En esta pràctica comparamos la velocidad de ordenación de vectores de dos algoritmos que ordenan vectores i parece que uno de ellos es por lo menos más rápido que el más lento de ellos. Bien echo, claro, con significación estadística.

Roser González Valeri, José Antonio Rius Cobo i Erik Carrasco Alastrué.

Novembre 2011

Resum

Objectiu: El nostre objectiu es fer la pràctica de l'últim bloc per millorar la nota –perquè molt ens sembla que ho necessitarem.

Conclusió: De forma estadísticament significativa, sembla que entre els mols algorismes possibles d'ordenació de vectors hi ha dos, com a mínim, que entre ells no son iguals en els seus temps d'ordenació de vectors que prèviament no estaven ordenats i al final sí.

Mètodes: Es van generar 50 vectors de grandàries distribuïdes uniformement entre 0 i 10^6 contenint números enters distribuïts també uniformement entre 0 i deu vegades la grandària del vector. Es varen aplicar els dos algorismes d'ordenació a tots els vector amb un ordre aleatori de execució. Com la diferència depenia de la magnitud del vector, però no així el rati, és varen transformar les dades amb el seu logaritme natural. La comparació de mitjanes es va fer amb la t-Student per dades aparellades.

Resultats: El temps d'execució va ser diferent.

Introducció

En l'àmbit de la informàtica, el disseny de programes té una importància cabdal. Aquests han de complir unes determinades característiques: llegibilitat, robustesa, reusabilitat, eficiència etc...

Volem veure si podem fer més ràpid el nostre sistema. Ara, tenim implementat l'algoritme Quicksort (Q) d'ordenació de vectors, però volem decidir si convé canviar-ho per Mergesort (M) – en cas de que tingui major eficiència temporal. Per això, intentarem rebutjar la hipòtesi nul·la de igualtat de temps a favor de la alternativa bilateral de que M és més ràpid que Q.

Material y mètodes

Es van generar 50 números al·leatoriament d'una mostra uniforme entre 0 i 10^6 per tal de definir les grandàries dels vectors a endreçar pels dos algorismes.

Posteriorment, s'endreçaven aquests 50 vectors amb els dos algorismes, alternant aleatòriament l'ordre d'execució en cada iteració, per tal que no influís l'ordre en la durada de l'execució.

Els experiments es van dur a terme utilitzant el programari R v.2.11.1 en un ordinador amb un microprocessador Intel Core 2 de 2.66GHz i 3.46 Gb de memòria RAM. El sistema operatiu emprat va ser el Microsoft Windows XP.

Les variables recollides van ser la grandària del vector a endreçar i el temps d'execució d'ambdós algorismes. Com la teoria diu que ambdós tenen un temps d'execució d'ordre O(*nlog(n)*) varen estudiar si per comparar el seu rendiment era més convenint utilitzar la seva diferència o el seu rati.

La descriptiva dels temps d'execució i de les grandàries es va realitzar mitjançant mitjanes i la desviació estàndard (DE). La comparació entre ambdós grups es va realitzar mitjançant l'ús de la t-Student per a mostres aparellades seguint el següent procediment.

I. Càlcul de la diferència dels temps d'execució

Creem la variable (d) com diferència dels temps d'execució (M-Q) amb els dos algorismes.

II. Premisses convenients

1. Normalitat per les dues variables estudiades (M i Q).
2. Homocedasticitat. Estudiarem si la diferència és raonablement similar per qualsevol grandària. Si depengués de la longitud de la llista de números a ordenar, estudiarien si el seu rati es pot considerar constant..
3. Obtenció a l'atzar de les grandàries i assignació a l'atzar dels ordres de execució.

III. Contrast d'hipòtesi a realitzar:

$$\begin{cases} H_0 : \mu_D = 0 \\ H_1 : \mu_D > 0 \end{cases}$$

IV. Càlcul de l'estadístic de la comparació:

Les dades no són independents, per tant s'ha de realitzar un test aparellat.

