レポート課題1

- 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。提出は pdf 形式でも結構です。
- 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、11/30（月曜日）一杯とします。

課題1-1

- n個の属性（属性値は各2個）を持つデータを2クラスに分類する問題を考える。Naïve Bayes分類器と（条件付き独立性を仮定しない）Bayes分類器のパラメータ数はそれぞれいくつか
- Training dataset/validation dataset/test dataset の違いを記して下さい。Test dataset は一度しか使ってはいけませんが、その理由を述べて下さい。また、時系列予測（例えば、日経平均の明日の終値を予測する）の時、training dataset の時刻は validation dataset の時刻より前でなければならないし、validation dataset の時刻は、test dataset の時刻より前でなければならない。その理由を述べて下さい。
- 時系列予測の時には、cross-validation が使えません。その理由を述べて下さい。
- 次のデータを学習データとして、決定木を作成するときの、根の属性の選択のみを行って下さい。手計算（決定木学習ツールは使わないという意味です。Excelがおすすめです）で行ってください。

日付	地域	タイプ	収入	既顧客	結果
2003/10/3	郊外	広い一戸建て	高	No	無答
2003/9/4	郊外	広い一戸建て	高	Yes	無答
2002/4/2	田園地帯	広い一戸建て	高	No	返信あり
2003/1/18	都会	一戸建て	高	No	返信あり
2003/4/3	都会	一戸建て	低	No	返信あり
2002/10/15	都会	一戸建て	低	Yes	無答
2002/10/15	田園地帯	一戸建て	低	Yes	返信あり
2001/3/2	郊外	マンション	高	No	無答
2003/5/4	郊外	一戸建て	低	No	返信あり
2003/1/2	都会	マンション	低	No	返信あり
2003/10/3	郊外	マンション	低	Yes	返信あり
2003/10/3	田園地帯	マンション	高	Yes	返信あり
2003/4/8	田園地帯	広い一戸建て	低	No	返信あり
2002/5/6	都会	マンション	高	Yes	無答

課題1-2

- 過学習の起こり方を、多項式近似と中間層1層のニューラルネットワークで調べてみて下さい。入力は1次元としましょう（直観的に分かるように）
  - RまたはWekaを使って下さい。勿論、C等のプログラミング言語で実装して下さっても結構です。
    - プログラムする場合、ニューラルネットワークの学習アルゴリズムは、普通のBPで結構です。活性化関数は tanh(x) がお薦めです
    - 言語は、何でもよい。Excel 内の visual basic でも結構です！
    - ただし、ソルバーを使う方法では、うまく行かない場合があります（有名なXOR問題がそうです）。
- データは何でもよいのですが、少なくとも次のデータで示してみて下さい（正解はy=2*x + 8 としましょう）。
  - x: -4, -3, -2, -1, 0,1,2,3,4,5
  - y: 0, 1.6, 3.6, 6.2, 7.4, 10.4, 11.6, 14.5, 15.5, 18.6
- 学習用データ(HW01-poly00.csv)とテスト用データ(HW01-poly-test.csv)をcsvファイルで用意しました。Weka用と考えて下さい（RやC等では、プログラムで生成すればよいので）。

課題1-2（続）

- 多項式回帰の場合、次数は、9次まで試してください。
  - Wekaでは線形回帰しかありませんので、入力値として前記xの1次～9次の冪乗の値を作ります(HW01-poly01.csv). しかし、これではうまくいかないので、正規化します。正規化した上に、直交化したものを HW01-poly02.csv におきます。HW01-poly02-test.csv がテストデータです。
  - Wekaでは、LinearRegressionを使って下さい。なお、パラメータattributeSelectionMethod を No attribute selection にして下さい。また Test optionsは "Use training set"にして下さい。テスト誤差を測るときは、"Supplied test set" にして下さい。
  - Rでも線形回帰しかありませんので、やはり線形回帰関数を用います。入力値は、Wekaと同様に前記xの1次～9次の冪乗の値を作ります(poly(…, raw=T)とする)。Weka同様に、うまくいかないので、正規化直交化をします(poly(…, raw=F)とする)。
- ニューラルネットワークによる回帰の場合、出力素子を線形素子にします。中間素子数は1から50くらいまで試してみて下さい。結合荷重初期値は乱数で決めるので、試行ごとに結果は異なるはずです。
  - 入力が一次元ですから、中間素子数を h とすると、自由度は 3h+1 となります。中間素子数50であれば、間違いなく過学習に陥ります（陥るはずです）。
  - しかし、学習を早期に停止すると、過学習を回避することが可能になる。停止する条件を、学習の繰り返し回数ではなく、誤差の絶対値（abstol）や相対値（reltol）にし、早期に停止することにより、過学習が抑えられることを、示してください。

