دانشکده علوم ریاضی
دانشگاه صنعتی شریف

الگوریتم‌های خلاصه‌سازی برای مه‌داده

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

تبدیل فوریه تنک

جلسه ۱۹

نگارنده: مجتبی استواری

۱ مقدمه

در جلسه قبل با استفاده از گستر گراف‌ها توانستیم ماتریس تنک RTP_1 را بسازیم و سپس با کمک آن بازیابی تنک را انجام دادیم. در این جلسه موضوع جدید با نام تبدیل فوریه تنک را مطرح خواهیم کرد.

هدف از تبدیل فوریه تنک پیدا کردن راهی سریع برای یافتن تقریبی ضرایب فوریه یک سیگنال تنک است. ابتدا تعریف فوریه گسسته را تعریف می‌کنیم و سپس تبدیل فوریه تنک را بیان می‌نماییم.

۲ تبدیل فوریه گسسته

فرض کنید عدد n توانی از ۲ باشد یعنی مقدار l ای وجود داشته باشد که $n = 2^l$ است. در اینجا ورودی دنباله n تایی از سیگنال گسسته $a = (a_0, a_1, ..., a_{n-1}) \in \mathbb{C}^n$ است و هدف محاسبه ضرایب فوریه $\hat{a} = (\hat{a}_0, \hat{a}_1, ..., \hat{a}_{n-1})$ است که برای هر $u = 0, 1, ..., n-1$ به صورت زیر تعریف می‌شود.

$$\hat{a}_u = \sum_{j=0}^{n-1} a_j e^{-\frac{2\Pi i}{n} uj}$$

جایی که $i = \sqrt{-1}$ است. برای سادگی $w = w_n = e^{\frac{2\Pi i}{n}}$ را تعریف می‌کنیم. پس خواهیم داشت:

$$\hat{a}_u = \sum_{j=0}^{n-1} a_j w^{-uj}$$

برای سادگی می‌توان تبدیل فوریه و معکوس تبدیل فوریه را به شکل ماتریسی نوشت به صورتی که داشته باشیم $\hat{a} = Fa$ و $a = F^{-1}\hat{a}$ برای ماتریس‌های $F, F^{-1} \in \mathbb{C}^{n\times n}$ که برای $u, j = 0, 1, ..., n-1$ به صورت زیر تعریف می‌شوند.

$$F_{uj} = \frac{1}{n} w^{-uj}$$

$$F^{-1}_{uj} = w^{uj}$$

ادعا ۱. *هر دو ماتریس F و F^{-1} تا مقیاسی معین متعامد هستند. یعنی برای هر $x \in \mathbb{C}^n$ خواهیم داشت:*

$$\| Fx \|_2^2 = \frac{1}{n} \| x \|_2^2$$

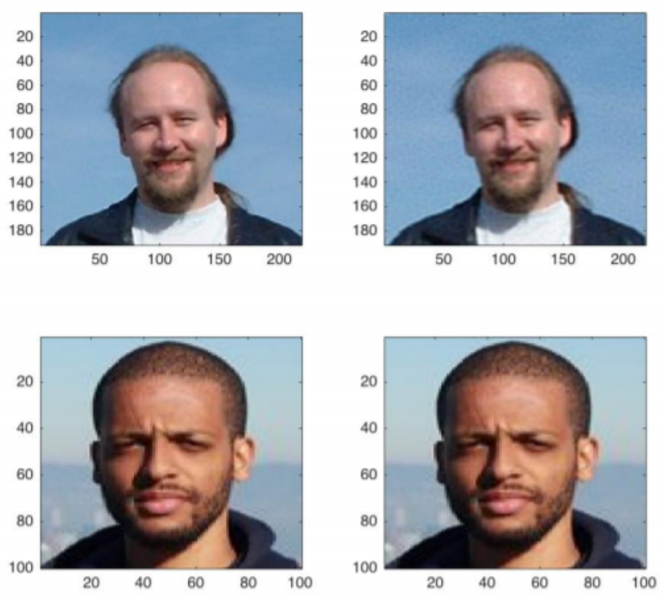
$$\| F^{-1}x \|_2^2 = n \| x \|_2^2$$

قضیه ۲. *فرض کنید a و $\hat{a}$ مثل قبل تعریف شده باشند و a' و $\hat{a'}$ سیگنالی جدید و تبدیل فوریه آن باشد. برای هر $b = 0, 1, ..., n-1$ خواهیم داشت:*

$$\forall j = 0, 1, ..., n-1, a'_j = a_j w^{bj} \Leftrightarrow \forall u = 0, 1, ..., n-1, \hat{a'}_u = \hat{a}_{u-b}$$

۳ کاربرد

یکی از مهم‌ترین کاربردهای تبدیل فوریه تکنیک فیلتر کردن در پردازش سیگنال و تصویر است. برای مثال یک تصویر حجیم را ابتدا روی آن تبدیل فوریه انجام می‌دهیم و ضرایب فوریه آن را به دست می‌آوریم. سپس ۹۰ درصد کوچکترین ضرایب فوریه آن را حذف می‌کنیم و روی ضرایب جدید تبدیل معکوس فوریه را اعمال می‌کنیم. تصویر جدید تقریبا مثل تصویر اول خواهد بود.



