




تحليل كلان داده‌ها

تمرین دوم

{Clustering, Data Stream}

مهلت تحویل

۱۴۰۲/۰۲/۲۲



برای ارسال تمرین به نکات زیر توجه کنید.

۱. ملاک اصلی انجام تمرین گزارش آن است و ارسال کد بدون گزارش فاقد ارزش است. برای این تمرین یک فایل گزارش در قالب pdf تهیه کنید و در آن برای هر سوال، تصاویر ورودی، تصاویر خروجی و توضیحات مربوط به آن را ذکر کنید. سعی کنید توضیحات کامل و جامعی تهیه کنید.
۲. زبان برنامه‌نویسی برای انجام تمرین‌ها، پایتون(Python) در نظر گرفته شده است.
۳. برای سهولت در انجام تمرینات، توصیه می‌شود که پلتفرم کولب گوگل استفاده نمایید.
۴. در نظر داشته باشید کدهای شما باید قابلیت اجرا در هنگام ارائه را داشته باشند. همچنین بر روی کدهای خود مسلط باشید
۵. کدهای ارسالی خود را برای افزایش خوانایی و درک بهتر به صورت مناسب کامنت‌گذاری کنید.
۶. پاسخ سوالات تشریحی، فایل‌های کد و گزارش خود را در یک فایل فشرده قرار داده و با نام با فرمت HW2_StudentID ارسال نمایید.
۷. در صورت وجود ابهام یا اشکال می‌توانید از طریق [کانال](#) با ما در ارتباط باشید(داخل کانال لینک گروه نیز موجود می‌باشد)

- ۱- در این بخش وظیفه شما این است که با استفاده از الگوریتم CURE به خوشه بندی داده‌های چند بعدی بپردازید. یک فایل CURE.ipynb در اختیار شما قرار داده شده است. در این فایل کد مربوط به load کردن دو مجموعه داده مورد نظر برای این بخش، از پیش نوشته شده است. برای حل این بخش لازم است این قطعه کد را در ابتدای کد خود کپی کرده و با استفاده از این دو مجموعه داده، نتایج خود را بیان نمایید. لازم به ذکر است که متغیر y موجود در قطعه کد صرفاً جهت مصورسازی بوده و به دلیل اینکه با یک مسئله بدون نظارت روبرو هستیم در ادامه از آن استفاده‌ای نخواهد شد.
- الف) برای پیاده سازی بخش اول الگوریتم CURE ابتدا ۲۰۰ داده از مجموعه داده‌ها را به صورت تصادفی انتخاب کرده و آن‌ها را بصورت سلسله مراتبی خوشه بندی کنید. سپس نقاط Representation را برای خوشه‌ها انتخاب کنید (تعداد نقاط را باید با استفاده از سعی و خطا به دست آورید و در انتها برای هر کدام از مجموعه داده‌ها به صورت جداگانه ذکر کنید).
- ب) نقاط Representation به دست آمده در مرحله قبل را ترسیم کنید.
- با انجام دو قسمت بالا شما مرحله اول الگوریتم CURE را با موفقیت پیاده سازی کرده‌اید.
- ج) مرحله دوم الگوریتم CURE را پیاده‌سازی کرده و دسته مربوط به هر یک نقاط را مشخص کنید و در نهایت آن‌ها را ترسیم نمایید.
- د) الگوریتم BFR را به اختصار توضیح دهید و مزیت‌ها و معایب آن را با CURE مقایسه کنید. (از بخش د به بعد لزومی به پیاده سازی وجود ندارد)
- ه) به نظر شما آیا الگوریتم BFR می‌تواند مجموعه داده‌های انتخابی برای این مسئله را خوشه بندی کند؟
- ی) یکی از روش‌های مطرح برای انتخاب پارامتر k در الگوریتم k -means، استفاده از روش elbow می‌باشد این روش را به اختصار توضیح دهید. به نظر شما آیا استفاده از این روش در برای تعیین k در الگوریتم K -NN نیز مناسب است؟

^۱ در این تمرین استفاده از Spark مجاز نمی‌باشد.

