الگوریتم‌های آنلاین

کاوه حسینی

بخش دوم - الگوریتم‌های آنلاین تصادفی[1]

۱ معرفی الگوریتم‌های آنلاین تصادفی

در بسیاری از مسایل از جمله مسئله‌ی صفحه‌بندی ،الگوریتم‌های آنلاین اگر انتخاب‌های تصادفی داشته باشند ممکن است کارایی بهتری داشته‌باشند.

تعریف ۱. *الگوریتم تصادفی آنلاین A یک توزیع احتمال $\{A_x\}$ روی فضای الگوریتم‌های قطعی آنلاین است.*

ضریب رقابتی یک الگوریتم تصادفی آنلاین ALG نسبت به یک دشمن خاص تعریف می‌شود. دشمن دنباله‌ی درخواست‌های σ را تولید می‌کند و هنگام تولید دنباله از ساز و کار الگوریتم ALG آگاه است. حال سوال اساسی این است: هنگام تولید درخواست‌ها آیا دشمن می‌تواند انتخاب‌های تصادفی انجام شده‌ی قبلی توسط ALG را ببیند یا نه؟ دشمن‌های فراموشکار[2] برخلاف دشمن‌های توافقی[3] این توانایی را ندارند. سه نوع دشمن توسط بن، دیوید و بقیه در [۲] معرفی شده‌اند که در زیر آورده شده است.

تعریف ۲. *دشمن فراموشکار:دشمن فراموشکار همه‌ی دنباله‌ی درخواست را باید از همان اول و قبل از اینکه هر درخواستی پاسخ داده‌شود تولید کند.ولی از نحوه‌ی توزیع احتمال روی الگوریتم‌های قطعی آگاه است.*

تعریف ۳. *دشمن توافقی آنلاین[4]: این دشمن می‌تواند الگوریتم آنلاین را ببیند و درخواست بعدی خود را برمبنای پاسخ الگوریتم به درخواست‌های قبلی بدهد.دشمن بایستی درخواست‌های خود را به صورت آنلاین و بدون اطلاع از پاسخ الگوریتم به درخواست‌های حال و آینده مطرح کند.*

تعریف ۴. *دشمن توافقی آفلاین[5]: این دشمن همانند دشمن توافقی آنلاین است با این تفاوت که می‌تواند دنباله را به شکل آفلاین تولید کند.*

تعریف ۵. *به الگوریتم تصادفی آنلاین ALG c - رقابتی نسبت به دشمن فراموشکار گفته می‌شود اگر ثابت b وجود داشته باشد به طوری که برای هر دنباله‌ی درخواست σ که توسط دشمن فراموشکار تولید شده‌است داشته‌باشیم، $E[ALG(\sigma)] \leq c.OPT(\sigma) + b$. امید ریاضی روی همه‌ی انتخاب‌های تصادفی ALG با توجه به تابع توزیع احتمال مربوطه گرفته‌می‌شود.*

[1] Randomized On-line Algorithm
[2] Oblivious Adversary
[3] Adaptive adversary
[4] Adaptive online adversary
[5] Adaptive offline adversary

فرض کنید الگوریتم تصادفی آنلاین ALG و دشمن توافقی آنلاین (توافقی آفلاین) ADV داده‌شده‌است و $E[ALG(\sigma)]$ و $E[ADV(\sigma)]$ به ترتیب امید ریاضی هزینه‌ی پاسخ به دنباله‌ی تولیدشده توسط ADV برای ALG و ADV باشد. به الگوریتم ALGc –رقابتی نسبت به دشمن توافقی آنلاین (آفلاین) گفته‌می‌شود اگر ثابت b وجود داشته باشد به طوری که برای همه‌ی دشمن‌های توافقی آنلاین (آفلاین) ADV، $E[ALG(\sigma)] \leq c.E[ADV(\sigma)] + b$ که امید ریاضی روی همه‌ی انتخاب‌های تصادفی ALG گرفته‌می‌شود.

