Progetto Simulazione

Jacopo Castellini

A.A. 2015/16

Testo del progetto

Il gestore del vostro supermercato (CONAD) ha problemi per determinare il numero di cassiere da avere in uno dei periodi di maggiore affluenza.
Poiché tua madre di lagna spesso delle attese alla cassa e ti ha sfidato a mettere in pratica la tua preparazione sulla teoria delle code, tu ti incarichi del compito di aiutare il gestore.
Decidi quindi – a causa della complessità di decisione e di diniego, dell'abilità di aggiungere o togliere cassiere (che incrementano o decrementano il μ di canale) e altre complessità – di simulare il modello di sistema.
Sviluppa il modello di simulazione e implementalo nel linguaggio che tu desideri.
Tenta di stimare, a partire da tue reali osservazioni su alcuni sabati mattina, i modelli di arrivo e di servizio, così come la disciplina delle code.
Usa parte delle osservazioni per sviluppare le distribuzioni empiriche e convalida poi il simulatore usando le rimanenti osservazioni (al 90% del livello di confidenza).
Quindi cerca di determinare le soluzioni per i problemi del gestore.

Raccolta ed analisi dei dati

Per prima cosa abbiamo raccolto dati campionati dal sistema reale per determinare i processi stocastici che governano il sistema stesso. Ci siamo quindi recati al supermercato per rilevare i tempi di interarrivo dei clienti alle varie casse ed i tempi di servizio delle stesse. Abbiamo raccolto 108 campioni per quanto riguarda l'arrivo in coda e 50 campioni per quanto riguarda il servizio alle casse. Da questi dati abbiamo innanzi tutto ricavato media e deviazione standard campionaria dei due processi per cercare di determinare quale distribuzione stocastica potesse modellarli al meglio:

$$E(x) = \sum x_i / n \qquad \text{con } x_i \text{ tempi misurati}$$
$$\sigma^2 = E(x^2) - E^2(x) \qquad \text{stima indiretta della varianza}$$

Media tempi di interarrivo = 25.09259259
Deviazione standard tempi di interarrivo = 18.98673272
Media tempi di servizio = 94.06
Deviazione standard tempi = 44.22641652

Da questi risultati abbiamo deciso di provare a modellare l'arrivo dei clienti in coda con una distribuzione esponenziale negativa di media $\lambda = 25$ (abbiamo scelto la distribuzione esponenziale poiché i valori di media e varianza sono simili) ed il servizio alle casse come una gaussiana di media $m = 95$ e deviazione standard $\sigma = 50$. Abbiamo poi applicato il test della Goodness of Fit alle due distribuzioni per verificare che queste possano effettivamente modellare i due fenomeni presi in considerazione. Questi sono i risultati:

Categorie	Intervalli	fi	Freq. Rel.	Fi	$(Fi-fi)^2/Fi$
1	0-15	42	0.451188364	48.7283433	0.929040483
2	15-30	34	0.247617424	26.74268181	1.96940941
3	30-45	18	0.135895324	14.67669496	0.752509773
4	45-60	7	0.074580935	8.054740973	0.138114748
5	60 o più	7	0.090717953	9.797538955	0.798794906
Totale		108	1	108	V = 4.587920851

Tabella 1: Test GoF tempi di interarrivo

Abbiamo poi confrontato il valore di V con i percentili della distribuzione χ-quadro con df = 5-1-1 = 3 gradi di libertà:

$$P_{10} = 0.584 \quad P_{90} = 6.25$$

Poiché V cade tra P_{10} e P_{90}, possiamo accettare la distribuzione.

Categoria	Intervalli	fi	Freq. Rel.	Fi	$(Fi-fi)^2/Fi$
1	0-60	11	0.2044	10.22	0.059530333
2	60-90	15	0.2435	12.175	0.655492813
3	90-120	12	0.2583	12.915	0.064825784
4	120 o più	12	0.2776	13.88	0.254639769
Totale		50	0.9838	49.19	V = 1.034488699

Tabella 2: Test GoF tempi di servizio

Confrontiamo V con i percentili della distribuzione χ-quadro con df = 4-1-2 = 1 gradi di libertà:

$$P_{10} = 0.0158 \quad P_{90} = 2.71$$

Anche in questo caso, V è compreso tra P_{10} e P_{90}, quindi possiamo accettare la distribuzione.
Quindi, date le due convalide sopra, possiamo vedere ogni cassa come un sistema M/G/1 con $\lambda = 1 / (\#casse * 25)$ e $\mu = 1 / 94$. È evidente quindi che per avere un sistema stazionario il numero di casse aperte deve essere almeno 4 (poiché deve valere $\rho = \lambda / \mu < 1$). La probabilità che un cliente si accodi su una delle casse è di 1 / #casse.

