Complementi di Algoritmi e Strutture Dati

Algoritmi Probabilistici

Docente: *Nicolò Cesa-Bianchi* versione 13 marzo 2018

Un **algoritmo probabilistico** è un algoritmo che ha accesso a un oracolo che, ad ogni chiamata, restituisce in tempo unitario un bit causale indipendente, cioè una variabile casuale Z tale che $\mathbb{P}(Z = 1) = \mathbb{P}(Z = 0) = \frac{1}{2}$. Esistono due principali tipi di algoritmi probabilistici:

- **Algoritmi Montecarlo.** Sono algoritmi che hanno un tempo di esecuzione deterministico (ovvero indipendente dai bit forniti dall'oracolo) ma che con una certa probabilità producono l'output errato. Nel caso dei problemi di decisione, 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.
  - Algoritmi con errore **two-sided**. Possono sbagliare su entrambi i possibili output.
- **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). Si richiede che, per ogni intero n, il tempo di esecuzione su una qualsiasi istanza di lunghezza n abbia un valore atteso finito.

Un algoritmo Montecarlo one-sided può essere facilmente trasformato in un algoritmo con probabilità di errore arbitrariamente piccola attraverso un meccanismo di **amplificazione**. Sia $1 - p_n < 1$ un maggiorante della probabilità di errore sulle istanze di taglia n quando l'output è 0 (ovvero, con probabilità al più $1 - p_n$ l'algoritmo ritorna 0 mentre l'output corretto è 1). 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 $(1 - p_n)^k \le e^{-p_n k}$. Perché questa probabilità sia al più un $\varepsilon > 0$ piccolo a piacere è sufficiente scegliere $k \ge \frac{1}{p_n} \ln \frac{1}{\varepsilon}$.

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.

Lemma 1 (Chernoff-Hoeffding) *Siano $Z_1, \dots, Z_k$ variabili casuali Bernoulliane (cioè con valori in $\{0, 1\}$), indipendenti e tali che $\mathbb{P}(Z_t = 1) = \mu$ per $t = 1, \dots, k$. Allora, per ogni $\varepsilon > 0$ fissato,*

$$\mathbb{P}\left(\frac{1}{k}\sum_{t=1}^{k} Z_t > \mu + \varepsilon\right) \le e^{-2\varepsilon^2 k} \qquad e \qquad \mathbb{P}\left(\frac{1}{k}\sum_{t=1}^{k} Z_t < \mu - \varepsilon\right) \le e^{-2\varepsilon^2 k} \ .$$

Siano $X_1, \dots, X_k \in \{0, 1\}$ le variabili casuali indipendenti che denotano gli output prodotti dalle k esecuzioni dell'algoritmo su un'istanza I del problema di decisione $(\mathcal{I}, q)$. 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,

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

dove $Z_t = 1$ 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, $Z_1, \ldots, Z_k$ sono variabili casuali indipendenti (perché le esecuzioni dell'algoritmo sono indipendenti), identicamente distribuite, con valori in $\{0, 1\}$ e tali che $\mathbb{P}(Z_t = 1) = \frac{1}{2} - p_n$ 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

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

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 $\frac{1}{2}$.

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 $(\mathcal{I}, q)$ e sia $T_A(I)$ la variabile casuale che indica il tempo di calcolo dell'algoritmo sull'istanza $I \in \mathcal{I}$. Sia $f : \mathbb{N} \to \mathbb{R}$ la funzione tale che

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

Dato che A è Las Vegas, $f(n) < \infty$ per ogni $n \in \mathbb{N}$. Quindi per ogni istanza I con $n = |I|$ posso trovare un intero $t(n)$ tale che

$$\frac{f(n)}{t(n)} \leq \frac{2}{3} \ .$$

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

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

Inoltre, dato che quando A non termina l'output di A' è 0, A' è one-sided dato che è l'output 1 è sempre corretto. Quindi A' è un algoritmo Montecarlo one-sided con probabilità di errore al più $\frac{2}{3}$.

Viceversa, un algoritmo Montecarlo two-sided A per un problema di decisione $X = (\mathcal{I}, q)$ può essere trasformato in un algoritmo Las Vegas qualora A produca in output anche un certificato $z \in \{0,1\}^*$ che può essere verificato da un algoritmo deterministico B. Ovvero, supponiamo che su input $I \in \mathcal{I}$ l'algoritmo A produca la risposta $y \in \{0,1\}$ e il certificato $z \in \{0,1\}^*$. Allora se B su input (I,z) restituisce 1 sappiamo che $y = q(I)$, mentre se B restituisce 0 allora $q(I) = 1$ e $q(I) = 0$ sono entrambi possibili. Supponiamo inoltre che la probabilità che A restituisca un certificato errato (ovvero tale che l'output di B sia 0) su un'istanza di taglia n sia al più $\frac{1}{2} - p_n$.

Per l'analisi abbiamo bisogno del lemma seguente.

Lemma 3 (Valore atteso distribuzione Geometrica) *Siano $Z_1, Z_2, \dots$ variabili casuali Bernoulliane, indipendenti e tali che $\mathbb{P}(Z_t = 1) = p$ per $t \geq 1$. Sia $G = \min\{k \,:\, Z_k = 1\}$. Allora*

$$\mathbb{E}[G] = \frac{1}{p} \; .$$

Dimostrazione.

$$\begin{aligned}
\mathbb{E}[G] &= \sum_{k=1}^{\infty} k(1-p)^{k-1} p = p \sum_{k=1}^{\infty} k(1-p)^{k-1} = -p \sum_{k=1}^{\infty} \frac{d}{dp}(1-p)^k \\
&= -p \, \frac{d}{dp} \sum_{k=1}^{\infty} (1-p)^k = -p \, \frac{d}{dp} \left(\frac{1}{1-(1-p)} - 1 \right) = -p \, \frac{d}{dp} \frac{1-p}{p} \\
&= -p \, \frac{-1}{p^2} = \frac{1}{p} \; .
\end{aligned}$$

□

Definiamo l'algoritmo A' che, su input I di lunghezza n, simula A sullo stesso input. Quando A si arresta con output (y,z) l'algoritmo A esegue B su input (I,z). Se B termina con 1 allora sappiamo che $q(I) = y$. Altrimenti, A' continua a ripetere la simulazione di A su input I finché B non termina con 1. Dato che su istanze di lunghezza n l'algoritmo A sbaglia con probabilità al più $\frac{1}{2} - p_n$, ci vorranno in media al più $\frac{1}{\frac{1}{2}+p_n}$ simulazioni di A prima che A produca un certificato valido. Quindi, il tempo atteso di calcolo di A' è

$$\mathbb{E}\big[T_{A'}(n)\big] = \frac{T_A(n) + T_B(n)}{\frac{1}{2} + p_n} < \infty$$

dove $T_A(n) + T_B(n)$ è il tempo che A' impiega per simulare A ed eseguire B per verificare se la soluzione è corretta. Quindi A' è un algoritmo Las Vegas.