_{11} = 生徒と教師の比率 (Ptratio)，x_{12} = アフリカ系米国人の比率を a とした $1000(a - 0.63)^2 (B)$，x_{13} = 低所得者層の割合 (Lstat)，x_{14} = 持ち家価格の中央値の対数 (LogCmedv)．


```
> ## 重回帰分析の例
> dat <- read.table("bostondata.txt") # テキスト形式（表形式）のデータ
> dim(dat) # 行数 列数
[1] 506  14
> colnames(dat) # 各列につけられた変数名 "LogCmdev"が住宅価格（の対数）
 [1] "Crim"    "Zn"      "Indus"   "Chas"    "Nox2"    "Rm2"
 [7] "Age"     "Dis"     "Rad"     "Tax"     "Ptratio" "B"
[13] "Lstat"   "LogCmedv"
> y <- dat[,14] # dat["LogCmedv"] でも同じ
> X <- as.matrix(dat[,-14]) # "data.frame"形式を"matrix"形式へ変換しておく
> X <- cbind(1,X) # 最初に 1 の列を追加
> beta <- solve(t(X) %*% X) %*% (t(X) %*% y) # 回帰係数の推定
> beta # 列ベクトル
                [,1]
          4.057090e+00
Crim     -1.017747e-02
Zn        1.216730e-03
Indus     2.859965e-03
Chas      1.018526e-01
Nox2     -5.695311e-01
Rm2       8.194266e-03
Age      -4.459517e-05
Dis      -4.604581e-02
Rad       1.321739e-02
Tax      -6.287862e-04
Ptratio -3.597845e-02
B         4.136065e-04
Lstat    -2.833569e-02
```


推定した回帰係数をつかって ϵ_t の影響を取り除いた予測値を計算するには

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{t1} + \hat{\beta}_2 x_{t2} \cdots + \hat{\beta}_m x_{tm}, \quad t = 1, \ldots, n$$

とする．予測値 $\hat{y}_t$ と観測値 y_t の差は，残差 (residual) と呼ばれ e_t とかく (誤差 ϵ_t の推定値とみなせる)．

$$e_t = y_t - \hat{y}_t, \quad t = 1, \ldots, n$$

```
> haty <- X %*% beta # 予測値
> resy <- y - haty # 残差
> plot(haty,y) ; abline(a=0,b=1) # 予測値と観測値
> plot(haty,resy) ; abline(h=0) # 予測値と残差
```

[課題 3.3]　課題 3.2の解でとくに $m = 1$ とおけば課題 3.1の解になることを示せ．

[課題 3.4]　$\{1, \ldots, n\} = A_1 \cup A_2 \cup \cdots A_m$ と分割する．t 番目の要素がグループ i に属することを $t \in A_i$ と書き，A_i の要素数を n_i とする $(n_1 + \cdots + n_m = n)$．グループ i の y_t の期待値を β_i とするモデルは，課題 3.2において，$x_{ti} = 1$ $(t \in A_i)$，$x_{ti} = 0$ $(t \notin A_i)$ と表現できる．このような 0/1 変数をダミー変数と呼ぶ．このとき，最小２乗法の解が，$\hat{\beta}_i = \sum_{t \in A_i} y_t / n_i$ となることを示せ．ただし $\boldsymbol{\beta} = (\beta_1 \ldots, \beta_m)'$，$\boldsymbol{X}$ も第１

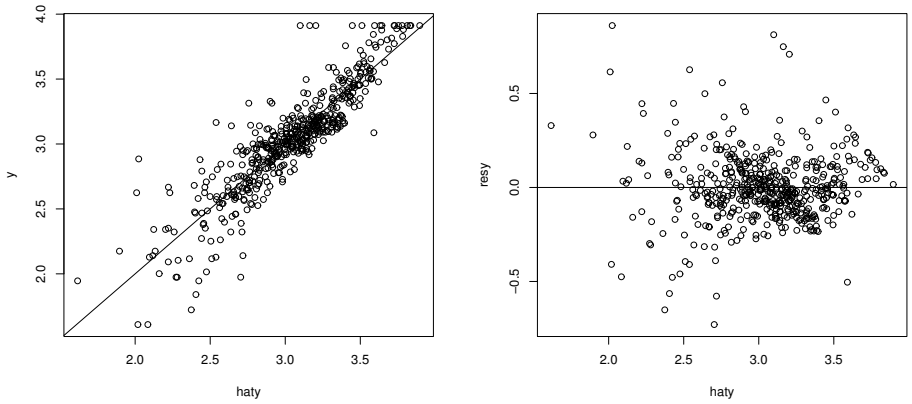


図 26　（左）$(\hat{y}_t, y_t)$ のプロット，（右）$(\hat{y}_t, e_t)$ のプロット

列を取り除いて再定義する（もしこうしないとどういう問題がおこるか？）

[課題 3.5]　誤差 $\epsilon_1, \ldots, \epsilon_n$ が互いに独立で，その分散が $V(\epsilon_t) = \sigma^2$ とする．推定した回帰係数 $\hat{\boldsymbol{\beta}}$ の期待値ベクトルと分散共分散行列が次式で与えられることを示せ．

$$E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}, \quad V(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{3.4}$$

ただし，利用した重回帰モデル (3.3) が正しいと仮定し，また説明変数 x_{ti} はすべて定数として扱う．

[課題 3.6]　課題 3.5において，任意の $m+1$ 次元ベクトル $\boldsymbol{d} = (d_0, d_1, \ldots, d_m)'$ をひとつ決める．$\gamma = \boldsymbol{d}'\boldsymbol{\theta}$ の不偏推定量が $\hat{\gamma} = \boldsymbol{d}'\hat{\boldsymbol{\theta}}$ で与えられること，および分散が $V(\hat{\gamma}) = \sigma^2\boldsymbol{d}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{d}$ であることを示せ．

[課題 3.7]　課題 3.5において，γ の不偏推定量として $y_1, \ldots, y_n$ の重み付き和を考える．重みベクトルを $\boldsymbol{f} \in \mathbb{R}^n$ とすれば，推定量は $\boldsymbol{f}'\boldsymbol{y}$ と書ける．このような線形不偏推定量のなかで分散を最小にするものが，課題 3.6で与えたものになることを示せ．ヒント： 任意の β で $E(\boldsymbol{f}'\boldsymbol{y}) = \boldsymbol{d}'\beta$ をみたすような $\boldsymbol{f}$ のうち $V(\boldsymbol{f}'\boldsymbol{y})$ を最小にするものが $\boldsymbol{f}^* = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{d}$ で与えられることを示す．

- **$\boldsymbol{X}'\boldsymbol{f} = \boldsymbol{d}$ をみたす $\boldsymbol{f}$ のうち $\|\boldsymbol{f}\|^2$ を最小にするものが $\boldsymbol{f}^*$ であることを示せばよい．**
- **ラグランジュの未定乗数法をつかえば，$\boldsymbol{f}^*$ で分散が極値を取ることはすぐに分かる．**
- **分散の最小性をいうには，$(\boldsymbol{f} - \boldsymbol{f}^*)'\boldsymbol{f}^* = 0$ に注意して，$\|\boldsymbol{f}\|^2 = \|\boldsymbol{f} - \boldsymbol{f}^*\|^2 + \|\boldsymbol{f}^*\|^2$ を示せばよい．**

[課題 3.8]　課題 3.5において，β の線形不偏推定量 $\boldsymbol{F}'\boldsymbol{y}$ を考える．ただし $\boldsymbol{F}$ は $n \times (m+1)$ 行列で $E(\boldsymbol{F}'\boldsymbol{y}) = \beta$ を満たすものである．このとき，$V(\boldsymbol{F}'\boldsymbol{y}) \geq V(\hat{\beta})$ であることを示せ（行列の差が非負正定値）．ヒント：前課題と同様に，$\boldsymbol{F}^* = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}$ とおいて $(\boldsymbol{F} - \boldsymbol{F}^*)'\boldsymbol{F}^* = 0$ を示せばよい．これで任意の $\boldsymbol{b} \in \mathbb{R}^{m+1}$ に対して $\|\boldsymbol{F}\boldsymbol{b}\|^2 = \|(\boldsymbol{F} - \boldsymbol{F}^*)\boldsymbol{b}\|^2 + \|\boldsymbol{F}^*\boldsymbol{b}\|^2$ がいえる．

