به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

پردازش داده‌های حجیم

پروژه پایانی

دانشجو: رضا ساجدی

استاد: دکتر چهرقانی

بهار ۱۴۰۱

فهرست مطالب

۱- خلاصه پژوهش

در این پژوهش، الگوریتم‌هایی برای تخمین تعداد مثلث‌ها و حلقه‌های به طول ۴ در جریان‌های گرافی ارائه می‌شود. اینها جز مسائل پایه‌ای در زمینه تجزیه و تحلیل گراف‌ها هستند که تاکنون پژوهش‌های زیادی به آنها پرداخته‌اند. در این پژوهش، از مدل‌های یادگیری ماشین که از آنها با نام اوراکل[۱] یاد شده است، در دو مدل لیست مجاورت[۲] و ترتیب دلخواه[۳] ورود یال‌ها بهره‌گیری می‌شود.

اوراکل‌ها اطلاعاتی راجع‌به اینکه یک یال در تعداد زیادی از مثلث‌ها مشارکت دارد یا نه پیش‌بینی می‌کنند. به این دسته از یال‌ها، یال‌های سنگین[۴] و به مدل پیش‌بینی‌گر مربوط به آن، «اوراکل یال سنگین»[۵] گفته می‌شود. این مورد، مشابه مسائل طبقه‌بندی دودویی در یادگیری ماشین است. گونه مشابه آن در رگرسیون[۶] نیز با نام «اوراکل مقداری»[۷] یاد شده است. نتایج ارزیابی تئوری و تجربی این پژوهش نشان می‌دهد که استفاده از این اوراکل‌ها باعث کاهش پیچیدگی حافظه‌ای الگوریتم‌های سنتی می‌شود.

شناسایی یال‌های سنگین و نمونه‌برداری از آنها در افزایش دقت تخمین تعداد مثلث‌ها از اهمیت بالایی برخوردار است. در پژوهش مربوطه، ابتدا مفهوم اوراکل کامل[۸] بیان می‌شود که با مشخص کردن یک آستانه، اگر یک یال حداقل به تعداد آستانه در مثلث‌ها دخیل باشد، به‌عنوان یال سنگین تشخیص داده می‌شود. اما این فرض در عمل واقعی نیست و در ادامه مفهوم اوراکل نویزدار مطرح می‌شود که برای یال‌های خیلی سبک یا خیلی سنگین با احتمال بالایی پیش‌بینی درست انجام می‌شود و اگر یک یال به آستانه نزدیک باشد، ممکن است اوراکل در تصمیم‌گیری دچار اشتباه شود.

---

[1] Oracle
[2] Adjacency list
[3] Arbitrary order
[4] Heavy edges
[5] Heavy edge oracle
[6] Regression
[7] Value oracle
[8] Perfect oracle

۱-۱- نقاط قوت

- مقاله نوشته شده از لحاظ تئوری بسیار قوی است و مفاهیم و اثبات‌های آن به‌خوبی با استفاده از ریاضیات و احتمال فرمول‌بندی شده است.
- فرض محدودکننده‌ای راجع‌به ویژگی گراف‌ها مانند پژوهش‌های اخیر وجود ندارد و الگوریتم‌های ارائه شده در اکثر کاربردها قابل استفاده هستند.
- برای اولین‌بار در مسئله مربوطه، افزودن مدل‌های یادگیری ماشین به الگوریتم‌های سنتی مورد بررسی قرار گرفته است.
- نتایج ارزیابی تئوری و تجربی بهره‌گیری از مدل‌های یادگیری، بیانگر کاهش حافظه مصرفی است و از این روش می‌توان در مسائل مشابه الهام گرفت.

۱-۲- نقاط ضعف

- پیچیدگی زمانی الگوریتم‌ها به‌صورت تئوری تحلیل شده است اما ارزیابی تجربی نشده است. بهتر بود این مورد نیز به‌صورت عملی آزمایش و با الگوریتم‌های ارائه شده در پژوهش‌های اخیر مقایسه می‌شد.
- در بخش F.8 توضیحات کمی راجع‌به سربار استفاده از مدل‌های یادگیری داده است. بهانه‌ای که به آن اشاره کرده این است که این سربار با توجه به فرآیند و مجموعه‌داده انتخاب شده متفاوت است. اما این دلیل قانع‌کننده‌ای به‌نظر نمی‌رسد. بهتر بود مانند سایر مفاهیم که به‌صورت تئوری با استفاده از ریاضیات و احتمال تحلیل و اثبات کرده است، این مورد نیز مورد توجه قرار می‌گرفت.
- بخش نتیجه‌گیری و کارهای آینده در مقاله موجود نبود.