۱- برای ثبت نام در برخی سایت‌ها نیاز به ایجاد نام کاربری می‌باشد. زمانی اجازه تعریف نام کاربری جدید داده می‌شود که آن نام کاربری از قبل توسط کاربر دیگری تعریف نشده باشد. پس هدف مساله این است که اجازه ایجاد نام کاربری جدید که مشابه یکی از نام‌های کاربری موجود در سایت می‌باشد، داده نشود. یک راه، مقایسه نام کاربری درخواستی با تمامی نام‌های کاربری در سایت هست که از مرتبه $O(n)$ می‌باشد و زمان بر است. همچنین چنین کاری نیاز به استفاده از حافظه زیادی دارد. در این بخش قصد داریم از الگوریتم bloom-filter برای حل این مساله استفاده کنیم تا هم زمان اجرا را کاهش دهیم و هم نیازی به مصرف حافظه زیاد نداشته باشیم.

الف) به صورت مختصر توضیح دهید چرا الگوریتم bloom-filter برای استفاده در این مساله مناسب است و هدف مساله را برآورده می‌کند؟

در این بخش مجموعه داده ی `user_dataset` که در اختیار شما قرار گرفته است، شامل یک میلیون نام کاربری ثبت نام شده در سایت ردیت می‌باشد. هر نام کاربری، یک رشته به صورت $S = \langle s_0 s_1 \dots s_{l-1} \rangle$ می‌باشد که l طول رشته مورد نظر می‌باشد. ابتدا لازم است کاراکترهای موجود در رشته را به ارقام تبدیل کنیم که برای اینکار از دستور `ord()` در پایتون استفاده کنید. در این صورت رشته S تبدیل به $C = [c_0, c_1, \dots, c_{l-1}]$ می‌شود که $c_i = ord(s_i)$ می‌باشد.

در ادامه دو نوع تابع هش معرفی می‌کنیم. از این توابع برای پیاده سازی الگوریتم bloom-filter استفاده می‌شود.

type I hash function:

$$h(S) = \left(\min(C) + \left(\prod_{i=0}^{l-1} c_i \right) \times p + \left(\sum_{i=0}^{l-1} c_i \right) \times p^{\lfloor l/2 \rfloor} + \max(C) \times p^{l-1} \right) \bmod M$$

type II hash function:

$$h(S) = \left(\sum_{i=0}^{l-1} (c_i \times p^i) \right) \bmod M$$

در عبارات بالا p یک عدد دلخواهی است که باید تعیین شود و معمولاً عدد اول انتخاب می‌شود. M اندازه جدول هش می‌باشد.

ب) الگوریتم bloom-filter را با استفاده از تابع هش نوع یک، روی مجموعه داده ی `user_dataset` پیاده سازی کنید. برای اینکار ۴ تابع هش با مقادیر $p = 17, 31, 47, 61$ تعریف کنید. در صورتی که بخواهیم احتمال `false positive` مورد انتظار، $9 \pm 0.2\%$ درصد باشد، اندازه جدول هش را چند برابر اندازه دیتاست تعیین کنیم (دو برابر، سه برابر، ...) ؟

ج) الگوریتم bloom-filter را این بار روی تابع هش نوع دو با همان شرایط و پارامترهای قسمت ب پیاده سازی کنید.

در این بخش مجموعه داده user_requests در اختیار شما قرار می‌گیرد. ستون اول نام‌های کاربری هست که در طول زمان برای ثبت نام در سایت درخواست داده شده‌اند. ستون دوم نشان می‌دهد که نام کاربری درخواستی در سایت وجود دارد یا خیر. مقدار ۱ نشان می‌دهد نام کاربری درخواستی در سایت موجود است و صفر یعنی اینکه نام کاربری آزاد هست و می‌تواند برای ثبت نام انتخاب شود.

د) با استفاده از توابع هش در قسمت ب و جدول هش که در این قسمت بدست آورده‌اید، مشخص کنید نام‌های کاربری دیتاست user_requests، برای ثبت نام آزاد هست یا خیر. در نهایت false positive را بدست آورید و در گزارش خود ذکر کنید.

ه) تمامی مراحل قسمت قبل را برای توابع و جدول هش قسمت ج نیز انجام دهید.

ی) مقادیر false positive که در قسمت‌های قبل به دست آورده‌اید را با مقدار مورد انتظار ($0.2 \pm 9\%$) مقایسه کنید. آیا FP بدست آمده برای هر کدام از قسمت‌های د و ه با مقدار مورد انتظار مطابقت دارد؟ در صورتی که برای هر کدام جواب منفی باشد، دلیل این عدم تطابق را توضیح دهید.