قضیه ۶. *اگر یک الگوریتم تصادفی آنلاین c – رقابتی نسبت به دشمن توافقی آفلاین وجودداشته‌باشد، یک الگوریتم آنلاین c – رقابتی قطعی وجود دارد.*[۲]

قضیه ۷. *اگر ALG یک الگوریتم تصادفی آنلاین c – رقابتی نسبت به دشمن توافقی آنلاین باشد، ALG یک الگوریتم تصادفی آنلاین $(c.d)$ – رقابتی نسبت به دشمن توافقی آفلاین است.*[۲]

به عبارتی دیگر از قضیه ۶ نتیجه می‌شود که تصادفی کردن الگوریتم در مقابل دشمن‌های توافقی تاثیری ندارد.

نتیجه ۸. *اگر یک الگوریتم تصادفی c –رقابتی نسبت به دشمن توافقی آنلاین وجود داشته‌باشد، آنگاه یک الگوریتم قطعی c^2 – رقابتی قطعی وجود دارد.*

راقاوان و سنیر [۵] نشان دادند در مقابل دشمن‌های توافقی هیچ الگوریتم تصادفی آنلاین برای مسئله‌ی صفحه‌بندی از k – رقابتی بهتر وجود ندارد. به همین دلیل روی دشمن‌های فراموشکار تمرکز می‌کنیم و نشان می‌دهیم می‌توان کران k برای الگوریتم‌های قطعی را با تصادفی کردن به شکل نمایی بهبود بخشید. یکی از این الگوریتم‌ها، علامت‌گذاری تصادفی[۶] است که توسط فیات و بقیه [۳] ارائه شد.

علامت‌گذاری تصادفی: الگوریتم از استراتژی علامت‌گذاری استفاده می‌کند. با هر بار بروز خطا یکی از بخش‌های بدون علامت به طور تصادفی انتخاب شده و حذف می‌شود.

الگوریتم علامت‌گذاری تصادفی

در ابتدا همه‌ی بخش‌ها علامت‌دار شده‌اند. با درخواست بخش p:

۱. اگر p در M_1 وجود ندارد:
   - اگر همه‌ی بخش‌ها در M_1 علامت‌دار هستند، علامت همه را بردار.
   - p را با بخشی که به طور تصادفی از بین بخش‌های بدون علامت انتخاب شده‌است عوض کن.

۲. p را علامت‌دار کن.

عدد $H_k = \sum_{i=1}^{k} 1/i$ را k–امین عدد هارمونیک بگیرید که می‌توان با $\ln k$ تقریب زد. داریم:

$$\ln(k+1) \leq H_k \leq \ln k + 1$$

قضیه ۹. *الگوریتم علامت‌گذاری تصادفی $2H_k$– رقابتی است.*[۳]

قضیه ۱۰. *ضریب رقابتی هیچ الگوریتم تصادفی آنلاین صفحه‌بندی نسبت به دشمن فراموشکار از H_k کمتر نیست.*[۳]

الگوریتم‌های پیچیده‌تری در [۱و۴] معرفی شده‌اند.

۲ تحلیل الگوریتم علامت‌گذاری تصادفی

مرجع [۶] را ببینید.

[۶] Randomized Marking

۳ کران پایین برای همه‌ی الگوریتم‌های آنلاین تصادفی

۱.۳ یک روش مفید

چگونه می‌توانیم یک کران پایین برای ضریب رقابتی هر الگوریتم تصادفی نسبت به دشمن فراموشکار بیابیم؟ این کار را با انتخاب یک توزیع D از دنباله‌های ورودی و محاسبه‌ی امید ریاضی هزینه‌ی بهترین الگوریتم آنلاین و مقایسه‌ی آن با امید ریاضی الگوریتم MIN روی توزیع D انجام می‌دهیم.

