Soluzione Esercizio di Statistica Foglio II
Draghici Ana Maria 2101044

Testo dell'esercizio 14. Siano $N \in \mathbb{N}$, $q \in (0,1)$. Sia $p_{\mathrm{Bin}}(N,q)$ la funzione di massa di probabilità (densità discreta) della distribuzione binomiale di parametri N e q, definita come:

$$p_{\mathrm{Bin}}(N,q)(k) = \begin{cases} \binom{N}{k} q^k (1-q)^{N-k} & \text{se } k \in \{0,\dots,N\}, \\ 0 & \text{altrimenti.} \end{cases}$$

1. Si mostri che, per $k \in \{0,\dots,N-1\}$, vale:
$$p_{\mathrm{Bin}}(N,q)(k+1) = \frac{q}{1-q} \cdot \frac{N-k}{k+1} \cdot p_{\mathrm{Bin}}(N,q)(k).$$
2. Si verifichi inoltre che, per ogni $k \in \{0,\dots,N\}$, vale:
$$p_{\mathrm{Bin}}(N,q)(k) = p_{\mathrm{Bin}}(N,1-q)(N-k).$$

Soluzione

1. Relazione ricorsiva tra termini consecutivi

Sia:
$$p_k = p_{\mathrm{Bin}}(N,q)(k) = \binom{N}{k} q^k (1-q)^{N-k},$$
e:
$$p_{k+1} = p_{\mathrm{Bin}}(N,q)(k+1) = \binom{N}{k+1} q^{k+1} (1-q)^{N-k-1}.$$

Vogliamo calcolare il rapporto tra questi due termini consecutivi:
$$\frac{p_{k+1}}{p_k} = \frac{\binom{N}{k+1} \cdot q^{k+1} \cdot (1-q)^{N-k-1}}{\binom{N}{k} \cdot q^k \cdot (1-q)^{N-k}}.$$

Separiamo i fattori:
$$= \left(\frac{\binom{N}{k+1}}{\binom{N}{k}} \right) \cdot \left(\frac{q^{k+1}}{q^k} \right) \cdot \left(\frac{(1-q)^{N-k-1}}{(1-q)^{N-k}} \right).$$

Calcoliamo ciascun rapporto:

- **Termine binomiale**: dalla definizione dei coefficienti binomiali:
$$\frac{\binom{N}{k+1}}{\binom{N}{k}} = \frac{\frac{N!}{(k+1)!(N-k-1)!}}{\frac{N!}{k!(N-k)!}} = \frac{k!(N-k)!}{(k+1)!(N-k-1)!}.$$

  Riscrivendo i fattoriali:
$$= \frac{k! \cdot (N-k) \cdot (N-k-1)!}{(k+1) \cdot k! \cdot (N-k-1)!} = \frac{N-k}{k+1}.$$

- **Potenza di q**:
$$\frac{q^{k+1}}{q^k} = q.$$

- **Potenza di $1-q$**:
$$\frac{(1-q)^{N-k-1}}{(1-q)^{N-k}} = \frac{1}{1-q}.$$

Combinando i risultati ottenuti:

$$\frac{p_{k+1}}{p_k} = \frac{N-k}{k+1} \cdot q \cdot \frac{1}{1-q} = \frac{q}{1-q} \cdot \frac{N-k}{k+1}.$$

Pertanto:

$$p_{k+1} = \frac{q}{1-q} \cdot \frac{N-k}{k+1} \cdot p_k,$$

come volevasi dimostrare.

2. Simmetria della distribuzione binomiale

Osserviamo:

$$p_{\text{Bin}}(N,q)(k) = \binom{N}{k} q^k (1-q)^{N-k},$$

e:

$$p_{\text{Bin}}(N,1-q)(N-k) = \binom{N}{N-k} (1-q)^{N-k} q^k.$$

Poiché:

$$\binom{N}{N-k} = \binom{N}{k},$$

segue che:

$$p_{\text{Bin}}(N,1-q)(N-k) = \binom{N}{k} (1-q)^{N-k} q^k = p_{\text{Bin}}(N,q)(k).$$

Pertanto:

$$p_{\text{Bin}}(N,q)(k) = p_{\text{Bin}}(N,1-q)(N-k),$$

come richiesto.