۲- پیاده‌سازی

کدهای پیاده‌سازی در کولب[9] موجود است. ابتدا با استفاده از کلاس Pre داده‌ها خوانده و پیش‌پردازش می‌شوند. این کلاس، نشانی فایل‌های آموزشی و آزمون را دریافت کرده و گراف مربوط به آنها را می‌سازد و در حافظه ذخیره می‌کند. هر خط از فایل‌ها به‌عنوان یک یال درنظر گرفته شده که دو عدد صحیح اول، نمایانگر گره‌ها یا رأس‌های یال مربوطه است. برای راحتی، هنگام ذخیره‌سازی یال‌ها، رأسی که شناسه کمتری دارد در ابتدا قرار می‌گیرد. گراف‌ها غیرجهت‌دار درنظر گرفته شده و از ذخیره‌سازی یال‌های تکراری صرف‌نظر می‌شود.

پس از آن، یک ترتیب تصادفی از رئوس ایجاد شده و براساس قوانین مربوط به مدل لیست مجاورت، یال‌ها انتخاب می‌شوند. در مدل لیست مجاورت، همه یال‌های مربوط به یک گره، به‌همراه یکدیگر می‌رسند. به‌عبارت دیگر، اگر در جریان‌داده گره x قبل از گره y بیاید، لیست مجاورت مربوط به x نیز قبل از لیست مجاورت y می‌آید. از روی گراف آزمون، تعداد مثلث‌های R_{uv} برای هر یال شمارش می‌شود. با استفاده از گراف آموزشی نیز مقادیر N_{uv} شمارش شده و زیرمجموعه کوچکی از آنها به‌ترتیب نزولی،.برای پیش‌بینی در الگوریتم‌های مبتنی بر یادگیری ذخیره می‌شود. N_{uv} تعداد مثلث‌های مجاور یال دلخواه uv را نشان می‌دهد و R_{uv} تعداد مثلث‌های به فرم uwv را نشان می‌دهد که در آن u قبل از w و w قبل از v در مدل لیست مجاورت می‌رسد.

در پژوهش مربوطه، سه الگوریتم برای تخمین مثلث‌ها در مدل لیست مجاورت پیاده‌سازی و آزمایش شده است. دو الگوریتم، مبتنی بر یادگیری هستند که در این پژوهش برای اولین بار پیشنهاد شده است. الگورریتم دیگر سنتی بوده و با نام MVV شناخته شده که در پژوهشی در سال ۲۰۱۶ ارائه شده است.

در نسخه اول الگوریتم مبتنی بر یادگیری، با یک‌مرتبه نمونه‌برداری تصادفی، یال‌ها به دو دسته سنگین و سبک تفکیک می‌شوند. تشخیص اینکه یک یال سنگین است یا سبک، با استفاده از مقادیر N_{uv} که در مرحله قبل محاسبه شد، پیش‌بینی می‌شود. تعداد مثلث‌های مربوط به یال‌های

[9] https://colab.research.google.com/drive/1Cuv6kolB78d5epuFGhBSySg53Nn-Dn9D?usp=sharing

سنگین و سبک نیز با استفاده از مقادیر R_{uv} حاصل از مرحله قبل، به‌طور جداگانه محاسبه شده و در نهایت با تجمیع[10] آنها تخمینی از تعداد مثلث‌ها حاصل می‌شود. لازم است درصد فضا برای ذخیره یال‌های سنگین توسط کاربر مشخص شود. فضای باقی‌مانده نیز به یال‌های سبک اختصاص داده می‌شود.

در نسخه دوم الگوریتم مبتنی بر یادگیری، زیرنمونه‌برداری چندلایه انجام می‌شود. با دومرتبه نمونه‌برداری، یال‌ها به سه دسته سنگین، سبک و متوسط تفکیک می‌شوند. تعداد مثلث‌ها برای هر سه دسته محاسبه شده و جواب‌ها تجمیع می‌شود. علاوه‌بر درصد فضای یال‌های سنگین، درصد فضا برای یال‌های سبک به‌همراه یک آستانه[11] نیز توسط کاربر تعیین می‌شود. فضای باقی‌مانده نیز مربوط به یال‌های متوسط خواهد بود.

