کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

محمد راشدی

آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

راهنمای تحویل

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت سوال اشاره‌ای به آن نشده باشد.**
- **کدهای ارسالی می‌بایست قابلیت اجرای دوباره داشته باشند، با این حال، دستیاران آموزشی ملزم به اجرای کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **در صورت استفاده از Jupyter لازم است تا تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد در غیر اینصورت ورودی‌ها و خروجی‌ها متناظر می‌بایست در گزارش آورده شوند.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **با این که بحث در مورد تمرین‌ها منعی ندارد اما راه‌حل شما می‌بایست توسط شما (و فقط شما) باشد. همچنین، تمامی مطالب جانبی در گزارش باید رفرنس داده شود. یادآوری می‌شود که عدم صداقت علمی[1] عواقب شدیدی را به همراه دارد.**
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست.**
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت‌کننده در آن، به میزان بارم سوال نمره منفی لحاظ می‌شود.**
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number]_[Lastname]_[StudentNumber].zip

---

[1] Academic dishonesty

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

محمد راشدی

آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

در صورت وجود سوال، ابهام و یا درخواست راهنمایی با دستیاران آموزشی مرتبط با هر پرسش از طریق ایمیل‌های آورده شده در سربرگ در ارتباط باشید.

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

محمد راشدی

آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

فهرست

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۱ – شمارش کلمات و شاخص وارونه[۱]

الف) هدف ما در این قسمت شمارش تعداد رخداد هر کلمه در تمام اسناد فراهم شده می‌باشد. در این تمرین ۵ فایل متنی داریم که هر کدام یک کتاب از سایت پروژه گوتنبرگ هستند. فایل‌های این تمرین در پوشه‌ی q1/data قرارداده شده اند. برای انجام این تمرین باید در هر تسک Map برای هر سند، رخداد تمام کلمات در آن را بشمارید و در Combiner با هم ترکیب کنید. در Reduce نیز باید برای تعداد رخداد هر کلمه در تمام اسناد را در خروجی چاپ کنید. در پایان ۲۰ کلمه با بیشترین تکرار را گزارش کنید.

توجه: قبل از نوشتن برنامه فایل‌های متنی باید تمیز شوند. برای مثال حذف علائم نگارشی، تبدیل شکل‌های مختلف کلمات به یک شکل واحد(با هر روش دلخواه) و غیره.

ب) در بخش قبل احتمالا پرتعداد ترین کلمات، ایست‌واژه[۲]هایی مانند a، the، I و ... بوده‌اند که تاثیر معنایی زیادی ندارند. در این مرحله لیستی از حداقل ۵۰ ایست‌واژه را تعیین کنید و آن‌ها را در نظر نگیرید. حالا عملیات شمارش را مانند گام قبل اجرا کنید و ۲۰ کلمه با بیشترین تکرار را چاپ کنید.

پ) یک شاخص وارونه داده ساختاری است که برای دریافت اسناد یا صفحات وبِ حاوی یک کلمه یا مجموعه‌ای از کلمات به صورتی کارا[۳] بسیار مناسب می‌باشد. در یک شاخص وارونه ، کلید هر خط یک کلمه و مقدار آن لیستی از اسناد حاوی آن کلید می‌باشند. در این قسمت باید با استفاده از معماری MapReduce برای هر کلمه، لیستی از اسنادی که این کلمه در آن‌ها رخ می‌دهد و همچنین تعداد تکرار کلمه در هر سند را چاپ کنید. برای مثال برای کلمه logic خروجی شما باید به شکل زیر باشد:

- logic [('0', '37'), ('1', '15'), ('3', '3')]

---

۱ Inverted Index
۲ Stopword
۳ Efficient

این پاسخ به این معنی است که کلمه logic در فایل اول ۳۷ بار تکرار شده، در فایل دوم ۱۵ بار و در فایل چهارم ۳ بار. همچنین این کلمه در فایل‌های سوم و پنجم رخ نداده است. البته به جای اعداد 0 تا 4 بهتر است که نام کتاب را چاپ کنید.

