CNP II

Processi Stocastici Discreti

Un ***processo markoviano*** descritto da una matrice di probabilità di transizione $\mathcal{P}$ tale che

$$\mathcal{P}_{ij}^{(n+m)} = \sum_{k\in\mathbb{I}} \mathcal{P}_{ik}^{(n)}\mathcal{P}_{kj}^{(m)}$$

(***Chapman-Kolmogorov***) ha delle soluzioni stazionarie $\{\pi_i\}$ (i.e. le probabilità di trovare il sistema nello stato i in un generico istante) ottenibili tramite le stationary equations

$$\begin{cases} \pi = \pi\mathcal{P} \\ \sum_i \pi_i = 1 \end{cases}$$

Se esistono le ***probabilità limite*** $\{\pi_j^*\}$ tali che

$$\lim_{n\to\infty} P_{ij}^{(n)} = \lim_{n\to\infty} (P^n)_{ij} = \pi_j^*$$

allora abbiamo che $\pi = \pi^*$ è l'unica soluzione del ***processo stocastico***.

La π^* esiste sicuramente se il sistema è ***ergodico***, ovvero se ha le seguenti proprietà:

- ***Aperiodicity*** - Il periodo di uno stato j è il massimo comune divisore del set $S \in \mathbb{N}$ formato da tutti gli $\{n \in S \,|\, \mathcal{P}_{jj}^n > 0\}$. In altre parole, se le probabilità di ritorno sono non nulle per un set di passi n "regolare", il sistema è periodico. Se il ***MCD è*** 1, allora il set di passi non presenta alcuna regolarità, ovvero lo stato è aperiodico;
  - Un sistema è aperiodico se ***tutti i suoi stati*** sono aperiodici;
  - Nota che ***non-aperiodico non significa periodico***. Se per uno stato la periodicità non è definita, l'intero sistema non ha una periodicità definita.
- ***Irreducibility*** - Esiste sempre un cammino dallo stato i allo stato j, $\forall\, i, j \in \mathbb{I}$ (i.e. la catena è ***connessa***);
  - Se il sistema non è irriducibile, potrei avere periodi diversi;
  - Nel caso di CdM con un numero finito di stati, queste prime due condizioni da sole sono sufficienti a dire che $\exists\, \pi^*$, in quanto si può mostrare che implicano la terza.
- ***Positive Recurrence*** - "Se finisco in uno stato, prima o poi ci ritorno". Sia f_j la probabilità di ritorno allo stato j (i.e. $f_j = \sum_n^\infty \mathcal{P}_{jj}^n$). Distinguiamo allora
  - ***Stati Positive Recurrent*** - $f_{jj} = 1$, i.e. il numero atteso di visite è ∞;
    - Il numero atteso di step tra due visite consecutive m_{jj} si dimostra essere pari a $\frac{1}{\pi_{jj}}$.
  - ***Stati Transienti*** - $f_{jj} < 1$, i.e. il numero atteso di visite è $< \infty$.
    - In questo caso non c'è positive recurrence, quindi $\nexists\, \pi^*$.

Una CdM irriducibile ha ***solo stati transienti o solo stati ricorrenti***.

Tutto questo assume che gli elementi della matrice $\mathcal{P}$ (e quindi le etichette degli archi) siano normalizzati (i.e. chiudano a 1). In genere però la CdM rappresenta una coda, e la probabilità di osservare un aumento (decremento) di popolazione è proporzionale a λ (μ), che però non è normalizzata (rispetto a cosa, poi?). Posso usare brutalmente λ e μ ?

- Sì, perché anche ammesso di che si definisca una normalizzazione, quando poi vado a scrivere le equazioni delle barriere questa si semplifica (i.e. vengono sempre cose del tipo $\pi_1 = \frac{\lambda}{\mu}\pi_0$), quindi posso in prima approssimazione ignorare il problema;
  - ***NON*** posso dire lo stesso se uso le equazioni stazionarie, perché non si semplifica proprio nulla!
  - Se volessi essere puntiglioso, potrei dire che prendo un periodo di osservazione talmente piccolo (i.e. una frequenza di osservazione Λ talmente grande) da rendere il tutto un ***processo poissoniano***, quindi tale che ogni stato ha una enorme loop probability e bassissime probabilità di uscire dallo stato (i.e. la probabilità di eventi simultanei come arrivo e completion nello stesso time slot è trascurabile, quindi assumo che possa avvenire al più un evento per ogni time slot).