در الگوریتم سنتی MVV نیز یال‌ها به دو دسته سبک و سنگین تفکیک می‌شوند. اما برای تشخیص دسته مربوطه از پیش‌بینی و مدل یادگیری استفاده نمی‌شود. به‌اندازه فضای موجود، یک‌بار از یال‌ها به‌صورت تصادفی نمونه‌برداری انجام می‌شود. یک آستانه برای یال‌های سنگین توسط کاربر تعیین می‌شود و پس از شمارش مثلث‌ها برای هر یال، اگر مقدار R_{uv} مربوط به آن یال بزرگتر یا مساوی با آستانه باشد، به‌عنوان یال سنگین درنظر گرفته می‌شود و در غیر این صورت در دسته سبک قرار می‌گیرد. آنگاه مانند قبل، تعداد مثلث‌ها برای هر دسته محاسبه شده و جواب‌ها تجمیع می‌شود.

[10] Aggregate
[11] Threshold

۳- ارزیابی

در این پژوهش، برای ارزیابی از معیار خطای نسبی[۱۲] استفاده شده است. خطای نسبی، نسبت تفاضل تعداد مثلث‌های تخمین زده شده توسط الگوریتم از تعداد واقعی مثلث‌ها به تعداد واقعی مثلث‌ها را نشان می‌دهد. بنابراین دقت[۱۳] الگوریتم‌ها، با استفاده از معیار خطای نسبی سنجیده می‌شود و هرچقدر خطا کمتر باشد، دقت بالاتر است.

$$RE = \left|1 - \frac{\tilde{T}}{T}\right|$$

برای ارزیابی تجربی عملکرد الگوریتم‌ها، نمودارهایی که خطای نسبی را برحسب فضای مصرفی نشان می‌دهد، ارائه شده است. منظور از فضای مصرفی، حداکثر تعداد یال‌هایی است که الگوریتم در هر اجرا می‌تواند نمونه‌برداری و ذخیره کند. به‌عبارت دیگر، اندازه باکت است که با Z نمایش داده می‌شود. به‌ازای هر مقدار Z ، الگوریتم ۵۰ بار اجرا می‌شود و خطای حاصل، میانگین[۱۴] همه اجراها خواهد بود.

۳-۱- تنظیمات

آزمایش براساس ترتیب تصادفی رسیدن رأس‌ها[۱۵] انجام می‌شود. برای الگوریتم‌های مبتنی بر یادگیری، لازم است مشخص شود که چه درصدی از فضای موجود، به نگهداری یال‌های سنگین اختصاص داده شود. این مقدار، ۱۰ درصد درنظر گرفته خواهد شد. بنابراین تعداد یال‌های سنگین، با k نمایش داده می‌شود و برابر با Z/10 است که مربوط به یال‌هایی است که تاکنون در جریان‌داده دیده شده‌اند و بیشترین مقدار پیش‌بینی شده $R_{u,v}$ را دارند. با ورود یال‌های جدید، به‌مرور

[12] Relative error
[13] Accuracy
[14] Mean
[15] Random vertex arrival order

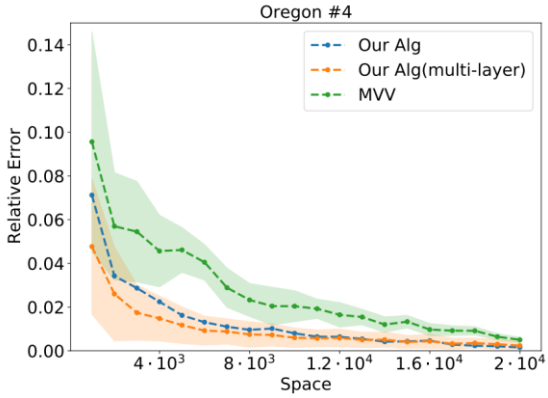
یالی‌های سنگینی که کمترین مقدار را دارند، خارج می‌شوند و جای خود را به یال‌های بزرگتر می‌دهند. برای سایر یال‌ها که ۹۰ درصد از فضا را در اختیار دارند با روشی مشابه نمونه‌برداری انجام می‌شود.

در نسخه چندلایه الگوریتم، علاوه‌بر مشخص شدن درصد فضای یال‌های سنگین، درصد یال‌های خیلی کوچک نیز مشخص شده و برابر با ۷۰ درنظر گرفته می‌شود. فضای باقی‌مانده نیز که ۲۰ درصد است، برای نمونه‌برداری از سایر یال‌ها اختصاص داده می‌شود که به آنها یال‌های متوسط گفته می‌شود. آستانه برای یال‌های کوچک نیز، ۵ درنظر گرفته می‌شود. در الگوریتم MVV نیز مقدار آستانه ۱۰۰ برای یال‌های سنگین تعیین می‌شود.