$T_D = \dfrac{\overline{D}}{s\sqrt{\dfrac{1}{n}}} \sim t_n$ on *s* és la estimació de la DE de les diferències Q-M i *n* és la grandària mostral.

V. Obtenció del valor que delimita la Regió Crítica (α = 0.05)

Si $|T| > t_{n,0.975}$,llavors es rebutja la hipòtesi nul·la.

VI. Construcció de l'interval de confiança per la diferència

$$IC(D, 1-\alpha) = \left[\overline{D} \mp t_{n-1,\ 0.975} \cdot \frac{s}{\sqrt{n}}\right]$$

Resultats

Descriptiva

La taula 1 conté la mitjana i la desviació tipus dels temps emprats pels dos algorismes i de la seva diferència. El mètode d'ordenació per mescla inverteix en la mostra un temps inferior

Algorisme	Mitjana (DE)
Merge	2777(1901)
Quick	3123(2039)
Diferència	346(1198)

Taula 1: Mitjana i desviació tipus dels temps d'execució

En l'annex II es mostren els resultats per les variables sense transformar. S'observa que les diferències no semblen seguir la D. Normal i que són més amples a mesura que s'augmenta la longitud del vector. Fem la transformació logaritme natural dels temps per intentar oferir un únic resultat que apliqui a qualsevol longitud. La figura 1 demostra que ara la variable és Normal i en la figura 2 observem una distribució homogènia de la diferències dels logaritmes al llarg del temps.

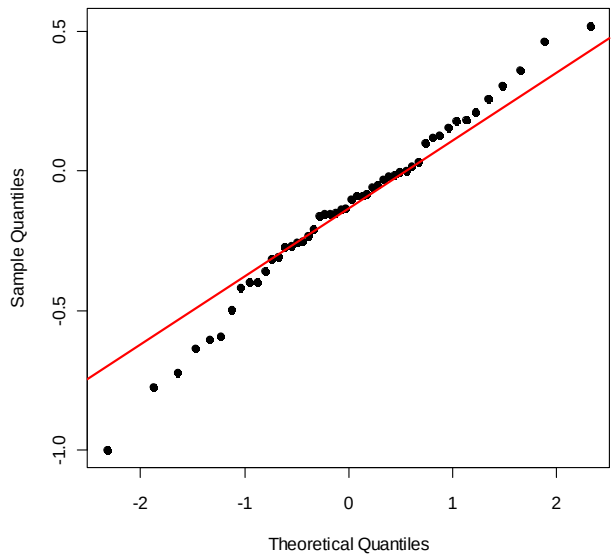


Figura 1. QQ-norm de la diferència dels logaritmes del temps (ln{ms}).

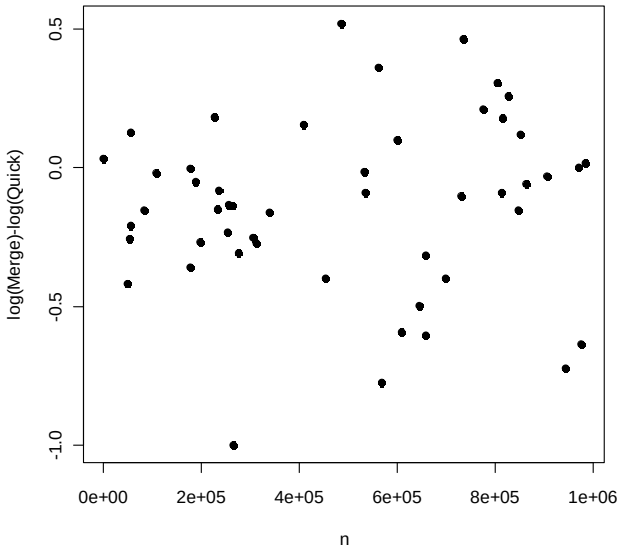


Figura 2: Representació de les diferències dels logaritmes del temps (ln{ms}).) en funció de la mida del vector.

