Capitulo 5

Modelos de Regresion Lineal

Los modelos predictivos sonmodelos matematicos ecuaciones que nos permiten predecir algo sobre una variable de interes, de una o mas variables que han sido relatadas.

Modeling the witeside data

The witeside data de el paquete MASS provee una simple y facil interpretacion de ejemplos ilustrando varios aspectos de los problemas de regresion linea, llamar estos data frame incluye tres variables

Temp una medida de la temperatura externa for cada semana durante dos diferentes temporadas de gas consumido cada semana

Insul una variable categorica con dos valores Before para esos, registros de la primera temporada antes de que se haya instalado la insolacion, y After para esos registros de la segunda temporada despues de la insolacion instalada.

Describiendo lineas en el plano

Antes de describir el problema de lineas, es necesario decir cuantas lineas en el plano son representadas matematicamente, diferentes representaciones son posibles.La funcion abline proove una manera extremadamente conveniente de mostrar las lineas en la grafica una vez que nosotros sabemos la pendinte e interceptamos los parametros que describen esas lineas pero nos da una coleccion de puntos.

Adecuando el whiteside data

```
library(MASS)
linearModelA <- lm(Gas ~ Temp, data = whiteside)
```

El primer argumento de la funcion GAS ~ Temp que usa R es la formula de interface que le dice a la funcion Lm para adecuar el modelo que predice Gas la variable de el simbolo ~ en este caso temperatura

```
names(linearModelA)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

Los coeficientes de los elementos contienen la iterseccion, y los parametros de la pendiente para el modelo lineal es el que mejor se adecua al dato, y la funcion abline a sido designada para mirar en esos parametros cuando el modelo es dado para la funcion lm.

```
linearModelA$coefficients
```

```
## (Intercept)        Temp
##   5.4861933  -0.2902082
```

Dado que el parámetro Temp es negativo, la línea se inclina hacia abajo y porque los valores de Gas son todos positivos, el parámetro de intercepción debe ser positivo para garantizar predicciones positivas. La función de resumen genérico proporciona una descripción más completa del modelo

```
summary(linearModelA)
```

```
##
## Call:
## lm(formula = Gas ~ Temp, data = whiteside)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.6324 -0.7119 -0.2047  0.8187  1.5327
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.4862     0.2357  23.275  < 2e-16 ***
## Temp         -0.2902     0.0422  -6.876 6.55e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8606 on 54 degrees of freedom
## Multiple R-squared:  0.4668, Adjusted R-squared:  0.457
## F-statistic: 47.28 on 1 and 54 DF,  p-value: 6.545e-09
```

Aplicada a un objeto de modelo de regresión lineal, la función de resumen muestra:

1. La llamada a la función utilizada para ajustar el modelo, identificando el tipo de modelo (es decir, la función lm), las variables incluidas en el modelo y el marco de datos del que se obtuvieron estas variables;
2. El resumen de cinco residuales de Tukey correspondiente a los errores de predicción {e k} en la ecuación, calculado a partir de los datos y los parámetros del modelo;
3. Los coeficientes del modelo discutidos anteriormente, junto con algunas caracterizaciones relacionadas discutidas en el siguiente párrafo;
4. Cinco caracterizaciones de bondad de ajuste, discutidas brevemente a continuación.

Las cinco caracterizaciones de ajuste incluidas en este resumen son el error estándar residual, el R cuadrado múltiple y el R cuadrado ajustado, y el estadístico F y su valor p asociado.

La parte del resultado resumido con la etiqueta “Coeficientes” es un marco de datos que proporciona los parámetros estimados del modelo en la columna Estimación, y tres características de estas estimaciones:

Std, Error, valor de t y Pr (> | t |). La primera de estas columnas proporciona el error estándar asociado con cada coeficiente, una medida de la precisión con la que se ha estimado.

La siguiente columna proporciona el estadístico t correspondiente derivado del error estándar, y la última columna proporciona el valor p asociado con este estadístico t.

Sobreajuste y division de datos

Un peligro de este enfoque es el sobreajuste: si aumentamos la complejidad de nuestro modelo al agregar más términos, se garantiza que se adaptará mejor a los datos.

Si hacemos que el modelo sea demasiado complejo, comenzamos a ajustar todos los detalles en los datos, incluidos aquellos que reflejan la presencia inevitable de ruido, y no solo el “comportamiento general” de interés.

Ejemplo de sobreajuste de datos

