Logica e Metodi Probabilistici per L'Informatica - Homework

Academic year 2022/2023

Instructor: Prof. Stefano Leonardi   Teaching Assistant: Federico Fusco

Deadline: 05/06/2023

Regole generali: Potete consegnare gli esercizi in gruppi da due studenti. Tuttavia, assicuratevi di aver compreso ogni dettaglio dell'elaborato che consegnate! Non copiate da altri studenti, da internet o da altre fonti, né condividete la vostra soluzione con altri gruppi; in caso contrario l'elaborato verrà invalidato e dovrete sostenere l'esame scritto. Non ci saranno eccezioni. Se avete dubbi sull'interpretazione dell'homework potete scrivere a fuscof@diag.uniroma1.it.

Consegna in ritardo: Il punteggio degli elaborati consegnati dopo la scadenza verrà penalizzato in base ai giorni di ritardo: un giorno di ritardo corrisponde ad una penalizzazione del 10% rispetto al punteggio effettivo, due giorni al 20%, tre giorni al 30%. Elaborati inviati con più di tre giorni di ritardo non verranno corretti.

Formato consegna: Gli elaborati dovranno essere obbligatoriamente scritti in LaTex (almeno 11 pt) e riportare il nome dei due componenti del gruppo su ogni foglio. Ogni risposta dovrà essere adeguatamente motivata e riportare il numero dell'esercizio a cui si riferisce. Consegnare gli elaborati via mail a fuscof@diag.uniroma1.it usando l'indirizzo istituzionale @studenti.uniroma1.it. Le mail dovranno riportare "LMPI 2022/2023 Homework" in oggetto e contenere un solo file pdf nominato "cognome1_cognome2.pdf". Il pdf non dovrà contenere più di 4 pagine, le ulteriori pagine saranno ignorate.

Testo degli esercizi

Exercise 1. Supponiamo di osservare una sequenza di oggetti che arrivano uno dopo l'altro, di cui ne possiamo mantenere solo k contemporaneamente in memoria. Ogni nuovo elemento può essere scartato o salvato in memoria (eventualmente rimuovendo un elemento salvato precedentemente per fargli spazio); elementi scartati o rimossi sono persi per sempre e non possono essere più considerati. Consideriamo l'algoritmo di campionamento descritto di seguito come algorithm 1.

Per ogni istante di tempo t, sia N_t l'insieme dei primi t elementi della sequenza e sia S_t la versione di S alla fine dell'iterazione del ciclo for relativo al t^o elemento della sequenza.

(a) Dimostrare che, per ogni t, l'insieme S_t è estratto uniformemente a caso tra tutti i sottinsiemi di N_t di k elementi. Formalmente, dimostrare che

$$\mathbb{P}\left(S_t = X\right) = \binom{t}{k}^{-1}, \ \forall t \geq k, \forall X \subseteq N_t, \ |X| = k.$$

Algorithm 1 Campionamento online di k elementi

```
S ← ∅, t ← 0
for ogni nuovo elemento e che arriva do
    t ← t + 1
    if |S| < k then
        Inserire e in S
    else
        Lanciare una moneta con probabilità di testa k/t
        if testa then
            Rimuovere un elemento u.a.c. da S ed inserire e al suo posto
```

(b) S_t e S_ℓ sono dipendenti o indipendenti, per $t \neq \ell$?

Exercise 2. Consideriamo un messaggio $b \in \{-1, 1\}$ inviato a un ricevitore, che lo riceve disturbato dal rumore prodotto da altre comunicazioni vicine, secondo il seguente modello semplificato. Ci sono n altri messaggi $b_i \in \{-1, 1\}$, per ogni $i = 1, \ldots, n$, ognuno dei quali è associato ad una potenza di segnale p_i. Il nostro ricevitore riceve il seguente messaggio:

$$s = b + \sum_{i=1}^{n} p_i b_i.$$

Se s è più vicino a $+1$ che a -1, allora il ricevitore decodifica il messaggio b come 1, altrimenti come -1.

Assumiamo che tutti i b_i siano variabili aleatorie indipendenti ed uniformi in $\{-1, 1\}$. Dare un risultato di concentrazione sulla probabilità che il ricevitore faccia un errore nel decodificare il messaggio.

