وحید مواجی 83205947

Fuzzy C-means Clustering

Fuzzy c-means (FCM) یک روش خوشه بندی است که اجازه می دهد یک داده به دو یا بیشتر خوشه تعلق داشته باشد. این روش (که توسط Dunn در 1973 توسعه داده شد و توسط Bezdek در 1981 بهبود یافت) غالبا در تشخیص الگو بکار می رود. این روش بر اساس مینیمم سازی توابع هدف زیر بنا شده است:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 \quad 1 \le m < \infty$$

که m یک عدد حقیقی بزرگتر از 1 ، u_{ij} درجه عضویت x_i در خوشه j، x_i iامین داده اندازه گرفته شده d-بعدی، c_j مرکز بعد d خوشه و ||*|| یک نرم است که شباهت بین داده اندازه گرفته شده و مرکز را بیان می دارد. تفکیک فازی بوسیله یک بهینه سازی تکراری از تابع هدف بالا انجام می گردد که درجه عضویت u_{ij} و مراکز خوشه ها را اینگونه به روز می کند:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|}\right)^{\frac{2}{m-1}}} \qquad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

این تکرار وقتی متوقف می شود که $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < \varepsilon$ که ε یک معیار خاتمه بین 0و 1 است و k تعداد مراحل تکرار است. این رویه به یک مینیمم محلی یا نقطه تکیدگی J_m همگرا می شود. الگوریتم از مراحل زیر تشکیل شده است:

1. ماتریس $U = [u_{ij}]$ را مقدار دهی اولیه کن، $U^{(0)}$
2. در k مرحله: بردارهای مراکز $C^{(k)} = [c_j]$ را با $U^{(k)}$ محاسبه کن.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

3. $U^{(k)}, U^{(k+1)}$ را به روز رسانی کن.

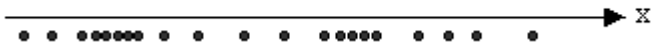
$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|}\right)^{\frac{2}{m-1}}}$$

4. اگر $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ آنگاه متوقف شو؛ در غیر اینصورت به مرحله 2 برو.

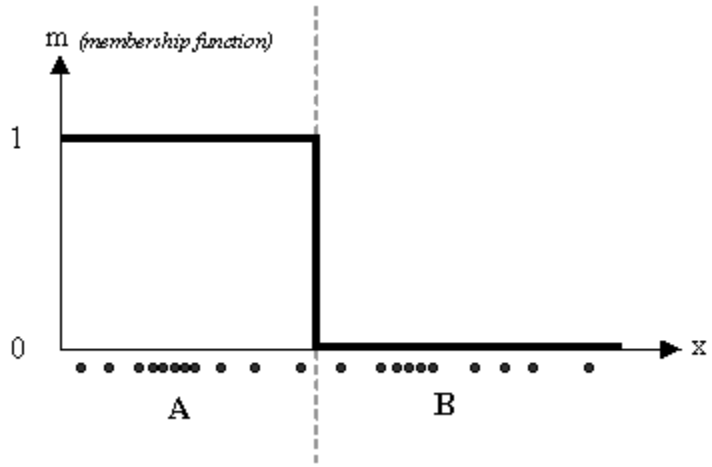
ملاحظات

همانگونه که گفته شد، داده ها به هر خوشه از طریق تابع عضویت محدود هستند که رفتار فازی این الگوریتم را بیان می دارد. برای انجام دادن آن، ما فقط نیاز داریم که یک ماتریس مناسب U بسازیم که فاکتورهای آن اعدادی بین 0و 1 هستند و درجه عضویت بین داده و مراکز خوشه ها را بیان می دارند.

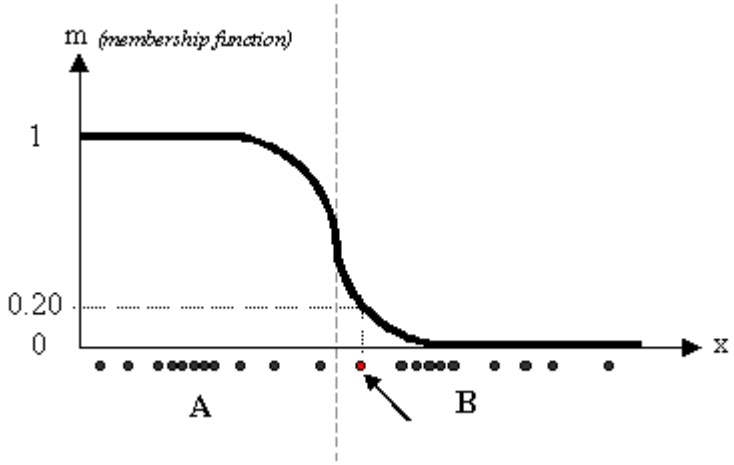
برای درک بهتر، می توانیم یک مثال ساده تک بعدی را درنظر بگیریم. با داشتن یک مجموعه داده مشخص، آنرا روی یک محور توزیع می کنیم.



با دیدن این تصویر، می توانیم دو خوشه را در مجاورت دو تجمع داده تشخیص دهیم. آنها را A, B می نامیم. با روش k-means ما هر داده را به یک مرکزثقل متسب می کردیم لذا این تابع عضویت بصورت زیر در می آمد:



در روش FCM در عوض، همان داده ها، بطور انحصاری به یک خوشه تعلق ندارند. در این حالت تابع عضویت یک خط هموارتر را دنبال می کند و بدین معنی است که هر داده می تواند به چند خوشه و با درجه عضویت های متفاوت تعلق داشته باشد.