[課題 3.9]　課題 3.5において，残差の２乗和 $\|\boldsymbol{e}\|^2 = \sum_{t=1}^n e_t^2$ の期待値が $E(\|\boldsymbol{e}\|^2) = (n-(m+1))\sigma^2$ となることを示せ．ヒント：$\boldsymbol{X}$ の列ベクトルがはる線形部分空間の直交補空間の正規直交基底を並べた $n \times (m+1)$ 行列 B をひとつ決めると，$\boldsymbol{X}'B = 0$, $B'B = \boldsymbol{I}$ である．これを使うと，$\|\boldsymbol{e}\|^2 = \|B'\boldsymbol{y}\|^2$ とかけ，また $V(B'\boldsymbol{y}) = \sigma^2\boldsymbol{I}$ である．

[例 3.3]　例 3.2で推定した $\hat{\beta}$ に (3.4) を適用して分散共分散行列を求める．ただし，σ^2 の値は未知であるから，データから次式で推定して

$$\hat{\sigma}^2 = \frac{1}{n-(m+1)}\sum_{t=1}^n e_t^2 \tag{3.5}$$

を代入したものを $\hat{V}(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$ とする．(3.5) の分母が n ではなく，$n-(m+1)$ となっているのは，推定量を「不偏」にするため．$E(\hat{\sigma}^2) = \sigma^2$ という意味．

```
> v <- sum(resy^2)/(nrow(X) - ncol(X)) # sigma^2の推定値
> V <- v*solve(t(X) %*% X) # 回帰係数の分散共分散行列の推定
> V[1:5,1:5] # 一部だけ表示してみる
                            Crim            Zn         Indus          Chas
  2.068560e-02 -1.020433e-05 -4.205819e-06  7.159956e-06 -2.333587e-04
```



```
Crim  -1.020433e-05  1.635833e-06 -6.025660e-08  1.150831e-07  2.115733e-06
Zn    -4.205819e-06 -6.025660e-08  2.861474e-07  1.452392e-07 -2.226609e-07
Indus  7.159956e-06  1.150831e-07  1.452392e-07  5.717345e-06 -7.989261e-06
Chas  -2.333587e-04  2.115733e-06 -2.226609e-07 -7.989261e-06  1.127054e-03
```

$\hat{V}(\hat{\boldsymbol{\beta}}) = \hat{\boldsymbol{\Sigma}}$ の対角成分 $(\hat{\sigma}_{00}, \ldots, \hat{\sigma}_{mm})$ を取りだすと，$\hat{V}(\hat{\beta}_i) = \hat{\sigma}_{ii}$ である．そして次の表を作成する．

$$\hat{\beta}_i, \quad \sqrt{\hat{\sigma}_{ii}}, \quad \frac{\hat{\beta}_i}{\sqrt{\hat{\sigma}_{ii}}}, \quad 2 \times \left[1 - F_{n-(m+1)}\left(\frac{|\hat{\beta}_i|}{\sqrt{\hat{\sigma}_{ii}}}\right)\right]$$

１列目は $\hat{\beta}_i$，２列目は標準誤差 $\sqrt{\hat{\sigma}_{ii}}$ である．３列目は t 統計量，４列目はその p 値であり，帰無仮説 $\beta_i = 0$ の検定に用いる．$p_i < 0.05$ なら $\beta_i \neq 0$ と判断する．このような検定を行うには，誤差のしたがう確率モデルを指定する必要がある．ここでは誤差が正規分布に従うことを暗に仮定している（課題 3.12参照）．

```
> sbeta <- sqrt(diag(V)) # 対角成分をとりだして，平方根をとる
> cbind(beta,sbeta, # 回帰係数とその標準誤差
+       beta/sbeta, # t統計量
+       2*pt(abs(beta/sbeta),df=nrow(X) - ncol(X),lower=F)) # p値
                          sbeta
         4.057090e+00 0.1438249067  28.2085350 7.361162e-105
Crim    -1.017747e-02 0.0012789967  -7.9573831  1.220186e-14
Zn       1.216730e-03 0.0005349275   2.2745693  2.336141e-02
Indus    2.859965e-03 0.0023910971   1.1960892  2.322378e-01
Chas     1.018526e-01 0.0335716323   3.0338883  2.541897e-03
Nox2    -5.695311e-01 0.1109698757  -5.1323039  4.127707e-07
Rm2      8.194266e-03 0.0012591585   6.5077320  1.882328e-10
Age     -4.459517e-05 0.0005083177  -0.0877309  9.301263e-01
Dis     -4.604581e-02 0.0076690524  -6.0041063  3.739856e-09
```

```
Rad      1.321739e-02 0.0025796054   5.1238039  4.308396e-07
Tax     -6.287862e-04 0.0001463849  -4.2954310  2.101708e-05
Ptratio -3.597845e-02 0.0051507944  -6.9850289  9.286912e-12
B        4.136065e-04 0.0001044979   3.9580378  8.670362e-05
Lstat   -2.833569e-02 0.0019652958 -14.4180269  1.462098e-39
```

これと全く同じ結果を得るには，R組み込みの `lm` を実行しても良い．

```
> f <- lm(LogCmedv ~ . , data=dat) # 線形モデル (linar model) による重回帰分析
> summary(f) # 結果の表示
Call:
lm(formula = LogCmedv ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72917 -0.09510 -0.01151  0.08944  0.86119

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.0570899  0.1438249  28.209  < 2e-16 ***
Crim        -0.0101775  0.0012790  -7.957 1.22e-14 ***
Zn           0.0012167  0.0005349   2.275  0.02336 *
Indus        0.0028600  0.0023911   1.196  0.23224
Chas         0.1018526  0.0335716   3.034  0.00254 **
Nox2        -0.5695311  0.1109699  -5.132 4.13e-07 ***
Rm2          0.0081943  0.0012592   6.508 1.88e-10 ***
Age         -0.0000446  0.0005083  -0.088  0.93013
Dis         -0.0460458  0.0076691  -6.004 3.74e-09 ***
Rad          0.0132174  0.0025796   5.124 4.31e-07 ***
Tax         -0.0006288  0.0001464  -4.295 2.10e-05 ***
```



```
Ptratio     -0.0359785  0.0051508  -6.985 9.29e-12 ***
B            0.0004136  0.0001045   3.958 8.67e-05 ***
Lstat       -0.0283357  0.0019653 -14.418  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1847 on 492 degrees of freedom
Multiple R-Squared: 0.8005,        Adjusted R-squared: 0.7953
F-statistic: 151.9 on 13 and 492 DF,  p-value: < 2.2e-16
```

[例 3.4]　(x_t, y_t) に多項式の関係があるとする．

$$y_t = \sum_{i=0}^{m} \beta_i x_t^i + \epsilon_t$$

これを多項式回帰と呼ぶ．形式的に $\boldsymbol{x}_t = (1, x_t, x_t^2, \ldots, x_t^{m-1})$ とおいて重回帰分析を適用すればよい．

$$\beta_0 = -1, \quad \beta_1 = 2, \quad \beta_2 = -0.5, \quad \sigma^2 = 2^2$$

としてデータを生成し，それに多項式回帰分析を適用する．

```
> ## データの生成
> truebeta <- c(-1,2,-0.5); truess <- 2^2; n <- 100 # パラメタ
> m <- length(truebeta)-1 # 多項式の次数
> x <- runif(n,min=0,max=5) # x ~ U(0,5) とする
> X <- outer(x,0:m,"^") # X 行列の作成
> y <- X %*% truebeta + rnorm(n,sd=sqrt(truess)) # 誤差が正規分布に従うと仮定して y の生成
> x0 <- seq(from=min(x),to=max(x),length=300) # プロット用に x の範囲を 300 等分
> X0 <- outer(x0,0:m,"^") # その X 行列
> ## 真の多項式とデータのプロット
```



```
> plot(x,y) # データ
> lines(x0,X0 %*% truebeta,col="green",lwd=2,lty=2) # 真の多項式
> ## 係数の推定
> A <- solve(t(X) %*% X) # A=(X'X)^-1 とおく
> beta <- A %*% (t(X) %*% y) # 最小二乗法
> ee <- y - X %*% beta # 残差
> sum(ee) # 残差の和は常に 0
[1] -2.85133e-13
> ss <- sum(ee^2)/(n-(m+1)) # 分散の不偏推定
> round(ss * A, 4) # 回帰係数の分散共分散行列
       [,1]    [,2]    [,3]
[1,]  0.1777 -0.1442  0.0242
[2,] -0.1442  0.1736 -0.0337
[3,]  0.0242 -0.0337  0.0070
> sbeta <- sqrt(diag(ss*A)) # 回帰係数の標準誤差
> cbind(beta,sbeta, # 回帰係数とその標準誤差
+       beta/sbeta, # t 統計量
+       2*pt(abs(beta/sbeta),df=n-(m+1),lower=F)) # p 値
                  sbeta
[1,] -0.7306907 0.42152234 -1.733457 0.0861920625
[2,]  1.2922445 0.41662026  3.101732 0.0025197182
[3,] -0.3243540 0.08359933 -3.879863 0.0001905010
> ## 推定した多項式のプロット
> lines(x0,X0 %*% beta,col="red",lwd=2)
```