Conclusione

Abbiamo dimostrato:

- **Relazione ricorsiva tra termini consecutivi:** Abbiamo mostrato che la densità discreta della distribuzione binomiale, $p_{\text{Bin}}(N,q)(k)$, soddisfa una relazione ricorsiva tra i termini consecutivi.

  In particolare, abbiamo derivato la formula che lega il valore $p_{\text{Bin}}(N,q)(k+1)$ al valore $p_{\text{Bin}}(N,q)(k)$, evidenziando il ruolo dei parametri N e q nella dinamica della distribuzione.

- **Simmetria della distribuzione binomiale:** Inoltre, abbiamo verificato la simmetria della distribuzione binomiale rispetto ai parametri q e $1-q$. Più precisamente, abbiamo dimostrato che la densità discreta della distribuzione binomiale per il parametro q è uguale a quella per $1-q$ se consideriamo i valori k e $N-k$.

Introduzione Esercizio 15

In un ballottaggio tra due candidati, A e B, votano $N + M$ persone, dove N è il numero di elettori indecisi che votano a caso e M è il numero di elettori che supportano il candidato A. L'obiettivo di questo esercizio è calcolare la probabilità che il candidato A vinca, dato che i voti degli elettori indecisi sono descritti tramite una variabile aleatoria binomiale. La probabilità che A vinca dipende dalla distribuzione del numero di voti che riceve dagli elettori indecisi, e in particolare vogliamo calcolare la probabilità che A ottenga più di metà dei voti totali, considerando anche i voti già sicuri che supportano A.

Giustificazione matematica della procedura

Sia S_N la variabile aleatoria che rappresenta il numero di voti che il candidato A riceve dai N elettori indecisi. Poiché ogni elettore indeciso ha una probabilità $\frac{1}{2}$ di votare per A e una probabilità $\frac{1}{2}$ di votare per B, S_N segue una distribuzione binomiale di parametri N e $\frac{1}{2}$, cioè:

$$S_N \sim \text{Bin}(N, \frac{1}{2}).$$

Il numero totale di voti per A è dato dalla somma dei voti dei M elettori che sostengono A, che sono già determinati, e dai voti dei N elettori indecisi, rappresentati dalla variabile S_N. Pertanto, la probabilità che A vinca è la probabilità che il numero totale di voti per A sia maggiore del numero di voti per B:

$$P(\text{Vittoria di A}) = P(S_N + M > \frac{N + M}{2}).$$

Questa disuguaglianza può essere riscritta come:

$$P(S_N > \frac{N - M}{2}).$$

Poiché S_N è una variabile aleatoria binomiale, la probabilità che S_N sia maggiore di $\frac{N-M}{2}$ è data dalla somma delle probabilità che S_N assuma valori da $\left\lfloor \frac{N-M}{2} \right\rfloor + 1$ a N:

$$P(S_N > \frac{N - M}{2}) = \sum_{k=\left\lfloor \frac{N-M}{2} \right\rfloor + 1}^{N} p_{\text{Bin}}(N, \frac{1}{2})(k),$$

dove $p_{\text{Bin}}(N, \frac{1}{2})(k)$ è la probabilità di ottenere esattamente k successi in una distribuzione binomiale di parametri N e $\frac{1}{2}$, ossia:

$$p_{\text{Bin}}(N, \frac{1}{2})(k) = \binom{N}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{N-k} = \binom{N}{k} \left(\frac{1}{2}\right)^N.$$

Quindi la probabilità che A vinca è:

$$P(\text{Vittoria di A}) = \sum_{k=\left\lfloor \frac{N-M}{2} \right\rfloor + 1}^{N} \binom{N}{k} \left(\frac{1}{2}\right)^N.$$

Conclusioni

Abbiamo determinato la probabilità che il candidato A vinca nel ballottaggio, utilizzando la distribuzione binomiale per modellare il comportamento degli elettori indecisi. La probabilità che A vinca è data dalla somma delle probabilità che il numero di voti ricevuti da A (avendo

N voti "già sicuri") superi una certa soglia (la metà dei voti $\frac{N+M}{2}$) garantendo la vittoria del candidato A rispetto al candidato B . Questo risultato è importante per comprendere il comportamento elettorale in scenari di voto casuale, dove ci possono essere degli elettori indecisi che votano senza preferenza tra i candidati.