```r
library(MASS)
set.seed(331)
x <- seq(1, 10, 1)
y <- rnorm(10)
full <- data.frame(x = x, y = y)
xT <- sort(sample(x, size = 5, replace = FALSE))
train <- full[xT, ]
```

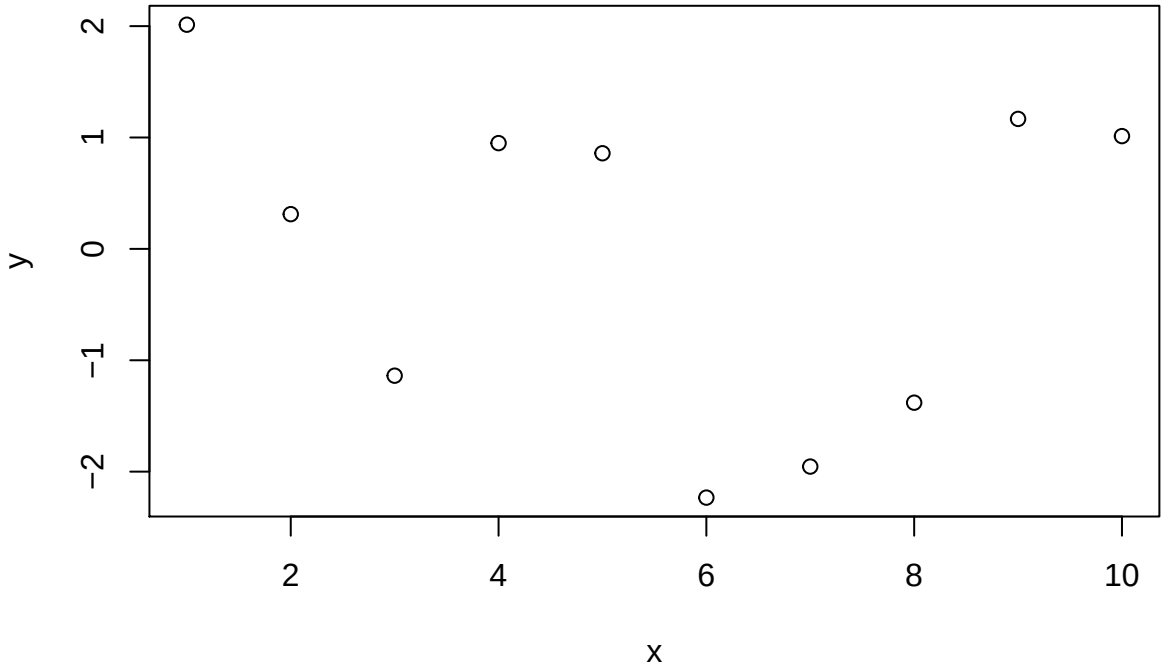
Aquí, x es la secuencia de enteros del 1 al 10, e y es 10 muestras de un media aleatoria simulada, variable aleatoria gaussiana de varianza unitaria.

Es bien sabido que “dos puntos determinan una línea”, y es más general que n puntos determinan un polinomio de grado n - 1. Por lo tanto, tres puntos son suficientes para determinar un polinomio cuadrático, cuatro puntos determinan un cúbico polinomio, y cinco puntos determinan un polinomio de cuarto orden de esta forma

y = a 0 + a 1 x + a 2 x 2 + a 3 x 3 + a 4 x 4 .

```
model0 <- lm(y ~ 1, data = train)
model1 <- lm(y ~ x, data = train)
model2 <- lm(y ~ x + I(x^2), data = train)
model3 <- lm(y ~ x + I(x^2) + I(x^3), data = train)
model4 <- lm(y ~ x + I(x^2) + I(x^3) + I(x^4), data = train)
```