任意の x における $E(y|x) = \sum_{i=0}^{m} \beta_i x^i$ の不偏推定は $\sum_{i=0}^{m} \hat{\beta}_i x^i$ である．一般に重みベクトル $\boldsymbol{w}$ を与えたとき，$\boldsymbol{w}'\boldsymbol{\beta}$ の不偏推定は $\boldsymbol{w}'\hat{\boldsymbol{\beta}}$ である．その分散は $V(\boldsymbol{w}'\hat{\boldsymbol{\beta}}) = \sigma^2 \boldsymbol{w}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{w}$ である．これを利用して $E(y|x)$ の信頼区間を計算する．


```
> ## 95% 信頼区間の計算
> q0 <- qt(0.975,df=n-(m+1)) # t 分布の両側 5% 点
> q0 # 約 2
[1] 1.984723
> ss0 <- ss*apply(X0,1,function(w) t(w) %*% A %*% w) # x0 の各点における E(y|x) の不偏分散
> plot(x,y,type="n") # データはプロットしないで枠だけ
> lines(x0,X0 %*% truebeta,col="green",lwd=2,lty=2) # 真の多項式
> lines(x0,X0 %*% beta + q0*sqrt(ss0),col="blue",lwd=2,lty=3) # 上側
> lines(x0,X0 %*% beta - q0*sqrt(ss0),col="blue",lwd=2,lty=3) # 下側
```

[課題 3.10]　重回帰モデルで説明変数 $\boldsymbol{x}_t$ については特に確率分布を想定せず，$\boldsymbol{x}_t$ を与えたときの y_t の条件付分布 $f(y_t|\boldsymbol{x}_t;\boldsymbol{\theta})$ を考えて尤度を

$$L(\boldsymbol{\theta}|\mathcal{X}) = f(y_1|\boldsymbol{x}_1;\boldsymbol{\theta}) \cdots f(y_n|\boldsymbol{x}_n;\boldsymbol{\theta}) \tag{3.6}$$

で定義する．誤差の従う分布が $\epsilon_t \sim N(0, \sigma^2)$ (i.i.d.) とすれば，確率モデルが

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n) \tag{3.7}$$

によって定義できる．ただし，モデルのパラメタは $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots, \beta_m, \sigma^2)$ である．このとき，回帰係数の最尤推定が最小二乗法に一致することを示せ．また，σ^2 の最尤推定が

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{t=1}^{n} e_t^2 \tag{3.8}$$

となることを示せ．(3.5) と (3.8) は分母が異なることに注意する．ヒント：$f(y_t|\boldsymbol{x}_t;\boldsymbol{\theta})$ は，

$$y_t|\boldsymbol{x}_t \sim N(\beta_0 + \beta_1 x_{t1} + \cdots + \beta_m x_{tm}, \sigma^2) \tag{3.9}$$

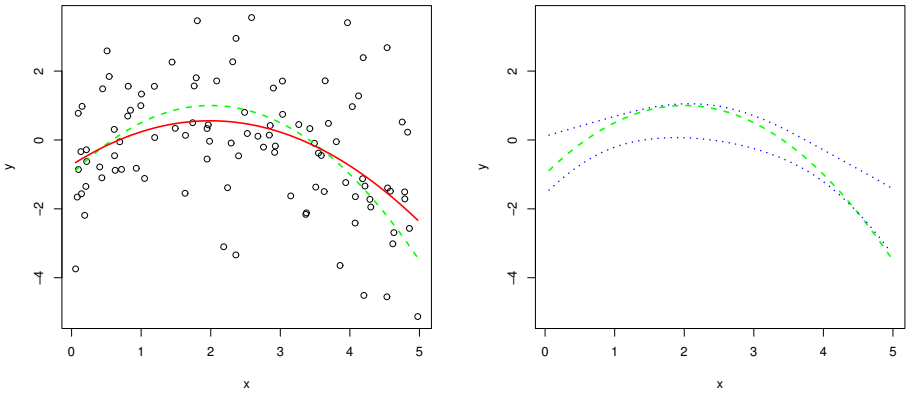


図 27　各点はデータ (x_t, y_t)．真の多項式は緑の破線，推定した多項式は赤の実線，信頼区間は青の点線．

と書ける．

[課題 3.11]　課題 3.10において，次式を示せ．

$$E\left(-\frac{\partial^2 \log L}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right) = \begin{bmatrix} \frac{1}{\sigma^2}\boldsymbol{X}'\boldsymbol{X} & \mathbf{0} \\ \mathbf{0} & \frac{n}{2\sigma^4} \end{bmatrix}$$

[課題 3.12]　課題 3.10において，回帰係数の最尤推定量が $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1})$ であることを示せ．また，(3.5) で与えられる不偏分散 $\hat{\sigma}^2$ が $\hat{\boldsymbol{\theta}}$ と独立な確率変数で $(n-(m+1))\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-(m+1)}$ であることを示せ．ヒント：課題 3.9のヒントで $\boldsymbol{B}'\boldsymbol{y} \sim N(\mathbf{0}, \sigma^2\boldsymbol{I})$ であることと，$\boldsymbol{X}'\boldsymbol{B} = \mathbf{0}$ であることを利用すればよい．

[課題 3.13]　重回帰モデルで誤差 $\epsilon_1, \ldots, \epsilon_n$ が独立であるがその分散が異なり $\epsilon_t \sim N(0, \sigma_t^2)$ に従うとする．(i) $\sigma_1^2, \ldots, \sigma_n^2$ が既知と仮定して，回帰係数の最尤推定量を求めよ．(ii) 既知の定数 $a_1, \ldots, a_n > 0$ と未知パラメタ γ を用いて $\sigma_t^2 = a_t\gamma$ と書けると仮定する．回帰係数と γ の最尤推定量を求めよ．

3.2　ロジスティック回帰分析

スパムメール判別を回帰分析とみなせば，$\boldsymbol{x}_t$ が t 番目のメールの特徴量（単語の有無情報）であり，スパムなら $y_t = 1$，非スパムなら $y_t = 0$ となる．このように，目的変数 y_t が２値しか取らない場合（とりあえず 0 または 1）を考える．(3.9) のように正規分布を想定するのは明らかにおかしい．

[定義 3.1]　$\boldsymbol{x}_t$ を与えたときの y_t の条件付分布 $f(y_t|\boldsymbol{x}_t;\boldsymbol{\beta})$ を次式であたえる．

$$f(1|\boldsymbol{x}_t;\boldsymbol{\beta}) = p_t, \quad f(0|\boldsymbol{x}_t;\boldsymbol{\beta}) = 1 - p_t$$

とおいて，

$$p_t = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{t1}+\cdots+\beta_m x_{tm})}} \tag{3.10}$$

これをロジスティック回帰モデルという．モデルのパラメタは $\boldsymbol{\theta} = \boldsymbol{\beta} = (\beta_0, \ldots, \beta_m)$．

[注意]　(3.10) に現れる

$$g(\eta) = \frac{e^\eta}{1+e^\eta} = \frac{1}{1+e^{-\eta}}$$

```
> eta <- seq(-10,10,length=300)
> prb <- 1/(1+exp(-eta))
> plot(eta,prb,type="l")
```

をロジスティック (logistic) 関数と呼ぶ（図 28左）．$g(-\infty) = 0$, $g(0) = 0.5$, $g(\infty) = 1$ である．単調増加関数で S 字型の関数である．これを使うと，(3.10) は

$$p_t = g(\eta_t), \quad \eta_t = \sum_{i=0}^{m} \beta_i x_{ti}$$

と表現できる．なお，ロジスティック関数の逆関数

$$g^{-1}(p) = \log\left(\frac{p}{1-p}\right)$$

をロジット (logit) 関数と呼ぶ．

[課題 3.14]　ロジスティック回帰モデルの対数尤度関数が次式で与えられることを確認せよ．

$$\log L(\boldsymbol{\beta}|\mathcal{X}) = \sum_{t=1}^{n} (y_t \log p_t + (1-y_t)\log(1-p_t)) \tag{3.11}$$