۱- مجموعه داده bitcoin.csv در اختیار شما قرار گرفته است که نشانگر قیمت لگاریتمی بیت کوین است. مطلوب است:

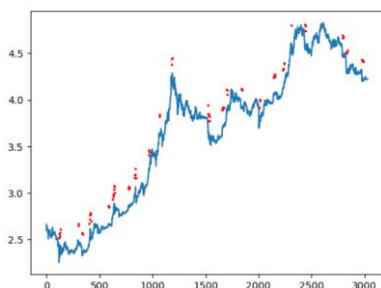
الف) میانگین قیمت را در طول زمان محاسبه کنید و همراه با قیمت بیت کوین، نمایش دهید. شکل ۱.
 ب) همانطور که در شکل ۱ مشاهده می کنید، به دلیل تغییر توزیع داده ها این نوع میانگین گرفتن مناسب نیست. این بار میانگین را طوری محاسبه کنید که روزهای اخیر از اهمیت بیشتری برخوردار باشند. بدین منظور دو رویکرد وجود دارد:
 ۱- لیستی با طول ثابت c در نظر بگیرید و همیشه c روز اخیر را در این لیست نگه دارید. این روش خیلی مناسب نیست. اولاً اینکه طول لیست را چقدر بگیریم؟ دوماً آیا تمام این c روز اخیر از وزن یکسانی برخوردار باشند؟ برای همین رویکرد دوم را نظر می گیریم.

۲- میانگین را طوری محاسبه کنید که وزن روزهای اخیر به صورت نمایی کاهش یابد و پارامتری تحت عنوان $stepsize$ را برای تعیین سرعت کاهش وزن های اخیر در نظر بگیرید در شکل ۲ محاسبه میانگین را با دو $stepsize$ متفاوت مشاهده می کنید. $stepsize$ را چه عددی در نظر بگیرید نتیجه مشابه قسمت الف خواهد شد؟

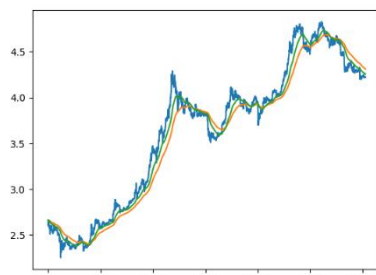
ج) در این قسمت می خواهیم با روشی متوجه شویم چه زمانی توزیع داده ها در حال عوض شدن است و به ما هشدار بدهد. یک روش ساده این است که واریانس یا انحراف معیار و همچنین میانگین آن را در طول زمان داشته باشیم و زمانی که واریانس داده های جدید تغییر محسوس با میانگین واریانس داشت به ما هشدار بدهد. این رویکرد را پیاده سازی کنید. در این قسمت باید نکاتی را مورد بررسی قرار بدهید و باهم مقایسه کنید.

۱- از چه میانگینی استفاده کنیم (وزن دار یا غیر وزن دار).
 ۲- چه زمانی هشدار بدهیم؟ با آمدن یک داده جدید که با میانگین واریانس تغییر محسوس داشت؟ یا بهتر است کمی صبر کنیم شاید داده پرتی باشد؟ چقدر صبر کنیم؟ در شکل ۳ هشدارها با رنگ قرمز نمایش داده شده اند. دقت کنید که شکل ۳ لزوماً بهترین حالت نیست.

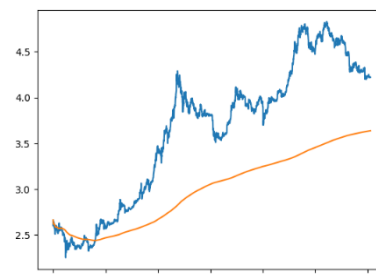
توجه: شما نمی توانید برای محاسبات خود از تمام دیتاست استفاده کنید. یعنی باید فرض کنید این یک جریان داده است. همچنین شما مجاز به ذخیره کل دیتاست برای محاسبات خود نیستید. تمام کدهایی که می نویسید باید از پیچیدگی فضایی $O(1)$ باشد. در قسمت ب مجاز به استفاده از آرایه با طول ثابت هم نخواهید بود.



شکل ۳



شکل ۲



شکل ۱