دانشکده علوم ریاضی
دانشگاه صنعتی شریف

# الگوریتم‌های خلاصه‌سازی برای مه‌داده

**محمدهادی فروغمنداعرابی**
پاییز ۱۳۹۹

# تقریب نرم $p$

جلسه نهم

**نگارنده: فاطمه کرمانی**

این جلسه یک جلسه‌ی کوتاه ضبطی است. مبحثی که در این جلسه پوشش داده می‌شود تقریب نرم $p$ یک بردار است.

## ۱ تعریف مسئله

بردار مورد نظر $x$ را به صورت مستقیم به ما نمی‌دهند. ساختمان داده‌ای که برای آن استفاده می‌شود مشابه مواردی که در گذشته داشتیم یک آرایه‌ی $x$ با عملیات افزایش و کاهش یک درایه است.

**مسئله** $F_p$: را تعریف می‌کنیم

$$\|x\|_p^p = \sum_{i=1}^{p} |x_i|^p$$

تا حالا با مسئله‌ی پیدا کردن $F_0$ (در حالت فقط افزایش $x$) سر و کار داشته‌ایم و با آن آشنا هستیم. هدف ما پیدا کردن یک تخمین‌گر $a$ است که با احتمال $1-\delta$، داشته باشیم $|a-F_p| \leq \epsilon F_p$ به عبارت دیگر هدف پیدا کردن یک برآوردگر $(1+\epsilon)$ـ تقریب برای $F_p$ است.

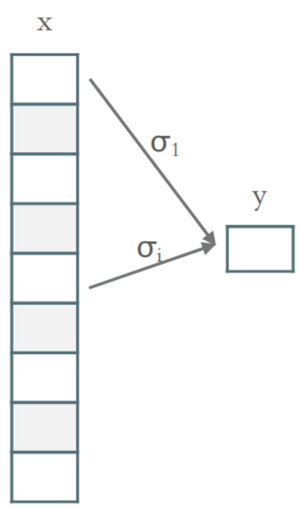
می‌توانیم یک زیرهدف برای رسیدن به هدف اصلی تعریف کنیم. این زیر هدف پیدا کردن یک الگوریتم برای پیدا کردن تخمین‌گر $a$ است که با احتمال ۲/۳ برای آن داشته باشیم $|a-F_p| \leq \epsilon F_p$. قبلاً دیدیم که به این روش چطور می‌شود $log(1/\delta)$ بار الگوریتم زیر هدف را تکرار کرد و به کران مورد نظر برای مسئله‌ی اصلی رسید.

# ۲ روش کلی

روشی که از آن استفاده می‌کنیم خلاصه‌سازی خطی است. به عبارت دقیق‌تر برای نگه‌داری بردار $x$، ماتریس $\Pi$ ای در نظر می‌گیریم و در هر گام فقط ماتریس $\Pi x$ را ذخیره می‌کنیم. $y = \Pi x \in \mathbb{R}^m$. (دقت داشته باشید که $m < n$). یک مزیت این روش به روزرسانی ساده‌ی آن است. با تغییر یک درایه از $x$، به میزان $\Delta$، کافیست ستون متناظر با آن درایه را از ماتریس $\Pi$ در $\Delta$ ضرب کنیم و حاصل را با مقدار گذشته جمع کنیم.

## ۱.۲ ایده‌ی کلی روش AMS

الگوریتم AMS این مسئله را برای حالت $p = ۲$ حل می‌کند. ایده‌ی کلی این الگوریتم این است که اطلاعات چند درایه را در یک خانه‌ی حافظه ذخیره کنیم. در حالت خیلی ساده‌تر، فرض کنید یک عدد $y$ را به نمایندگی تمام اطلاعات $x$ ذخیره کنیم. به این شکل که هر کدام از درایه‌ها را در یک عدد ۱+ یا ۱− ضرب می‌کنیم. و با $y$ جمع می‌کنیم. به عبارت دیگر یک متغیر تصادفی $\sigma_i$ را به طور تصادفی و یکنواخت برابر ۱− و ۱+ قرار می‌دهیم. مطابق شکل ؟؟