ただし p_t は (3.10) で与える．

[例 3.5]　簡単なモデルでデータを生成して，ロジスティック回帰分析を行う．$\boldsymbol{\beta} = (\beta_0, \beta_1)$ とおいて，

$$\beta_0 = -4, \quad \beta_1 = 10$$

からサンプルサイズ $n = 100$ のデータを生成する．$x_t \sim U(0,1)$ としておく．

```
> ### データの生成
> truebeta <- c(-4,10) # パラメタの真値 (beta0,beta1) の設定
> n <- 100 # サンプルサイズ
> x <- runif(n) # x の生成
> ## beta=c(beta0,beta1) から p(y=1|x) のベクトルを計算する関数を準備 (x を参照する)
> mylogistic2 <- function(beta) 1/(1+exp(-(beta[1]+beta[2]*x)))
> ## 以下で y を生成
> trueprb <- mylogistic2(truebeta) # p(y=1|x) の計算
> y <- (runif(n) < trueprb)+0 # y の生成
```

```
> plot(x,y) # (x,y) データ点を黒でプロット
> points(x,trueprb,col="green") # p(y=1|x) を緑でプロット（パラメタは真値）
```

次に optim を利用して最尤法を実行する．(3.11) を数値的に最大化して $\hat{\beta}$ を計算する．optim は目的関数の２階微分して得られる行列も一緒に返す (オプションで hessian=TRUE を指定) ので，(2.6) から $\hat{V}(\hat{\beta})$ が得られる．結果を表にまとめる．表の形式は例 3.3における重回帰分析と同じ．

```
> ## 対数尤度関数*(-1) の定義 (x と y を参照する)
> mylik <- function(beta) {
+   prb <- mylogistic2(beta)
+   -sum(y*log(prb)+(1-y)*log(1-prb)) # -log L
+ }
> ## 数値的最適化
> a <- optim(c(0,0),mylik,method="BFGS",hessian=TRUE,control=list(trace=1))
initial  value 69.314718
iter  10 value 35.447467
final  value 35.447424
converged
> beta <- a$par # beta の最尤推定量
> sd <- sqrt(diag(solve(a$hessian))) # beta の標準誤差
> cbind(beta,sd,beta/sd,2*pnorm(abs(beta/sd),lower=F)) # 結果を表にまとめる
          beta        sd
[1,] -3.972583 0.8107357 -4.899973 9.585004e-07
[2,]  9.104391 1.7340507  5.250360 1.518022e-07
> points(x,mylogistic2(beta),col="red") # p(y=1|x) を緑でプロット（パラメタは最尤推定量）
```

[例 3.6]　R 組み込みの glm を用いて例 3.5を再計算する．利用法は lm と同様．なお，glm は GLM (Genelalized Linear Model) のこと．

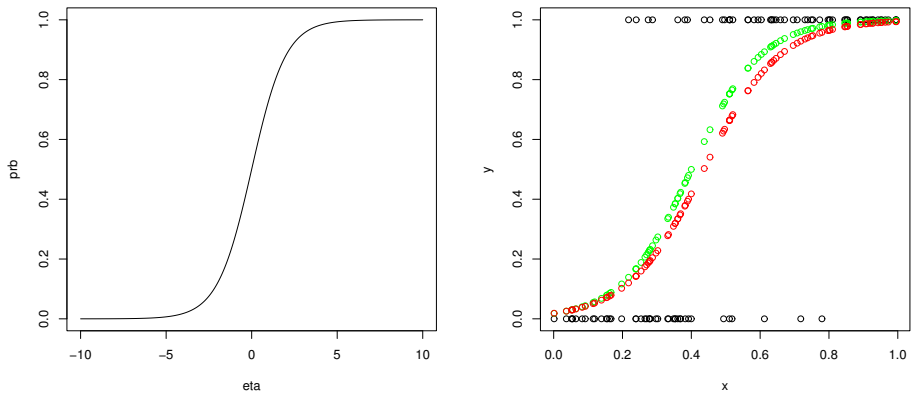


図 28　（左）ロジスティック関数，（右）$\beta = (-4, 10)$ から生成した (x_t, y_t)


```
> f <- glm(y ~ x, binomial, data.frame(x,y)) # bionomial を指定するとロジスティック回帰
> summary(f) # 結果を表にまとめる
Call:
glm(formula = y ~ x, family = binomial, data = data.frame(x,
    y))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5172  -0.5635   0.1384   0.5042   2.0589

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.9724     0.8107   -4.90 9.59e-07 ***
x             9.1040     1.7340    5.25 1.52e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 136.663  on 99  degrees of freedom
Residual deviance:  70.895  on 98  degrees of freedom
AIC: 74.895

Number of Fisher Scoring iterations: 6
```


結果は optim とほぼ同じ．glm の内部では，optim のような汎用の最適化アルゴリズムではなく，Fisher Scoring 法（一種のニュートン法）が用いられている．これを理解するために，まず以下の課題を準備する．

[課題 3.15]　次式を示せ．

$$-\frac{\partial \log L}{\partial \beta_i} = \sum_{t=1}^{n}(p_t - y_t)x_{ti}, \quad -\frac{\partial^2 \log L}{\partial \beta_i \partial \beta_j} = \sum_{t=1}^{n} p_t(1-p_t)x_{ti}x_{tj}$$

[注意]　これを行列表示すると，

$$-\frac{\partial \log L}{\partial \boldsymbol{\beta}} = \boldsymbol{X}'(\boldsymbol{p} - \boldsymbol{y}), \quad -\frac{\partial^2 \log L}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X} \tag{3.12}$$

ただし $\boldsymbol{W}$ は対角行列でその対角成分が $(p_1(1-p_1), \ldots, p_n(1-p_n))$ とする．最尤推定量を求めるために尤度方程式

$$\frac{\partial \log L}{\partial \beta_i} = 0, \quad i = 0, \ldots, m$$

をニュートン法で計算するには，次の反復を行えばよい．

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} - \left(\boldsymbol{X}'\boldsymbol{W}^{(k)}\boldsymbol{X}\right)^{-1}\boldsymbol{X}'(\boldsymbol{p}^{(k)} - \boldsymbol{y})$$

GLM のスコア法では目的関数の２階微分（ヘシアン）をその期待値で置き換えるのだが，ロジスティック回帰では (3.12) にあるように両者が一致して単にニュートン法になる．なお最尤推定量 $\hat{\beta}$ の分散共分散行列は

(2.6) より次式で推定できる．

$$\hat{V}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X}'\hat{\boldsymbol{W}}\boldsymbol{X})^{-1}$$

```
> X <- cbind(1,x) # データ行列の準備
> betak <- c(0,0) # パラメタの初期値
> nr <- 100 # 最大反復数
> lik <- rep(0,nr) # 目的関数の系列を保存する配列（あとでプロットするため）
> for(k in 1:nr) {
+   etak <- X %*% betak # eta の計算
+   prbk <- 1/(1+exp(-etak)) # p(y=1|x) の計算
+   wk <- as.vector(prbk*(1-prbk)) # 行列 W の対角成分
+   A <- solve(t(X * wk)  %*% X) # = (t(X) %*% diag(wk) %*% X) の逆行列
+   betak <- betak - A %*%(t(X) %*% (prbk - y)) # パラメタの更新
+   lik[k] <- -sum(y*log(prbk)+(1-y)*log(1-prbk)) # 目的関数の計算
+   if(k>1 && lik[k-1]-lik[k]<1e-10) break # 終了判定
+ }
> lik[1:k] # 目的関数の系列
[1] 69.31472 39.71577 35.97845 35.46445 35.44745 35.44742 35.44742
> sdk <- sqrt(diag(A)) # 標準誤差
> cbind(betak,sdk,betak/sdk,2*pnorm(abs(betak/sdk),lower=F)) # 結果を表にまとめる
                sdk
  -3.972395 0.8107003 -4.899955 9.585862e-07
x  9.103970 1.7339637  5.250381 1.517852e-07
```

[例 3.7]　例 2.7のスパムメール判別をロジスティック回帰分析で行う．ロジスティック回帰では最尤推定量が解析的に得られず，数値的最適化によって $\hat{\beta}$ を求める．R 組み込みの `glm` を利用してこれを実行する．学

