۱۳۰۷
دانشگاه صنعتی خواجه نصیرالدین طوسی

تمرین سوم (پروژه پایانی) درس تحلیل ها و سیستم های داده های حجیم – استادگرامی جناب آقای دکتر خواسته

حل مسئله پیج رنک در نگاشت کاهش هدوپ و موارد مرتبط با الگوهای تکرار شونده

دستیاران: آذرکسب- زکی زاده

ترم اول سال تحصیلی ۰۲-۱۴۰۱

مقدمه:

در این تمرین ما می خواهیم یک مسئله واقعی بیگ دیتایی را حل کنیم. در مجموعه اسلاید چهارم، مسئله پیج رنک در صفحات ۷۱ به بعد مطرح شده است. در ادامه برای راهنمایی دوستان به **بیان مسئله و ارائه راه حل** به زبان ساده پرداخته می شود. می خواهیم ببینیم که گوگل چگونه سایت اصلی مورد جستجو را پیدا می کند؟ ایده پیج رنک است که بر اساس آن صفحات دسته بندی می شوند. هر صفحه خودش اعتباری دارد اعتبار اولیه همه برابر است بر اساس میزان لینک خروجی که دارد قسمتی از اعتبار خودش را به اون لینک خروجی اختصاص می دهد. در عین حال هرچه سایت معتبر تر باشد لینکش از امتیاز بالایی برخوردار خواهد بود.

بهترین نوع پیاده سازی پیج رنک با مپ ردیوس است. به بیان دیگر رنکینگ، نتایج رو دارد به تولید کننده صفحات وابسته می کند. یک مقدار دلخواه اولیه برای صفحات در نظر می گیرد. هر پیجی امتیاز خودش رو بر اساس فرمول مربوطه به لینکهای خروجی خودش توزیع می کند و هر پیج امتیازهای دریافتی را تجمیع می کند. مقدار جدید را محاسبه می کند و دوباره این کار را تکرار می کند. نکته این است که امتیاز پیج رنک ایکس x به امتیاز پیج رنک باقی صفحات تی t بستگی دارد و چه بسا بالعکس. پس این فرایند یک فرایند تکراری است. اینکار ادامه پیدا می کند تا اینکه همگرا شود و از یک حدی دیگر تغییر پیدا نکند. هر تغییری لوکال است یعنی هر مرحله به مرحله قبلی آن بستگی دارد. در هر تکرار نیز می توانیم امتیاز نودها رو مستقل از بقیه نودها حساب کنیم. خروجی یک مرحله باید به فرمت ورودی مرحله بعدی باشد. یک ماتریس خلوت می گیریم برای نمایش یوار ال ها و لینکهایی که بینشان هست. ماتریس خلوت است از آن جهات که تعداد یو آر الها خیلی زیاد است اما از هر یو آر ال به یو آر ال دیگر لینک نداریم. مرحله اول پیش پردازش است در این مرحله لینکها از اچ تی ام ال سایتهای مختلف توسط یک خزنده وب استخراج می شود. فایلهای مربوط برای انجام این تمرین را با مراجعه به آدرس زیر دانلود نمایید:

https://github.com/google-research-datasets/wiki-links

در ادامه یکسری مپ تسک داریم که یو آر ال و محتویات پیج را می گیرد و آنها را به یو آر ال به عنوان کلید نگاشت کرده و پیج رنک اولیه و لیست خروجی ها را استخراج می کند. مقدار اولیه یک مقدار سیگ است که می تواند به روش الکسا به دست آید. در این مرحله ردیوس تسک نداریم.

مرحله دوم توزیع پیج رنک هاست در این مرحله ردیوس تسک داریم که که یو آر ال و لیست یو آر ال ها را می گیرد و ولیوها رو با هم جمع می کند. چون انجام کار به صورت اتریتیو است در نتیجه خروجی باید مثل ورودی باشد.

المانهای مپ ردیوس:

- توزیع و جمع محاسبه امتیازها،
- یکپارچه سازی پایگاه داده ها،
- ایجاد یکسری فایلهای جانبی.

در مپ، کلید میانی اون یو آر ال هایی هستند که بهشون لینک خروجی رفته که در نهایت بتوان امتیازها را تجمیع کرد. خروجی میانی می شود لینک خروجی، و امتیازی که از پیش داشتیم تقسیم بر تعداد. شافل و سورت بر اساس ای دی های مختلف که همان یو آر ال ها هستند آنها را مرتب می کند و می فرستد به ردیوسر یکسان. در ادامه ردیوسر می آید و یک ایدی می گیرد با تعداد زیادی امتیاز، و امتیازها رو با هم جمع می کند. و خروجی را در اختیار ما قرار می دهد. یک کار دیگر باید انجام بدهیم! ما می خواهیم که ساختار لینکها از دست نرود! در ردیوسر، ای دی را می گیریم به علاوه امتیازها، یکی از این امتیازها به صورت لیست است. در کلید ولیوهای میانی، کلید همیشه یو آر ال است. ولیوها، تمامشان امتیاز هستند غیر از یکی، که اون یکی لیستی از لینکهاست پس در ردیوسر آن یک المان را کنار می گزاریم و بقیه را محاسبه می کنیم و شکل نهایی را می سازیم.

