Complementi di Algoritmi e Strutture Dati

Algoritmi Probabilistici

Docente: *Nicolò Cesa-Bianchi* versione 12 marzo 2025

Un **algoritmo probabilistico** è un algoritmo che ha accesso a un oracolo che, ad ogni chiamata, restituisce in tempo unitario un bit causale, cioè una variabile casuale Y tale che $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 0) = \frac{1}{2}$. Inoltre, i bit restituiti in una sequenza di chiamate all'oracolo sono indipendenti. Nel seguito, indicheremo con $Z \in \{0,1\}^*$ la stringa di bit casuali indipendenti che l'oracolo restituisce in una sequenza di chiamate. Indicheremo anche con $A(I, Z) \in \{0, 1\}$ la variabile casuale che rappresenta l'output dell'algoritmo probabilistico A per un problema di decisione $X = (\mathcal{I}, q)$ e avente come input l'istanza $I \in \mathcal{I}$ e i bit casuali Z dell'oracolo. In modo simile, indichiamo con $T_A(I, Z)$ la variabile casuale che rappresenta il tempo di esecuzione di A con input $I \in \mathcal{I}$ e bit casuali Z forniti dall'oracolo.

Esistono due principali tipi di algoritmi probabilistici.

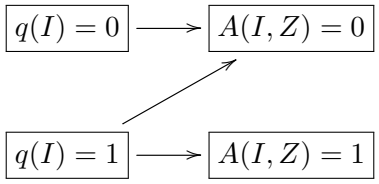
Algoritmi Montecarlo. Sono algoritmi A tali che:

- Per ogni $I \in \mathcal{I}$, $T_A(I, Z)$ dipende solo da I (ovvero il tempo di esecuzione è deterministico).
- Esiste $I \in \mathcal{I}$ per cui $\mathbb{P}\big(A(I, Z) \neq q(I)\big) > 0$ (ovvero l'output non è sempre corretto).

Gli algoritmi Montecarlo si dividono ulteriormente in:

- Algoritmi con errore **one-sided**. Un algoritmo ha errore one-sided quando è sempre corretto almeno su uno dei suoi due possibili output. Convenzionalmente, assumeremo che l'algoritmo sia sempre corretto quando il suo output è 1. Ovvero, A è Montecarlo one-sided quando, per ogni $I \in \mathcal{I}$, $\mathbb{P}\big(A(I, Z) = q(I) \mid A(I, Z) = 1\big) = 1$ e $\mathbb{P}\big(A(I, Z) = q(I) \mid A(I, Z) = 0\big) > 0$.
- Algoritmi con errore **two-sided**. Possono sbagliare su entrambi i possibili output. Ovvero, A è Montecarlo two-sided quando $\mathbb{P}\big(A(I, Z) = q(I)\big) > \frac{1}{2}$ per ogni $I \in \mathcal{I}$.

Una rappresentazione grafica della relazione fra $q(I)$ e $A(I, Z)$ per un algoritmo Montecarlo one-sided è la seguente



Questo mostra come $A(I, Z) = 1$ può solo corrispondere a $q(I) = 1$ mentre $A(I, Z) = 0$ lascia incertezza sul valore di $q(I)$. Si noti anche che quando $q(I) = 0$ l'algoritmo è sempre corretto.

Algoritmi Las Vegas. Sono algoritmi che producono sempre l'output corretto ma che hanno un tempo di esecuzione probabilistico (ovvero che dipende dai bit forniti dall'oracolo). Ovvero, un algoritmo A è Las Vegas quando $\mathbb{P}\big(A(I, Z) = q(I)\big) = 1$ per ogni $I \in \mathcal{I}$ ma il tempo di calcolo di A su una qualunque istanza $I \in \mathcal{I}$ è una variabile casuale $T_A(I, Z)$ tale che $\mathbb{E}\big[T_A(I, Z)\big] < \infty$.

Amplificazione. Un algoritmo Montecarlo one-sided può essere facilmente trasformato in un algoritmo con probabilità di errore arbitrariamente piccola attraverso un meccanismo di amplificazione.

Sia

$$\mathbb{P}\big(A(I,Z) \neq q(I) \mid A(I,Z) = 0\big) \leq 1 - p_n \qquad \text{per ogni } I \in \mathcal{I} \text{ con } |I| = n$$

Se su una data istanza l'algoritmo produce 0 in output possiamo eseguirlo nuovamente per amplificare la probabilità di avere l'output corretto. Se k esecuzioni indipendenti producono sistematicamente la risposta 0 allora la probabilità che la risposta corretta sia 1 è al più $(1-p_n)^k \leq e^{-p_n k}$. Perché questa probabilità sia al più un $\varepsilon > 0$ piccolo a piacere è sufficiente scegliere $k \geq \frac{1}{p_n} \ln \frac{1}{\varepsilon}$.

Un meccanismo di amplificazione simile ma leggermente più complesso esiste anche per gli algoritmi Montecarlo two-sided. Supponiamo che su istanze di taglia n l'algoritmo fornisca la risposta errata con probabilità al più $\frac{1}{2} - p_n < \frac{1}{2}$. Per amplificare la probabilità di ottenere la risposta corretta, possiamo ripetere l'esecuzione k volte e utilizzare un voto di maggioranza sui k output prodotti (per semplicità, supponiamo che k sia dispari).

Per analizzare il voto di maggioranza utilizzeremo il seguente lemma (dimostrazione omessa).