Pseudocodice del programma che calcola la probabilità di vittoria del candidato A

Input: Numero totale di elettori $N + M = 10^6$, dove:

- M: numero di voti sicuri per il candidato A (elettori che votano con certezza A),
- $N = 10^6 - M$: numero di elettori indecisi che votano casualmente.

1. Per ogni $M \in \{0, 10, 20, \ldots, 5000\}$:
   (a) Calcolare il numero di elettori indecisi:
   $$N = 10^6 - M$$
   (b) Calcolare la soglia minima di voti casuali necessari affinché A ottenga la maggioranza:
   $$k = \left\lfloor \frac{N - M}{2} \right\rfloor + 1$$
   (c) Calcolare la probabilità che A ottenga più di k voti dal gruppo degli indecisi, modellato come variabile aleatoria binomiale con parametri N e $\frac{1}{2}$:
   $$P = \sum_{i=k}^{N} \binom{N}{i} \left(\frac{1}{2}\right)^N$$
   (d) Salvare il valore calcolato di P
2. Al termine, tracciare un grafico che rappresenta la probabilità di vittoria di A in funzione del numero di voti sicuri M.

Conclusione:
Questo algoritmo consente di stimare la probabilità di vittoria del candidato A in un'elezione in cui una parte dell'elettorato è incerta e vota in modo equiprobabile. Il modello binomiale fornisce una descrizione del comportamento aleatorio degli elettori indecisi. Il grafico risultante mostra l'evoluzione della probabilità di vittoria in funzione del supporto iniziale garantito a A (ossia M).

Commenti sul grafico generato dal codice Python

Nel grafico prodotto dal codice Python, si osserva chiaramente come la probabilità di vittoria del candidato A cresca rapidamente al crescere del numero di voti sicuri M, pur restando M estremamente piccolo rispetto al numero totale di elettori $N + M = 10^6$.

Questo fenomeno è una conseguenza diretta delle proprietà della distribuzione binomiale. Gli N elettori indecisi votano in modo equiprobabile (probabilità 0.5), quindi il numero di voti che A ottiene tra gli indecisi segue una binomiale $B(N, 0.5)$, concentrata attorno al valore medio $N/2$, con una deviazione standard pari a $\sigma = \sqrt{N}/2$.

Anche un piccolo numero di voti certi per A (es. $M = 1000$) sposta la soglia di vittoria a suo favore, riducendo il numero minimo di voti necessari dagli indecisi.

Il grafico mostra una transizione rapida: per valori bassi di M (tra 0 e 500), la probabilità cresce lentamente da 0.5 a valori vicini a 1, per poi saturarsi rapidamente vicino a 1 già intorno a $M = 3000$. Per valori maggiori di $M = 3000$, la probabilità di vittoria di A è talmente prossima a 1 da risultare indistinguibile da essa nel grafico.

Ottimizzazione e Stabilizzazione del Calcolo Ricorsivo della Distribuzione Binomiale

Nel problema proposto, il numero di elettori indecisi N può arrivare fino a 10^6, rendendo inefficiente e instabile il calcolo diretto delle probabilità binomiali:

$$pBin(N, q)(k) = \binom{N}{k} q^k (1-q)^{N-k}.$$

Nel caso simmetrico $q = \frac{1}{2}$, questa si semplifica a:

$$pBin(N, \tfrac{1}{2})(k) = \binom{N}{k} \cdot \left(\tfrac{1}{2}\right)^N.$$

Per determinare la probabilità che il candidato A vinca, è necessario sommare i termini binomiali dalla soglia $k = \left\lfloor \frac{N-M}{2} \right\rfloor + 1$ fino a N:

$$P = \sum_{i=k}^{N} \binom{N}{i} \cdot \left(\tfrac{1}{2}\right)^N = \left(\tfrac{1}{2}\right)^N \sum_{i=k}^{N} \binom{N}{i}.$$

