3　時系列データの周期解析への応用

天文学では時系列データの周期解析をすることがよくあります。例えば、変動天体の光度曲線や恒星の視線速度変化などが対象となります。周期解析といえば時系列データをフーリエ変換してスペクトルを推定する手法が一般的です。しかし、天文学のデータの場合、天候、昼夜、季節の影響によって欠測部分が多く発生し、サンプリング間隔は非一様なものとなります。このような場合、窓関数は複雑な形状となり、観測値から計算されるパワースペクトルには偽の信号（エイリアス）が発生します。本章ではこの問題に対する 1 次ノルム最小化のメリットについて紹介します。

まず、時系列データからスペクトルを推定する問題の基本事項を復習しましょう。時系列データとして、ある等しい時間間隔 Δt でサンプリングされた N 個のデータ点 $\boldsymbol{y} = \{y_1, y_2, \ldots, y_N\}$ からなる光度曲線を考えます。また、周波数軸データとして、ある周波数間隔 $\Delta\nu$ の $2M$ 個のデータ $\boldsymbol{x} = \{a_1, a_2, \ldots, a_M, b_1, b_2, \ldots, b_M\}$ を考えます。このとき、$\boldsymbol{y}$ と $\boldsymbol{x}$ の関係は以下のように書くことができます。

$$\boldsymbol{y} = \mathcal{F}\boldsymbol{x} \tag{1}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \cos(2\pi t_1\nu_1) & \cdots & \cos(2\pi t_1\nu_M) & \sin(2\pi t_1\nu_1) & \cdots & \sin(2\pi t_1\nu_M) \\ \cos(2\pi t_2\nu_1) & \cdots & \cos(2\pi t_2\nu_M) & \sin(2\pi t_2\nu_1) & \cdots & \sin(2\pi t_2\nu_M) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cos(2\pi t_N\nu_1) & \cdots & \cos(2\pi t_N\nu_M) & \sin(2\pi t_N\nu_1) & \cdots & \sin(2\pi t_N\nu_M) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \\ b_1 \\ b_2 \\ \vdots \\ b_M \end{pmatrix} \tag{2}$$

$$t_i = i\Delta t \quad (i = 1 \cdots N) \tag{3}$$

$$\nu_j = j\Delta\nu \quad (j = 1 \cdots M) \tag{4}$$

この表式の場合、パワースペクトル $P(\nu)$ は同じ周波数の cos と sin の係数の 2 乗和 $P_i = a_i^2 + b_i^2$ で与えられます。フーリエ変換には他に複素数での表記なども可能ですが、ここでは線形変換としてわかりやすいこの表式にします。本質的には各周波数信号の「振幅項」と「位相項」を時系列データから推定することになります。このことからわかるように、$M = N/2$ の時に $\boldsymbol{x}$ と $\boldsymbol{y}$ の要素数は等しくなります。このとき、時系列データ $\boldsymbol{y}$ はナイキスト周波数 $1/2\Delta t$ 以下を分解能 $\Delta\nu = 1/N\Delta t$ で得た周波数に対する係数ベクトル $\boldsymbol{x}$ で再構成できます。これがナイキスト・シャノンのサンプリング定理に対応します。N 個の時系列データから $M = N/2$ 点以上のパワースペクトルを推定することはできません。

実際には、観測から推定されるパワースペクトルは真のパワースペクトルと窓関数のパワースペクトルとのコンボリューションになります。つまり、無限に長く均一にサンプリングされた連続データ $Y_{\rm org}(t)$ があれば真のパワースペクトルを計算できるわけですが、実際には $Y_{\rm obs}(t) = w(t)Y_{\rm org}(t)$ しかデータは得られません。ここで w は観測データがとれた区間を 1、それ以外の区間を 0 とする関数で、窓関数と呼ばれます。観測データ $Y_{\rm obs} = wY_{\rm org}$ のフーリエ変換 ($\mathcal{F}$) はフーリエ変換の畳み込み定理をつかうと、$\mathcal{F}(wY_{\rm org}) = \mathcal{F}(w) * \mathcal{F}(Y_{\rm org})$ （$*$ は畳み込みの意味）となります。いま本当に知りたいのは $\mathcal{F}(Y_{\rm org})$ ですが、実際に観測データから計算されるのは $\mathcal{F}(wY_{\rm org})$ です。

天文学データの場合、時系列データのサンプリング間隔が一定でないことが多くあります。天体は主に夜間しか観測できませんし、天候にも大きく左右されます。その結果、窓関数は複