شکل ۱: نحوه‌ی تعریف متغیر $y$

امید ریاضی $y$ برابر صفر است چون هر کدام از درایه‌های $x$ با احتمال $\frac{۱}{۲}$ مثبت و با احتمال $\frac{۱}{۲}$ با علامت منفی ظاهر شده اند. اما امید $y^۲$ ناصفر است زیرا هر گاه که $\sigma_i^۲$ ظاهر شود، مقدار آن مستقل از $\sigma$، مثبت می‌شود. در نتیجه جملات مثبت بیش‌تری خواهیم داشت.

$$\mathbb{E}(Y^۲) = \mathbb{E}[\sum_{j,j'} \sigma_j \sigma_{j'} x_j x_{j'}] = \mathbb{E}[\sum_j \sigma_j^۲ x_j^۲ + \sum_{j \neq j'} \sigma_j \sigma_{j'} x_j x_{j'}] = \sum_j \mathbb{E}[x_j^۲] + \sum_{j \neq j'} \mathbb{E}[\sigma_j \sigma_{j'} x_j x_{j'}] = \|x\|_۲^۲$$

در محاسبات بالا تساوی آخر به این دلیل است که $\sum_{j \neq j'} \mathbb{E}[\sigma_j \sigma_{j'} x_j x_{j'}]$ صفر است زیرا $\sigma_j$ و $\sigma_{j'}$ از هم مستقل هستند پس امید ضربشان برابر ضرب امیدشان است و امید تک تکشان صفر است. پس امید $Y^۲$ همان تابع هدف ماست.

برای این‌که این برآوردگر را به عنوان خروجی برگردانیم درباره‌ی واریانس آن نیز باید کنترل داشته باشیم. واریانس در این حالت ممکن است زیاد باشد. برای حل این مشکل، الگوریتم را چند بار تکرار می‌کنیم. به طور دقیق‌تر، ماتریسی $m \times m$ از $\sigma$ ها تولید می‌کنیم، به طوری که انتخاب $\sigma$ ها استقلال ۴ طرفه داشته باشند. قبلاً دیدم که می‌توان در صورت وجود استقلال ۴ طرفه در درایه‌ها، این تعداد عدد را با $O(\log(mn))$ بیت حافظه ذخیره کنیم.

- شروع:
  - $(\sigma \in \{-۱, ۱\}^{m \times n})$
  - $(\sigma_{i,j}/\sqrt{m} = \Pi_{i,j})$
- به‌روز رسانی:
  - همیشه $y = \Pi x$ را نگه دار. (برای $x_i = +۱$، ستون $i$ام $\Pi$ را به $y$ اضافه کن.)

– داریم: $y_i = \sum_{i=1}^{n} \sigma_{i,j} x_j / \sqrt{m}$

- تخمین‌گر:

– $\|\Pi x\|_2^2 = \|y\|_2^2$

درایه‌ی $ij$ ماتریس را $\sigma/\sqrt{m}$ قرار می‌دهیم که مخرج کسر در واقع حکم ضریب نرمال کننده را دارد.

# ۳ تحلیل AMS

## ۱.۳ تحلیل امید ریاضی تخمین‌گر

امید هر کدام از درایه‌های بردار $y^2$ را محاسبه می‌کنیم. در واقع هر کدام از درایه‌ها، تکراری از ایده‌ی اولیه است.

$$\begin{aligned}
\mathbb{E} y_r^2 &= \frac{1}{m}\mathbb{E}\left(\sum_{j=1}^{n} \sigma_{r,j} x_j\right)^2 \\
&= \frac{1}{m}\left[\|x\|_2^2 + \mathbb{E}\sum_{j\neq j'} \sigma_{r,j}\sigma_{r,j'} x_j x_{j'}\right] \\
&= \frac{1}{m}\left[\|x\|_2^2 + \sum_{j\neq j'} (\sum \sigma_{r,j}\sigma_{r,j'}) x_j x_{j'}\right] \\
&= \frac{1}{m}\left[\|x\|_2^2 + \sum_{j\neq j'} (\sum \sigma_{r,j})(\sum \sigma_{r,j'}) x_j x_{j'}\right] \\
&= \frac{1}{m}\|x\|_2^2
\end{aligned}$$

