



خوشه‌بندی فازی بزدک

تابع هدف، ماتریس درجه تعلق و نقش پارامتر فازی ساز

Bezdek's Fuzzy Clustering & Objective Functional

نام و نام خانوادگی: محمدعلی پیشوا

درس: شناسایی الگو

رشته: مهندسی و علم کامپیوتر گرایش هوش مصنوعی

استاد درس: دکتر هادی صدوقی یزدی

تاریخ: ۵ آبان ۱۴۰۵

فهرست مطالب

۱	مقدمه: مسئله خوشه‌بندی چندگانه و چالش تعلق نمونه‌ها	۲
۱.۱	چالش وابستگی متقابل شاخص و مجموعه	۳
۲	رویکرد بزدک: ایده عضویت فازی و ماتریس تعلق	۳
۱.۲	ساختار ماتریس تعلق، قید نرمال‌سازی و شمارش مجهولات مسئله	۴
۳	فرمول‌بندی تابع هدف بزدک و نقش پارامتر فازی‌ساز	۶
۱.۳	بسط تابع هدف به تفکیک خوشه‌ها	۶
۲.۳	تحلیل نقش «پیچ تنظیم کاربر» (m) و نحوه دیده شدن خوشه‌ها	۷
۴	پیاده‌سازی عملی و شبیه‌سازی در پایتون	۹
۱.۴	کد کامل شبیه‌سازی و تولید نمودارها	۱۰
۵	تحلیل خروجی‌ها و نتیجه‌گیری	۱۵
۶	منابع	۱۸

۱ مقدمه: مسئله خوشه‌بندی چندگانه و چالش تعلق نمونه‌ها

در مسئله خوشه‌بندی^۱، به جای اینکه تنها با یک مجموعه داده سر و کار داشته باشیم، با c زیرمجموعه یا خوشه مختلف $(C_1, C_2, \dots, C_c)$ روبه‌رو هستیم و می‌خواهیم محاسبه شاخص^۲ را برای هر یک از این خوشه‌ها انجام دهیم. در این حالت، هدف کمینه‌سازی امید ریاضی تودرتو^۳ روی تابع زیان^۴ یعنی $E\{E\{l(e)\}\}$ است؛ که در آن یک امید ریاضی^۵ روی نمونه‌های درون هر مجموعه و امید ریاضی دیگر روی کل مجموعه‌ها تعریف می‌شود:

$$J = \sum_{j=1}^c \sum_{x_i \in C_j} (x_i - \mu_j)^2 = \sum_{x_i \in C_1} (x_i - \mu_1)^2 + \sum_{x_i \in C_2} (x_i - \mu_2)^2 + \dots + \sum_{x_i \in C_c} (x_i - \mu_c)^2 \quad (۱)$$

که در آن:

- J : تابع هزینه کل^۶ روی تمامی مجموعه‌ها
 - c : تعداد کل خوشه‌ها (مجموعه‌ها)
 - j : شمارنده خوشه‌ها که از ۱ تا c تغییر می‌کند
 - C_j : مجموعه نمونه‌های متعلق به خوشه j -ام
 - $x_i \in C_j$: شرط تعلق نمونه i -ام به مجموعه C_j در زیر سیگمای داخلی
 - μ_j : بردار شاخص (مرکز) خوشه j -ام $(\mu_1, \mu_2, \dots, \mu_c)$
- البته در حالت کلی‌تر، حتی ضرورتی ندارد که معیار خطای همه خوشه‌ها یکسان و به فرم مربع فاصله باشد؛ بلکه می‌توان تابع زیان هر خوشه را متناسب با ساختار آن متفاوت انتخاب کرد:

$$J_{\text{General}} = \sum_{x_i \in C_1} l_1(x_i, \mu_1) + \sum_{x_i \in C_2} l_2(x_i, \mu_2) + \dots + \sum_{x_i \in C_c} l_c(x_i, \mu_c) \quad (۲)$$

که در آن:

- J_{General} : تابع هزینه خوشه‌بندی با توابع زیان مجزا
- $l_1(\cdot), l_2(\cdot), \dots, l_c(\cdot)$: توابع زیان یا معیارهای فاصله اختصاصی برای خوشه‌های اول تا c -ام