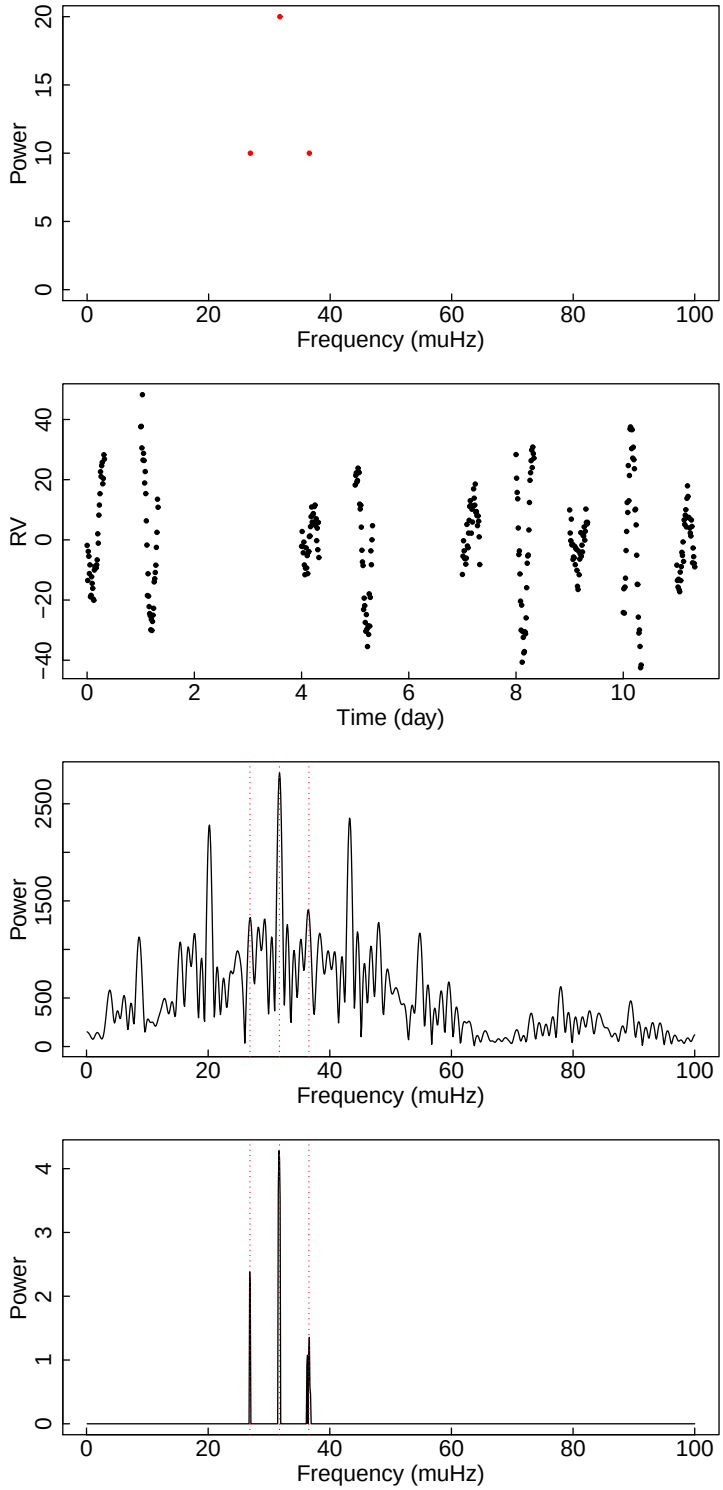

図 1: 光度曲線のフーリエ変換の実験。上から、仮定したパワー、光度曲線、通常の計算で得られたパワースペクトル、1 次ノルム最小化（LASSO）で推定されたパワースペクトル。

雑な形状をとり、観測から得られるパワースペクトルには偽の信号ができます。この信号をエイリアスといいます。例えば、興味の対象が 1 つの卓越した周期にのみある場合はエイリアスは問題になりません。しかし、周波数が近い複数の信号が存在するような場合、パワースペクトルに現れる信号が真の信号かエイリアスか、区別がつかなくなるため問題です。具体的な問題としては、複数周期をもつ脈動星やそれを使った星震学などが挙げられます。図 1 はこのような光度曲線のフーリエ変換に関する簡単なシミュレーションです。一番上のパネルは仮定したパワースペクトルを表しています。強い信号の両側に弱い信号を仮定しています。これは星振学の分野で実際に巨星で期待される周波数パターンを想定したものです。その下はシミュレートした観測データを表しています。時間の単位は「日」、1 日のうちの観測期間を 8 時間とし、さらに数日間は天候が悪かったせいでデータがとれていません。その下は通常の方法で推定されたパワースペクトルです。ここで、データ間隔が等しく欠損データが無い状態に対応して係数の数を設定しているため、観測数は係数の数よりも少なくなります。そのため、このままでは式 (2) の逆問題は解けません。つまり、式 (2) に関して次式のような状況です。