Per comparar la velocitat d'ambdós algorismes d'ordenació fem un contrast de hipòtesi d'igualtat de variàncies amb dades independents. La sortida proporcionada per R és la següent:

```
        Paired t-test

data:  LogMerge and LogQuick
t = -3.1065, df = 49, p-value = 0.003146
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.22774047 -0.04882862
sample estimates:
mean of the differences
             -0.1382845
```

El valor de l'estadístic T és -3.11. Els graus de llibertat son 49, que corresponen al nombre d'observacions menys un. Amb un nivell de significació del 5%, es rebutja la hipòtesi nul·la d'igualtat de mitjanes (p = 0.0031). La mitjana del logaritme del temps de M esta 0.14 log{ms} per sota de Q amb un $IC_{95\%}$ de [0.05 a 0.23]. Si desfem els logaritmes:

$$\log(Merge) - \log(Quick) = -0.1382845 \Rightarrow \log\left(\frac{Merge}{Quick}\right) = -0.1382845 \Rightarrow \frac{Merge}{Quick} = e^{-0.1382845} = 0.8708509$$

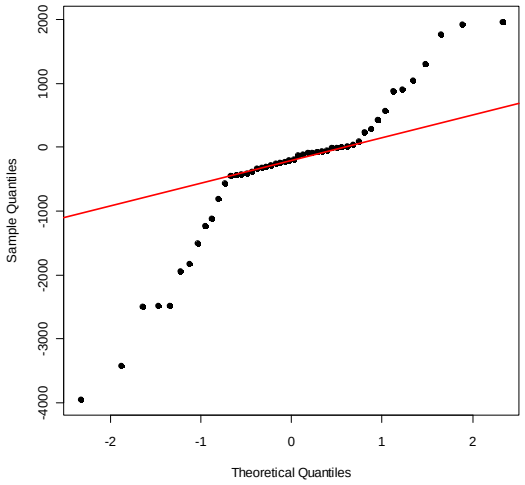
$$Eficiencia\ relativa = \left(1 - \frac{Merge}{Quick}\right) \cdot 100 = 12.91491\%$$

equival a dir que el algorisme M és un 12.9% ($IC_{95\%}$: de 4.8 a 20.4%) més eficient que Q.

Discussió

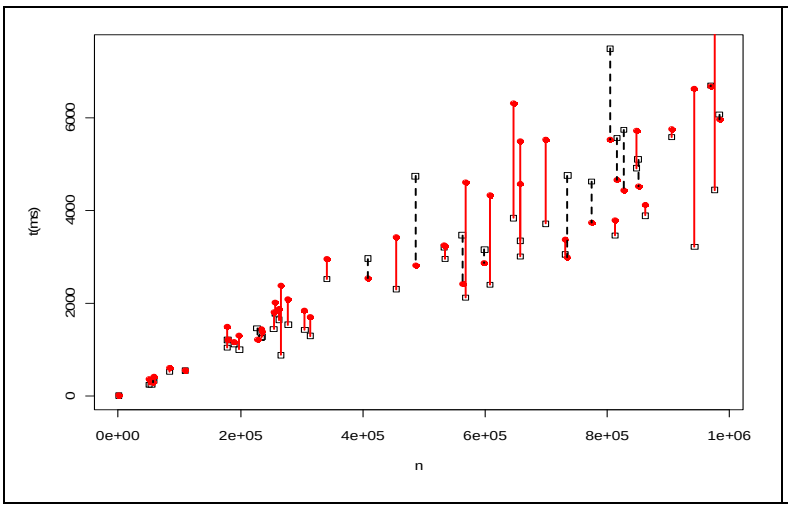
A partir d'ara hem d'utilitzar sempre M en lloc de Q.