Lemma 1 (Chernoff-Hoeffding) *Siano $Y_1, \ldots, Y_k$ variabili casuali Bernoulliane (cioè con valori in $\{0,1\}$), indipendenti e tali che $\mathbb{P}(Y_t = 1) \leq \mu$ per $t = 1, \ldots, k$. Allora, per ogni $\varepsilon > 0$ fissato,*

$$\mathbb{P}\left(\frac{1}{k}\sum_{t=1}^{k} Y_t > \mu + \varepsilon\right) \leq e^{-2\varepsilon^2 k}$$

Sia $p_E = \mathbb{P}\big(A(I,Z) \neq q(I)\big)$ la probabilità di errore dell'algoritmo su un'istanza I del problema di decisione $(\mathcal{I}, q)$. Siano $X_1, \ldots, X_k \in \{0,1\}$ le variabili casuali indipendenti che denotano gli output prodotti dalle k esecuzioni dell'algoritmo. Sia $M_k \in \{0,1\}$ la variabile casuale che denota il voto di maggioranza su $X_1, \ldots, X_k$ (ovvero $M_k = 1$ se e solo se $\sum_{t=1}^{k} X_t > \frac{k}{2}$). Allora,

$$M_k = q(I) \quad \Longleftrightarrow \quad \sum_{t=1}^{k} Y_t < \frac{k}{2}$$

dove $Y_t = 1$ se e solo se $X_t \neq q(I)$, per $t = 1, \ldots, k$. In altre parole, il voto di maggioranza M_k è corretto se e solo se l'algoritmo genera l'output errato in non più di $\left\lfloor \frac{k}{2} \right\rfloor$ delle k esecuzioni. Ora, $Y_1, \ldots, Y_k$ sono variabili casuali indipendenti (perché le esecuzioni dell'algoritmo sono indipendenti), identicamente distribuite, con valori in $\{0,1\}$ e tali che $\mathbb{P}(Y_t = 1) = p_E \leq \frac{1}{2} - p_n$ per ogni $t = 1, \ldots, k$. Applicando il lemma di Chernoff-Hoeffding otteniamo che la probabilità che il voto di maggioranza sia sbagliato è limitata da

$$\mathbb{P}\left(\sum_{t=1}^{k} Y_t > \frac{k}{2}\right) = \mathbb{P}\left(\frac{1}{k}\sum_{t=1}^{k} Y_t > \left(\frac{1}{2} - p_n\right) + p_n\right) \leq \mathbb{P}\left(\frac{1}{k}\sum_{t=1}^{k} Y_t > p_E + p_n\right) \leq e^{-2p_n^2 k} \ .$$

Perché la probabilità $e^{-2p_n^2 k}$ sia al più un $\varepsilon > 0$ piccolo a piacere è sufficiente scegliere $k \geq \frac{1}{2p_n^2} \ln \frac{1}{\varepsilon}$.

Si noti che nel caso one-sided possiamo usare l'amplificazione per ridurre qualsiasi probabilità di errore strettamente minore di 1 mentre nel caso two-sided la stessa cosa vale per qualsiasi probabilità strettamente minore di $\frac{1}{2}$.

Da Las Vegas a Montecarlo one-sided. Un algoritmo Las Vegas per un problema di decisione può essere trasformato in un algoritmo Montecarlo one-sided. Per far ciò utilizziamo la disuguaglianza di Markov.

Lemma 2 (Markov) *Sia Z una variabile casuale non negativa tale che $\mathbb{E}[Z] < \infty$. Allora per ogni $c > 0$,*

$$\mathbb{P}\big(Z > c\big) \leq \frac{\mathbb{E}[Z]}{c} \ .$$

Dimostrazione. Sia $\mathcal{A}$ l'insieme di numeri non negativi tali che $Z \in \mathcal{A}$. Allora

$$\begin{aligned}\mathbb{E}[Z] &= \sum_{a \in \mathcal{A}} a\,\mathbb{P}(Z = a) = \overbrace{\sum_{a \in \mathcal{A}\,:\,a \leq c} a\,\mathbb{P}(Z = a)}^{\geq 0} + \sum_{a \in \mathcal{A}\,:\,a > c} a\,\mathbb{P}(Z = a) \\ &\geq c \sum_{a \in \mathcal{A}\,:\,a > c} \mathbb{P}(Z = a) = c\,\mathbb{P}(Z > c)\end{aligned}$$

che conclude la dimostrazione. □

Sia A un algoritmo Las Vegas per un problema di decisione $(\mathcal{I}, q)$. Sia $f : \mathbb{N} \to \mathbb{R}$ la funzione tale che

$$f(n) = \max\big\{\mathbb{E}\big[T_A(I, Z)\big] \ : \ I \in \mathcal{I},\, |I| = n\big\} \ .$$

Dato che A è Las Vegas, $f(n) < \infty$ per ogni $n \in \mathbb{N}$. Quindi posso trovare una funzione $t : \mathbb{N} \to \mathbb{N}$ tale che

$$t(n) \geq \frac{3}{2} f(n) \qquad n \in \mathbb{N}$$

Posso quindi costruire un algoritmo A' che simula A sull'istanza I arrestando la simulazione dopo $t(n)$ passi. Se A non ha terminato allora A' produce 0 in output. Dato che A è Las Vegas, A' sbaglia solo quando A non termina entro $t(n)$ passi. Per la disuguaglianza di Markov, la probabilità che ciò accada è al più

$$\mathbb{P}\big(T_A(I, Z) > t(n)\big) \leq \frac{\mathbb{E}\big[T_A(I, Z)\big]}{t(n)} \leq \frac{f(n)}{t(n)} \leq \frac{2}{3} \ .$$

Inoltre, dato che quando A non termina l'output di A' è 0, A' è one-sided dato che l'output 1 è sempre corretto. Quindi A' è un algoritmo Montecarlo one-sided con probabilità di errore al più $\frac{2}{3}$. Infine, si noti che il tempo di esecuzione di A' soddisfa $T_{A'}(I) \leq t(|I|) = \mathcal{O}\big(f(|I|)\big)$.