習データから $\hat{\beta}$ を計算して，結果を表にまとめる．表の形式は例 3.3における重回帰分析と同じ．

```
> ## spambase データ（単語の有無だけ）の読み込み
> load("spam1.rda") # dat1.train,spam.train,dat1.test,spam.test
> ## glm (Genelalized Linear Model = 一般化線形モデル) による回帰係数の推定
> f1 <- glm(spam.train ~ ., binomial, cbind(dat1.train, spam.train))
> summary(f1) # 回帰係数などのサマリ
Call:
glm(formula = spam.train ~ ., family = binomial, data = cbind(dat1.train,
    spam.train))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-4.20928  -0.15042  -0.00927   0.07687   3.90505

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         -1.9434     0.1609 -12.080  < 2e-16 ***
Iword_freq_make     -0.5845     0.2577  -2.268 0.023304 *
Iword_freq_address  -0.1196     0.2540  -0.471 0.637867
Iword_freq_all      -0.3027     0.2000  -1.514 0.130152
Iword_freq_3d        1.2322     0.8801   1.400 0.161470
Iword_freq_our       1.1013     0.1929   5.708 1.14e-08 ***
Iword_freq_over      0.3230     0.2519   1.282 0.199765
Iword_freq_remove    2.4263     0.3097   7.835 4.68e-15 ***
Iword_freq_internet  0.6153     0.2856   2.154 0.031221 *
Iword_freq_order     0.8631     0.3191   2.704 0.006842 **
Iword_freq_mail      0.3888     0.2218   1.753 0.079645 .
Iword_freq_receive  -0.2933     0.3066  -0.957 0.338721
Iword_freq_will     -0.2534     0.1779  -1.424 0.154334
```



```
Iword_freq_people      -1.0419     0.2951  -3.530 0.000415 ***
Iword_freq_report       0.9001     0.3962   2.272 0.023115 *
Iword_freq_addresses    1.7282     0.6639   2.603 0.009244 **
Iword_freq_free         1.6063     0.2042   7.865 3.68e-15 ***
Iword_freq_business     0.9090     0.2805   3.241 0.001192 **
Iword_freq_email       -0.5401     0.2370  -2.279 0.022683 *
Iword_freq_you          0.2260     0.1947   1.160 0.245875
Iword_freq_credit       0.6876     0.4516   1.523 0.127841
Iword_freq_your         0.7106     0.1944   3.655 0.000257 ***
Iword_freq_font         1.2074     0.5055   2.388 0.016927 *
Iword_freq_000          1.5723     0.3731   4.214 2.51e-05 ***
Iword_freq_money        1.2646     0.3328   3.800 0.000145 ***
Iword_freq_hp          -3.4287     0.4251  -8.065 7.34e-16 ***
Iword_freq_hpl         -0.6228     0.4927  -1.264 0.206188
Iword_freq_george      -5.1857     0.7839  -6.615 3.71e-11 ***
Iword_freq_650          2.3686     0.4711   5.027 4.98e-07 ***
Iword_freq_lab         -0.4928     0.5398  -0.913 0.361335
Iword_freq_labs         0.1928     0.5227   0.369 0.712230
Iword_freq_telnet      -3.2020     1.3612  -2.352 0.018657 *
Iword_freq_857         -0.5321     1.8535  -0.287 0.774060
Iword_freq_data        -0.7921     0.3912  -2.025 0.042880 *
Iword_freq_415          1.4776     1.3720   1.077 0.281508
Iword_freq_85          -1.9679     0.6886  -2.858 0.004264 **
Iword_freq_technology   0.4962     0.3612   1.374 0.169543
Iword_freq_1999        -1.1901     0.3173  -3.751 0.000176 ***
Iword_freq_parts        1.5060     0.6305   2.389 0.016912 *
Iword_freq_pm          -0.8053     0.4148  -1.941 0.052220 .
Iword_freq_direct      -0.2871     0.6091  -0.471 0.637323
Iword_freq_cs          -6.2489     4.7763  -1.308 0.190768
```



```
Iword_freq_meeting     -2.4746     0.4899  -5.051 4.39e-07 ***
Iword_freq_original    -0.9624     0.6164  -1.561 0.118438
Iword_freq_project     -1.8310     0.4874  -3.756 0.000172 ***
Iword_freq_re          -0.9277     0.2000  -4.637 3.53e-06 ***
Iword_freq_edu         -2.9250     0.3886  -7.527 5.20e-14 ***
Iword_freq_table        0.6027     0.9187   0.656 0.511780
Iword_freq_conference  -1.6802     0.6067  -2.769 0.005616 **
`Ichar_freq_;`         -0.2049     0.2536  -0.808 0.419232
`Ichar_freq_(`          0.1743     0.1747   0.998 0.318371
`Ichar_freq_[`         -0.2466     0.4152  -0.594 0.552647
`Ichar_freq_!`          1.3012     0.1624   8.012 1.12e-15 ***
`Ichar_freq_$`          2.0727     0.2334   8.882  < 2e-16 ***
`Ichar_freq_#`         -0.3522     0.3102  -1.135 0.256221
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4848.3  on 3600  degrees of freedom
Residual deviance: 1224.0  on 3546  degrees of freedom
AIC: 1334

Number of Fisher Scoring iterations: 10
```

次に，学習データで spam 判別を行ったとき，ham を spam と誤判別する確率が 0.01 になるように閾値を調整する．

```
> pp1.train <- predict(f1,dat1.train,type="response") # 学習データで予測
> round(pp1.train[1:20],3) # p(y=1|x) 最初の 20 個だけ表示
```

```
    1     2     3     4     5     6     7     8     9    10    11    12    13
0.984 1.000 0.934 0.000 0.396 0.993 0.740 0.003 0.039 0.231 0.633 0.999 0.022
   14    15    16    17    18    19    20
0.923 0.009 0.984 0.997 0.999 0.001 0.040
> spam.train[1:20] # spam=1, ham=0
 [1] 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 1 1 0 0
> pth1 <- quantile(pp1.train[spam.train==0],p=0.99) # 閾値の計算
> pth1 # もし p(y=1|x)>pth1 なら，spam と判定する．
      99%
0.8951595
```

先ほど求めた $\hat{\beta}$ と，この閾値を用いて，テストデータで spam 判別を行う．

```
> pp1.test <- predict(f1,dat1.test,type="response") # テストデータで予測
> round(pp1.test[1:20],3) # p(y=1|x) 最初の 20 個だけ表示
    1     2     3     4     5     6     7     8     9    10    11    12    13
0.886 0.781 0.045 0.999 0.000 0.008 0.000 0.998 0.268 0.122 0.000 0.005 1.000
   14    15    16    17    18    19    20
0.005 0.000 0.000 0.845 0.999 0.998 0.000
> spam.test[1:20] # spam=1, ham=0
 [1] 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 1 0
> myPPplot(spam.test,pp1.test,pth1) # 閾値 pth で判別（左図）
       pth         p0         p1
0.89515955 0.01430843 0.70619946
```

例 2.7より予測精度が若干向上した．

[例 3.8]　これまで各単語がメールに含まれているか，その有無だけの情報で spam 判別を行った．ここで利用している spambase データセットでは，単語の有無だけでなく，その頻度 (パーセント) 情報が利用できる．

これを説明変数に利用して予測を行う．有無データ x_{ti} と頻度データ x'_{ti} には，$x_{ti} = I(x'_{ti} > 0)$ の関係がある．

```
> ## spambase データ（単語頻度）の読み込み
> load("spam2.rda") # dat2.train,spam.train,dat2.test,spam.test
> dim(dat2.train) # dat1.train と同じサイズ
[1] 3601   54
> t(dat2.train[1:10,1:5]) # 最初の 10 個のメール，5 個の変数だけ表示
                  1    2    3 4    5    6    7 8    9 10
word_freq_make    0 0.00 0.00 0 0.00 0.00 0.05 0 0.00  0
word_freq_address 0 0.10 0.00 0 0.00 0.00 0.30 0 0.00  0
word_freq_all     0 0.30 0.53 0 0.19 0.68 0.40 0 0.47  0
word_freq_3d      0 0.00 0.00 0 0.00 0.00 0.00 0 0.00  0
word_freq_our     0 1.02 0.53 0 0.00 0.00 0.10 0 1.91  0
> ## glm (Genelalized Linear Model = 一般化線形モデル）による回帰係数の推定
> f2 <- glm(spam.train ~ ., binomial, cbind(dat2.train, spam.train))
> ## p(y=1|x) の計算
> pp2.train <- predict(f2,dat2.train,type="response") # 学習データ
> pp2.test <- predict(f2,dat2.test,type="response") # テストデータ
> ## 学習用データで ham を spam と判別する確率を 0.01 になるように閾値を決める
> pth2 <- quantile(pp2.train[spam.train==0],p=0.99)
> myPPplot(spam.test,pp2.test,pth2) # 閾値 pth で判別（図は省略）
      pth        p0        p1
0.9385936 0.0190779 0.5633423
```