---

۲ – پیاده‌سازی الگوریتم PageRank با MapReduce

PageRank الگوریتمی است که توسط Google Search برای امتیازدهی به نتایج موتور جستجوی گوگل به کار می‌رود. این الگوریتم به افتخار Larry Page یکی از هم‌بنیان‌گذاران گوگل که این الگوریتم را ارائه داد، نامگذاری شده است. این الگوریتم روشی برای اندازه‌گیری اهمیت[۱] صفحات وب(به طور کلی‌تر اهمیت نودها در یک گراف) است. در این الگوریتم اهمیت هر نود متناسب است با تعداد و کیفیت لینک‌هایی که به آن نود وارد می‌شوند. فرض ما این است که صفحات مهم‌تر به احتمال زیاد لینک‌های زیادی از دیگر صفحات دریافت خواهند کرد.

الگوریتم PageRank را می‌توان به صورت زیر پیاده‌سازی کرد:

۱. در ابتدا اهمیت هر نود را ۱ یا ($\frac{1}{\text{تعداد صفحات}}$) در نظر بگیرید (بسته به روشی که استفاده می‌کنید).

۲. سپس اهمیت هر نود را مطابق رابطه‌ی زیر آپدیت می‌کنیم. در اینجا r اهمیت یک نود را نشان می‌دهد و N(j) یعنی مجموعه‌ی همسایه‌های نود j . d_i نیز تعداد یال‌های نود i را مشخص می‌کند.

$$r_j = \sum_{i \in N(j)} \frac{r_i}{d_i}$$

۳. گام دوم را آنقدر تکرار کنید تا مقدار اهمیت نودها پایدار شود(دیگر تغییر نکند).

---

[۱] Importance(Rank)

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

در این تمرین شما باید با استفاده از الگوی MapReduce الگوریتم PageRank را بر روی دیتاست Google Web Graph پیاده‌سازی کنید. این دیتاست در پوشه‌ی q2/data فراهم شده است. دیتاست ما شامل تعاملات بخشی از وب است که در سال ۲۰۰۲ توسط گوگل برای یک مسابقه‌ی برنامه‌نویسی منتشر شد. این دیتاست یک گراف جهت‌دار متشکل از 875713 نود و 5105039 یال است. در اینجا نودها همان صفحات وب و یال‌ها نیز لینک‌های بین صفحات می‌باشند. هر خط از دیتاست ما نشان‌دهنده‌ی یک یال جهت‌دار از نود سمت چپ به نود سمت راست است.

می‌خواهیم با استفاده از کتابخانه‌های پردازش موازی مانند اسپارک اهمیت هر نود را توسط الگوریتم PageRank به دست آوریم. اسپارک یک موتور تحلیلی یکپارچه و متن‌باز برای پردازش موازی داده‌ها در مقیاس بزرگ است. برای استفاده از اسپارک در زبان پایتون می‌تواند از کتابخانه‌ی pyspark استفاده کنید. Pyspark یک رابط برنامه‌نویسی برای اسپارک در پایتون است. برای آشنایی بیشتر با اسپارک و pyspark فایل spark_tutorial.ipynb در پوشه q2 قرار داده شده است.

توجه داشته باشید که الگوریتم PageRank یک الگوریتم بازگشتی است و باید چندبار اجرا شود تا اهمیت نودها به میزان درست خود همگرا شوند.

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۳ - رمز بی‌مانند (NFT[1])

هدف از این تمرین آزمایش دانش و مهارت شما در برنامه نویسی MapReduce با استفاده از یک کتابخانه مدرن پایتون یعنی Dask می‌باشد. Dask کتابخانه‌ای برای انجام محاسبات موازی بر روی داده‌های بزرگ می‌باشد. مزیت Dask هماهنگی عالی آن با کتابخانه‌های معروف پایتون مانند Pandas و Numpy است که می‌تواند کار شما را بسیار راحت کند. Dask همچنین می‌تواند به ماشین‌های چند هسته‌ای و به خوشه‌های توزیع شده مقیاس‌پذیر شود، که منجر به بهبود عملکرد و کارایی برنامه مبتنی بر MapReduce می‌شود.