فرض کنید $C_H(\sigma)$ هزینه‌ی الگوریتم قطعی H روی دنباله‌ی σ باشد. می‌گوییم $A\alpha$–رقابتی است اگر برای هر دنباله‌ی σ:

$$Exp[C_{A_x}(\sigma)] \leq \alpha C_{MIN}(\sigma) + c$$

فرض کنید σ^j، j عضو اول σ باشند. فرض کنید یک توزیع D روی دنباله‌های ورودی σ داریم. j را ثابت بگیرید. از دو طرف نامساوی بالا می‌توان نسبت به دنباله‌ی σ روی توزیع D امید ریاضی گرفت.

$$Exp_y[Exp_x[C_{A_x}(\sigma_y^j)]] \leq \alpha Exp_y[C_{MIN}(\sigma_y^j)] + c$$

با استفاده از قضیه‌ی فوبینی[۷] می‌توان امید ریاضی‌ها را جابه‌جا کرد.

$$Exp_x[Exp_y[C_{A_x}(\sigma_y^j)]] \leq \alpha Exp_y[C_{MIN}(\sigma_y^j)] + c$$

قرار دهید

$$m_j = \min_H(Exp_y[C_H(\sigma_y^j)])$$

داریم

$$m_j \leq \alpha Exp_y[C_{MIN}(\sigma_y^j)] + c$$

بنابراین

$$\frac{m_j}{Exp_y[C_{MIN}(\sigma_y^j)]} \leq \alpha + \frac{c}{Exp_y[C_{MIN}(\sigma_y^j)]}$$

فرض کنید D طوری انتخاب شده‌است که

$$\lim_{j\to\infty} Exp_y[C_{MIN}(\sigma_y^j)] = \infty$$

بنابراین

$$\lim_{j\to\infty} \frac{m_j}{Exp_y[C_{MIN}(\sigma_y^j)]} \leq \alpha$$

این نامساوی بیانگر چیست؟ یعنی ضریب رقابتی هر الگوریتم تصادفی آنلاین حداقل به اندازه‌ی نسبت امید ریاضی هزینه‌ی بهترین الگوریتم قطعی آنلاین به امید ریاضی بهترین الگوریتم قطعی آفلاین است هنگامی که امید ریاضی روی دنباله‌های به اندازه‌ی کافی طولانی گرفته‌می‌شوند. می‌توانیم D را به طور دلخواه انتخاب کنیم تا نسبت را بیشینه کنیم.

حال قضیه ۱۰ را ثابت می‌کنیم:

اثبات. برای اینکه نامساوی

$$m_j \leq Exp_x[Exp_y[C_{A_x}(\sigma_y^j)]]$$

تا حد ممکن محکم[۸] باشد D را طوری انتخاب می‌کنیم که هر الگوریتم قطعی آنلاین به اندازه‌ی یکسان بد عمل کنند. در این حالت می‌توانیم این را با انتخاب σ_i به طور یکنواخت از بین $k+۱$ بخش موجود انتخاب کرد. با توجه به اینکه $M_۱$ فقط شامل k بخش است هر الگوریتم قطعی برای هر درخواست σ_i هزینه‌ی $\frac{۱}{k+۱}$ می‌پردازد پس $m_j = \frac{j}{k+۱}$.

با استفاده از روش ارائه‌شده نتیجه می‌گیریم

$$\alpha \geq \lim_{j\to\infty} \frac{j}{(k+۱)Exp_y[C_{A_x}(\sigma_y^j)]}$$

[۷] Fubini Theorem
[۸] Tight

حکم قضیه را می‌توان به شکل زیر نوشت

$$\lim_{j\to\infty} \frac{j}{Exp_y[C_{A_x}(\sigma_y^j)]} = (k+۱)H_k$$

برای اثبات این ادعا بایستی رفتار الگوریتم MIN را بررسی کنیم. σ را به مرحله‌های تصادفی تقسیم می‌کنیم. مرحله‌ی i شامل درخواست‌هایی با اندیس در $[X_i, X_i+۱, \dots, X_{i+۱}-۱]$ هستند که $X_۰ = ۱$ و

$$X_{i+۱} = \min\{t : \{\sigma_{X_i}, \sigma_{X_i+۱}, \dots, \sigma_t\} = \{۱, \dots, k+۱\}$$