結果を見ると，（期待に反して？）単語の有無だけを利用して判別したほうが，頻度情報を用いるより予測精度が高かった．頻度に比例して $\beta'_i x'_{ti}$ という形で spam の可能性が高まるのではなくて，特定の単語が少しでもあれば spam である可能性が高まることを意味している．このように，特徴量の選択こそが現実のアプリ

ケーションでは重要になる．念のため，単語の有無と頻度の両方を同時に利用して判別したらどうなるかを以下で確かめてみる．

```
> f3 <- glm(spam.train ~ ., binomial, cbind(dat1.train, dat2.train, spam.train))
> pp3.train <- predict(f3,cbind(dat1.train,dat2.train),type="response") # 学習データ
> pp3.test <- predict(f3,cbind(dat1.test,dat2.test),type="response") # テストデータ
> pth3 <- quantile(pp3.train[spam.train==0],p=0.99)
> myPPplot(spam.test,pp3.test,pth3) # 閾値 pth で判別（右図）
       pth         p0         p1
0.87828914 0.01589825 0.77088949
```

これまでで一番予測精度が高い結果である．

3.3　主成分分析

- 回帰分析では x によって y を説明することを試みた．主成分分析では y を明示的に与えず，x だけからデータをよく説明するような合成変量 y を与える．
- データ行列は

$$\boldsymbol{X} = \underbrace{\left.\begin{bmatrix} x_{11} & \dots & x_{1m} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ x_{n1} & \dots & x_{nm} \end{bmatrix}\right\} n}_{m} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{x}^{(n)} \end{bmatrix} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_m]$$

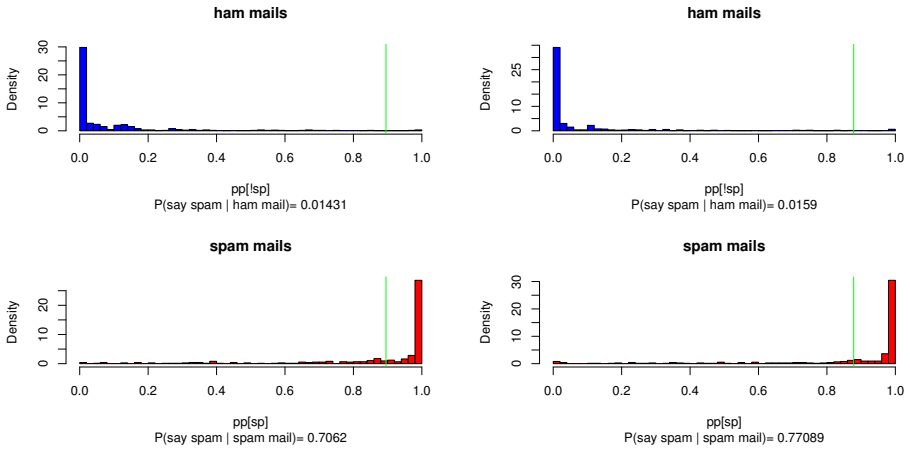


図 29　テストデータに関して，（左）単語の有無だけで予測，（右）単語頻度も使って予測

回帰分析では（とくに断らない限り）データ行列 X の最初の列が 1 であることを仮定したが，主成分分析ではそれを仮定しない．

- データはあらかじめ「中心化」されているものとする．すなわち，各列から平均値を引いて 0 にしてあると仮定する．各変数 $i = 1, \ldots, m$ について，

$$x_{ti} \leftarrow x_{ti} - \frac{1}{n}\sum_{t=1}^{n} x_{ti}, \quad t = 1, \ldots, n$$

または，X の i 番目の n 次元列ベクトル $\boldsymbol{x}_i$ と n 次元列ベクトル $\mathbf{1}_n = (1, \ldots, 1)'$ を使って

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\boldsymbol{x}_i, \quad \text{または} \quad X \leftarrow X - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'X$$

R では `dat <- scale(dat,scale=F)` でよい．中心化後は $\mathbf{1}_n'X = \mathbf{0}$ である．

- 応用ではデータをあらかじめ「標準化」することが多い．まず中心化してから，

$$x_{ti} \leftarrow x_{ti} \Big/ \left(\frac{1}{n-1}\sum_{t=1}^{n} x_{ti}^2\right)^{1/2} \quad \text{または} \quad \boldsymbol{x}_i \leftarrow \boldsymbol{x}_i \Big/ \sqrt{\frac{1}{n-1}\|\boldsymbol{x}_i\|^2}$$

R では単に `dat <- scale(dat)` でよい．標準化後は，$\|\boldsymbol{x}_i\|^2 = n - 1$ である．

[例 3.9]　例 3.2で用いたボストン市の住宅価格データ (bostondata.txt) の主成分分析をまずやってみる．計算の中身は後ほど説明．


```
> ## データ行列の準備
> dat <- read.table("bostondata.txt") # テキスト形式（表形式）のデータ
> dim(dat) # 行数 列数
[1] 506  14
> colnames(dat) # 変数名
 [1] "Crim"     "Zn"       "Indus"    "Chas"     "Nox2"     "Rm2"
 [7] "Age"      "Dis"      "Rad"      "Tax"      "Ptratio"  "B"
[13] "Lstat"    "LogCmedv"
> X <- scale(dat) # 標準化
> round(var(X)[1:5,1:5],4) # 標本分散共分散行列（最初の 5*5 だけ表示）
         Crim      Zn   Indus    Chas    Nox2
Crim   1.0000 -0.2005  0.4066 -0.0559  0.4013
Zn    -0.2005  1.0000 -0.5338 -0.0427 -0.4626
Indus  0.4066 -0.5338  1.0000  0.0629  0.7347
Chas  -0.0559 -0.0427  0.0629  1.0000  0.1008
Nox2   0.4013 -0.4626  0.7347  0.1008  1.0000
> pairs(X,pch=".") # 各ペアで散布図
> ## R の組み込み関数を用いた主成分分析と結果の表示（以下の２行だけでよい）
> f <- princomp(dat,cor=T) # 主成分分析: cor=T オプションで自動的に scale を適用する
> biplot(f,cex=0.5) # バイプロット
```

ペアごとの散布図をみると，14 変数の相互に関係がみられるが，14 次元空間に全体としてどのような構造があるかを見抜くのは困難である．バイプロットでは，14 変数の各軸は（２次元の）矢印として表現されていて，関連する変数は向きや長さがそろっている．各点は２個の合成変量を表す．

[課題 3.16]　m 次元単位ベクトル v 方向へ n 個の点 $\boldsymbol{x}^{(i)\prime}$, $i = 1, \ldots, n$ の射影を考える．合成変量 $y_i = \sum_{j=1}^{m} v_j x_{ij}$, $i = 1, \ldots, n$ を用いれば，$\boldsymbol{x}^{(i)\prime}$ を射影した点は $y_i\boldsymbol{v}$ とかける．この射影した点ともとの点の２乗

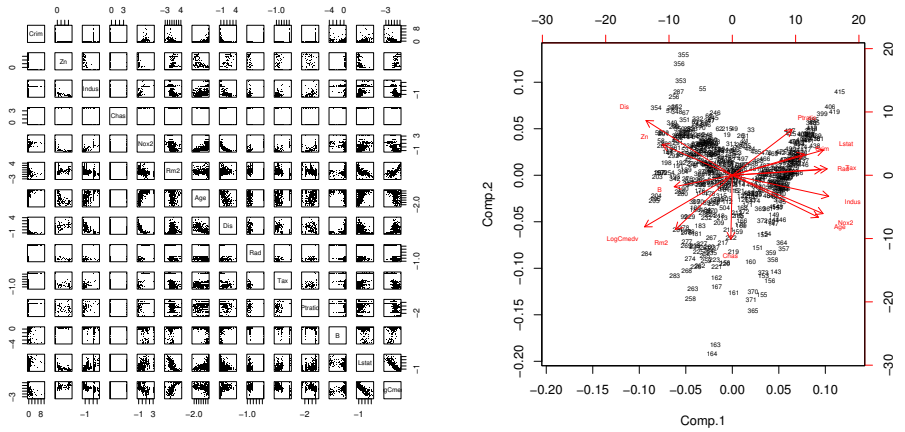


図 30　ボストン住宅価格データ（14 変数）．左図はペア毎の散布図，右図は主成分分析のバイプロット．

誤差の和が

$$\sum_{i=1}^{n} \|\boldsymbol{x}^{(i)\prime} - y_i\boldsymbol{v}\|^2 = \mathrm{tr}(X'X) - \boldsymbol{y}'\boldsymbol{y} \tag{3.13}$$