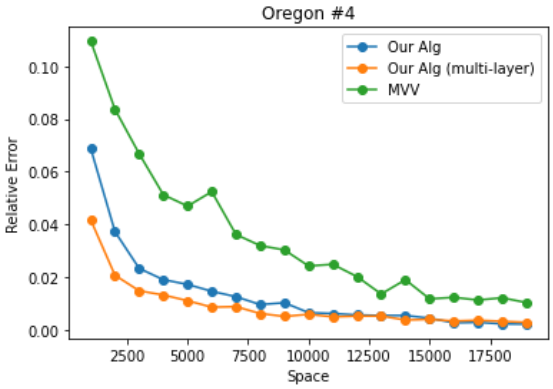
۳-۲- نتایج

برای مجموعه‌داده‌های Oregon و CAIDA که شامل دنباله‌ای از گراف‌ها هستند، گراف اول آنها انتخاب شده و پس از شمارش مقدار دقیق تعداد مثلث‌ها برای هر یال (N_e)، ۱۰ درصد از سنگین‌ترین یال‌ها ذخیره شده و به‌عنوان پیش‌بینی‌گر سنگینی یال‌ها در گراف‌های بعدی استفاده می‌شود. اگر یال مورد پرس‌وجو ذخیره نشده باشد، مقدار پیش‌بینی شده N_e برای آن صفر خواهد بود.

ابتدا چهارمین گراف از دنباله Oregon برای ارزیابی انتخاب می‌شود. این گراف که با نام oregon1_010421 در مجموعه مربوطه قرار گرفته، به‌عنوان مجموعه آزمون درنظر گرفته می‌شود. گراف اول با نام oregon1_010331 نیز به‌عنوان مجموعه آموزش در دو الگوریتمی که مبتنی بر یادگیری هستند استفاده می‌شود. پارامتر فضا، از ۱۰۰۰ تا ۲۰۰۰۰ با مقدار ۱۰۰۰ تغییر می‌کند. نمودارهای زیر نتیجه پیاده‌سازی و اجرای الگوریتم‌ها در این گزارش و پژوهش اصلی را در کنار یکدیگر نمایش می‌دهد.

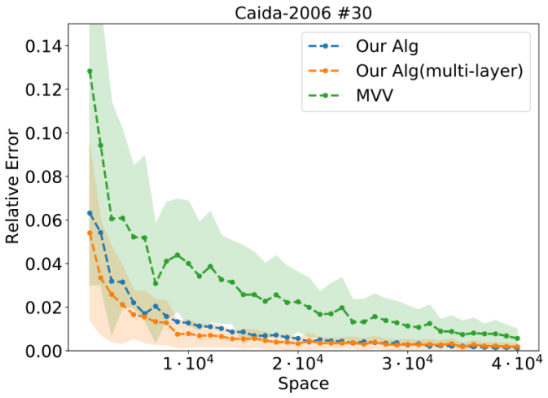


نتیجه پژوهش اصلی

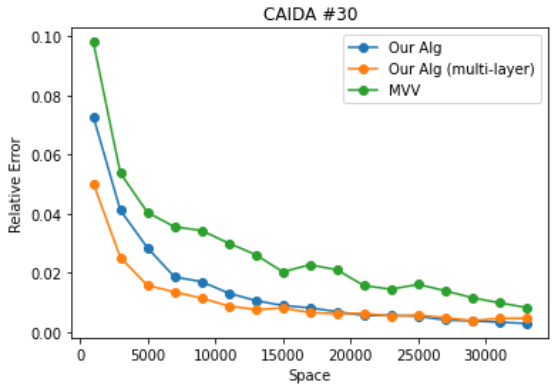


نتیجه این گزارش

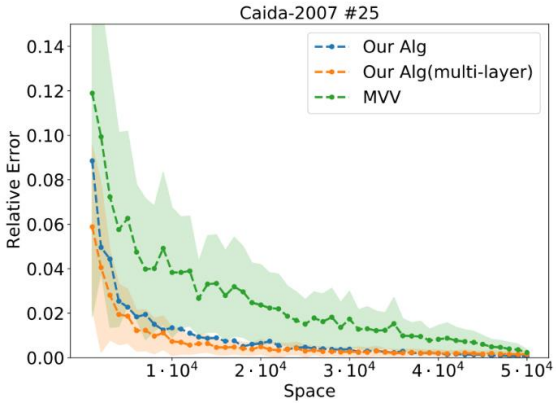
از مجموعه CAIDA گراف اول با نام as-caida20040105 به‌عنوان داده‌های آموزشی درنظر گرفته می‌شود. گراف شماره ۳۰ با نام as-caida20060206 و گراف شماره ۲۵ با نام as-caida20060102 به‌عنوان داده‌های آزمون بررسی می‌شوند. پارامتر فضا، از ۱۰۰۰ تا ۳۵۰۰۰ با مقدار ۲۰۰۰ تغییر می‌کند. نمودارهای زیر، نتایج را نشان می‌دهد.