Exercise 3. Si consideri l'algoritmo randomizzato per l'estrazione del k^o elemento in una lista non ordinata di numero distinti presentato di seguito come algorithm 2 e rispondere alle seguenti domande:

1. Si dimostri che l'algoritmo restituisce la risposta corretta.
2. Sia N il numero (casuale) di confronti operato dall'algoritmo. Quanto vale N nel caso peggiore?
3. Dimostrare che $\mathbb{E}[N] \in O(n)$.
4. Dimostrare che $N \in O(n)$ con alta probabilità. *Suggerimento*: abbiamo visto a lezione un algoritmo simile...

Exercise 4 (Stima del coefficiente di clustering)**.** Sia $G = (V, E)$ un grafo non diretto. Per ogni vertice $v \in V$, denotiamo con $N(v)$ il suo vicinato, i.e. $N(v) = \{u \in V | \{u, v\} \in E\}$. Il coefficiente di clustering C_v di un vertice v di G è definito come la probabilità che due vicini scelti a caso siano connessi da un arco:

$$C_v = \frac{|\{\{u, w\} \in E : u \in N(v) \text{ and } w \in N(v)\}|}{\binom{|N_v|}{2}}.$$

Se $|N(v)| < 2$, adottiamo la convenzione che $C_v = 0$. Il coefficiente di clustering C_G di un grafo G è la media tra i coefficienti di clustering dei suoi vertici:

$$C_G = \frac{1}{|V|} \sum_{v \in V} C_v$$

Algorithm 2 Algoritmo randomizzato per l'estrazione del k^o elemento

Input: Una lista $S = \{x_1, x_2, \ldots, x_n\}$ di n numeri distinti
Output: il k^o numero più piccolo in S
Seleziona un elemento x di S uniformemente a caso. ▷ x è il pivot!
Confronta ogni elemento di S con x e dividi $S \setminus \{x\}$ in due liste:

(a) S_1 contiene tutti gli elementi in S più piccoli di x

(b) S_2 tutti quelli maggiori di x

Sia $i = |S_1|$
if $i = k - 1$ **then**
    Restituisci x
else if $i \geq k$ **then**
    Richiama ricorsivamente l'algoritmo per trovare il k^o elemento di S_1
else
    Richiama ricorsivamente l'algoritmo per trovare il $(k - i - 1)^o$ elemento di S_2

1. Descrivere un algoritmo deterministico che calcoli il coefficiente di clustering in tempo polinomiale (in $n = |V|$ ed $m = |E|$). Dimostrare la sua correttezza ed analizzare il suo costo computazionale.
2. Si consideri ora l'algoritmo randomizzato descritto di seguito. Dimostrare che, in valore atteso, restituisce una stima corretta del coefficiente di clustering, i.e. $\mathbb{E}[X_i] = C_G$.
3. Si dimostri che, per ogni $\varepsilon > 0$, $\ell \geq \frac{3}{\varepsilon^2 C_G} \ln(\frac{2}{\delta})$ campionamenti sono sufficienti ad ottenere, con probabilità almeno $1 - \delta$, una $(1 \pm \varepsilon)$ approssimazione di C_G, cioè:

$$(1 - \varepsilon)C_G \leq X \leq (1 + \varepsilon)C_G.$$

Algorithm 3 Algoritmo randomizzato per il calcolo del coefficiente di clustering

Input: Un grafo $G = (V, E)$, il numero ℓ di campionamenti da eseguire
Output: X, una stima di C_G
for $i = 1, \ldots, \ell$ **do**
    Sia u un nodo estratto uniformemente a caso
    if $|N(u)| > 1$ **then**
        Estrarre una coppia di nodi v, w da $N(u)$ uniformemente a caso (senza sostituzione)
        if $\{v, w\} \in E$ **then**
            $X_i \leftarrow 1$
        else
            $X_i \leftarrow 0$
Restituire $X \leftarrow \frac{1}{\ell} \sum_{i=1}^{\ell} X_i$

Exercise 5 (Tightrope walking)**.** Frederick likes to do tightrope walking when he is completely drunk. The rope is $a + b$ meters long and there is land at position 0 and a river at position $a + b$. Frederick starts at position a of a rope, and he makes with equal probability a step of 1 meter on the left or on the right. Frederick stops when he is safe on lands or when he falls in water.

1. Model this problem as a Markov Chain. Specify the states and the transition matrix.

2. Is the chain aperiodic? Is it irreducible? Argue about its ergodicity.
3. Which is the probability that Frederick falls into water?
4. Which is the expected number of steps Frederick makes before he falls into water or he is safe on land.
5. Which is the expected number of steps needed by Frederick to reach one of the two endpoints when he is slightly less drunk and moves to the left with probability $2/3$ and to the right with probability $1/3$?