LVQuicksort – Federica Tamerisco

Codice

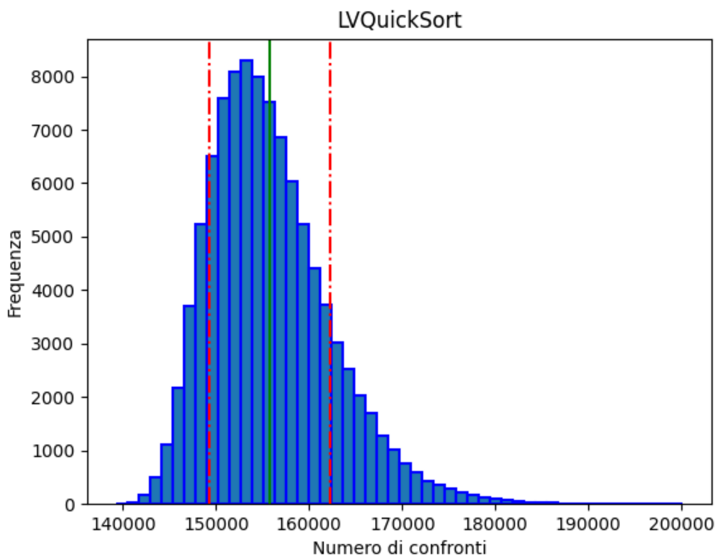
```
import random
import numpy
import matplotlib.pyplot as plt

Xr = 0

def switch(v, i, j):
    tmp = v[j]
    v[j] = v[i]
    v[i] = tmp

def partition(v, start, end):
    pivot_idx = random.randrange(start, end)
    switch(v, pivot_idx, start)
    i = start + 1

    for j in range(start + 1, end + 1):
        global Xr
        Xr += 1

        if(v[j] < v[start]):
            switch(v, i, j)
            i += 1

    switch(v, start, i-1)
    return i - 1

def LVQuickSort(v, start, end):
    if(start < end):
        pivot = partition(v, start, end)
        LVQuickSort(v, start, pivot-1)
        LVQuickSort(v, pivot+1, end)

if __name__ == "__main__":
    R = pow(10, 5)
    numConfronti = numpy.zeros(R)

    N = pow(10, 4)
    s = []

    for i in range(1, N+1):
        s.append(i)

    random.shuffle(s)

    for i in range(0, R):
        tmp = numpy.copy(s)
```

```
        LVQuickSort(tmp, 0, len(tmp) - 1)
        numConfronti[i] = Xr
        Xr = 0

    u = 0
    u = numpy.mean(numConfronti)
    print("Il valore medio è: ", u)

    d = 0
    d = numpy.std(numConfronti)
    print("La deviazione standard empirica è: ", d)

    plt.hist(numConfronti, bins = 50, edgecolor = 'blue', linewidth = 1.5)
    plt.axvline(x=u, color='g', linestyle='solid')
    plt.axvline(x=u - d, color='r', linestyle='dashdot')
    plt.axvline(x=u + d, color='r', linestyle='dashdot')
    plt.title('LVQuickSort')
    plt.xlabel('Numero di confronti')
    plt.ylabel('Frequenza')
    plt.show()
```

Relazione

Dalla produzione del codice troviamo:

```
Il valore medio è:  155804.47844
La deviazione standard empirica è:  6481.371080110686
```



Nel grafico la linea verde indica il valore medio mentre le linee rosse la deviazione standard.

Usando le disuguaglianze di Markov e Chebyshev possiamo porre un limite massimo alla probabilità che la variabile casuale X>0 assuma valori elevati.

Se prendiamo v come multiplo del valore atteso μ:

- Dalla disuguaglianza di Markov otteniamo

$$\Pr\{X \geq v\mu\} \leq \begin{cases} \dfrac{1}{2} \; per\ il\ doppio\ del\ valore\ atteso\ (v = 2) \\ \dfrac{1}{3} \; per\ il\ triplo\ del\ valore\ atteso\ (v = 3) \end{cases}$$

- Dalla disuguaglianza di Chebyshev otteniamo

$$\Pr\{X \geq v\mu\} \leq \begin{cases} 0{,}0017305092 \; per\ il\ doppio\ del\ valore\ atteso\ (v = 2) \\ 0.0004326273 \; per\ il\ triplo\ del\ valore\ atteso\ (v = 3) \end{cases}$$

Le due disuguaglianze, quindi, esprimono la piccola probabilità di avere un numero di confronti pari al doppio o al triplo del valore atteso.

Ma i calcoli sulla deviazione standard mostrano come il numero di confronti è molto vicino al valore atteso: questo indica che l'algoritmo è efficiente nella scelta di un pivot randomizzato, permettendo quindi di evitare il caso peggiore in cui il pivot è il minimo o il massimo della sequenza da ordinare (quindi si evita di ottenere un albero ricorsivo sbilanciato).