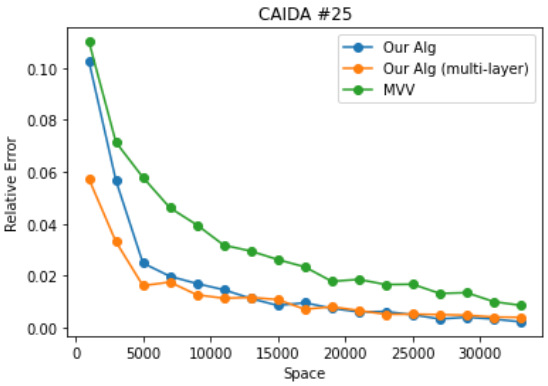
نتیجه پژوهش اصلی



نتیجه این گزارش

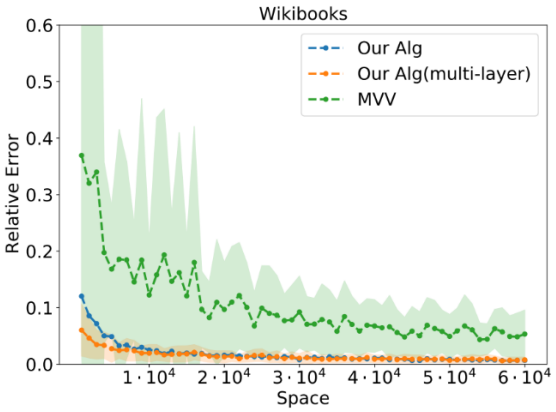


نتیجه پژوهش اصلی

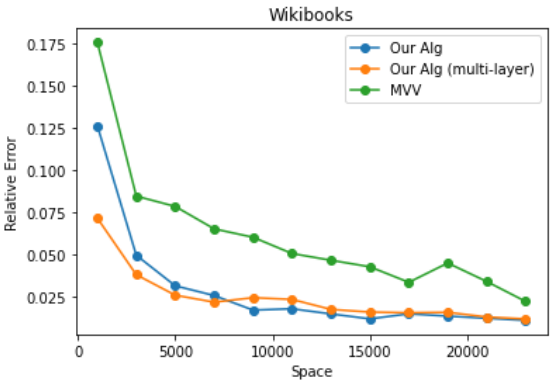


نتیجه این گزارش

مجموعه Wikibooks تشکیل شده از یک فایل حاوی ۱,۱۶۴,۵۷۶ خط است. این فایل به دو قسمت تفکیک می‌شود. نیمی از آن برای آموزش و نیمه دیگر آن به‌عنوان آزمون استفاده می‌شود. در پژوهش مربوطه، چیزی ذکر نشده است که چند درصد داده‌های یادگیری ذخیره شود. بنابراین مانند موارد قبل مقدار این پارامتر را ۱۰ درصد درنظر می‌گیریم. پارامتر فضا، از ۱۰۰۰ تا ۲۵۰۰۰ با مقدار ۲۰۰۰ تغییر می‌کند. نمودارهای زیر، نتایج را نشان می‌دهد.



نتیجه پژوهش اصلی



نتیجه این گزارش

۳-۳- تحلیل

نتایج حاصل از پیاده‌سازی و اجرای الگوریتم‌ها در این گزارش، مشابه نتایج ارائه شده در پژوهش اصلی است. این مورد می‌تواند نشان‌دهنده صحت پیاده‌سازی و تنظیم صحیح پارامترها باشد. البته نتایج حاصل، تفاوت‌های جزئی با یکدیگر دارند که به‌دلیل دخیل بودن عامل‌های تصادفی و احتمالات در الگوریتم‌ها، این مورد طبیعی بوده و قابل اغماض است؛ زیرا در هربار اجرا ممکن است نتایج حاصل کمی تفاوت داشته باشد.

نتایج نشان می‌دهد که الگوریتم‌های مبتنی بر یادگیری برای تخمین تعداد مثلث‌ها در مدل‌های لیست مجاورت که در این پژوهش ارائه شده است به‌ازای فضای یکسان، خطای کمتری نسبت الگوریتم‌های سنتی (MVV) دارد و عملکرد آنها بهتر است. در میان الگوریتم‌های مبتنی بر یادگیری نیز، نسخه‌ای که به‌صورت چندلایه زیرنمونه‌برداری انجام می‌دهد در مجموع عملکرد بهتری دارد. با افزایش فضا نیز خطا کاهش می‌یابد.