Processi Stocastici Continui

Se al posto delle probabilità ho i rate, come detto, in prima approssimazione non mi importa. Dopodiché definisco un rate di osservazione Λ così grande da escludere eventi simultanei. Le probabilità diventano $\lambda \to \frac{\lambda}{\Lambda}$ e aggiungo loop di probabilità $\frac{\Lambda-\lambda}{\Lambda}$.

Questo però ***assume che il rate di arrivo dei job sia memory-less***, i.e. detto in parole povere dato un arrivo non c'è alcuna correlazione con il successivo. Questa cosa non è vera, ad esempio, se c'è un loop: un job che quando completato viene rimandato alla coda con $p \sim 1$ presenta una fortissima correlazione tra arrivi, quindi si perde l'assenza di memoria.

Assumendo però sia vero, un ***rate di arrivo*** distribuito come ***Poisson*** ci dice che

$$P(k \text{ arrivals in } T) = \frac{(\lambda T)^k e^{-\lambda T}}{k!}$$

mentre l'***inter-arrival time*** x si trova come

$$P(x \leq T) = 1 - P(0 \text{ arrivals in T}) := F_x(t) = 1 - e^{-\lambda T} \Rightarrow f_x(t) = \partial_t F_x(t) = \lambda e^{-\lambda T}$$

ed è distribuito ***esponenziale***. Dal momento che

$$\text{Numero di arrivi poissoniano} \iff \text{Inter-arrival time esponenziale}$$

si usano indistintamente questi due termini per intendere che è rispettata la condizione di assenza di memoria, formalmente scritta come

$$P(x > t_1 + t \,|\, x > t_2) = P(x > t)$$

Per indicare rapidamente questi sistemi continui si usa la ***Kendall Notation***

$$\text{Arrivals Distribution} \,/\, \text{Departures Distribution} \,/\, \#\text{Servers} \,/\, \text{Buffer Size}$$

dove nei primi due campi si usa la lettera M per indicare processi poissoniani (esponenziali).

Per un sistema $\mathrm{M}/\mathrm{M}/1/\infty$ valgono ad esempio le leggi

- $\pi_0 = 1 - \rho, \quad \pi_i = \rho^i(1-\rho)$;
- $N = \frac{\rho}{1-\rho}, \quad N_{Queue} = N - \rho = \frac{\rho^2}{1-\rho}$
- $X = \lambda, \quad R = \frac{1}{\mu}, \quad T_w = \frac{\rho}{\mu - \lambda}$

Una coda, un server

Nel caso di un solo server e di una sola coda, la modellizzazione prevede che

- Gli stati sono la popolazione totale del sistema. Si riempie prima il server, poi la coda;
- Le transizioni sono le probabilità di arrival o completion, o i rispettivi rate.

Posto che il sistema sia ergodico, tramite le stationary equations o tramite la barrier technique trovo il set di probabilità stazionarie $\{\pi_i\}$.

- L'***utilization*** del server è quindi $\rho = P(\text{Server is busy}) = \sum_{j=1}^{N} \pi_j = 1 - \pi_0$;
  - Nota che uno sarebbe tentato di dire $\rho = \frac{\lambda}{\mu}$ Questo è vero solo per ***coda infinita***. In caso contrario, devo sempre calcolarlo come $1 - \pi_0$.
- Il ***throughput*** è $X = P(\text{Completion}) \cdot P(\text{Server is busy}) = \mu(1-\pi_0) = \frac{\rho}{S}$ (i.e. ***utilization law***);
- Per ***coda finita*** e π_N probabilità che il sistema sia pieno, il ***drop rate*** è $\lambda \pi_N$.