Implementazione del simulatore

Una volta raccolti e convalidati i dati, il passo seguente è stato quello di implementare il simulatore vero e proprio. Per farlo abbiamo utilizzato Python 2.7.10 e la libreria per simulazione ad eventi discreti SimPy (https://pypi.python.org/pypi/simpy). Il simulatore, tramite l'ambiente fornito da SimPy, simula per un tempo prefissato un processo che gestisce gli arrivi dei clienti alle casse ed un processo per ogni utente che arriva alle casse e paga la spesa. Il numero di casse è un parametro da passare al sistema: ogni cassa è rappresentata come una risorsa limitata a se stante (sempre grazie alle strutture dati per le risorse limitate offerte da SimPy). Le distribuzioni statistiche generano dati grazie alle funzioni predefinite nella libreria random di Python, basandosi sulle distribuzioni convalidate precedentemente tramite il test della Goodness of Fit. Infine, al termine del tempo di simulazione, vengono calcolati vari parametri del sistema, come tempo medio di attesa in coda o numero medio di utenti nel sistema. Ecco l'output di un run di 10.000 unità di tempo con 5 casse aperte:

```
###################################
###         Statistics          ###
###################################

Total customers number:  431
Enqueued customers number:  13
Customer at the cash registers:  2
Completed customers number:  416
Mean arrival time:  23.0934113499
Mean response time:  311.067812632
Mean service time:  95.6851332222
Mean waiting time:  215.38267941
Mean utilization:  0.796100308409
Mean throughput:  0.0104509443246
Lambda:  0.00866047882529
Mu:  0.0104509443246
Rho:  0.828679070167  -> Stable
```

Stazionarietà del sistema

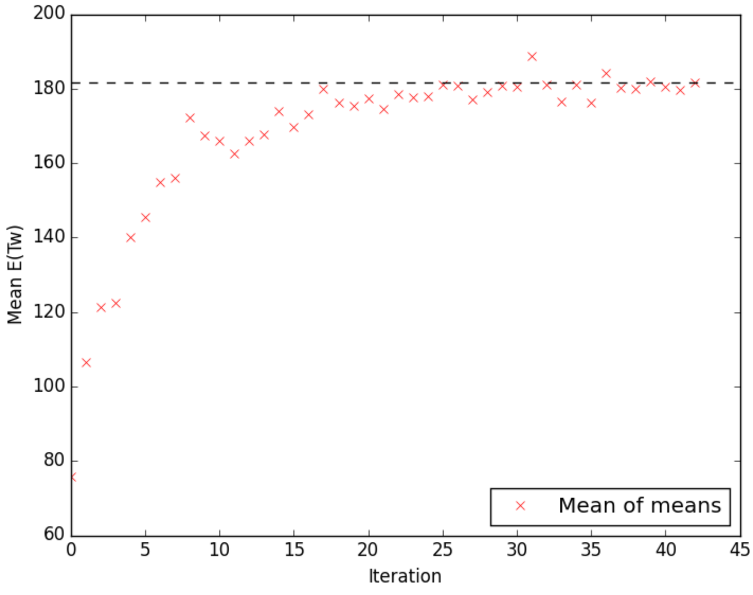
Per poter valutare i risultati del nostro simulatore è prima necessario che le simulazioni raggiungano un punto stabile in cui certi parametri di interesse non subiscono più variazioni considerevoli. Per fare ciò si esegue la simulazione fino ad un punto in cui lo stato iniziale (cioè con zero clienti nel sistema) non abbia più influenza sullo stato attuale, e di conseguenza sui parametri del sistema. Assicuratici quindi che il sistema sia stazionario (quindi che $\rho < 1$), si eseguono p run a partire dallo stesso stato iniziale ma con diversi numeri casuali generati dalle distribuzioni (questo è garantito dalla casualità con cui vengono generati i valori presi dalle distribuzioni). Calcoleremo poi la media delle varie osservazioni dei vari run (ognuno dei p run è formato da n osservazioni):