¹Clustering

²Prototype / Representative

³Nested Expectation

⁴Loss Function

⁵Mathematical Expectation

⁶Total Cost Function

۱.۱ چالش وابستگی متقابل شاخص و مجموعه

برای نمایش دقیق‌تر مانع ریاضی در روابط (۱) و (۲)، می‌توان شرط زیر سیگما ($x_i \in C_j$) را با استفاده از تابع مشخصه باینری^۷ بازنویسی نمود:

$$\chi_{C_j}(x_i) = \begin{cases} 1, & x_i \in C_j \\ 0, & x_i \notin C_j \end{cases} \implies J = \sum_{j=1}^c \sum_{i=1}^n \chi_{C_j}(x_i) (x_i - \mu_j)^2 \quad (3)$$

که در آن:

• $\chi_{C_j}(x_i) \in \{0, 1\}$: تابع مشخصه گسسته با مقادیر صفر یا یک

• n : تعداد کل نمونه‌های داده

اشکال اساسی در حل مستقیم: در فرمول‌بندی بالا، خودِ مجموعه C_j برای ما نامعلوم است؛ یعنی از ابتدا نمی‌دانیم هر نمونه x_i متعلق به کدام خوشه است و خودِ این تعلق یک مجهول در مسئله است. اگر می‌دانستیم کدام نمونه‌های x_i در C_1 ، کدام در C_2 و کدام در C_c قرار دارند، به‌سادگی درون هر مجموعه شاخص‌های μ_1 تا μ_c را به صورت مستقل محاسبه می‌کردیم؛ اما نامشخص بودن تعلق نمونه‌ها مانع از حل مستقیم معادله می‌شود.

۲ رویکرد بزدک: ایده عضویت فازی و ماتریس تعلق

پروفسور جیمز بزدک^۸ برای برطرف کردن مشکل نامشخص بودن تعلق نمونه‌ها به خوشه‌ها ($x_i \in C_j$)، دیدگاه عضویت فازی^۹ را جایگزین افراز قطعی^{۱۰} کرد:

ایده بنیادین بزدک: «همه نمونه‌ها متعلق به همه خوشه‌ها هستند، اما هر نمونه به هر خوشه یک درجه تعلق^۹ مشخص و مجهول دارد.»

^aMembership Degree

در این رویکرد، به جای اینکه بپرسیم آیا نمونه x_i در مجموعه C_1 یا C_2 قرار دارد یا خیر، فرض می‌کنیم هر نمونه به تک‌تک خوشه‌ها میزانی از تعلق را داراست که مقدار آن برای ما مجهول است:

- نمونه اول به خوشه اول تعلق u_{11} و تا خوشه c —ام تعلق u_{1c} دارد.
- نمونه دوم به خوشه اول تعلق u_{21} و تا خوشه c —ام تعلق u_{2c} دارد.