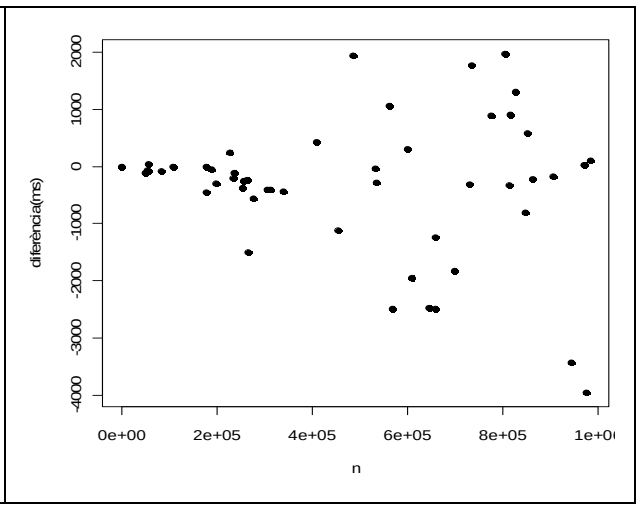
ANNEX I. Test aparellat de les diferències sense transformar



El QQ-plot adjunt mostra un desviament important de la Normalitat.

La següent figura mostra que els temps d'execució dels dos programes i la seva diferència augmenten segons les grandàries mostrals dels vectors a ordenar. La ultima figura ressalta que les diferències augmenten amb la grandària. Aquesta situació té dues implicacions pràctiques: (1) no té sentit estimar un valor comú per totes les granaries; i (2) no es pot aplicar el test per comparació de mitjanes amb dades aparellades ja que totes les observacions no tenen la mateixa variància i l'estadístic no s'ajustaria a la distribució de la t-Student.





De totes formes, cal dir que també s'hauria rebutjat la hipòtesi d'igualtat.

```
        Paired t-test

data:  dades$Merge and dades$Quick
t = -2.0439, df = 49, p-value = 0.04635
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -686.489824   -5.819776
sample estimates:
mean of the differences
              -346.1548
```

ANNEX II. Script en r

```
### Lectura de les dades
dades <- read.table('SortTimes.txt',header=TRUE,sep='\t')

### Variable diferència
dif <-  dades$Merge-dades$Quick

### Descriptiva
```

```
apply(dades[,2:3],2,mean)
apply(dades[,2:3],2,sd)

mean(dif)
sd(dif)

### Normalitat
qqnorm(dif,pch=19)
qqline(dif,lwd=2,col=2)

### boxplot
boxplot(dades[,2:3])

### Gràfic diferències
plot(dades$n,dades$Merge,pch=22,cex=1,xlab="n",ylab="t(ms)")
points(dades$n,dades$Quick,pch=19,col=2,cex=1)
for (i in 1:n){
     if(dades$Merge[i]>dades$Quick[i]){co <- 1 ; lt <- 2}
     else{co <- 2 ; lt <- 1}

segments(dades$n[i],dades$Merge[i],dades$n[i],dades$Quick[i],col=co,lty=lt,lwd=2
)
}

# Heterocedasticitat
plot(dades$n,dif,xlab="n",ylab="diferència(ms)",pch=19)

# boxplot diferencies
boxplot(dif)

### Test aparellat pels temps sense transformar
t.test(dades$Merge,dades$Quick,paired=TRUE)


# Treure logaritmes
LogMerge <- log(dades$Merge)
LogQuick <- log(dades$Quick)
dif <-  LogMerge-LogQuick

plot(dades$n,LogMerge,pch=22)
points(dades$n,LogQuick,pch=19,col=2)

qqnorm(dif,pch=19)
qqline(dif,lwd=2,col=2)

plot(dades$n,dif,pch=19,xlab="n",ylab="log(Merge)-log(Quick)")
boxplot(dif)

### Test aparellat pels temps transformats
```

t.test(LogMerge,LogQuick,paired=TRUE)