تعریف مساله: دیتاست ما متشکل از ۴۴هزار تصویر از محصولات پوشیدنی است(لینک به مجموعه‌داده‌گان در Kaggle). هر تصویر از یک شناسه منحصر به فرد و یک برچسب که تصویر مدنظر را توصیف می‌کند تشکیل شده است. پیشنهاد می‌شود که قبل از انجام تمرین، به خوبی با فایل‌ها و فیلدهای مختلف هر فایل آشنا شوید. قصد داریم تا به کمک MapReduce دو وظیفه زیر را انجام دهیم:

۱. برای هر تصویر، شبیه‌ترین تصویر را براساس ظاهر آن پیدا کنید(از نظر مقادیر پیکسل تصاویر). برای اندازه‌گیری فاصله بین پیکسل‌ها می‌تواند از انواع روش‌ها مانند فاصله اقلیدسی و یا فاصله کسینوسی استفاده کنید. خروجی این مرحله برای هر تصویر، شناسه آن تصویر و شبیه‌ترین تصویر به آن و فاصله آن‌ها باشد.
۲. برای هر تصویر بر اساس آنچه که تصویر نشان میدهد یا بنظر می‌آید، یک کد خاص بسازید(این کد به NFT ID معروف است). شناسه NFT یک دارایی دیجیتال منحصر به فرد است، که مالکیت یا دارا بودن حق/حقوقی از یک موجودیت مورد منحصر به فرد را نشان می‌دهد. شما میتوانید از هر روشی (مانند SHA-256 یا MD5) برای ساختن این کد از ورودی استفاده کنید، تا یک رشته منحصر به فرد از کاراکترها برای هر شناسه NFT ایجاد کنید. خروجی باید شناسه NFT و شناسه تصویر متناظر آن باشد.

[1] Non-Fungible Token

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

هدف از ایجاد شناسه‌های NFT برای هر تصویر، آماده‌سازی بستری است که کاربران را قادر می‌سازد، تصاویر را به عنوان موارد منحصر به فرد خرید و فروش کنند، همچنین ساختار شناسه‌ها به کاربران کمک می‌کند تا تصاویر مورد علاقه خود را راحت‌تر پیدا و مقایسه کنند. بنابراین، فرایند ایجاد شناسه‌های NFT (استفاده از برچسب و شبیه‌ترین تصویر به هر تصویر به عنوان ورودی‌های ساخت کد) به شما اطمینان می‌دهد که شناسه‌های NFT ایجادشده، بیانگر آنچه تصاویر نشان می‌دهند و چگونه به نظر می‌رسند، هستند.

ورودی: شما می‌بایست مجموعه‌دادگان را دریافت و به فرمت ورودی درآورید. ساختار دادگان ورودی، یک فایل متنی است به طوری که هر تصویر در یک خط و هر خط با مقدارهای شناسه تصویر، برچسب (انتخاب صحیح برچسب بر اساس فراداده[1]های موجود در فایل styles.csv انجام می‌گردد، دلیل انتخاب یک، دو یا چند برچسب ارائه شده به عنوان برچسب تصویر را در گزارش خود ذکر کنید) و مقادیر پیکسل که با کاما «،» از هم جدا شده‌اند، برای مثال؛

- 1234,car,255,0,0,0,255,0,…
- 5678,dog,0,255,0,0,0,255,…

خروجی مورد انتظار: با در نظر گرفتن اینکه داده‌های خروجی در فایل دیگری نوشته می‌شوند، انتظار میرود که هر خط شامل یک تصویر و با مقدارهای شناسه تصویر، شبیه‌ترین شناسه تصویر، فاصله و شناسه NFT که با کاما «،» از هم جدا شده‌اند، تکمیل گردد، برای مثال؛

- 1234, 3456, 12.34, a1b2c3d4e5f6...
- 56789, 7890, 56.78, f9e8d7c6b5a4...

[1] Metadata

همچنین موارد خواسته شده را در گزارش خود ذکر کنید.