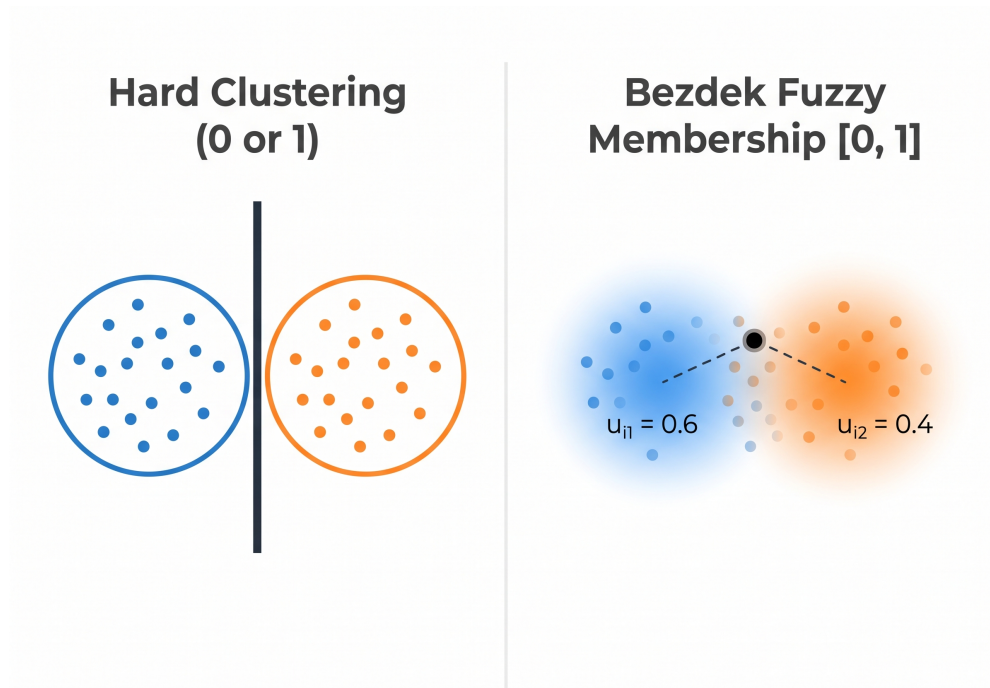
⁷Binary Characteristic Function

⁸James C. Bezdek

⁹Fuzzy Membership

¹⁰Hard Partitioning

- نمونه n - ام به خوشه اول تعلق u_{n1} و تا خوشه c - ام تعلق u_{nc} دارد.



شکل ۱: مقایسه مفهومی افراز قطعی مبتنی بر مجموعه‌های مجزا با ایده عضویت فازی بزدک.

منبع تصویر: تولیدشده توسط هوش مصنوعی Nano Banana 2.

۱.۲ ساختار ماتریس تعلق، قید نرمال‌سازی و شمارش مجهولات مسئله

با این تغییر دیدگاه، شرط $x_i \in C_j$ از زیر سیگما حذف شده و سیگمای داخلی روی تمامی نمونه‌ها از $i = 1$ تا n بسته می‌شود. در مقابل، یک ماتریس افراز فازی^{۱۱} با ابعاد $n \times c$ از متغیرهای مجهول به مسئله اضافه می‌گردد:

$$U_{n \times c} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1c} \\ u_{21} & u_{22} & \cdots & u_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nc} \end{bmatrix} \in [0, 1]^{n \times c}, \quad \mu = \{\mu_1, \mu_2, \dots, \mu_c\}, \quad \mu_j \in \mathbb{R}^d \quad (4)$$

که در آن:

¹¹Fuzzy Partition Matrix

- $U_{n \times c}$: ماتریس درجات تعلق نمونه‌ها به خوشه‌ها (شامل n سطر و c ستون)
- u_{ij} : درجه تعلق نمونه i - ام به خوشه j - ام در بازه پیوسته $[0, 1]$
- μ_j : بردار شاخص خوشه j - ام در فضای ویژگی d - بعدی ($\mathbb{R}^d$)

قید نرمال‌سازی درجات تعلق (شرط افراز واحد بزدک)

در فرمول‌بندی بزدک، درایه‌های ماتریس $U_{n \times c}$ نمی‌توانند مقادیر دلخواه و رها داشته باشند؛ بلکه برای تشکیل یک افراز فازی معتبر، مجموع درجات تعلق هر نمونه x_i به تمامی c خوشه باید دقیقاً برابر با یک باشد (جمع سطری ماتریس U برابر با ۱):

$$\sum_{j=1}^c u_{ij} = u_{i1} + u_{i2} + \dots + u_{ic} = 1, \quad \forall i \in \{1, 2, \dots, n\} \quad (5)$$

همچنین برای اطمینان از اینکه هیچ خوشه‌ای کاملاً تهی نماند و هیچ خوشه‌ای به تنهایی تمامی داده‌ها را تصاحب نکند، قید عدم تبهگنی ستونی زیر نیز برقرار است:

$$0 < \sum_{i=1}^n u_{ij} < n, \quad \forall j \in \{1, 2, \dots, c\} \quad (6)$$

چرا قید $\sum_{j=1}^c u_{ij} = 1$ از نظر ریاضی و مفهومی حیاتی است؟

۱. جلوگیری از جواب بدیهی و تبهگن صفر (Preventing the Trivial Zero Solution):

از آنجا که فواصل در تابع هدف همواره نامنفی هستند ($d(x_i, \mu_j) \geq 0$)، اگر قید نرمال‌سازی رابطه (۵) وجود نداشته باشد، الگوریتم بهینه‌سازی برای صفر کردن خطا، تمامی درجات تعلق را برابر با صفر قرار می‌دهد ($J = 0 \implies u_{ij} = 0$) و مسئله بدون یافتن هیچ خوشه‌ای فرو می‌پاشد!

۲. توزیع کامل سهم حضور نمونه (Simplex Constraint): این قید بیان می‌کند که هر نمونه

x_i مجموعاً یک واحد کامل (۱۰۰٪) ظرفیت تعلق دارد که باید به طور کامل بین خوشه‌ها توزیع شود (برای مثال در حالت دو خوشه‌ای، اگر $u_{i1} = 0/6$ باشد، حتماً $u_{i2} = 1 - 0/6 = 0/4$ خواهد بود).