در ادامه جهت تکمیل تر شدن مطالب، مباحثی را پیرامون **پارتیشنر** و **کامباینر** و مزیتهای استفاده از خدمت دوستان ارائه می کنیم که در حل این تمرین سودمند می باشد.

مباحث تکمیلی در رابطه با کامباینر

کامباینر در خروجی مپر قرار می گیرد برای تجمیع نتایج میانی در عمل یک مینی ردیوسر است که برای کارهای جابجایی داده ها استفاده می شود. هرچقدر که جابجایی داده ها رو کمتر کنیم کارایی برنامه را بالاتر می بریم و نتیجه بهتری را به دست می آوریم. کامباینرها معمولا در حافظه و بلافاصله بعد از مپ اجرا می شوند تا نتیجه را میانی را تجمع کنند، و حجم داده ای که قرار است از خروجی مپر به ورودی ردیوسر جابجا بشود را کاهش می دهند. کامباینر در همان ورکر، دیتا نود، اسلیو نودی که مپر روی آن اجرا می شود قرار دارد در حالیکه ردیوسر در جایی دیگر قرار دارد داده باید از خروجی مپر به ورودی ردیوسر جابجا بشود. طبیعتا در مپر جابجایی داده نداریم. پارتیشن داده های رو جابجا می کند و مشخص می کند که هر کی ولیو می بایست به کدام ردیوسر برود. این کاری است یه مقدار بالاتر از کاری که فریم ورک انجام می دهد. فریم ورک فقط تضمین می کرد که داده های باکلید میانی یکسان بروند به ردیوسر یکسان، حالا اگر بخواهیم تقسیم بندی بیشتری انجام بدهیم می توانیم با کمک تابع پارتیشن اینکار را انجام بدهیم ردیوسر هم که کار تجمیع را انجام می دهد. یکسری مانع بین فاز مپ و فاز ردیوسر نیاز است. یعنی قبل از اینکه تمام مپ تسکها تمام بشود. نباید هیچکدام از ردیوس تسکها شروع بشود. تمام زوجهای کلید مقدار با کی ولیو یکسان باید به ردیوس یکسان بروند چون ممکن است خروجی آخرین مپ تسک نیاز باشد به اولین مپ تسک برسد. اما کپی کردن داده ها رو از خروجی مپر به ورودی ردیوسر می توانیم زودتر شروع کنیم.

نکته دیگر اینکه معمولا تعداد ردیوسها از تعداد مپرها کمتر است لذا پردازش چند کلید میانی به یک ردیوسر می رسند لذا ترتیب نیز مهم می شود. به طور خلاصه کامباینر، یک تجمیع کننده محلی است که کلیدهایی را که در خروجی مپر یکسان هستند را تجمیعشان را انجام می دهد. و برای کارهای انجمنی نظیر ماکس، جمع گرفتن و تعداد گرفتن مناسب است.