شکل ۱: تصاویر سمت راست تصاویر اصلی هستند و تصاویر سمت چپ فیلتر شده آن‌ها هستند. می‌توان دید که تصاویر بعد از فیلتر مقدار کمی تار شده‌اند.

سوال: چقدر سریع می‌توان تبدیل فوریه را انجام داد؟

هدف اصلی این درس این موضوع است. در نگاه اول می‌توان گفت که ضرب ماتریس در بردار در $O(n^2)$ برای تبدیل فوریه قابل انجام است. الگوریتمی سریع برای تبدیل فوریه وجود دارد که که در زمان $O(n \log n)$ می‌تواند این کار را انجام دهد. در این درس تمرکز ما بر روی حالتی است که ورودی k تنک است برای $\log n < k \ll n$ که به آن تبدیل فوریه تنک می‌گوییم.

۴ تبدیل فوریه تنک

فرض کنید $a \in \mathbb{C}^n$ سیگنال ورودی باشد. و $\hat{a}$ ضرایب فوریه آن باشد. برای هر مقدار $k = 1, 2, ..., n$ $\hat{a}^{(k)}$ را برداری تعریف می‌کنیم که شامل k بزرگترین مقادیر $\hat{a}$ است. هدف تبدیل فوریه تنک پیدا کردن $\hat{a'}$ است که برای آن داشته باشیم:

$$\| \hat{a} - \hat{a'} \|_2 \leq C \| \hat{a} - \hat{a}^{(k)} \|_2$$

جایی که $C > 0$ یک مقدار ثابت است.

هدف نهایی پیدا کردن الگوریتم تصادفی با زمان $O(k \log n)$ است. توجه کنید که k بسیار کوچکتر از n است و الگوریتم ممکن است که حتی تمام ورودی را نخواند!

۱.۴ شدنی بودن؟

قبل از آن که الگوریتم را مطرح کنیم، شدنی بودن چنین الگوریتمی را با طراحی الگوریتمی با گارانتی L_1 به جای L_2 بررسی می‌کنیم. در حقیقت این موضوع با استفاده از مباحث جلسات قبل به دست می‌آید.

از قبل می‌دانیم که زیرماتریس‌های ماتریس تبدیل معکوس فوریه RIP F^{-1} است. با بیانی دقیق‌تر فرض کنید $S \subseteq [n]$ و $|S| = O(\epsilon^{-2} k \log^4 n)$ باشد. پس $A = F_S^{-1}$ دارای خاصیت $(\epsilon, O(k)) - RIP$ است. همین‌طور اگر A دارای خاصیت RIP باشد، می‌توان x را از روی Ax بازسازی کرد.

در اینجا می‌توانیم x را به عنوان ضرایب فوریه $\hat{a}$ در نظر بگیریم. پس $Ax = F_S^{-1} \hat{a}$ سیگنال ورودی متناظر با S است. پس نتیجه می‌دهد که کافیست به تعداد $|S| = O(\epsilon^2 k \log^4 n)$ نمونه از سیگنال ورودی را داشته باشیم تا با استفاده از آن $\hat{a}$ را بازیابی کنیم. اگرچه زمان اجرای این الگوریتم چندجمله‌ای بر حسب n است.

۲.۴ حالت خاص $k = 1$

برای شروع در این قسمت الگوریتم تبدیل فوریه سریع را در حالتی که تنها یک ضریب فوریه غیر صفر داریم ارائه می‌کنیم. در این صورت دو حالت می‌توانیم داشته باشیم:

- **با نویز:** وجود دارد $u = 0, 1, ..., n-1$ به طوری که برای هر $u' \neq u, \hat{a}_{u'} = 0$. در این حالت هدف ما پیدا کردن u و تقریب زدن $\hat{a}_u$ است.
- **بدون نویز:** هدف پیدا کردن $\hat{a'}$ است به طوری که $\| \hat{a} - \hat{a'} \|_2 \leq C \| \hat{a} - \hat{a}^{(1)} \|_2$ برای مقدار $C > 0$ ثابت.

۳.۴ حالت بدون نویز برای $k = 1$

ایده الگوریتم بر اساس ادعا زیر است.

ادعا ۳. *با استفاده از تبدیل وارون فوریه و این نکته که تنها یک ضریب غیر صفر فوریه وجود دارد، برای هر مقدار $j = 0, 1, ..., n-1$ $a_j = \hat{a} w^{uj}$ را خواهیم داشت.*

حال با استفاده از نکته قبل و نمونه‌گیری a_0 و a_1 می‌توانیم u و $\hat{a}_u$ را به صورت زیر بازیابی کنیم:

$$\hat{a}_u = a_0 \text{ and } u = n \times A(\frac{a_1}{a_0}) .$$

که منظور از $A(\frac{a_1}{a_0})$ زاویه بین $\frac{a_1}{a_0}$ و محور اعداد حقیقی است. صحت الگوریتم از این نتیجه می‌شود که $a_0 = \hat{a}_u$ و $w^u = \frac{a_1}{a_0}$. این الگوریتم نمونه‌گیری دو نقطه‌ای نامیده می‌شود و نیاز به زمانی ثابت دارد. هرچند این الگوریتم دو مشکل دارد:

- این الگوریتم را نمی‌توان برای مقادیر k غیر از یک تعمیم داد پس در حالت کلی الگوریتم $O(k)$ نداریم.
- این الگوریتم نمی‌تواند حالت بدون نویز را شامل شود.