بنابراین در این مسئله بهینه‌سازی^{۱۲} مقید، تعداد کل متغیرهای مجهول به صورت زیر محاسبه می‌شود:

$$N_{\text{unknowns}} = \underbrace{(n \times c)}_{U \text{ مجهولات تعلق}} + \underbrace{(c \times d)}_{\mu \text{ مجهولات شاخص‌ها}} = c(n + d) \quad (7)$$

¹²Optimization

که در آن:

- N_{unknowns} : تعداد کل متغیرهای اسکالر مجهول در مسئله (با احتساب n معادله قید خطی رابطه (۵)، تعداد درجات آزادی مستقل ماتریس تعلق برابر با $n(c-1)$ خواهد بود)
- $n \times c$: تعداد ضرایب تعلق مجهول اضافه شده به ازای n نمونه و c خوشه
- $c \times d$: تعداد مؤلفه های مجهول مربوط به c بردار شاخص (μ_1 تا μ_c)

۳ فرمول بندی تابع هدف بزدک و نقش پارامتر فازی ساز

در فرمول بندی بزدک، عبارت داخلی هر خوشه دقیقاً همان معیار فاصله قبلی است که تنها در ضریب تعلق توان دار ضرب می شود. علاوه بر این، برای اینکه کاربر در تنظیم رفتار الگوریتم آزادی عمل داشته باشد، ضریب تعلق به توان پارامتر m می رسد و مسئله بهینه سازی مقید با تابع هدف^{۱۳} زیر حاصل می شود:

$$\min_{U, \mu} J(U, \mu) = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m d(x_i, \mu_j) \quad \text{به شرط آنکه} \quad \sum_{j=1}^c u_{ij} = 1 \quad (\forall i = 1, \dots, n) \quad (۸)$$

که در آن:

- $J(U, \mu)$: تابع هدف خوشه بندی فازی بزدک بر حسب دو دسته مجهول U و μ
- j : شمارنده خوشه ها ($j = 1, \dots, c$)
- i : شمارنده تمامی نمونه های داده ($i = 1, \dots, n$)
- u_{ij} : درجه تعلق نمونه i - ام به خوشه j - ام
- m : پارامتر تعریفی کاربر^{۱۴} یا ضریب فازی ساز^{۱۵} ($m > 1$)
- $d(x_i, \mu_j)$: معیار فاصله^{۱۶} یا عدم شباهت بین نمونه x_i و شاخص خوشه μ_j

۱.۳ بسط تابع هدف به تفکیک خوشه ها

اگر رابطه (۸) را روی خوشه های اول تا c - ام باز کنیم، ساختار مجموع خطاهای وزن دار به شکل زیر نمایش داده می شود:

¹³Objective Function

¹⁴User-Defined Parameter

¹⁵Fuzzifier / Weighting Exponent

¹⁶Distance Metric

$$J(U, \mu) = \sum_{i=1}^n u_{i1}^m d(x_i, \mu_1) + \sum_{i=1}^n u_{i2}^m d(x_i, \mu_2) + \dots + \sum_{i=1}^n u_{ic}^m d(x_i, \mu_c) \quad (9)$$

که در آن:

- $\sum_{i=1}^n u_{i1}^m d(x_i, \mu_1)$: مجموع خطای وزن دار تمامی n نمونه نسبت به شاخص خوشه اول (μ_1)
 - $u_{i1}, u_{i2}, \dots, u_{ic}$: ضرایب تعلق نمونه i -ام به خوشه‌های اول، دوم تا c -ام که در شرط $\sum_{j=1}^c u_{ij} = 1$ صدق می‌کنند
- در حالت استاندارد که معیار فاصله همان مجذور فاصله اقلیدسی است، جمله مربوط به خوشه اول به صورت زیر نوشته می‌شود:

$$J_1(u_{i1}, \mu_1) = \sum_{i=1}^n u_{i1}^m (x_i - \mu_1)^2 \quad (10)$$

که در آن:

- $J_1(u_{i1}, \mu_1)$: سهم خوشه اول در تابع هدف کل
- $(x_i - \mu_1)^2$: مربع فاصله نمونه x_i از شاخص خوشه اول (μ_1)