- **اندازه کاهش‌دهنده**[۱]، که تعداد کاهش‌دهنده‌هایی است که نتایج میانی را از mapها پردازش می‌کند. شما باید اندازه کاهش‌دهنده را طوری انتخاب کنید که حجم بار[۲] را تعدیل کرده و سربار ارتباط[۳] را به حداقل برساند.
- **نرخ تکرار**[۴]، تعداد کپی‌های هر فایل ورودی است که در گره‌های خوشه توزیع می‌شود. شما باید نرخ تکرار مناسبی را انتخاب کنید که تحمل خطا و در دسترس بودن داده‌ها را تضمین کند.
- **عملکرد**[۵]، به عوامل مختلفی مانند پیچیدگی map و reduce، اندازه و فرمت دادگان ورودی و خروجی و پیکربندی خوشه‌ی شما بستگی دارد.

---

۱ Reducer Size

۲ Workload

۳ Communication Overhead

۴ Replication Rate

۵ Performance

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۴ – پیشنهاد دوستی

در این قسمت باید با استفاده از اسپارک برنامه‌ای برای پیشنهاد ((افرادی که ممکن است بشناسید)) بنویسید. ایده‌ی اصلی این است که اگر دو فرد، دوستان مشترک زیادی دارند، آنگاه سیستم باید پیشنهاد دهد که این دو نفر باید با هم دوست شوند.

مجموعه داده‌گان

فایل friendships.txt در پوشه‌ی q4/data یک لیست مجاورت برای شبکه‌ی اجتماعی ما می‌باشد و هر خط آن به فرم زیر می‌باشد:

<کاربر><TAB><دوستان>

در اینجا <کاربر> یک فرد است که با یک ID یکتا مشخص شده است و <دوستان> لیستی از ID دوستان آن فرد می‌باشد که با کاما «،» از هم جدا شده‌اند. توجه کنید که گراف ما جهت دار **نیست** و روابط دوستی دوطرفه هستند. در فایل بالا هر دو طرف دوستی ذکر شده‌اند(مثلا 0 → 1 و 1 → 0 هر دو نوشته شده‌اند).

روش کار

روش کار به این صورت است که باید برای هر کاربر مانند U، 10 نفر از کسانی که از دوستان U نیستند اما بیشترین دوست مشترک با U دارند را بیابید و به U پیشنهاد دهید.

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

خروجی

خروجی باید شامل یک خط به ازای هر کاربر به فرم زیر باشد:

<کاربر><TAB><پیشنهادها>

در اینجا نیز <کاربر> یک فرد است که با یک ID یکتا مشخص شده است و <پیشنهادها> لیستی از ID دوستان پیشنهادی برای آن فرد می‌باشد که با کاما از هم جداشده‌اند. پیشنهادها باید براساس تعداد دوستان مشترک و به صورت نزولی مرتب شوند.

حتی اگر یک کاربر کمتر از ۱۰ دوست درجه ۲ داشته باشد، همه‌ی آن‌ها را براساس تعداد دوستان مشترک و به ترتیب نزولی لیست کنید. اگر یک کاربر هیچ دوستی نداشته باشد، می‌تواند لیست پیشنهادها را خالی بگذارید. اگر پیشنهادهایی تعداد دوستان مشترک یکسانی داشته باشند، ID آن‌ها را به صورت صعودی از نظر عددی بنویسید.

نکات

- برای راحتی کار با اسپارک می‌توانید از Google Colab استفاده کنید.
- پیشنهاد می‌شود که از ساختارهای Pyspark Dataframe و یا RDD در اسپارک استفاده کنید.
- برای اطمینان از جواب، ۱۰ پیشنهاد برتر برای کاربر شماره‌ی ۱۱ باید به صورت زیر باشند:

27552,7785,27573,27574,27589,27590,27600,27617,27620,27667

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

محمد راشدی

آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۵ – پیاده‌سازی LSH بر روی دیتاست MovieLens 100k

هدف این تمرین پیاده‌سازی LSH و min-hashing بر روی دیتاست MovieLens 100k می‌باشد. این دیتاست متشکل از 100,000 امتیاز است که توسط ۹۴۳ کاربر به ۱۶۸۲ فیلم داده شده‌اند. فایل این دیتاست در پوشه‌ی q5/data فراهم شده است. برای اطلاع از محتویات این دیتاست فایل README آن را بخوانید و براساس نیاز پردازش‌های لازم را انجام دهید. چیزی که در این تمرین برای ما مهم است، مجموعه‌ی فیلم‌هایی است که یک کاربر به آن‌ها امتیاز داده است و نه خود امتیازات. می‌خواهیم که شباهت ژاکارد را بین کاربران محاسبه کنیم.