Strategie di Ottimizzazione e Stabilizzazione

1. **Ricorsione tra termini consecutivi:** i coefficienti binomiali successivi possono essere calcolati in modo efficiente mediante la relazione:

   $$p_{k+1} = \frac{N-k}{k+1} \cdot p_k.$$

   Questo evita di calcolare direttamente ogni $\binom{N}{k}$, riducendo il costo computazionale da $\mathcal{O}(N)$ a $\mathcal{O}(1)$ per ogni termine successivo.

2. **Calcolo del primo termine:** si può iniziare il calcolo ricorsivo ottenendo direttamente solo il primo termine p_k della coda:

   $$p_k = \binom{N}{k} \cdot \left(\tfrac{1}{2}\right)^N.$$

3. **Spazio logaritmico:** per evitare problemi di underflow dovuti a termini molto piccoli (ad esempio per grandi N), si opera nello spazio logaritmico:

   $$\log(p_k) = \log \binom{N}{k} - N \log 2,$$

   $$\log(p_{k+1}) = \log(p_k) + \log\left(\frac{N-k}{k+1}\right).$$

4. **Somma stabile con log-sum-exp:** per sommare le probabilità in spazio logaritmico, si usa la tecnica numericamente stabile detta *log-sum-exp*:

$$\log\left(\sum_i e^{x_i}\right) = a + \log\left(\sum_i e^{x_i - a}\right), \quad a = \max_i x_i.$$

Questo permette di sommare probabilità piccole evitando l'approssimazione a zero nei calcoli in virgola mobile.

Conclusione

Combinando la relazione ricorsiva tra coefficienti binomiali con il calcolo in spazio logaritmico, si ottiene una strategia robusta e scalabile per calcolare code di distribuzioni binomiali anche per $N \sim 10^6$. Questi accorgimenti permettono di:

- ridurre significativamente i tempi di calcolo;
- prevenire instabilità numerica (underflow/overflow);
- evitare il calcolo diretto di ogni coefficiente binomiale;
- garantire la correttezza dei risultati anche per somme molto squilibrate.

Questa tecnica è essenziale per l'implementazione efficiente di algoritmi di statistica computazionale in presenza di modelli stocastici ad alta precisione.

Applicazione al problema elettorale (Esercizio 15)

Per calcolare la probabilità di vittoria di A:

$$P(S_N > \tfrac{N-M}{2}) = \sum_{i=k}^{N} \binom{N}{i} \left(\tfrac{1}{2}\right)^N,$$

dove $k = \left\lfloor \frac{N-M}{2} \right\rfloor + 1$. Le ottimizzazioni proposte sono critiche per:

- Evitare overflow/underflow con $N = 10^6$.
- Ridurre la complessità da $\mathcal{O}(N^2)$ a $\mathcal{O}(N)$.

Pseudocodice del programma che calcola la probabilità di vittoria del candidato A ottimizzato

Contesto: Il programma calcola la probabilità che il candidato A vinca il ballottaggio, dati:

Input: Numero totale di elettori $N + M = 10^6$, dove:

- M: numero di voti sicuri per il candidato A,
- $N = 10^6 - M$: numero di elettori indecisi che votano con probabilità $\frac{1}{2}$.

Output: Probabilità P che il candidato A ottenga la maggioranza dei voti.

