

۱- نحوه‌ی پیاده سازی الگوریتم به این صورت است که ابتدا ماتریس V که مراکز خوشه‌هاست مقدار دهی اولیه می‌شود. مقدار دهی اولیه‌ی آن با انتخاب چند نقطه از دیتاست به صورت تصادفی انجام می‌شود. مقدار دهی اولیه‌ی partition matrix هم به این صورت است که برای همه‌ی داده‌ها مقدار تعلق به خوشه‌ی اول ۱ و به بقیه‌ی خوشه‌ها صفر در نظر گرفته می‌شود. این کار برای این است که شرط زیر برقرار باشد:

$$\sum_{i=1}^C u_{ik} = 1 \text{ for all } k$$

سپس حلقه‌ی الگوریتم اجرا می‌شود که در آن با توجه به فرمول‌ها partition matrix به روز رسانی شده و از روی آن مراکز جدید به دست می‌آیند. شرط خاتمه همگرا شدن مراکز خوشه‌ها است. به این صورت وقتی مراکز تغییر می‌کنند، اختلاف آنها را قبلی را در نظر می‌گیریم و از مقدار مشخصی (اپسیلون) کمتر بود یعنی الگوریتم خاتمه یافته است.

۲- پارامتر m را ۲ در نظر گرفتیم. این پارامتر fuzzifier نام دارد و افزایش آن باعث می‌شود که خوشه‌بندی ما فازی تر بشود. یعنی هر داده تعلقش بین خوشه‌ها پخش تر بشود.

-۳

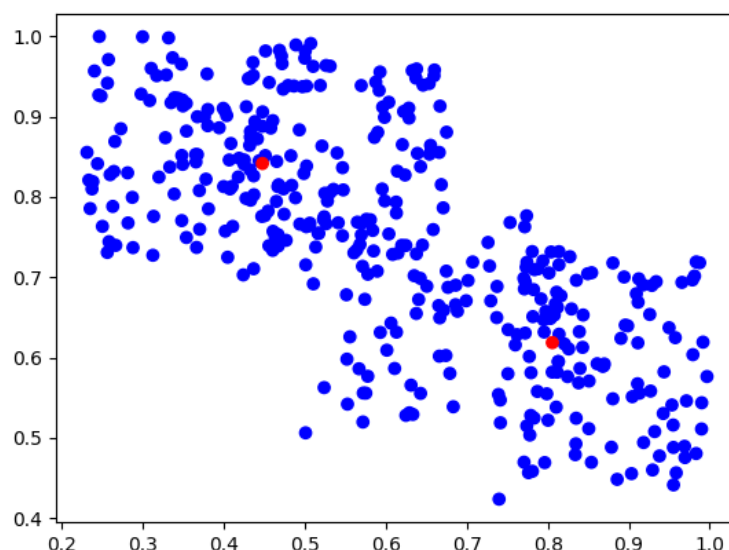
$$J(U, V) = \sum_{k=1}^n \sum_{i=1}^C (u_{ik})^m d^2(x_k, v_i)$$

رابطه‌ی تابع هزینه با تعداد خوشه‌ها این است که با کاهش تعداد خوشه‌ها مقادیر کمتری می‌گیرد تابع هزینه.

نتایج:

تعداد مراکز: ۲

هزینه: ۵۲.۱۴



تعداد مراکز: ۴

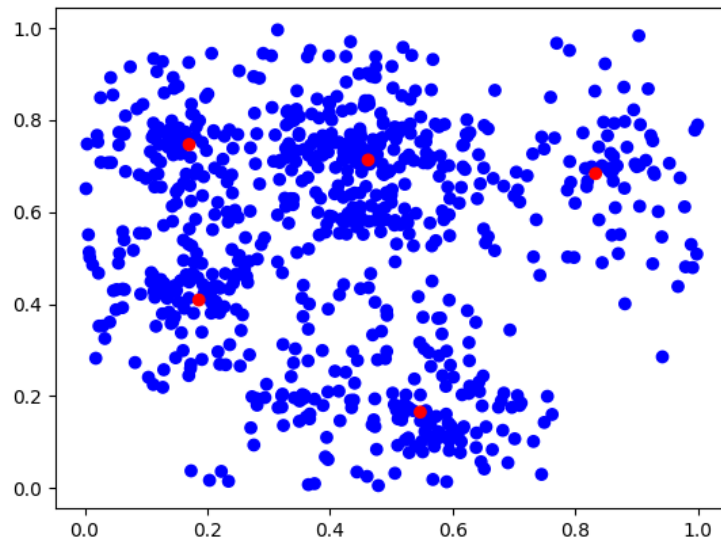
هزینه: ۶۱.۸۸

مراکز:

(0.633, 0.630, 0.632, 0.632) (0.400, 0.399, 0.399, 0.400) (0.111, 0.113, 0.112, 0.111) (0.797, 0.792, 0.792, 0.793)

تعداد مراکز: ۵

هزینه: ۹۲.۶۱



تعداد مراکز: ۳

هزینه: ۱۷.۱۱

مراکز:

(0.749, 0.750, 0.751) (0.887, 0.888, 0.885) (0.511, 0.515, 0.505)

تعداد مراکز: ۳

هزینه: ۵۹.۹۷