で与えられることを示せ．ヒント：合成変量をベクトル表現すれば $\boldsymbol{y} = X\boldsymbol{v}$ である．

[課題 3.17]　課題 3.16の (3.13) を最小にする v が，$\frac{1}{n-1}X'X$ の最大固有値の固有ベクトルとして得られることを示せ．ヒント：合成変量の標本分散 $\frac{1}{n-1}\sum_{i=1}^{n} y_i^2 = \|\boldsymbol{y}\|^2/(n-1)$ を最大にすればよい．

[定義 3.2]　$\boldsymbol{x}$ の標本分散共分散行列 $\frac{1}{n-1}X'X$ の固有値を大きい順に並べたものを

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$$

これに対応する固有ベクトル（長さ１）を

$$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m$$

とする．これを並べた行列を $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$ とする．このときデータ行列 X の $i = 1, \ldots, n$ 行目の要素に対応する $j = 1, \ldots, m$ 番目の主成分を

$$y_{ij} = \sum_{k=1}^{m} x_{ik}v_{kj}$$

で与える．j 番目の主成分を $\boldsymbol{y}_j = (y_{1j}, \ldots, y_{nj})'$，それを並べた行列を $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$ とすれば，

$$\boldsymbol{y}_j = \boldsymbol{X}\boldsymbol{v}_j, j = 1, \ldots, m, \quad \boldsymbol{Y} = \boldsymbol{X}\boldsymbol{V}$$

である．j 番目の主成分までの累積寄与率は

$$\frac{\lambda_1 + \ldots + \lambda_j}{\lambda_1 + \ldots + \lambda_m}, \quad j = 1, \ldots, m.$$

[課題 3.18]　主成分 $\boldsymbol{y}_j$ の標本分散が $\frac{1}{n-1}\|\boldsymbol{y}_j\|^2 = \lambda_j$ と表されることを示せ．また，$i \neq j$ に対して $\boldsymbol{y}_i$ と $\boldsymbol{y}_j$ の標本共分散が $\frac{1}{n-1}\boldsymbol{y}_i'\boldsymbol{y}_j = 0$ であることを示せ．ヒント：対角行列 $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ を用いて，$\frac{1}{n-1}\boldsymbol{Y}'\boldsymbol{Y} = \boldsymbol{\Lambda}$ を示せばよい．

[課題 3.19]　最初の r 個の主成分を用いる主成分分析は誤差最小の r 次元射影を求めていることを示せ．

[例 3.10]　例 3.9で行った主成分分析を，組み込み関数をもちいないで再度行う．

```
> Sigma <- var(X) # 標本分散共分散行列 Sigma
> a <- eigen(Sigma) # 固有値，固有ベクトルの計算
> names(a) # 固有値は values，固有ベクトルは vectors に格納される
[1] "values"  "vectors"
> rownames(a$vectors) <- colnames(X) # 変数名を設定
> colnames(a$vectors) <- names(a$values) <- paste("PC",seq(along=a$values),sep="") # PC1,PC2,...
> round(a$values,4) # 固有値をならべたベクトル (lambda_1,...,lambda_m)
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10   PC11
6.5969 1.5940 1.3118 0.9014 0.8588 0.6604 0.5521 0.4102 0.2767 0.2439 0.2209
```



```
  PC12   PC13   PC14
0.1874 0.1272 0.0582
> round(cumsum(a$values)/sum(a$values),4) # = summary(f) 累積寄与率
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10   PC11
0.4712 0.5851 0.6788 0.7432 0.8045 0.8517 0.8911 0.9204 0.9402 0.9576 0.9734
  PC12   PC13   PC14
0.9868 0.9958 1.0000
> round(a$vectors[,1:5],4) # 固有ベクトルをならべた行列 V （最初の 5 列）
              PC1     PC2     PC3     PC4     PC5
Crim       0.2462  0.1467  0.3673 -0.0764 -0.0415
Zn        -0.2397  0.2143  0.3850 -0.3615 -0.0297
Indus      0.3294 -0.1462 -0.0519  0.0292 -0.0150
Chas      -0.0051 -0.4446 -0.0144 -0.4098 -0.7704
Nox2       0.3102 -0.2666  0.0196 -0.1862  0.1731
Rm2       -0.1943 -0.3768  0.4522  0.2846  0.1315
Age        0.2943 -0.2922 -0.1558  0.0192  0.1689
Dis       -0.2951  0.3808  0.0992 -0.1443 -0.1721
Rad        0.3038  0.0380  0.4068  0.1715 -0.1877
Tax        0.3242  0.0446  0.3295  0.1154 -0.1353
Ptratio    0.2045  0.3178 -0.0981  0.5478 -0.4299
B         -0.1986 -0.0799 -0.3597  0.2617 -0.2378
Lstat      0.3117  0.1750 -0.2071 -0.3062  0.0571
LogCmedv -0.2999 -0.3587  0.1383  0.2295  0.0025
> Y <- X %*% a$vectors # 主成分の計算
> round(var(Y)[1:5,1:5],4) # Y の分散共分散行列（最初の 5*5 成分）
      PC1   PC2    PC3    PC4    PC5
PC1 6.5969 0.000 0.0000 0.0000 0.0000
PC2 0.0000 1.594 0.0000 0.0000 0.0000
PC3 0.0000 0.000 1.3118 0.0000 0.0000
```



```
PC4 0.0000 0.000 0.0000 0.9014 0.0000
PC5 0.0000 0.000 0.0000 0.0000 0.8588
> pairs(Y,pch=".") # ペアごとの散布図
> plot(Y[,1],Y[,2],type="n") # (y1,y2) の範囲で枠だけ
> text(Y[,1],Y[,2],seq(length=nrow(Y)),cex=0.5) # (y1,y2) のプロット
```

[定義 3.3]　各主成分 $\boldsymbol{y}_j$ を標準化して標本分散を 1 にしたものを $\boldsymbol{z}_j$ とする．

$$\boldsymbol{z}_j = \frac{\boldsymbol{y}_j}{\sqrt{\lambda_j}}, \quad j = 1, \ldots, m$$

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m] = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(n)} \end{bmatrix}, \quad \boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m] = \begin{bmatrix} \boldsymbol{z}^{(1)} \\ \vdots \\ \boldsymbol{z}^{(n)} \end{bmatrix}$$

ここで $\frac{1}{n-1}\|\boldsymbol{y}_j\|^2 = \lambda_j$，$\frac{1}{n-1}\|\boldsymbol{z}_j\|^2 = 1$ である．各個体 $\boldsymbol{x}^{(i)}$, $i = 1, \ldots, n$ に対応して $\boldsymbol{z}^{(i)}$ を主成分得点と呼ぶ．$\boldsymbol{\Lambda}^{-1/2} = \mathrm{diag}(\lambda_1^{-1/2}, \ldots, \lambda_p^{-1/2})$ をつかって行列表現すると，

$$\boldsymbol{Z} = \boldsymbol{Y}\boldsymbol{\Lambda}^{-1/2}$$

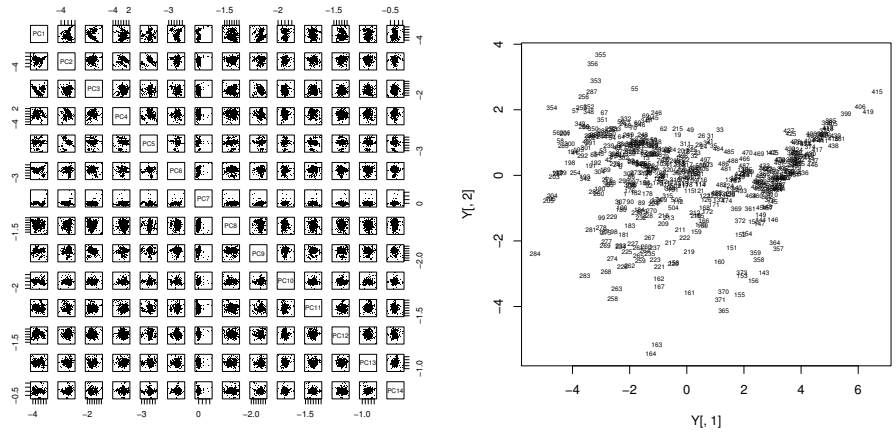


図 31　ボストン住宅価格データの主成分．（左）第 1 主成分から第 14 主成分までのペア毎の散布図．（右）第 1 と第 2 主成分のプロット．

[定義 3.4]　データセットの変量 x_j と標準化した主成分 z_k の共分散 $\frac{1}{n-1}x_j'z_k$ を並べた行列を B とする．

$$B = \frac{1}{n-1}X'Z, \quad B = [b_1, \ldots, b_m] = \begin{bmatrix} b^{(1)} \\ \vdots \\ b^{(m)} \end{bmatrix}$$