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \cancel{y_a} \\ \cancel{y_b} \\ \cancel{y_c} \\ y_d \\ y_e \\ \cancel{y_f} \\ \cancel{y_g} \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \cos(2\pi t_1\nu_1) & \cdots & \cos(2\pi t_1\nu_{N/2}) & \sin(2\pi t_1\nu_1) & \cdots & \sin(2\pi t_1\nu_{N/2}) \\ \cos(2\pi t_2\nu_1) & \cdots & \cos(2\pi t_2\nu_{N/2}) & \sin(2\pi t_2\nu_1) & \cdots & \sin(2\pi t_2\nu_{N/2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cancel{\cos(2\pi t_a\nu_1)} & \cancel{\cdots} & \cancel{\cos(2\pi t_a\nu_{N/2})} & \cancel{\sin(2\pi t_a\nu_1)} & \cancel{\cdots} & \cancel{\sin(2\pi t_a\nu_{N/2})} \\ \cancel{\cos(2\pi t_b\nu_1)} & \cancel{\cdots} & \cancel{\cos(2\pi t_b\nu_{N/2})} & \cancel{\sin(2\pi t_b\nu_1)} & \cancel{\cdots} & \cancel{\sin(2\pi t_b\nu_{N/2})} \\ \cancel{\cos(2\pi t_c\nu_1)} & \cancel{\cdots} & \cancel{\cos(2\pi t_c\nu_{N/2})} & \cancel{\sin(2\pi t_c\nu_1)} & \cancel{\cdots} & \cancel{\sin(2\pi t_c\nu_{N/2})} \\ \cos(2\pi t_d\nu_1) & \cdots & \cos(2\pi t_d\nu_{N/2}) & \sin(2\pi t_d\nu_1) & \cdots & \sin(2\pi t_d\nu_{N/2}) \\ \cos(2\pi t_e\nu_1) & \cdots & \cos(2\pi t_e\nu_{N/2}) & \sin(2\pi t_e\nu_1) & \cdots & \sin(2\pi t_e\nu_{N/2}) \\ \cancel{\cos(2\pi t_f\nu_1)} & \cancel{\cdots} & \cancel{\cos(2\pi t_f\nu_{N/2})} & \cancel{\sin(2\pi t_f\nu_1)} & \cancel{\cdots} & \cancel{\sin(2\pi t_f\nu_{N/2})} \\ \cancel{\cos(2\pi t_g\nu_1)} & \cancel{\cdots} & \cancel{\cos(2\pi t_g\nu_{N/2})} & \cancel{\sin(2\pi t_g\nu_1)} & \cancel{\cdots} & \cancel{\sin(2\pi t_g\nu_{N/2})} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cos(2\pi t_N\nu_1) & \cdots & \cos(2\pi t_N\nu_{N/2}) & \sin(2\pi t_N\nu_1) & \cdots & \sin(2\pi t_N\nu_{N/2}) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N/2} \\ b_1 \\ b_2 \\ \vdots \\ b_{N/2} \end{pmatrix} \tag{5}$$

つまり、逆フーリエ変換に相当する行列は本来正方行列であるのに、一部の行が欠けている状況です。そこで通常は、観測されていない時間のデータに 0 を入れて（0-padding)、無理やりデータを完結させてパワースペクトルを推定します。つまり、