در شکل بالا، داده ای که با فلش نشان داده شده است، بیشتر به خوشه B تعلق دارد تا A. مقدار 0.2 برای m درجه عضویت A برای این داده را نشان می دهد. حال به جای استفاده از یک ارائه گرافیکی، یک ماتریس U را معرفی می کنیم که فاکتورهای آن از توابع عضویت گرفته شده اند:

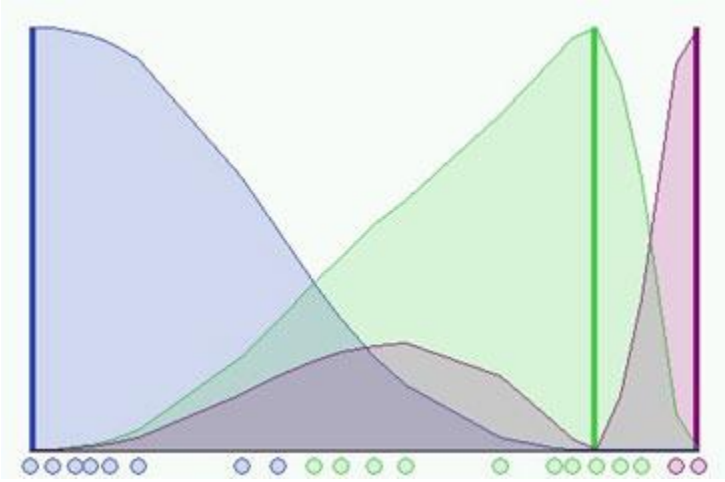
$$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

تعداد سطرها و ستونها به این بستگی دارد که چقدر داده و خوشه را در نظر داریم. بطور دقیقتر، ما C=2 ستون (خوشه) و N سطر داریم که C کل تعداد خوشه ها و N کل تعداد داده ها است. عنصر کلی u_{ij} است. در مثال بالا ما روش k-means و FCM را درنظر گرفتیم. می توانیم توجه کنیم که در حالت اول ضرایب همیشه واحد هستند. این بدین معنی است که هر داده فقط می تواند به یک خوشه تعلق داشته باشد. دیگر ویژگی ها اینگونه هستند:

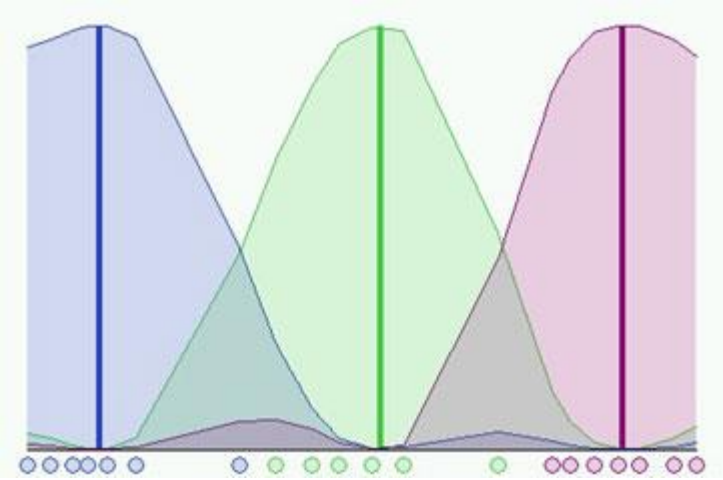
- $u_{ij} \in [0,1] \quad \forall i, j$
- $\sum_{j=1}^{C} u_{ik} = 1 \quad \forall i$
- $0 < \sum_{i=1}^{N} u_{ij} < N \quad \forall N$

یک مثال

حال یک مورد ساده از کاربرد تک بعدی FCM را درنظر می گیریم. بیست داده و سه خوشه استفاده شده اند برای مقداردهی اولیه الگوریتم و محاسبه ماتریس U. شکل های زیر مقدار عضویت را برای هر داده و هر خوشه نشان می دهند.



در شبیه سازی نشان داده شده در شکل بالا ما از یک ضریب فازی بودن m = 2 استفاده کردیم و الگوریتم را موقعی تمام میکنیم که $\max_{ij}\left\{\left|u_{ij}^{(k+1)} - u_{ij}^{(k)}\right|\right\} < 0.3$. تصویر، شرط اولیه ای را نشان می دهد که توزیع فازی به موقعیت خاص خوشه ها بستگی دارد. هیچ مرحله ای هنوز تکرار نشده است. حال می توانیم الگوریتم را اجرا کنیم تا زمانیکه به شرط خاتمه برسیم. شکل زیر شرط نهایی را که در مرحله هشتم و با m=2 و ε = 0.3 =به آن رسیدیم نشان می دهد:



آیا می شود کار را بهتر کرد؟ قطعا می توانستیم از دقت بالاتری استفاده کنیم ولی باید بهای بیشتری برا ی محاسبات می پرداختیم. در شکل بعدی می توانیم یک نتیجه بهتر را مشاهده کنیم که از شرایط اولیه مشابه قبل و ε = 0.01 استفاده کردیم ولی 37 مرحله طول کشید.