```
yHat0 <-predict(model0, newdata=full)
yHat  <-predict(model1, newdata=full)
yHat2 <-predict(model2, newdata=full)
yHat3 <-predict(model3, newdata=full)
yHat4 <-predict(model4, newdata=full)
plot(x,y)
```



El entrenamiento la validacion Division

En términos más generales, el enfoque de división de datos en el aprendizaje automático la comunidad divide aleatoriamente nuestro conjunto de datos original en tres, excluyéndose mutuamente 1. un conjunto de entrenamiento utilizado para ajustar los parámetros del modelo; 2. un conjunto de validación utilizado para tomar decisiones de estructura del modelo; 3. un conjunto de prueba o conjunto de retención, utilizado para la evaluación final del modelo.

Aplicaciones para whiteside data

La única desventaja es que este conjunto de datos es demasiado pequeño para proporcionar un ejemplo “rico en datos” para la estrategia de partición de datos de tres vías que se acaba de describir. Sin embargo, ilustra algunas de las desventajas de aplicar particiones aleatorias a pequeños conjuntos de datos, al tiempo que ilustra claramente las ideas básicas detrás de este enfoque y sus resultados típicos.

```
#TVHflags <- TVHsplit(whiteside, split = c(0.5, 0.5, 0))
#trainSub <- whiteside[which(TVHflags == "T"), ]
#validSub <- whiteside[which(TVHflags == "V"), ]
```

```
#linearModelB <- lm(Gas ~ Temp, data = trainSub)
#GasHatBT <- predict(linearModelB, newdata = trainSub)
#GasHatBV <- predict(linearModelB, newdata = validSub)
```

```
#GasHatBT <- predict(linearModelB, newdata = trainSub)
#GasHatBV <- predict(linearModelB, newdata = validSub)
#ek <- GasHatBT - trainSub$Gas
#mseBT <- mean(ek^2)
#mseBT
## [1] 0.5179938
```

Dos modelos utiles de validacion

La primera herramienta es una caracterización gráfica simple que nos permite ver desviaciones sistemáticas de varios tipos, así como puntos de datos individuales que el modelo predice mal.

La segunda herramienta es una medida numérica de qué tan bien el modelo predice los datos de validación, basado en ideas de estadísticas clásicas.

```
library(MASS)
badTrain <- whiteside[1:28, ]
badValid <- whiteside[29:56, ]
badModel <- lm(Gas ~ Temp, data = badTrain)
badGasHatT <- predict(badModel, newdata = badTrain)
badGasHatV <- predict(badModel, newdata = badValid)
```

Regresion con multiples predictores

Los modelos de regresión lineal considerados hasta ahora predicen una respuesta numérica variable de una sola covariable numérica o variable de predicción.

A menudo tenemos varias variables que son capaces de proporcionar predicciones parciales y, en estos casos, generalmente podemos lograr mejores predicciones incorporando dos o más de estas variables.

```
#Cars93Flag <- TVHsplit(Cars93, split = c(0.5, 0.5, 0.0))
#Cars93T <- Cars93[which(Cars93Flag == "T"), ]
#Cars93V <- Cars93[which(Cars93Flag == "V"), ]
#Cars93Model5 <- lm(Horsepower ~ Price, data = Cars93T2)
#summary(Cars93Model5)
##
## Call:
## lm(formula = Horsepower ~ Price, data = Cars93T2)
```

Usando predictores categoricos

Los problemas de modelado de regresión lineal discutidos hasta ahora intentan predecir una variable de respuesta numérica a partir de una o más variables predictoras numéricas.En particular, una clase extremadamente importante son las variables categóricas como la variable Insul en el marco de datos del lado blanco, que toma dos valores distintos: Before, lo que significa que las observaciones de datos asociadas se realizaron antes de instalar el aislamiento del hogar, y After, lo que significa que las observaciones de datos fueron hecho después de que se instaló el aislamiento.

Una característica clave de las variables categóricas es que no podemos realizar operaciones aritméticas como sumar o multiplicar con ellas. Aún así, como lo ilustra el siguiente ejemplo, las variables categóricas

pueden incorporarse en los modelos de regresión lineal como predictores, y estas variables pueden mejorar significativamente la calidad de la predicción.