از طرفی داریم $\|y\|_2^2 = \sum_{r=1}^{n} y_r^2$، پس در نهایت به دست می‌آید

$$\|x\|_2^2 = \mathbb{E}[\|y\|_2^2]$$

## ۲.۳ تحلیل واریانس تخمین‌گر

$$\begin{aligned}
\mathbb{E}(\|y\|_2^2 - \mathbb{E}\|y\|_2^2)^2 &= \frac{1}{m}\mathbb{E}(\sum_{r=1}^{m}\sum_{j\neq j'} \sigma_{r,j}\sigma_{r,j'} x_j x_{j'})^2 \\
&= \frac{1}{m}\sum_{r_1,r_2}^{m} \sum_{\substack{j_1\neq j_2 \\ j_3\neq j_4}} (\mathbb{E}\sigma_{r_1,j_1}\sigma_{r_1,j_2}\sigma_{r_2,j_3}\sigma_{r_2,j_4}) x_{r_1,j_1} x_{r_1,j_2} x_{r_2,j_3} x_{r_2,j_4} \\
&= \frac{2}{m}\sum_{j_1\neq j_2} x_{j_1}^2 x_{j_2}^2 \neq \frac{2}{m}\|x\|_2^4
\end{aligned}$$

دقت کنید که برای این‌که بتوانیم ضرب امید را به صورت امید ضرب بنویسیم باید فرض کنیم که مستقل هستند. پس نباید با هم برابر و تکراری باشند. پس حالت‌هایی باقی می‌ماند که توان هیچ کدام فرد نیست. پس باید $j_3 = j_1$ و $j_4 = j_2$ یا $j_4 = j_1$ و $j_3 = j_2$ و تساوی آخر از همین نکته نتیجه می‌شود. نامساوی آخر نیز با اضافه کردن عناصر دیگر به دست می‌آید و کران بالای مناسبی برای واریانس به دست می‌دهد.

## ۳.۳ تحلیل نهایی

با استفاده از نابرابری چبیشف و جاگذاری مقادیر مطابق زیر به دست می‌آوریم

$$\begin{aligned}
\mathbb{P}\left[\left|\|y\|_2^2 - \|x\|_2^2\right| > \epsilon\|x\|_2^2\right] &< \mathrm{Var}(\|y\|_2^2)/(\epsilon\|x_2^2\|)^2 \\
&< \frac{2}{m}\|x\|_2^4 \frac{1}{\epsilon^2\|x\|_2^4} \\
&= \frac{2}{\epsilon^2 m}
\end{aligned}$$

(نابرابری چبیشف (برای یادآوری) $\forall\lambda > 0, \mathbb{P}(|X - \mathbb{E}(X)| > \lambda) < \frac{\mathbb{E}(X-\mathbb{E}X)^2}{\lambda^2}$ )

دقت کنید که در صورتی که $m$ به اندازه‌ی کافی بزرگ باشد($m = 6/\epsilon^2$) می‌توانیم فرض کنیم از کران فوق حداکثر ۱/۳ است.

حافظه‌ی مورد نیاز برای این الگوریتم $O(1/\epsilon^2 \log(1/\delta))$ عدد است برای حالتی که $1/\delta$ بار موازی آن را اجرا کنیم.

می‌توانیم الگوریتم را ارتقا دهیم به این روش که در هر ستون تعداد کمی درایه‌ی ناصفر داشته باشیم. درایه‌های ناصفر با تابع درهم‌ساز $h : [n] \to [m]$ انتخاب کنیم. که تابع درهم استقلال ۲ــطرفه داشته باشد. و مانند قبل مقدار درایه، به صورت تصادفی با استقلال ۴ــطرفه به صورت $\sigma \in \{-1, +1\}^n$ انتخاب می‌شود. به عبارت دیگر در هر ستون $j$، $\Pi_{h(j),j} = \sigma_j$ و بقیه‌ی ستون ۰ باشد. در این حالت مشابه قبل می‌توانیم امید ریاضی را محاسبه کنیم

$$\mathbb{E}\|\Pi z\|_2^2 = \|z\|_2^2$$

و همچنین مشابه حالت قبل برای واریانس داریم

$$\mathrm{Var}(\|\Pi z\|_2^2) = O(1/m)\|z\|_2^4$$

و تمام جزئیات دیگر به صورت مشابه تعریف می‌شود.