

レポート課題1-2

- レポート課題の 1-1 と分けて提出して下さい
- レポートの提出は、keio.jp を使って下さい。
 - TeX, MsWord 等で作成して下さい。
 - プログラムはテキストファイルで提出して下さい。すべてを合わせて圧縮して下さい。
- 書くべき内容に関しては特には述べません。すべて常識的に判断してください。
- 締め切りは、12/18(月曜日)一杯とします。

課題1-2

プログラムはほとんど書かれています。

- AIC規準とBIC規準を用いたモデル選択を試してみましょう。
 - 真のモデルは、ある次数の1変数多項式とします
 - 次数を変えた1変数多項式で回帰します
 - AIC, BIC, leave-one-out cross validation を用いて、最も適したモデルを推定します。
 - BICは、結果的にMDLと同じ式で用います。
 - 真のモデルと比較します
 - 真のモデルの次数や係数を変えて、複数回実験してみましょう
- 言語はなんでも結構ですが、次に、Rを用いたプログラム例を示します。
- 問題は次のようになります。
 - BICとは何か、簡略に説明してください。
 - プログラム(各関数および実行)が何を目的とするものか、簡略に説明して下さい。
 - R以外の言語を用いる場合は、そのプログラムを示して下さい。
 - 乱数は、各自、set.seed を用いて、いくつかの異なる乱数を元に試してみてください

課題1-2 (続)

- 問題は次のようになります(続)。
 - 真のモデルの次数を変えて試して下さい
 - 真のモデルである多項式は2次、回帰する多項式は3次として、AIC, BIC、を自分で計算して下さい(他の条件でも結構です。計算しやすい条件を設定してください)。言うまでもないことですが、ノイズは、正規分布です。プログラムでは、平均0、標準偏差0.1です。
 - 真のモデルとして、三角関数も試して下さい。
 - lapply と sapply という(ある人にとっては)風変わりな関数を使っています。説明が必要であれば、申し出て下さい。

```
# 線形回帰モデルに LOOCV を適用する
# 入力: lm() を用いて回帰したモデル
# 出力: leave-one-out cross validation のスコア
loocv.lm <- function(m) mean((residuals(m)/(1-hatvalues(m)))^2)

# 一変数多項式の値を返す
# sum_i (-1)^i x^i + epsilon
# 入力:
# 出力:
polynomialValues <- function(nPoints, degree) {
  x <- runif(nPoints, min=-2, max=2)
  monomialValues <- poly(x, degree=degree, raw=TRUE)
  coefs <- rep(c(-1,1), length.out=degree)
  y.true <- monomialValues %*% coefs
  y <- y.true + rnorm(nPoints,0,0.1)
  return(data.frame(x=x,y=y))
}
```

```

# 機能:
# 入力:
# 出力:
# 前提: data は、名称 x,y のコラムを持つデータフレーム、degreeは整数かつ>=1
polynomialFit <- function(data, maxDegree)
  lapply(1:maxDegree, function(d) lm(y~poly(x, degree=d), data=data) )

# 機能:
# 入力:
# 出力:
# 前提: (もしあれば)
selectors <- function(mList) {
  Rsq <- which.max(sapply(mList, function(m) summary(m)$r.sq ))
  Rsq.adj <- which.max(sapply(mList, function(m) summary(m)$adj.r.sq ))
  loocv <- which.min(sapply(mList, loocv.lm))
  aic <- which.min(sapply(mList, AIC))
  bic <- which.min(sapply(mList, BIC))
  selections <- c(Rsq=Rsq,Rsq.adj=Rsq.adj,LOOCV=loocv,AIC=aic,BIC=bic)
  return(selections)
}

selectors(polynomialFit(polynomialValues(nPoints=20, degree=3),
  maxDegree=10))
summary(t(replicate(1000, selectors(polynomialFit
  (polynomialValues(nPoints=20, degree=3), maxDegree=10))))))
summary(t(replicate(1000, selectors(polynomialFit
  (polynomialValues(nPoints=1000, degree=3), maxDegree=10))))))

```