```
#library(MASS)
#linearModelC <- lm(Gas ~ Temp + Insul, data = trainSub)
#summary(linearModelC)
#GasHatCV <- predict(linearModelC, newdata = validSub)
```

Interaccion en modelos de regresiones lineares

El modelo de regresión lineal que incluye tanto la variable real Temp y la variable categórica Insul podrían escribirse como dos modelos lineales estándar, cada uno de los cuales predice Gas de Temp con las mismas pendientes, pero diferentes términos de intercepción.

En términos prácticos, esto significa que la dependencia de Gas en Temp en este modelo no implica el valor de la otra variable Insul, si esta dependencia de Temp variara con el valor de Insul, diríamos que existe una interacción entre las variables Temp e Insul.

Variable de transformacion en regresiones lineales

El término "regresión lineal" se refiere al problema de ajustar modelos predictivos que dependen linealmente de los parámetros desconocidos.

Esto significa que es posible y, a veces extremadamente útil para construir modelos de regresión lineal que implican transformaciones no lineales de una o más de las variables de predicción.

Regresiones Robustas

La base para el ejemplo considerado aquí es el marco de datos del lado blanco del paquete MASS, con los valores de Temp y Gas ambos modificados para una sola observación.

Esta modificación en particular fue motivada por la práctica popular, aunque desafortunada, de representar valores numéricos faltantes con un único código numérico absurdamente grande como 9999. Esta convención probablemente se origina en un momento en que los conjuntos de datos eran relativamente pequeños, principalmente consistentes en números datos que abarcan un rango de valores relativamente limitado y se comparten con un grupo relativamente pequeño de compañeros de trabajo. En tal entorno, los analistas sabrían buscar estos grandes números y dejarlos fuera de sus análisis o aplicar alguna estrategia de imputación simple, pero la práctica ha persistido hasta cierto punto incluso hoy en día, donde los conjuntos de datos a menudo son extremadamente grandes y la recopilación de datos y las comunidades de análisis de datos están completamente separadas. Aquí, suponga que los valores de Temp y Gas para la observación 40 del marco de datos en el lado blanco no se observaron, pero se registraron utilizando el código de valor faltante 9999. A continuación, comparamos los coeficientes del modelo obtenidos en los siguientes cuatro escenarios:

1. un modelo ordinario de mínimos cuadrados ajustado utilizando el procedimiento lm, aplicado al datos originales en blanco (sin modificar).
2. un modelo ordinario de mínimos cuadrados ajustado utilizando el procedimiento lm, aplicado al datos modificados en blanco.
3. un modelo robusto de regresión ajustado usando el procedimiento lmrob, aplicado al datos originales en blanco.
4. un modelo robusto de regresión ajustado usando el procedimiento lmrob, aplicado al datos modificados en blanco.

```
library(MASS)
head(whiteside)
```

```
##    Insul Temp Gas
## 1 Before -0.8 7.2
## 2 Before -0.7 6.9
## 3 Before  0.4 6.4
## 4 Before  2.5 6.0
```

```
## 5 Before  2.9 5.8
## 6 Before  3.2 5.8
```

```
whitesideMod <- whiteside
whitesideMod$Temp[40] <- 9999
whitesideMod$Gas[40] <- 9999
model1 <- lm(Gas ~ Temp, data = whiteside)
model1 <- lm(Gas ~ Temp, data = whitesideMod)
model1 <- lm(Gas ~ Temp, data = whiteside)
model1 <- lm(Gas ~ Temp, data = whitesideMod)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2), data = train)
##
## Residuals:
##       1       3       5       6       7
##  0.4084 -1.3247  1.8220 -0.7956 -0.1101
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.41404    2.90631   0.831    0.494
## x           -0.84326    1.72877  -0.488    0.674
## I(x^2)       0.03356    0.21330   0.157    0.889
##
## Residual standard error: 1.716 on 2 degrees of freedom
## Multiple R-squared:  0.5699, Adjusted R-squared:  0.1398
## F-statistic: 1.325 on 2 and 2 DF,  p-value: 0.4301
```

Esencialmente, el procedimiento lmrob detecta observaciones de datos periféricos y las pondera hacia abajo, ajustando un modelo de regresión lineal que refleja el comportamiento de la porción nominal o "no periférica" de los datos. Aquí, esta porción nominal de los datos representa el marco de datos del lado blanco original, por lo que esperamos ver el robusto procedimiento de regresión es aproximadamente el mismo resultado en ecenarios (3) y (4) que obtenemos utilizando la regresión lineal estándar aplicada al conjunto de datos no contaminado en el Escenario (1).

Como lo demuestran los siguientes resultados, los resultados obtenidos en el Escenario (2) son dramáticamente diferentes, y están totalmente dominados por los valores de datos periféricos en el marco de datos modificado.