

# الگوریتم های آنلاین

On-line Algorithms

مهدی محمدی

ایمیل مهدی محمدی

**چکیده:** یک الگوریتم آنلاین الگوریتمی است که دنباله ای از درخواست ها را دریافت می کند و برای پاسخ به هر درخواست یک عمل آنی انجام می دهد. الگوریتم های آنلاین در هر جایی که باید تصمیمی گرفته شود که منابع آن بدون دانشی از آینده تخصیص داده می شوند، ظاهر می شوند. موثر بودن یک الگوریتم آنلاین با استفاده از نرخ رقابتی<sup>۱</sup> آن، که به صورت بدترین حالت نسبت بین هزینه آن و هزینه الگوریتم آفلاین<sup>۲</sup> فرضی آن که در ابتدا کل دنباله را می داند و رفتارش را به صورت بهینه تعیین می کند تعریف می شود، اندازه گیری می شود.

واژه های کلیدی: الگوریتم های آنلاین یادگیری آنلاین

## ۱ مقدمه

### ۱.۱ الگوریتم های آنلاین در مقابل الگوریتم های آفلاین

شامل بازیابی<sup>۴</sup> داده ها از پایگاه داده و احتمالاً تغییر آن ها است. در یک موقعیت سرمایه گذاری، یک درخواست ممکن است شامل تعیین قیمت برای یک کالا باشد و اقدام ممکن است خرید یا فروش مقداری از کالا باشد. الگوریتم های آنلاین ممکن است با الگوریتم های آفلاین، که کل دنباله درخواست ها را از قبل دریافت می کنند، مقایسه شوند. یک الگوریتم آفلاین برای انجام یک عمل در پاسخ به هر درخواست مورد نیاز است، اما انتخاب هر اقدام می تواند براساس کل دنباله درخواست ها باشد. به عبارت دیگر، یک الگوریتم آفلاین آینده را می داند، اما یک الگوریتم آنلاین نمی داند. واضح است که در بسیاری از موارد، ناآگاهی از آینده یک ضرر بزرگ است (مثلاً سرمایه گذاری در بازار سهام را در نظر بگیرید). بنابراین، الگوریتم های آنلاین اغلب بسیار بهتر از الگوریتم های آفلاین خود عمل می کنند. به راحتی می توان گفت که منظور از یک الگوریتم آفلاین بهینه چیست. برای هر

یک الگوریتم آنلاین الگوریتمی است که دنباله ای از درخواست ها را دریافت می کند و برای پاسخ به هر درخواست یک عمل آنی انجام می دهد. هر دنباله ای از درخواست ها و اعمال مربوط به آن دارای یک هزینه مرتبط است. الگوریتم های آنلاین در بسیاری از زمینه های مختلف ظاهر می شوند. به عنوان مثال، در مورد بسته بندی آنلاین، هر درخواست یک شی است که باید بسته بندی شود و اقدام مربوط به آن قرار دادن یک شی در یک بسته است. در زمینه یک سیستم پایگاه داده، هر درخواست ممکن است یک پرس و جو<sup>۳</sup> یا یک به روز رسانی باشد و اقدام مربوطه

<sup>۱</sup>competitive ratio  
<sup>۲</sup>offline  
<sup>۳</sup>query  
<sup>۴</sup>retrieve

برای هر دنباله درخواست  $r \in R$ ,  $Opt(r)$  را به صورت  $\min_{a \in A^n} c(r, a)$  تعریف می کنیم.

یک الگوریتم آنالین  $A$  توسط  $f_A : R^+ \rightarrow A$  تعریف می شود، که در آن  $R^+$  مجموعه همه ی دنباله های متناهی و غیر تهی از درخواست ها است. در پاسخ به دنباله ای از درخواست ها  $r = r_1, r_2, r_3, \dots, r_t$  الگوریتم دنباله ای از اقدامات  $A(r) = f_A(r_1), f_A(r_1, r_2), f_A(r_1, r_2, r_t)$  را انجام می دهد و هزینه  $c(r, A(r))$  را متحمل می شود.