課題1-2（続）

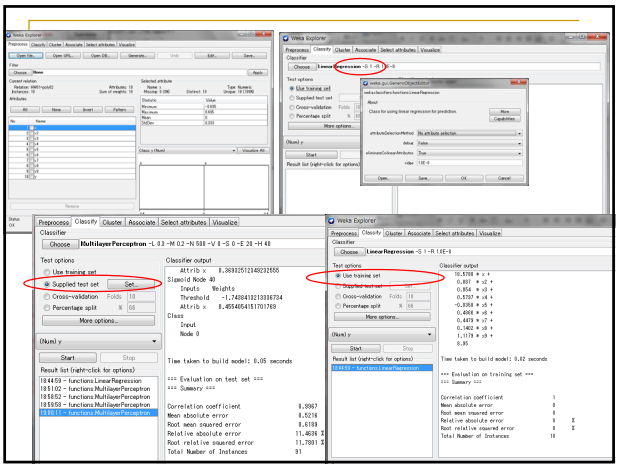
- 誤差を調べるにあたって、データの説明変数の値範囲 [0,9] の外まで調べて下さい。多項式回帰とニューラルネットワークによる回帰の性格が異なることが分かります。
  - 例えば、プログラムでの印字を次のように変えたら、挙動がよく分かります。
    （肝心のプログラム本体は、後の方のスライドにあります）

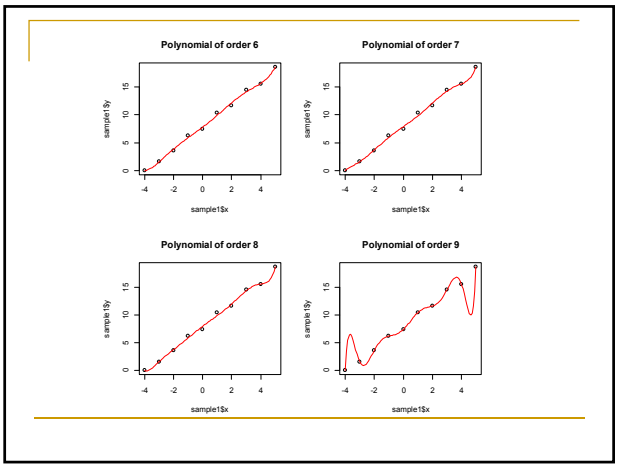
```
plot(sample1$x, sample1$y, type="p", xlim=c(-6,7), ylim=c(-5,25))

curve(pol2, col="red", add=T, xlim=c(-6,7), ylim=c(-5,25))
```

```
plot(sample1$x, sample1$y, type="p", xlim=c(-6,7), ylim=c(-5,25))

curve(plotf, col="red", add=T, xlim=c(-6,7), ylim=c(-5,25))
```






```
x <- seq(-4,5)
y <- c( 0, 1.6, 3.6, 6.2, 7.4, 10.4, 11.6, 14.5, 15.5, 18.6)

sample1 <- data.frame(x, y)
plot(sample1$x, sample1$y, type="b")

# 線形回帰の方法は2つあります。
# これが第一
fit1 <- lm(sample1$y ~ sample1$x)
fit2 <- lm(sample1$y ~ sample1$x + I(sample1$x^2))
fit3 <- lm(sample1$y ~ sample1$x + I(sample1$x^2) + I(sample1$x^3))

# これが第二
fit2b <- lm(sample1$y ~ poly(sample1$x, 2, raw=TRUE))
fit3b <- lm(sample1$y ~ poly(sample1$x, 3, raw=TRUE))

# プロット方法です。
plot(sample1$x, sample1$y, type="p")
pol2 <- function(x) fit2$coefficient[3]*x^2 + fit2$coefficient[2]*x + fit2$coefficient[1]
curve(pol2, col="red", add=T)
```

```
# ニューラルネットワークの場合
x <- seq(-4,5)
y <- c(0, 1.6, 3.6, 6.2, 7.4, 10.4, 11.6, 14.5, 15.5, 18.6)
sample1 <- data.frame(x, y)

library(nnet)
## helpt(nnet) とすればnnetに関わるヘルプ情報が見られます
# 学習は、abstolやreltolを指定することによっても止められます
sample1.nn <- nnet(sample1$x, sample1$y, size=50, linout=T, maxit=100)

plot(sample1$x, sample1$y, type="p")
# curveを使ってプロットするには工夫が必要
plotf <- function(x) predict(sample1.nn, matrix(x))
curve(plotf, col="red", add=T)

# テストエラーを求めるには、例えば、
tx <- seq(-4,5,by=0.1)
ty <- tx*2+8
ty.pred <- predict(sample1.nn, matrix(tx))
mean((ty-ty.pred)^2)

# 次のようにもできるのだが、後が面倒なので、今回は省略
sample1.nn <- nnet(sample1$y ~ sample1$x, size=20, linout=T, maxit=1000)
```

このパラメータでは、過学習します

```
# 学習エラーは
mean((y-predict(sample1.nn,matrix(x)))^2)
```