مباحث تکمیلی در رابطه با پارتیشن

پارتیشن می تواند کلیدهای میانی یکسان را به چند ردیوسر بفرستد. **کامباینر** در خروجی مپر قرار می گیرد برای تجمیع نتایج میانی. و در عمل یک مینی ردیوسر است که برای کارهای جابجایی داده ها استفاده می شود. هرچقدر که جابجایی داده ها رو کمتر کنیم کارایی برنامه را بالاتر می بریم و نتیجه بهتری را به دست می]وریم. پارتیشنها **معمولا** در حافظه اجرا و بلافاصله بعد از مپ می شوند تا نتیجه را میانی را تجمیع کنند. و حجم داده ای که قرار است از خروجی مپر به ورودی ردیوسر جابجا بشود را کاهش می دهند. کامباینر در همان ورکر، دیتا نو ، اسلیو نودی که مپر روی آن اجرا می شود قرار دارد در حالیکه ردیوسر در جایی دیگر قرار دارد. داده باید از خروجی مپر به ورودی ردیوسر جابجا بشود. طبیعتا در مپر جابجایی داده نداریم. پارتیشن داده ها رو جابجا می کند و مشخص می کند که هر کی ولیو می بایست به کدام ردیوسر برود. این کاری است یه مقدار بالاتر از کاری که فریم ورک انجام می دهد. فریم ورک فقط تضمین می کرد که داده های باکلید میانی یکسان بروند به ردیوسر یکسان، حالا اگر بخواهیم تقسیم بندی بیشتری انجام بدهیم می توانیم با کمک تابع پارتیشن اینکار را انجام بدهیم. ردیوسر هم که کار تجمیع را انجام می دهد. در اینجا یکسری مانع بین فاز مپ و فاز ردیوسر نیاز است. یعنی قبل از اینکه تمام مپ تسکها تمام بشود. نباید هیچکدام از ردیوس تسکها شروع بشود. ما گفتیم که تمام زوجهای کلید مقدار با کی ولیو یکسان به ردیوس یکسان می رود. چون ممکن است خروجی اخرین مپ تسک نیاز باشد به اولین ردیوس تسک برسد. با این رویکرد می توانیم کپی کردن داده ها رو از خروجی مپر به ورودی ردیوسر، زودتر شروع کنیم. نکته دیگر اینکه معمولا تعداد ردیوسها از تعداد مپرها کمتر است لذا پردازش چند کلید میانی به یک ردیوسر می رسند لذا ترتیب نیز مهم می شود. همانطور که در مطالب ارائه شده در کلاس آموخته شد در سکندری سورت دو روش استفاده شده است یکی اینکه ولیو رو در کلید می گذاریم دوم اینکه از پارتیشنر استفاده می کنیم پاریشنر می گوید که کدام کلید به کدام ردیوسر برود. در سکندری سورت بافر کار خوبی نبود و ما آنرا انجام نمی دادیم. در عوض ولیو تو کی کانورژن انجام می دادیم و کار نیاز نباشد کار دیگری انجام شود. خودش تمام جمع ها را برای ما انجام می داد. در اینجا پارتیشنیگ لازم است چراکه وقتی ولیو تو کی کانورژن انجام می دهیم دیگر کلیدها فقط به جزء اول حساس نیستند کلید دو جزئی می شود و کلید دو جزئی که مولفه های

دوم ان یکسان نیستند را لزوما فریم ورک انها را به یک ردیوسر نمی فرستد. برای اینکه مطمئن باشیم که این دو به یک ردیوسر می روند باید از ابزار پارتیشنر استفاده کنیم. که کلیدها رو تقسیم بندی کند و به ردیوسر مدنظر ارسال کند.

سوالات:

با استفاده از دیتاست معرفی شده در بالا، که شامل ۱۰ توکن از هر داکیومنت است موارد زیر را بدست آورید:

۱ -تعداد تکرار هر کلمه را بدست آورده و ۱۰ کلمه با بیشترین تکرار را گزارش کنید.

۲ -برای هر کلمه مقدار inverse document frequency یا همان idf را با استفاده از Partitioner محاسبه کنید.

۳ -احتمال وقوع همزمان دو به دوی توکن ها در یک سند را با استفاده از In-Mapper Combiner محاسبه کنید.

۴- با توجه به اینکه معمولا کلمات داخل یک سند با هم مرتبط هستند، بین ۱۰ کلمه هر سند یک رابطه دو سویه در نظر بگیرید یعنی فرض کنید هر کدام از توکن های داخل یک سند به سایر توکن‌های آن سند لینک دارد و با استفاده از الگوریتم page rank رتبه هر کلمه را محاسبه کنید و ۱۰ کلمه با بالاترین رتبه را گزارش کنید.

نکات:

- **ضمن توجه و تاکید بر نکات عنوان شده در تمرینهای قبلی به خصوص موضوع کپی!، به اطلاع دوستان می رسانیم که ارائه این تکلیف نیز به صورت گروههای ۲ نفره امکان پذیر می باشد.**
- **ارائه گزارش کامل و مصور به همراه توضیح مراحل انجام کار، بسیار حائز اهمیت است و در مقدار نمره دهی بسیار تاثیر گذار است.**
- **تثبیت نمره تکلیف، از طریق ارائه توضیحات به صورت حضوری توسط هر دو عضو گروه، انجام می شود که زمان آن متعاقبا اعلام خواهد شد.**
- **در پایان خاطر نشان می شود که انجام این تمرین نیازمند صرف زمانی بیشتری از دو تمرین قبل، در مراحل مختلف خود می باشد. از اینرو مدت زمان تحویل این تمرین طوری در نظر گرفته شده است که دوستان بتوانند زمان کار خود را مدیریت بفرمایند. لذا خواهشمندیم که زمان انجام این تمرین را به روزهای پایانی، که به تبع تلاقی با امتحانات پایان ترم دارد موکول نفرمایند. تا انشاالله مشکلی از لحاظ تحویل تمرین، برایشان پیش نیاید.**

با تشکر و آرزوی سلامتی برای تمام دوستان و عزیزان

موفق و سربلند باشید