

iteration 1:

$$\mu = (9, 3) \quad \mu_r = (11, 3)$$

Class 1:

از عبار حاصله استفاده می‌کنیم

$$\{(1, 0), (1, 3), (3, 0), (3, 3), (5, 0), (5, 3), (7, 0), (7, 3), (9, 0), (9, 3)\}$$

Class 2:

$$\{(11, 0), (11, 3), (13, 0), (13, 3), (15, 0), (15, 3), (17, 0), (17, 3)\}$$
iteration 2: $\mu_1 = (5, 1.5) \quad \mu_2 = (14, 1.5)$

Class 1:

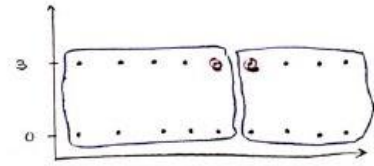
$$\{(1, 0), (1, 3), (3, 0), (3, 3), (5, 0), (5, 3), (7, 0), (7, 3), (9, 0), (9, 3)\}$$

Class 2:

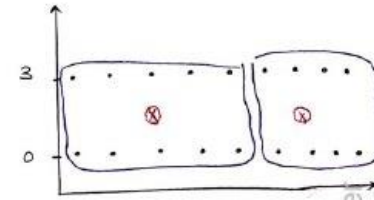
$$\{(11, 0), (11, 3), (13, 0), (13, 3), (15, 0), (15, 3), (17, 0), (17, 3), (19, 0), (19, 3)\}$$

دسته بندی تلاش اول ۲ در مرحله ۲ با مرکز تلاش
است یعنی مرکز اسناد تلاش بندی تغییر نکرده

First iteration:



Second iteration:



2.

(الف)

در زمینه تقسیم بندی مشتری، تحلیل خوشه ای استفاده از یک مدل ریاضی برای کشف گروه هایی از مشتریان مشابه بر اساس یافتن کوچکترین تغییرات در بین مشتریان در هر گروه است.

(ب) برای دسته بندی به گروه های مشابه هم و استفاده از الگویی برای نمایش ان خوشه و کاهش ابعاد

(ج) ایده اصلی پشت استفاده از خوشه بندی برای تشخیص ناهنجاری، یادگیری حالت (های) عادی در داده های موجود (آموزش) و سپس استفاده از این اطلاعات برای اشاره به غیر عادی بودن یا نبودن یک نقطه هنگام ارائه داده های جدید (تست) است.

(د) روشی برای انجام بخش بندی تصویر از تقسیم بندی پیکسلی است. در این نوع تقسیم بندی، سعی می کنیم پیکسل هایی را که در کنار هم قرار دارند، خوشه بندی کنیم. دو رویکرد برای انجام بخش بندی با خوشه بندی وجود دارد: خوشه بندی با ادغام و خوشه بندی توسط Divisive

3. هایپر پارامترهای DBSCAN تأثیر زیادی در خوشه های نهایی دارد. اگر بخواهیم خوشه های بزرگ را استخراج کنیم شعاع بالایی لازم است و از طرفی با انتخاب یک شعاع کوچک تعداد خوشه ها افزایش می یابد. از منظر دیگر اگر از قبل چگالی خوشه ها را بدانیم، می توانیم minpoint را به درستی انتخاب کنیم. Minpoint کم منجر به خوشه هایی با چگالی کم می شوند و مجموعه ای از خوشه های مترکم minpoint بالاتری نیاز دارند. در غیاب دانش قبلی قوی از ساختار خوشه ها، می توانیم از خطای مجموعه جستجو و اعتبارسنجی شبکه به عنوان روشی برای یافتن هایپر پارامترها استفاده کنیم.

4. در policy iteration ما با یک policy ثابت شروع میکنیم و برعکس در value iteration با انتخاب تابع مقدار شروع میکنیم و سپس در هر دو الگوریتم به طور مکرر تکرار میکنیم تا بهبود یابند و همگرا شوند

a)

Single-Linkage algo: The distance between two clusters, The smallest distance between two points such that:

$$D_{SL}(C_i, C_j) = \min_{a \in C_i, b \in C_j} d(a, b)$$

Complete-Linkage: The distance between two clusters, The largest distance between two points such that:

$$D_{CL}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a, b)$$

Average-Linkage: The distance between two clusters, The average distance between two points such that:

$$D_{AL}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} d(a, b)$$

The time complexity of Single-Linkage and complete-Linkage : $O(n^2)$

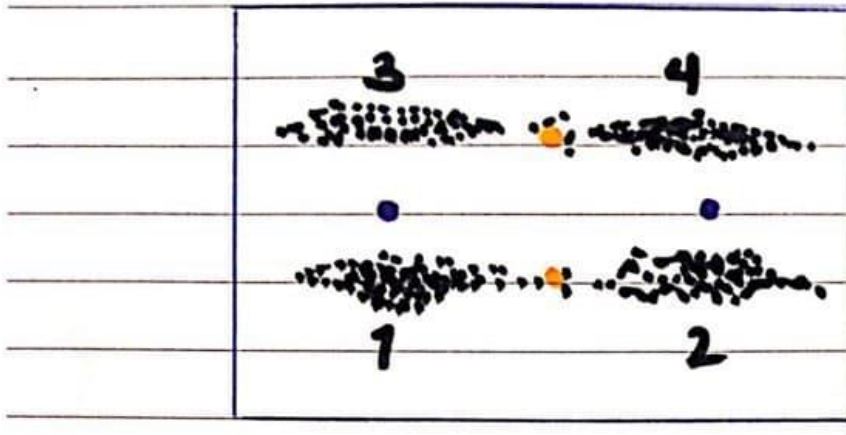
The time complexity of Average-Linkage: $O(n^2 \cdot \ln n)$

Sensitivity to outlier : Complete-Linkage < Average-Linkage < Single-Linkage

Because The complete-link will merge outliers late. Because they increase maximum distances much. And it's more robust than average-linkage and single-linkage

b)

در حالت طری ممکن است اتفاق بیفتد :



• وقتی دو نقطه ناراضی به عنوان مرکز ۲ خوشه در نظر بگیریم ← Complete Linkage $\{1, 3\}, \{2, 4\}$
 Single Linkage $\{1, 2\}, \{3, 4\}$
 Average Linkage $\{1, 2\}, \{3, 4\}$

• وقتی با بیش از سه بنی می کنیم و مرکز اول و دوم در نظر بگیریم ← Complete Linkage $\{1, 2\}, \{3, 4\}$
 Single Linkage $\{1, 3\}, \{2, 4\}$
 Average Linkage $\{1, 3\}, \{2, 4\}$

c) We can classify figure b, c with single-linkage because it decides based on the shortest distance and in figure b it definitely gives the correct answer, but in figure c we may have some error at the edge of the moon.