各変量 x_j, $j = 1, \ldots, p$ に対応して $b^{(j)}$ を主成分負荷と呼ぶ．

[課題 3.20]　$B = V\Lambda^{1/2}$ および $X = ZB'$ を示せ．

[注意]　各個体の主成分得点 $z^{(i)}$, $i = 1, \ldots, n$ と各変量の主成分負荷 $b^{(j)}$, $j = 1, \ldots, m$ を同時にプロットしたものを「バイプロット」と呼ぶ．ただし実際の表示では最初の数個の次元だけを使う．$x_{ij} = z^{(i)}b^{(j)\prime}$ である．

[例 3.11]　例 3.10で行った主成分分析の続き．

```
> Z <- Y %*% diag(1/sqrt(a$values)) # 主成分得点
> colnames(Z) <- colnames(Y) # 変数名の設定
> round(var(Z)[1:5,1:5],4) # 単位行列になる
    PC1 PC2 PC3 PC4 PC5
PC1   1   0   0   0   0
PC2   0   1   0   0   0
PC3   0   0   1   0   0
PC4   0   0   0   1   0
PC5   0   0   0   0   1
```



```
> B <- a$vectors %*% diag(sqrt(a$values)) # 主成分負荷
> colnames(B) <- colnames(a$vectors) # 変数名の設定
> biplot(Z[,c(1,2)],B[,c(1,2)],cex=0.5) # = biplot(f) 「バイプロット」
> biplot(Z[,c(1,3)],B[,c(1,3)],cex=0.5) # = biplot(f,choi=c(1,3))
```

[注意]　「因子分析」と呼ばれる手法は，しばしば主成分分析と混同されやすい．中身は全く違うのだが，ユーザーの立場からは形式的に類似ともいえる．あらかじめ「因子数」を p と定めておき，個体 i の $x^{(i)}$ が次のようなモデルに従うと仮定する．

$$x^{(i)} = \mu + f^{(i)}A + \epsilon^{(i)}, \quad i = 1, \ldots, n \quad \text{(i.i.d.)}$$

ここで，$x^{(i)}$ と $\epsilon^{(i)}$ は m 次元行ベクトル，$f^{(i)}$ は p 次元行ベクトル，A は $p \times m$ 行列であり因子負荷と呼ばれる．$f^{(i)}$ と $\epsilon^{(i)}$ は独立と仮定し，他にもいくつかの仮定をして因子負荷を推定する．さらに，解釈を容易にするための「回転」という操作を行う．

[例 3.12]　例 3.9で行った主成分分析を，形式的に因子分析に置き換えてみる．

```
> ## 因子分析のバイプロットを定義しておく
> biplot.factanal <- function(f,choices=1:2,...)
+   biplot(f$scores[,choices],f$loadings[,choices],...)
> ## 因子分析（因子数＝２，回転＝直交変換「バリマックス法」）
> f1 <- factanal(dat,factors=2,scores="Bartlett",rotation="varimax")
> f1
Call:
factanal(x = dat, factors = 2, scores = "Bartlett", rotation = "varimax")
```

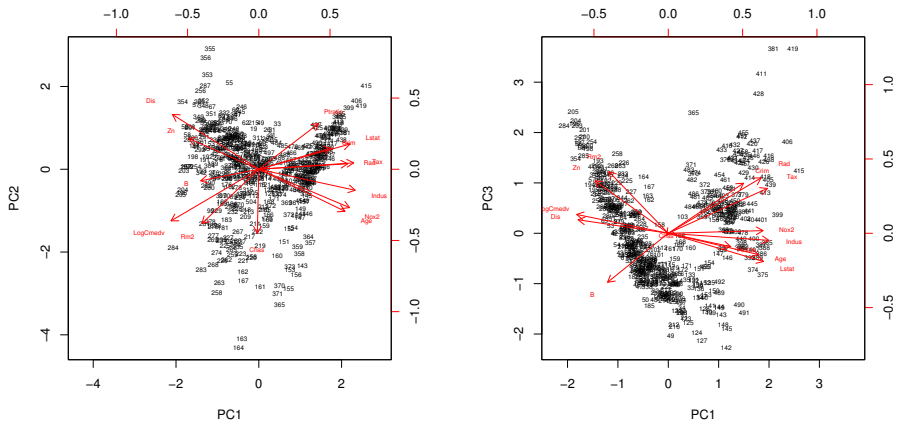



図 32　ボストン住宅価格データのバイプロット


```
Uniquenesses:
   Crim      Zn   Indus    Chas    Nox2     Rm2     Age     Dis
  0.615   0.502   0.262   0.981   0.310   0.850   0.301   0.263
    Rad     Tax Ptratio       B   Lstat LogCmedv
  0.134   0.044   0.770   0.777   0.495   0.586

Loadings:
        Factor1 Factor2
Crim     0.597   0.168
Zn      -0.197  -0.678
Indus    0.606   0.609
Chas             0.123
Nox2     0.532   0.638
Rm2     -0.234  -0.308
Age      0.371   0.750
Dis     -0.396  -0.762
Rad      0.920   0.141
Tax      0.956   0.204
Ptratio  0.459   0.140
B       -0.439  -0.173
Lstat    0.469   0.534
LogCmedv -0.521  -0.378

               Factor1 Factor2
SS loadings      4.053   3.057
Proportion Var   0.290   0.218
Cumulative Var   0.290   0.508

Test of the hypothesis that 2 factors are sufficient.
```

```
The chi square statistic is 1237.22 on 64 degrees of freedom.
The p-value is 9.61e-217
> biplot(f1,cex=0.5)
> ## 因子分析（因子数＝２，回転＝斜交変換「プロマックス法」）
> f2 <- factanal(dat,factors=2,scores="Bartlett",rotation="promax")
> f2
Call:
factanal(x = dat, factors = 2, scores = "Bartlett", rotation = "promax")

Uniquenesses:
    Crim       Zn    Indus     Chas     Nox2      Rm2      Age      Dis
   0.615    0.502    0.262    0.981    0.310    0.850    0.301    0.263
     Rad      Tax  Ptratio        B    Lstat LogCmedv
   0.134    0.044    0.770    0.777    0.495    0.586

Loadings:
         Factor1 Factor2
Crim               0.615
Zn       -0.843    0.222
Indus     0.601    0.324
Chas      0.189   -0.160
Nox2      0.667    0.214
Rm2      -0.331
Age       0.876
Dis      -0.884
Rad      -0.147    1.027
Tax                1.029
Ptratio            0.465
B                 -0.418
```



```
Lstat     0.548    0.209
LogCmedv -0.319   -0.379

               Factor1 Factor2
SS loadings      3.647   3.307
Proportion Var   0.260   0.236
Cumulative Var   0.260   0.497

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 1237.22 on 64 degrees of freedom.
The p-value is 9.61e-217
> biplot(f2,cex=0.5)
```

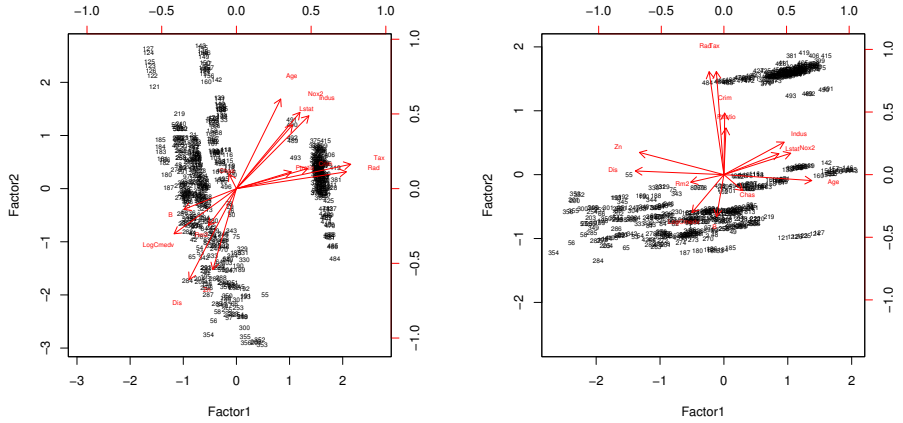



図 33　（左）因子の直交変換，（右）因子の斜交変換．