برای هر ثابت مثبت  $d$ ، می گویند الگوریتم آنالین  $A$   $d$ -competitive است اگر ثابت  $b$  وجود داشته باشد که به ازای همه ی دنباله های درخواست  $r$   $c(r, A(r)) \leq d \cdot Opt(r) + b$  باشد. نسبت رقابتی  $A$  به عنوان بزرگترین کران پایین مجموعه  $c$  تعریف می شود به طوری که  $A$   $c$ -competitive باشد. برخی از نویسندگان از گونه ای از این تعاریف استفاده می کنند که در آن  $b$  باید صفر باشد.

## ۲ چند مثال

### ۱.۲ مسئله اجاره چوب اسکی

برای نشان دادن مفهوم نسبت رقابتی رودلف<sup>۷</sup> مسئله اجاره چوب اسکی را مطرح کرد. فرض کنید هزینه کرایه یک جفت چوب اسکی برای یک بار رفتن به اسکی ۱ و هزینه خرید یک جفت اسکی  $s$  است. بدون اینکه بدانیم چند بار به اسکی خواهیم رفت، چگونه تصمیم می گیریم اجاره یا خرید کنیم؟ در اینجا فقط یک درخواست ممکن وجود دارد ("رفتن به اسکی") و سه اقدام ممکن ("اجاره"، "خرید" و "استفاده از چوب اسکی از قبل خریداری شده")، به ترتیب با هزینه های ۱،  $t$  و  $s$ ، که در آن اقدام سوم می تواند فقط در صورتی فراخوانی شود که اقدام دوم قبلاً اتفاق افتاده باشد. واضح است که هر الگوریتم آنالین معقولی به شکل «برای  $k$  بار اول اجاره کنید، سپس بخرید، سپس از چوب اسکی هایی که قبلاً خریداری کرده اید استفاده کنید» است. در دنباله از  $t$  درخواست هزینه متحمل شده توسط الگوریتم آنالین:

$$\begin{cases} t, & \text{if } t \leq k \\ k + s, & \text{otherwise} \end{cases}$$

و هزینه متحمل شده توسط یک الگوریتم آفلاین بهینه  $\min(s, t)$  است. مشکل انتخاب پارامتر  $k$  برای به حداقل

دنباله ای از درخواست ها، چنین الگوریتمی آن دنباله از اقدامات را انتخاب می کند که هزینه را به حداقل می رساند. تعریف یک معیار عملکرد برای الگوریتم های آنالین مشکل ساز تر است، زیرا معمولاً اینطور است که هر اقدامی هم که یک الگوریتم آنالین در پاسخ به یک سری درخواست های اولیه انجام دهد، دنباله ای از درخواست های بعدی وجود خواهد داشت که باعث می شود الگوریتم احمقانه به نظر برسد. شاید دلیل این مشکل در تعریف عملکرد «عقلانی» یا «بهینه» است که الگوریتم های آنالین در زمینه هایی مانند علوم کامپیوتر، تحقیق در عملیات و اقتصاد نسبتاً نادیده گرفته شده اند. برای مثال، تئوری زمان بندی بر مدل های آفلاین متمرکز شده است، که اغلب به طور نامناسب فرض می کنند که کل جریان مشاغلی که به یک مرکز خدماتی می رسند از قبل شناخته شده است. رایج ترین رویکرد برای ارزیابی الگوریتم های آنالین، فرض یک مدل تصادفی خاص از منبع درخواست ها است. در چنین مدلی، یک الگوریتم آنالین ممکن است بهینه در نظر گرفته شود اگر اقدامات خود را به گونه ای انتخاب کند که هزینه مورد انتظار<sup>۵</sup> را به حداقل برساند.

### ۲.۱ نرخ رقابتی

نسبت رقابتی یک الگوریتم آنالین به عنوان حداکثر نسبت بین هزینه های متحمل شده توسط الگوریتم آنالین و هزینه متحمل شده توسط یک الگوریتم آفلاین بهینه، در تمام توالی های ورودی ممکن تعریف می شود. یک الگوریتم آنالین بهینه الگوریتمی است که نسبت رقابتی آن کمتر باشد. مفهوم نسبت رقابتی به مفهوم کمترین بیشینه<sup>۶</sup> پشیمانی در تئوری بازی ها مربوط می شود و ما اغلب موقعیت را به عنوان یک بازی بین یک بازیکن آنالین که الگوریتم آنالین را انتخاب می کند و یک حریف که دنباله درخواست را برای به حداکثر رساندن نسبت بین هزینه الگوریتم آنالین و الگوریتم آفلاین بهینه انتخاب می کند نگاه می کنیم. به صورت دقیق تر یک الگوریتم آنالین عبارت است از:

• یک مجموعه از درخواست ها  $R$ ؛

• یک مجموعه از اقدام ها  $A$ ؛

• یک تابع هزینه  $c$ ؛

$$\bigcup_{n=1,2,\dots} R^n \times A^n \rightarrow \mathbb{R}^+, \text{ نشان دهنده اعداد حقیقی مثبت است,}$$

expected cost<sup>۵</sup>  
minimax<sup>۶</sup>  
Rudolph<sup>۷</sup>

### ۳.۲ یادگیری آنلاین: حداقل مربعات بازگشتی<sup>۸</sup>

یادگیری آنلاین، زیرمجموعه ای از یادگیری ماشینی است که بر این واقعیت تأکید دارد که داده های تولید شده از محیط ها می توانند در طول زمان تغییر کنند. در واقع، مدل های یادگیری ماشینی سنتی ثابت در نظر گرفته می شوند: هنگامی که یک مدل بر روی مجموعه ای از داده ها آموزش داده می شود، پارامترهای آن دیگر تغییر نمی کنند. اگرچه ممکن است یک محیط و داده های تولید شده آن در طول زمان تغییر کند، بنابراین، مدل از پیش آموزش دیده ما دیگر قابل اعتماد نیست، بهترین پاسخ برای این مشکل یادگیری آنلاین است. حداقل مربعات بازگشتی (RLS) یک تکنیک متداول است که برای مطالعه داده های بلادرنگ استفاده می شود. بنابراین، RLS را می توان به عنوان معادل بازگشتی الگوریتم حداقل مربعات استاندارد در نظر گرفت. در حداقل مربعات بازگشتی، هر تکرار الگوریتم یک نقطه داده جدید تجزیه و تحلیل می شود تا تخمین پارامترهای مدل ما را بهبود بخشد. در مقایسه با سایر تکنیک های رگرسیون، RLS معمولاً سریع تر همگرا می شود اما هزینه های محاسباتی بالاتری دارد. در صورتی که به این موضوع علاقه مند هستید می توانید به مرجع دوم مراجعه کنید.

## ۳ نتیجه گیری

الگوریتم های آنلاین در هر موقعیتی که باید تصمیم گیری شود و منابع با دانش ناقص از آینده تخصیص داده شود، به وجود می آیند. معمولاً در زمینه های برنامه ریزی حرکت ربات، تجزیه و تحلیل سرمایه گذاری، مدیریت حافظه پنهان، مهاجرت فایل، زمان بندی، رنگ آمیزی گراف، نگهداری ساختمان داده ها و پایگاه های داده، مکان یابی امکانات و افزایش ظرفیت شبکه ها ظاهر می شوند. در تمام این زمینه ها، استدلال های ریاضی جالب، کران های پایین و بالا را برای نسبت های رقابتی قابل دستیابی به دست آورده اند. اثبات کران پایین معمولاً به ساختارهای رقیب بستگی دارد و کران بالا معمولاً با تجزیه و تحلیل الگوریتم ها با کمک توابع پتانسیل به دست می آیند. در ارتباط با هر کاربرد خاص، این سوال مطرح می شود که آیا نسبت رقابتی یک الگوریتم با عملکرد آن در عمل مرتبط است یا خیر. متأسفانه، کاملاً روشن نیست که چنین باشد. استفاده از نسبت رقابتی به عنوان یک معیار عملکرد تقریباً معادل این فرض است که توالی درخواست ها توسط یک دشمن شیطانی با قدرت محاسباتی نامحدود و دانش کامل از الگوریتمی که او سعی در خنثی کردن آن دارد، تولید می شود.