توجه شود که X_iها متغیرهای تصادفی هستند. هر مرحله دارای درخواست به فقط k بخش است و اگر MIN در مرحله‌ای خطا داشته‌باشد، خطای بعدی نمی‌تواند قبل از مرحله‌ی بعد رخ‌دهد. بنابراین تعداد خطاهای MIN روی σ حداکثر برابر تعداد خطاهایی است که تا زمان j رخ می‌دهند. پس امید ریاضی هزینه‌ی MIN حداکثر برابر امید ریاضی تعداد مراحلی است که تا زمان j وجود دارند. پس

$$Exp_y[C_{MIN}(\sigma_y^j)] \leq ۱ + Exp[\max\{p : X_p \leq j\}]$$

با توجه به اینکه متغیرهای تصادفی $\{Y_i\} = \{X_{i+۱} - X_i : j \geq ۰\}$ مستقل و هم توزیع هستند، یک فرایند تجدید[۹] بدست می‌دهند. با استفاده از قضایای مقدماتی در نظریه‌ی فرایندهای تجدید داریم:

$$\begin{aligned}\lim_{j\to\infty} \frac{j}{۱ + Exp[\max\{p : X_p \leq j\}]} &= \lim_{j\to\infty} \frac{j}{Exp[\max\{p : X_p \leq j\}]} \\ &= Exp[\text{length of phase}] \\ &= Exp[X_۱ - ۱] \\ &= Exp[X_۱] - ۱\end{aligned}$$

نشان دادیم

$$\alpha \geq \frac{Exp[X_۱] - ۱}{k+۱}$$

حال بایستی $Exp[X_۱]$ را محاسبه کنیم. بنابر تعریف

$$X_۱ = \min\{t : \{\sigma_۱, \dots, \sigma_t\} = \{۱, \dots, k+۱\}\}$$

هر σ_i به طور یکنواخت بین $k+۱$ بخش توزیع شده است و از بقیه‌ی درخواست‌ها مستقل است. محاسبه‌ی $Exp[X_۱]$ در این حالت با مسئله‌ی جمع کننده‌ی کوپن[۱۰] هم‌ارز است. $Exp[X_۱]$ نشان‌دهنده‌ی امید ریاضی تعداد کوپن‌های لازم برای یک جمع کننده‌ی کوپن است به طوری که همه‌ی کوپن‌های مجزا را بدست آورد. برای حل مسئله تعریف می‌کنیم $Z_i = \min\{t :| \{\sigma_۱, \dots, \sigma_t\} |= i\}$ که $i = ۱\dots, k+۱$. حال داریم

$$\begin{aligned}Exp[X_۱] = Exp[Z_{k+۱}] &= \sum i = ۱^k (Exp[Z_{i+۱}] - Exp[Z_i]) + Exp[Z_۱] \\ &= \sum_{i=۱}^{k} Exp[Z_{i+۱} - Z_i] + ۱ \\ &= (k+۱)(۱ + \frac{۱}{۲} + \frac{۱}{۳} + \dots + \frac{۱}{k}) + ۱ \\ &= (k+۱)H_k + ۱\end{aligned}$$

پس

$$\alpha \geq \frac{[(k+۱)H_k + ۱] - ۱}{k+۱}$$

□

[۹] Renewal Process

[۱۰] Coupon collector problem

۴ تحلیل الگوریتم انتخاب تصادفی

مرجع [۶] را ببینید.

مراجع

[1] D. Achlioptas, M. Chrobak and J. Noga, *Competitive Analysis of Randomized Paging Algorithms* , Theoretical Computer Science, 234:203-218, 2000.

[2] S. Ben-David, A. Borodin, R.M. Karp, G. Tardos and A. Wigderson , *On the Power of Randomization in On-line Algrorithms* ,Algorithmica, 11:2-14, 1994.

[3] A. Fiat and RM. Karp and LA. McGeoch and DD. Sleator and NE. Young , *Competitive paging algorithms* , Journal of Algorithms 12:685–699, 1991.

[4] LA. McGeoch and DD. Sleator, *A Strongly Competitive Randomized Paging Algorithms* , Algorithmica, 6:816-825, 1991.

[5] P. Raghavan and M. Snir , *Memory Versus Randomization in On-line Algorithms* , IBM Journal of Research and Development, 38:683-708, 1994.

[6] M. X. Goemans , *Advanced Algorithms Cours* , Lecure Notes, September 1994.