$x_j(n) = (1 / n)\ \Sigma x_{ij}$ media campionaria di un run (i = 1...n)

$E(n) = (1 / p)\ \Sigma x_j(n)$ media totale dei p run (j = 1...p)

$s^2(x) = (1 / (p - 1))\ \Sigma(x_j(n) - E(n))^2$ varianza totale dei p run (j = 1...p)

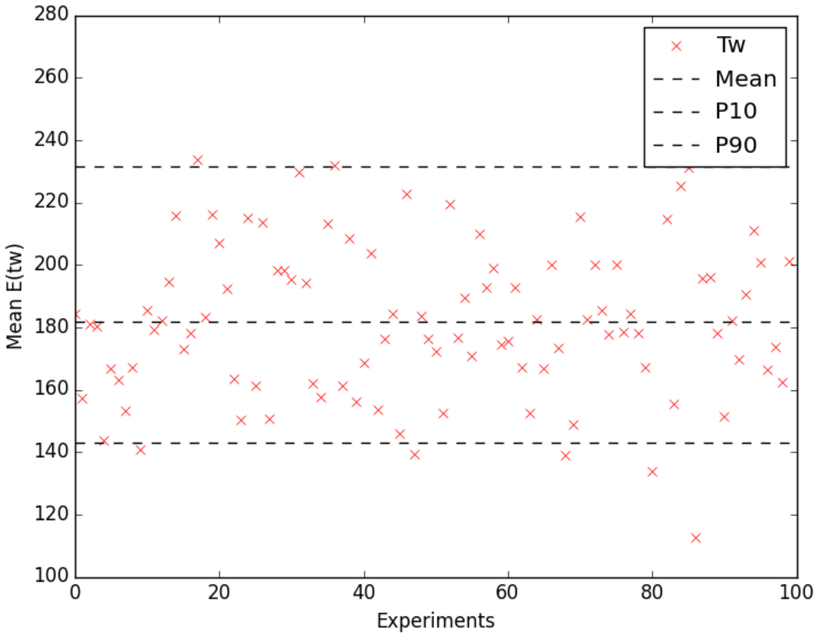
Si aumenta poi la durata della simulazione finché il valore della media totale E(n) non comincia subire variazioni piccole o nulle. Questo è il risultato per il nostro simulatore (con p = 100 e durata n = 1000 unità di tempo, incrementata di 1000 ogni volta) nell'osservare il tempo medio di attesa in coda:



Il valore della media totale del tempo di attesa in coda comincia a stabilizzarsi intorno ad un valore di 182 alla durata di 40.000 unità di tempo. Da questo istante in poi il sistema può essere considerato stabile e può essere valutato in relazione al sistema reale.

Metodo delle prove ripetute

Il metodo delle prove ripetute è un metodo usato per verificare l'aderenza del simulatore al sistema reale che si intende modellare. Per fare ciò, dopo aver stabilizzato il sistema, si eseguono p run del simulatore e si cominciano a prendere i risultati a partire da quando il sistema si è stabilizzato. Se questi risultati rientrano in un certo intervallo di confidenza rispetto al valore della media totale calcolata in fase di stabilizzazione allora il simulatore è convalidato e può effettivamente essere usato per verificare le richieste sul sistema fatto. Anche in questo caso la variabile osservata è stato il tempo medio di attesa in coda. Si sono eseguiti p = 100 run indipendenti, ognuno con una durata compresa tra 60.000 e 80.000 unità di tempo. Questi sono i risultati:




```
Stabilized at iteration 43
Mean: 181.662437
10° percentile: 143.102869
90° percentile: 231.627245
Experiments between P10 and P90: 92
Experiments out of range: 8
Validated!
```

Poiché i valori delle medie dei p run rimangono compresi almeno al 90% tra il 10° ed il 90° percentile della distribuzione empirica del tempo medio di attesa in coda calcolata alla fine della fase di stabilizzazione, il simulatore è convalidato e può essere utilizzato per effettuare osservazioni e stime sul sistema reale.