Per ***coda infinita*** ho una $\mathcal{P}$ infinita. Posso però definire $\mathcal{P}_{ij}$. Assumiamo che tutti gli arrival rate λ siano uguali, stessa cosa per i completion rate μ. Se definisco $\rho = \frac{\lambda}{\mu}$ trovo dalle barriere che$\pi_i = \rho^i \pi_0$, e dalla normalizzazione che $\pi_0 = 1 - \rho$ (***in questo caso*** ρ è proprio l'utilization), quindi

$$\pi_i = \rho^i(1-\rho)$$

Ripetiamolo insieme: la regola generale per l'utilization in un processo stocastico è $\rho = 1 - \pi_0$, dopodiché in alcuni casi può incidentalmente risultare proprio uguale a λ/μ.

Più code, più server

Se ho più di un server, ci sono due casi:

- Tutti i server condividono la stessa coda, formando un parallelo. Assumendo abbiano tutti la stessa μ e routing equiprobabile, la CdM si costruisce nel seguente modo:
  - L'***arrival rate*** λ è uguale per tutti gli stati;
  - Il ***completion rate*** è μ per la transizione $1 \to 0$, 2μ per la transizione $2 \to 1$ e così via fino a $N\mu$ per la transizione $N \to N-1$, dove N è il numero di server. A partire dallo stato $N+1$ inizio a riempire la coda, e da qui in poi (i.e. fino al suo completo riempimento) il completion rate resta $N\mu$;
  - La ***utilization*** va necessariamente calcolata come

$$U = \frac{\overline{N}_{\text{Busy Servers}}}{N_{\text{Servers}}} \quad \text{dove} \quad \overline{N}_{\text{Busy Servers}} = 0 \times \pi_0 + 1 \times \pi_1 + \ldots N \sum_{i=N}^{M} \pi_i$$

  - Il ***throughput*** va necessariamente calcolato come

$$X = \overline{N}_{\text{Busy Servers}} \times \mu = U \mu N_{\text{Servers}}$$

    assumendo ovviamente che la μ sia uguale per tutti i server del parallelo (credo che in caso contrario si potrebbe calcolarne un valore medio pesato secondo le probabilità di routing, ma forse questo è oltre gli scopi del corso);
- Ogni server ha la sua coda. Assumendo ***un unico input*** cui segue un routing probabilistico, mi limito a trattare separatamente i vari server secondo le probabilità di routing.
  - Se mi interessa la ***utilization*** del sistema complessivo (e.g. ho un incoming rate diviso in modo probabilistico tra due server ognuno avente la propria coda) la devo trovare come nel caso precedente. In generale questa cosa ha senso se ho un ***parallelo***, sia esso con code divise o meno;

Jackson Networks

Se ad un server ho più incoming rates poissoniani, non è detto che sia poissoniana anche la loro somma.

Se ad un sistema $\mathrm{M}/\mathrm{M}/1/\infty$ aggiungiamo un loop, gli arrivi sono quelli dall'esterno più quelli dal loop. Questi però dipendono dal completion rate, quindi si perde la proprietà di assenza di memoria.

Questo potrebbe essere un problema, ma se il sistema è tale che

- Ogni server ha una coda di lunghezza infinita con priorità FCFS;
- Ogni input che proviene dall'esterno è poissoniano;
- Ogni output di ogni singolo server k è poissoniano;

- Il routing tra uscite ed ingressi dei vari server è probabilistico.

allora posso applicare la teoria di Jackson. Si dimostra sotto queste condizioni che nonostante il sistema sia un delirio posso ancora trovare una soluzione per π_i in forma di prodotto come nel caso $\mathrm{M}/\mathrm{M}/1/\infty$

$$\vec{\pi}_{\{\cdots(n_i \text{ jobs at server } i)\cdots\}} = \prod_{i=1}^{k} \rho_i^{n_i}(1-\rho_i)$$

i.e. $\vec{\pi}$ è la probabilità di trovarsi nello stato $\{n_1, n_2, \ldots, n_k\}$.

Segue che per ogni server i possiamo calcolare la probabilità che questo abbia una popolazione di esattamente n_i^* jobs come, ad esempio,

$$P(n_1^* \text{ jobs a server } 1) = \rho_1^{n_1^*}(1-\rho_1)$$

Segue che per le Jackson Networks posso calcolare la popolazione attesa ad ogni server come

$$\overline{N_i} = \frac{\rho_i}{1-\rho_i}$$

che di base è tutto quello che serve sapere: per ogni server i trovo la λ_i effettiva e calcolo N con questa formula. Fine.