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ 0 \\ 0 \\ 0 \\ y_d \\ y_e \\ 0 \\ 0 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \cos(2\pi t_1\nu_1) & \cdots & \cos(2\pi t_1\nu_{N/2}) & \sin(2\pi t_1\nu_1) & \cdots & \sin(2\pi t_1\nu_{N/2}) \\ \cos(2\pi t_2\nu_1) & \cdots & \cos(2\pi t_2\nu_{N/2}) & \sin(2\pi t_2\nu_1) & \cdots & \sin(2\pi t_2\nu_{N/2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cos(2\pi t_a\nu_1) & \cdots & \cos(2\pi t_a\nu_{N/2}) & \sin(2\pi t_a\nu_1) & \cdots & \sin(2\pi t_a\nu_{N/2}) \\ \cos(2\pi t_b\nu_1) & \cdots & \cos(2\pi t_b\nu_{N/2}) & \sin(2\pi t_b\nu_1) & \cdots & \sin(2\pi t_b\nu_{N/2}) \\ \cos(2\pi t_c\nu_1) & \cdots & \cos(2\pi t_c\nu_{N/2}) & \sin(2\pi t_c\nu_1) & \cdots & \sin(2\pi t_c\nu_{N/2}) \\ \cos(2\pi t_d\nu_1) & \cdots & \cos(2\pi t_d\nu_{N/2}) & \sin(2\pi t_d\nu_1) & \cdots & \sin(2\pi t_d\nu_{N/2}) \\ \cos(2\pi t_e\nu_1) & \cdots & \cos(2\pi t_e\nu_{N/2}) & \sin(2\pi t_e\nu_1) & \cdots & \sin(2\pi t_e\nu_{N/2}) \\ \cos(2\pi t_f\nu_1) & \cdots & \cos(2\pi t_f\nu_{N/2}) & \sin(2\pi t_f\nu_1) & \cdots & \sin(2\pi t_f\nu_{N/2}) \\ \cos(2\pi t_g\nu_1) & \cdots & \cos(2\pi t_g\nu_{N/2}) & \sin(2\pi t_g\nu_1) & \cdots & \sin(2\pi t_g\nu_{N/2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cos(2\pi t_N\nu_1) & \cdots & \cos(2\pi t_N\nu_{N/2}) & \sin(2\pi t_N\nu_1) & \cdots & \sin(2\pi t_N\nu_{N/2}) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N/2} \\ b_1 \\ b_2 \\ \vdots \\ b_{N/2} \end{pmatrix} \tag{6}$$

図 1 をみると、仮定した 3 つの周波数成分のうち最も強いものは確認できますが、その両側の

弱い信号は埋もれてしまい、さらに離れたところにむしろ強いエイリアスが現れていることがパワースペクトルからわかります。これではどれが本当の信号か判別できません。

1 次ノルム最小化はこの問題に対して強力なツールを与えてくれます。上記のような問題の場合、周波数空間で信号はスパースだと考えられます。つまり、周波数空間のほとんどの成分が 0 で、意味のある信号はわずかしかありません。観測データの誤差が正規分布のような対称な分布だとすると、この問題はこれまでと同様、1 次ノルム最小化をつかって、

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \mathcal{F}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \tag{7}$$

というモデルが考えられます。行列 $\mathcal{F}$ は今回は前章までのようなランダム行列ではなく、式 (2) のような逆フーリエ変換に相当する要素が入ります。データに対するこのモデルの最適化部分は 1 章で紹介した R の glmnet で実装できます。Kato & Uemura (2012)[1] にはそのサンプルスクリプトが Appendix に記載されています。それを使って、上述のシミュレーションデータを再解析すると図 1 の一番下のパワースペクトルが得られます。その上の従来法のものと比較するとその違いは明らかで、真の信号以外のエイリアスが消えており、仮定したパワースペクトルがよく再構成されています。これは 1 次ノルム最小化によって周波数空間でスパースな解が選択されたためです。エイリアスの信号を入れるとスパース度は減り、データとの適合率も減るので、そのような解は選択されなかったわけです。

天文データの周期解析に関するこのようなモデルについてより詳しくは文献 [1] をご覧ください。もちろん、このモデルは「時系列データは複数の周期信号の重ね合わせで説明できる」(つまり、周波数空間でスパース）という事前情報が正しい場合にのみ、適応可能です。一方で、例えばパワースペクトルがべき乗型 ($P \propto \nu^{\alpha}$) の場合はこのモデルは使えません。また、上の例では $\boldsymbol{x}$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を全て独立に考えて 1 次ノルム最小化を使いましたが、周波数空間で信号がスパースであることを考えると、より正確には a_i と b_i が同時にゼロになる、もしくは非ゼロの値をもつ、ようなモデルも考えられます。これは group LASSO と呼ばれる手法によって可能になります。group LASSO とは係数ベクトル $\boldsymbol{x}$ を N 個のグループに分けて、$\sum_{j=1}^{N}\sqrt{\sum_{i\in G_j} x_i^2}$（ただし G_j は j 番目のグループに含まれる添字の集合）を制約項とするモデルです。今回の場合は、$\sum_{j=1}^{N}\sqrt{a_j^2+b_j^2}$ となります。