شباهت ژآکارد دقیق را برای تمام جفت کاربران محاسبه کرده و آن‌هایی که شباهت حداقل 0.5 دارند را چاپ کنید. سپس امضای min-hash کاربران را محاسبه کرده و شباهت ژاکارد تقریبی را با روش ذکر شده در کلاس به دست آورید. از ۵۰، ۱۰۰ و ۲۰۰ تابع hash استفاده کنید. برای هر مقدار، جفت‌هایی که حداقل شباهت تخمینی 0.5 را دارند، چاپ کرده و تعداد False Negativeها و False Postiveها را گزارش کنید. برای تعداد False Negativeها و False Postiveها متوسط تعداد ۵ اجرا را گزارش کنید.

سپس جدول امضاها را به b باند تقسیم کرده و برای هر باند r تابع هش در نظر بگیرید و LSH را پیاده‌سازی کنید. هدف یافتن جفت‌های کاندید با حداقل شباهت 0.6 است. برای جدول دارای ۵۰ تابع هش، b = 10 و r = 5 در نظر بگیرید. برای جدول دارای ۱۰۰ تابع هش، b = 40 و r = 5 و برای جدول دارای ۲۰۰ تابع هش b = 20 و r = 10 در نظر بگیرید. تعداد False Negativeها و False Postiveها را گزارش کنید. برای تعداد False Negativeها و False Postiveها متوسط تعداد ۵ اجرا را گزارش کنید. اگر حداقل شباهت 0.8 باشد تعداد False Negativeها و False Postiveها چه تغییری می‌کنند. آستانه[1] تابع Sigmoid چقدر خواهد بود.

[1] Threshold

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۶ – پیاده‌سازی الگوریتم SimHash

در این تمرین شما باید الگوریتم SimHash را پیاده‌سازی کنید. SimHash الگوریتمی برای تولید یک اثرانگشت یا هش با طول ثابت از ورودی‌هایی با طول متغیر مانند اسناد متنی می‌باشد. SimHash تا حدی مشابه توابع هش معمولی می‌باشد. SimHash نوعی از توابع LSH است اما به گونه‌ای طراحی شده تا نسبت به تصادم مقاوم‌تر باشد. روش کار SimHash به این صورت است که ابتدا ورودی را به توکن‌های کوچک‌تر می‌شکند و سپس برای هر توکن یک هش ایجاد می‌کند. سپس این هش‌ها را با هم ترکیب می‌کند تا هش نهایی آن ورودی را به دست آورد.

الگوریتم SimHash اکثرا برای یافتن اسناد تکراری یا تقریبا تکراری، تشخیص اسپم و ... به کار می‌رود. مزیت اصلی SimHash این است که محاسبه آن از دیگر الگوریتم‌های یافتن شباهت متنی سریع‌تر و به‌صرفه‌تر است.

روش کار SimHash

- ابتدا ورودی را به تعدادی توکن تبدیل کنید و علائم نگارشی را حذف کرده و دیگر پردازش‌ها را برحسب نیاز انجام دهید.
- به هر توکن یک وزن اختصاص دهید. برای محاسبه وزن می‌تواند از فرکانس رخداد کلمات یا روش TF-IDF استفاده کنید.
- با استفاده از توابع استاندارد هش مانند MD5 یا SHA-1 و ... برای هر توکن یک هش ایجاد کنید. سپس این هش را باید به باینری تبدیل کنید.
- در هش هر کلمه تمام 0ها را به 1- تبدیل کرده و سپس مقادیر هر ستون را با هم جمع بزنید تا هش نهایی به دست آید. اگر حاصل جمع یک ستون بزرگتر از ۰ باشد آنگاه حاصل ۱ بوده و در غیر اینصورت ۰ می‌باشد.