رساندن نسبت رقابتی است. برای یک  $k$  معین، یک دنباله درخواست به طول  $k + 1$  نسبت بین هزینه الگوریتم آنلاین و یک الگوریتم بهینه آفلاین را به حداکثر می رساند. به عبارت دیگر، حریف باید اسکی رفتن را تا زمانی که الگوریتم آنلاین یک جفت چوب اسکی خریداری کند ادامه دهد و سپس متوقف شود. هزینه الگوریتم آنلاین در چنین دنباله ای  $k + s$  است، هزینه الگوریتم آفلاین بهینه  $\min(k + 1, s)$  و نسبت رقابتی برابر با  $\frac{k+s}{\min(k+1,s)}$  است. با فرض اینکه  $s$  یک عدد صحیح است، نسبت رقابتی با تنظیم  $k = s - 1$  به حداقل می رسد، بنابراین نسبت رقابتی  $\frac{2s-1}{s}$  به دست می آید. یکی از راه های مشاهده این موضوع این است که بازیکن آنلاین باید تا زمانی که سفرهای اسکی کافی رخ نداده باشد، اجاره کند تا به جایی برسد که با نگاهی به گذشته، بهتر باشد در ابتدا چوب اسکی را بخرد.

### ۲.۲ مسئله $k$ -سرور

مسئله در یک فضای متریک تعریف شده است. یعنی مجموعه ای از نقاط  $M$  همراه با تابع فاصله  $d$ ، تعریف شده برای هر جفت  $xy$  از نقاط در  $M$  که در شرایط زیر صدق می کند:

$$\begin{cases} d(x, y) = d(y, x) & (\text{symmetry}) \\ d(x, z) \leq d(x, y) + d(y, z) & (\text{triangle inequality}) \end{cases}$$

در این مسئله، یک الگوریتم آنلاین باید حرکت مجموعه ای از  $k$  سرور را که به صورت نقاط در یک فضای متریک نمایش داده می شوند، کنترل کند و درخواست ها را که به شکل نقاط در فضا هستند، پاسخ دهد. با رسیدن هر درخواست، الگوریتم باید تعیین کند که کدام سرور را به نقطه درخواستی منتقل کند. هزینه عمل، مسافت طی شده توسط سرور است. سرور ثابت می ماند مگر اینکه برای انتقال به نقطه درخواست انتخاب شود. بسیاری از الگوریتم های معمولی برای مسئله  $k$ -سرور در دستیابی به یک نسبت رقابتی محدود شکست خورده است. به عنوان مثال، الگوریتم حریصانه زیر را در نظر بگیرید: "به هر درخواست نزدیک ترین سرور را اختصاص دهید و آن را به آن نقطه منتقل کنید." الگوریتم حریصانه را می توان با قرار دادن درخواست ها به طور متناوب در دو نقطه که به اندازه کافی نزدیک به هم هستند شکست داد، الگوریتم حریصانه با جابجایی یک سرور به جلو و عقب برای همیشه هزینه نامحدودی را ایجاد می کند، اما، در همان دنباله درخواست، یک الگوریتم آفلاین بهینه هزینه خود را با استقرار یک سرور به طور دائم در هر یک از دو نقطه محدود نگه می دارد.

## مراجع

- [1] Karp, Richard M. On-line algorithms versus off-line algorithms: How much is it worth to know the future. *ICSI Technical Report*, 1992.
- [2] L. Rosasco, T. Poggio. Machine learning: a regularization approach, 2015. MIT-9.520 Lectures Notes. Chapter 7 - Online Learning.

این دیدگاه به سختی قابل توجیه است. یک چالش برای آینده محدود کردن حریف یا تقویت الگوریتم آنلاین به نحوی است که تمام راه ها به این فرض کلاسیک که دنباله درخواست ها توسط یک منبع تصادفی ثابت تولید می شوند، تقلیل داده نشوند (در ارتباط با برخی مسائل این فرض منطقی است)؛ سپس الگوریتم آنلاین با وظیفه استنباط الگوریتم تولید درخواست دشمن، بدون متحمل شدن هزینه اضافی روبرو می شود. رویکرد دیگر ترکیب نسبت رقابتی با تحلیل حالت متوسط<sup>۹</sup> است با یافتن الگوریتم های آنلاینی که نه تنها نسبت رقابتی کمی دارند، بلکه در برابر توالی درخواست های «معمولی» نیز عملکرد خوبی دارند.

---

average-case analysis<sup>۹</sup>