次回は同じフーリエ変換でも 2 次元画像のフーリエ変換である電波干渉計への応用を紹介します。

参考文献

[1] Kato, T., Uemura, M.: Period Analysis using the Least Absolute Shrinkage and Selection O perator (Lasso), *Publ. Astrn. Soc. Japan*, 64, 122 (2012)

以下は本章で示した光度曲線と周期解析のシミュレーションに用いた R のスクリプトです。周期解析部分（perlasso）は文献 [1] の Appendix にあるものと同じです。

```
library(lars)
library(glmnet)
```

```
### Functions from Kato & Uemura (2012)
seqfreq <- function(a,b,...) {
return(1/seq(1/b,1/a,...))
}

makematlasso <- function(d,p,ndiv=2) {
  nd <- length(p)
  m <- matrix(0,nrow(d),nd*ndiv)
  for (i in 1:nd) {
    ph <- (((d$V1)/p[i]) %% 1)*pi*2
    for (j in 0:(ndiv-1)) {
      m[,i+nd*j] <- sin(ph+pi*j/ndiv)
    }
  }
  return(m)
}

perlasso <- function(d,p,ndiv=2,alpha=1,cv=FALSE) {
  nd <- length(p)
  mat <- makematlasso(d,p,ndiv)
  y <- d$V2 - mean(d$V2)
  m <- glmnet(mat,y,alpha=alpha)
  ndim <- m$dim[2]
  pow <- matrix(0,nd,ndim)
  for (i in 1:ndim) {
    v <- m$beta[,i]
    for (j in 0:(ndiv-1)) {
      pow[,i] <- pow[,i] + v[(nd*j+1):(nd*(j+1))]^2
    }
  }
  nmin <- NULL
  gcv <- NULL
  if (cv) {
    gcv <- cv.glmnet(mat,d$V2,alpha=alpha)
    minl <- gcv$lambda.min
    nmin <- which.min(abs(m$lambda-gcv$lambda.min))
  }
  r <- list(pow=pow,p=p,lambda=m$lambda,nmin=nmin,m=m,gcv=gcv,mat=mat)
  class(r) <- c("lassopow",class(r))
  return(r)
}

### Demo
```

```
nf <- 500
fnyq <- 50

# Assuming a spectrum
n <- 2*nf
x <- rep(0,n)
x[9] <- 1.0
x[29] <- 1.0

# Simulating data
df <- fnyq/nf
f <- seq(df,fnyq,df)
p <- 1/f
dt <- 1/(2*fnyq)
t <- seq(dt,dt*n,dt)

m <- matrix(0,n,n)
for (i in 1:nf) {
  ph <- ((t*f[i]) %% 1)*pi*2
  for (j in 0:1) {
    m[,i+2*j] <- sin(ph+pi*j/2)
  }
}

y <- m %*% x

# Window and data
id1 <- which(t<0.3 | (t>1.0&t<1.3) | (t>2.0&t<2.3) | (t>7.0&t<7.3) | (t>8.0&t<8.3) | (t>9.0&t<9
w1 <- rep(0,n)
w1[id1] <- 1.0
y1 <- w1*y

# Solution of a simple linear regression
dft <- lm(y1~m)
py1<- (dft$coefficients[1:nf])^2
py1 <- data.frame(V1=f,V2=py1)

# LASSO analysis
d1 <- data.frame(V1=t[id1],V2=y1[id1])
l1 <- perlasso(d1,p)

### Figures and plots
```

```
# postscript("perlasso_demo.eps",horizontal=FALSE,height=8,width=6,paper="special")
par(mar=c(3.2, 3.2, 1.5, 0.5), mgp=c(2.0, 0.7, 0), mfrow=c(4,1))
plot(f,x[1:nf],xlim=c(0,6),type="l",xlab="Frequency",ylab="Power")
plot(d1,xlab="Time", ylab="Flux",ylim=c(-2.0,+2.0),pch=16,cex=0.8)
lines(t,y)
plot(py1,xlim=c(0,6),type="l",xlab="Frequency",ylab="Power")
plot(f,l1$pow[,length(l1$lambda)],xlim=c(0,6),type="l",col="red",xlab="Frequency",ylab="Power")

# dev.off()
```