در پایان باید با مقایسه‌ی فاصله بین دو متن باید نشان دهید که الگوریتم شما به درستی کار می‌کند. برای محاسبه‌ی فاصله می‌توانید از فاصله‌ی همینگ استفاده کنید.

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

۷ – LSH برای جستجوی تقریبی همسایگان نزدیک[1]

در این مسئله، پیاده‌سازی LSH برای حل مسئله‌ی جستجوی تقریبی همسایگان نزدیک را بررسی خواهیم کرد. فرض کنید که یک دیتاست به نام A در فضای متریک داریم که از n نقطه تشکیل شده است و d(.,.) نیز یک معیار فاصله در این فضا است. همچنین فرض کنید c یک ثابت بزرگتر از ۱ باشد. آنگاه مسئله‌ی (c, λ)-ANN[2] به صورت زیر تعریف می‌شود:

- اگر یک نقطه‌ی پرس‌وجو[3] مانند z داشته باشیم که فاصله‌ی آن از یک نقطه مانند x در دیتاست A به صورت $d(x, z) \leq \lambda$ باشد، آنگاه نقطه‌ای مانند x' در دیتاست را بیابید که فاصله آن از z به صورت $d(x',z) \leq c\lambda$ باشد(این نقطه را یک (c,λ)-ANN می‌نامیم). بنابراین پارامتر c نشان‌دهنده‌ی حداکثر فاکتور تقریب مجاز در مسئله می‌باشد.

حالا اگر یک خانواده از توابع LSH به نام H را در نظر بگیریم که برای معیار فاصله‌ی d(.,.)، (λ, cλ, p1, p2)-sensitive باشد و داشته باشیم:

$$Let\ G = H^k = \{g = (h1, \dots, h_k) | h_i \in H, \forall\ 1 \leq i \leq k\}\ \ where\ \ K = log_{\frac{1}{p_2}}(n)$$

نکته: تساوی $G = H^k$ به این معنی است که هر تابع از G یک And-Construction از k تابع از H می‌باشد.

---

[1] Approximate Near Neighbor Search
[2] Approximate Near Neighbor
[3] Query Point

کاوش داده‌گان انبوه

تمرین «یک»

دکتر سامان هراتی‌زاده
دانشگاه تهران - دانشکده سامانه‌های هوشمند
نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دستیاران آموزشی
محمد راشدی
آرمین رحیمی دره‌باغ

ددلاین: ساعت ۵۹:۲۳ | ۱۴۰۳/۰۳/۱۵

حال روال زیر را در نظر بگیرید:

۱. به تعداد $L = n^{\rho}$ تابع تصادفی gL ,…, g1 از G انتخاب کنید، به طوری که $\rho = \frac{\log\left(\frac{1}{p_1}\right)}{\log\left(\frac{1}{p_2}\right)}$ باشد.
۲. تمام نقاط درون دیتاست و نقطه‌ی پرس‌وجو را با تمام توابع g_i هش کنید ($1 \le i \le L$).
۳. حداکثر 3L نقطه از مجموعه L باکتی که نقطه‌ی پرس‌وجو به آن‌ها هش شده است، انتخاب کنید(انتخاب باید به صورت تصادفی و یکنواخت باشد). اگر تعداد نقاطی که به باکت‌های نقطه‌ی پرس‌وجو هش شده‌اند کمتر از 3L باشد، به همان تعداد، نقطه در نظر بگیرید.
۴. از میان نقاط انتخاب شده در مرحله‌ی ۳، نزدیک‌ترین نقطه به نقطه‌ی پرس‌وجو به عنوان یک (c,λ)-ANN را گزارش کنید.