Per ogni $M \in \{0, 10, 20, \ldots, 5000\}$:

1. Calcolare il numero di indecisi:
$$N = 10^6 - M$$
2. Calcolare la soglia minima di voti casuali necessari per la maggioranza:
$$k = \left\lfloor \frac{N-M}{2} \right\rfloor + 1$$
3. Calcolare il logaritmo del primo termine binomiale:
$$\log(p_k) = \log \binom{N}{k} - N \log 2$$
$$\log \binom{N}{k} = \log \left(\frac{N!}{k!(N-k)!} \right)$$
4. Inizializzare la lista dei log-probabilità con $\log(p_k)$
5. **Per** $i = k+1$ **fino a** N:
   (a) Aggiornare ricorsivamente:
$$\log(p_i) = \log(p_{i-1}) + \log \left(\frac{N-(i-1)}{i} \right)$$
   (b) Aggiungere $\log(p_i)$ alla lista dei log-probabilità
   (*) *Evita underflow:* $\log(p_i)$ rimane finito anche per N grandi.
6. Calcolare la somma dei log-probabilità in modo stabile:
   - Sia $a = \max\{\log(p_k), \log(p_{k+1}), \ldots, \log(p_N)\}$
   - Calcolare:
$$\log(P) = a + \log \left(\sum_{i=k}^{N} e^{\log(p_i) - a} \right)$$
   - Ottenere la probabilità finale:
$$P = e^{\log(P)}$$

Salvare il valore calcolato di P

Confronto tra calcolo diretto e ottimizzato

Esempio:

Parametri

Consideriamo:

- $N = 10^6$ (numero totale di prove)
- $M = 1000$ (parametro aggiustamento)
- Soglia $k = \left\lfloor \frac{10^6 - 1000}{2} \right\rfloor + 1 = 499501$

Calcolo diretto (problemi)

Il calcolo diretto della probabilità:

$$P = \sum_{i=k}^{N} \binom{N}{i} \left(\frac{1}{2}\right)^N$$

genera errori di overflow perché:

- $\binom{10^6}{499501} \approx 10^{301029}$ (numero enorme)
- $\left(\frac{1}{2}\right)^{10^6} \approx 10^{-301029}$ (numero piccolissimo)

Il prodotto dà teoricamente ≈ 1, ma numericamente causa problemi.

Calcolo ottimizzato (soluzione)

Usando logaritmi ed esponenziali:

$$\begin{aligned} \log p_{499501} &= \log \binom{10^6}{499501} - 10^6 \log 2 \approx 0 \\ \log P &= \text{log-sum-exp}(\{\log p_i\}_{i=k}^{N}) \approx 13.12 \\ P &= e^{13.12} \approx 1.0 \end{aligned}$$

Questo approccio evita problemi numerici e dà il risultato corretto.

Conclusione

Nel corso dell'analisi, sono stati presentati due approcci per il calcolo della probabilità di vittoria del candidato A, basati su modelli binomiali.
Il primo approccio, che utilizza direttamente le funzionalità offerte dalla libreria `numpy`, permette di ottenere risultati in modo rapido e con codice semplice, beneficiando implicitamente di alcune ottimizzazioni interne fornite dalla libreria stessa.
Questo metodo si rivela particolarmente utile quando l'obiettivo è comprendere il comportamento qualitativo del sistema o visualizzare rapidamente l'andamento della probabilità al variare dei parametri.
Il secondo approccio, invece, introduce tecniche più avanzate di ottimizzazione numerica, come il calcolo nello spazio logaritmico, l'uso della funzione `logsumexp` per la somma stabile di probabilità molto piccole, e la valutazione dei coefficienti binomiali tramite la funzione `lgamma`.
Queste tecniche garantiscono una maggiore precisione e stabilità numerica, specialmente quando si opera con valori molto grandi di N, come nel caso considerato.
Tuttavia, ciò comporta un incremento del costo computazionale e, di conseguenza, tempi di esecuzione più elevati.
In conclusione, la scelta tra i due approcci dipende dal contesto applicativo: se si è interessati a una valutazione qualitativa e veloce, il primo metodo è più che sufficiente; se invece si richiede un'analisi quantitativa accurata e robusta, allora è preferibile adottare il secondo approccio, accettando un maggiore onere computazionale in cambio di una precisione superiore.