

تمرین چهارم

99243056

آرمان غفاریانیا

سوالات تحلیلی :

(1)

(1) مشکل نفرین ابعاد که تعداد $dimension$ ها خیلی زیاد می شود یا به عبارتی ویژگی های زیادی داریم که باعث می شود تشخیص نزدیکی ۲ دیتا یا فاصله آنها کار سختی شود و باعث فهم نادرست الگوریتم از نزدیکی یا فاصله دیتا ها شود و دچار $overfit$ شویم. (در کنار اینکه به نوبت حساس می شود)

(مفهوم فاصله نفا و مبهم می شود و همچنین محاسبات برای این تعداد $feature$ الگوریتم را بر هزینه و کند می کند)

راه حل :

- یک راه ساده این است که ابعاد مهم دیتای را حفظ کنیم و سایر بعدها را دور بریزیم تا حجم ابعاد کم شود.
- اما راه موثر تر استفاده از الگوریتم های است که $Dimensionality aware$ هستند.

مثل $DBSCAN$ که با اساس $density$ (تراکم نقاط داخل ناحیه) کاری کند.

می توانست $cluster$ های نزدیک را در ابعاد بالا شناسایی کند.

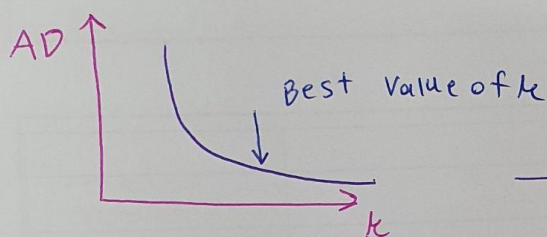
یک راه حل موثر دیگر استفاده از تجزیه ماتریس SVD است

راستاهایی که بیشترین شدت تغییرات داده در آنها اتفاق افتاده است را حفظ میکنیم و سایرین را حذف میکنیم که باعث کاهش ابعاد میشود و تقریب خوبی با خطای کمینه به ما میدهد.

(۲) یک حالت مهم انتخاب k یا تعداد خوشه ها است.

یک راه که در اسلاید مطرح شده این است که یک رنج از k ها را تست کنیم و میانگین فاصله نقاط با centroid را برای هر k بدست بیاوریم.
نقطه میانگین

طبیعتاً با افزایش تعداد خوشه ها این عدد کم می شود اما از یکجایی به بعد این کاهش چشمگیر نیست.
این نقطه = نقطه مناسب k است.



شکل اسلاید

(۳) برای انجام فشرده سازی به کمک k -means باید یک مترآیند clustering انجام دهیم

به این صورت که رنگ های که در یک رنج هستند را در یک cluster قرار دهیم و مرکز خوشه را به عنوان نماینده آن رنگ فرض کنیم.

تعداد رنگ های که در حالت فشرده ذخیره می کنیم
تعداد cluster $k \rightarrow$

برای انجام clustering، با دریافت تصویر ورودی که شامل پیکسل هاست، هر پیکسل را یک نقطه فرض می کنیم که شامل ۳ تا فیلتر B و G و R است. که به کمک آنها الگوریتم را اجرا می کنیم.

(۴) DBSCAN و زیر داده‌های موجود در شکل حالت *nesting* دارند

و احتمالاً سایر الگوریتم‌های clustering دچار مشکل می‌شوند. و دسته‌بندی نادرست می‌کنند.
مثلاً *k-means* که بر اساس فاصله اقلیدسی نقاط کار می‌کند طبیعتاً نمی‌تواند نقاط آبی پیرزگی را به درستی دسته‌بندی کند. (چون در حد اکثر فاصله هستند یکسری از نقاط)

اما الگوریتم DBSCAN که بر اساس *Density* کار می‌کند می‌تواند داده‌های *nesting* را

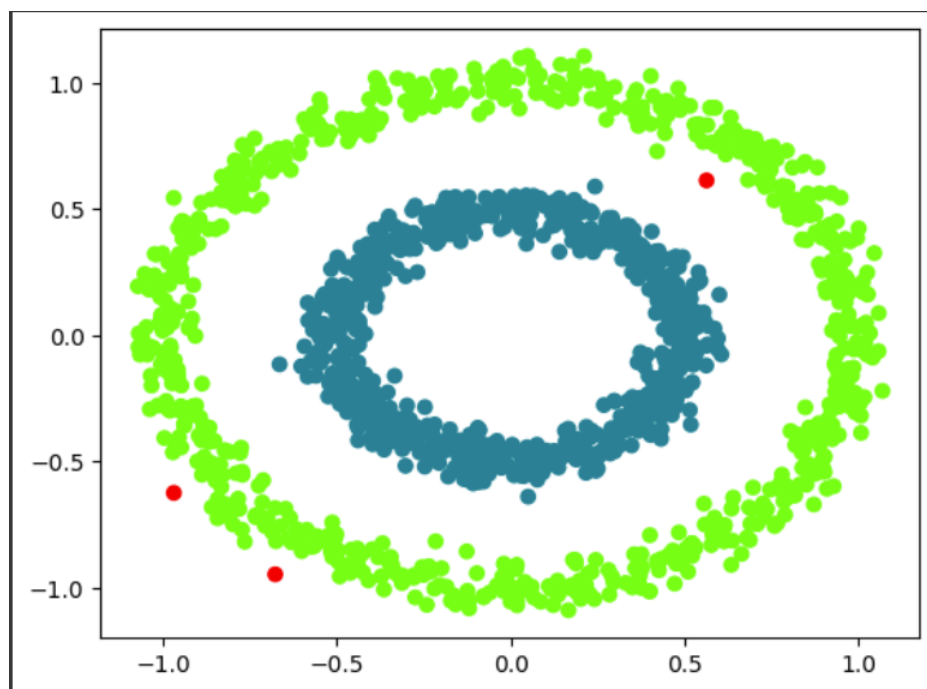
تراکم نقاط
در ناحیه

دسته‌بندی مناسب کند.

سوال کدی :

(1)

خروجی دیتاست اول :



نقاط قرمز noise هستند.

خروجی دیتاست دوم :