سوالات

الف) هدف این بخش نشان دادن این است که روال بالا به یک پاسخ درست با احتمال ثابت منجر خواهد شد. اگر $W_j = \{x \in A \,|g_i(x) = g_j(z)\}\ (1 \le j \le L)$ مجموعه‌ی نقاط x باشند که توسط تابع هش g_j به مقداری یکسان با نقطه‌ی پرس‌وجو یعنی z هش می‌شوند، آنگاه تعریف می‌کنیم: $T = \{x \in A | d(x,z) > c\lambda\}$

اثبات کنید(Pr همان احتمال است):

$$\Pr\left[\sum_{j=1}^{L} |T \cap W_j| \ge 3L\right] \le \frac{1}{3}$$

راهنمایی: از نامساوی مارکوف[1] استفاده کنید.

[1] Markov's Inequality

ب) اگر $x^* \in A$ نقطه‌ای باشد به طوری که $d(x^*, z) \leq \lambda$، *آنگاه اثبات کنید:*

$$\Pr[\forall\, 1 \leq j \leq L, g_j(x^*) \neq g_j(z)] < \frac{1}{e}$$

پ) اثبات کنید که با احتمالی بیشتر از یک مقدارِ ثابتِ مشخص، نقطه‌ی گزارش‌شده یک (c, λ)-ANN واقعی است.

ت) یک دیتاست از تکه‌های تصاویر[۱] در پوشه‌ی q7/data فراهم شده است. هر سطراز این دیتاست بخشی از یک تصویر با ابعاد ۲۰ * ۲۰ می‌باشد که به صورت یک بردار ۴۰۰بعدی بازنمایی شده است. برای تعریف شباهت تصاویر از معیار فاصله‌ی L1 استفاده خواهیم کرد. می‌خواهیم که عملکرد مسئله‌ی ANN مبتنی بر LSH را با جستجوی خطی مقایسه کنیم. باید از کد فراهم شده در کنار دیتاست برای انجام این بخش استفاده کنید.

کد **lsh.py** تمام نقاطی که به کدنویسی شما نیاز دارد را با TODO مشخص کرده است. به طور خاص، باید از توابع lsh_setup و lsh_search استفاده کنید و جستجوی خطی را خودتان پیاده‌سازی کنید. مقادیر پیش‌فرض به صورت L =10 و k = 24 هستند اما می‌توانید از مقادیر دیگر نیز استفاده کنید به شرطی که دلیل انتخاب مقادیر جدید را توضیح دهید.

- برای هر یک از تکه تصاویر در سطرهای 100, 200, 300,…, 1000 سه نزدیک‌ترین همسایه(به غیر از خود تصویر) را با روش LSH و جستجوی خطی بیابید. متوسط زمان جستجو را برای LSH و جستجوی خطی، گزارش کنید.
- با فرض اینکه $\{z_j | 1 \leq j \leq 10\}$ *مجموعه‌ی تکه تصاویر باشد و* $\{x_{ij}\}_{i=1}^{3}$ ۳ *همسایه یافته شده برای هر تصویر با روش مبتنی بر LSH باشند و* $\{x_{ij}^*\}_{i=1}^{3}$ ۳ *نزدیک‌ترین همسایه واقعی* z_j *باشند که توسط جستجوی خطی یافت شده‌اند، آنگاه، مقدار خطا را طبق رابطه‌ی زیر محاسبه کنید:*

---

[۱] Image Patches

$$erro = \frac{1}{10}\sum_{j=1}^{10}\frac{\sum_{i=1}^{3} d(x_{ij}, z_j)}{\sum_{i=1}^{3} d(x_{ij}^{*}, z_j)}$$

مقدار خطا را به عنوان تابعی از L رسم کنید(برای L = 10, 12, …, 20 و k = 24) . به طور مشابه، مقدار خطا را به عنوان تابعی از k نیز رسم کنید(k = 16, 18, 20, 22, 24 و L = 10). به طور مختصر توضیحاتی را برای هر نمودار ارائه دهید.

- در نهایت، ۱۰ بهترین همسایه را که توسط دو روش برای تصویر شماره ۱۰۰ یافت شدند(با استفاده از مقادیر پیش‌فرض L = 10 و k = 24 یا مقادیری که خودتان انتخاب کردید) را به همراه خود تصویر، رسم کنید. آیا تصویر و همسایه‌های یافت‌شده از نظر بصری به هم نزدیک هستند؟