۲.۳ تحلیل نقش «پیچ تنظیم کاربر» (m) و نحوه دیده شدن خوشه‌ها

توان m در عبارت u_{ij}^m یک پارامتر اضافی است که به مسئله افزوده شده تا محل اعمال نظر کاربر باشد (مانند پیچ تنظیم موج در رادیوهای قدیمی که به کاربر اجازه می‌دهد پارامتر را به صورت دستی یا خودکار تنظیم کند). این پارامتر یک فیلتر میرایی غیرخطی^{۱۷} روی درجات تعلق ایجاد می‌کند:

$$w_{ij}(m) = u_{ij}^m \quad (u_{ij} \in [0, 1], m > 1) \quad (11)$$

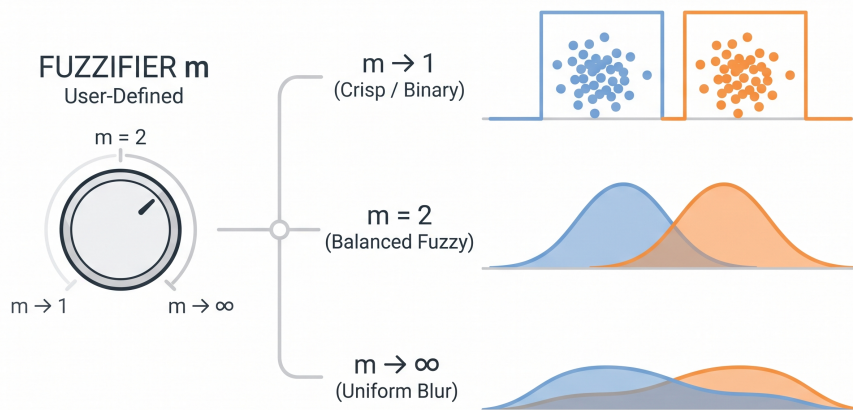
که در آن:

- $w_{ij}(m)$: وزن مؤثر نمونه i -ام در محاسبه خطای خوشه j -ام پس از اعمال توان m
- با بهینه‌سازی تابع هدف بزدک تحت قید $\sum_{j=1}^c u_{ij} = 1$ به روش ضرایب لاگرانژ، رابطه تحلیلی درجات تعلق بر حسب فاصله‌ها و پارامتر m به صورت زیر به دست می‌آید:

¹⁷Non-Linear Damping Filter

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - \mu_j\|}{\|x_i - \mu_k\|} \right)^{\frac{2}{m-1}}} \implies \lim_{m \rightarrow \infty} u_{ij} = \frac{1}{c} \quad (12)$$

رابطه (۱۲) نشان می‌دهد که وقتی $m \rightarrow 1^+$ میل کند، توان کسر به بی‌نهایت رفته و درجات تعلق به مقادیر باینری $\{0, 1\}$ تبدیل می‌شوند؛ اما وقتی $m \rightarrow \infty$ میل کند، توان $\frac{2}{m-1}$ به سمت صفر رفته و در نتیجه پایه کسری به توان صفر برابر با یک می‌شود ($(d_{ij}/d_{ik})^0 = 1$). بنابراین در حد بی‌نهایت، درجه تعلق تمامی نمونه‌ها به تمامی خوشه‌ها بدون توجه به فاصله مکانی، به مقدار ثابت و یکنواخت $\frac{1}{c}$ (که در مسئله دو خوشه‌ای حاضر برابر با $0.5 = \frac{1}{2}$ است) میل می‌کند.



شکل ۲: تأثیر مفهومی پیچ تنظیم کاربر (m) بر نحوه شکل‌گیری توابع عضویت و نرمی مرز خوشه‌ها.

منبع تصویر: تولیدشده توسط هوش مصنوعی Nano Banana 2.

نحوه دیده شدن داده‌ها توسط هر خوشه با حضور ماتریس U

پیش از معرفی ضریب u_{ij} توسط بزدک، راهی برای تفکیک ریاضی نمونه‌های هر خوشه داخل سیگمای $\sum_{i=1}^n$ وجود نداشت؛ اما اکنون با حضور u_{i1}^m در رابطه (۱۰)، خوشه اول عملاً داده‌های کل مجموعه را به شکل زیر می‌بیند:

$$u_{i1} \approx \begin{cases} 1, & \text{برای داده‌هایی که مربوط به خوشه اول هستند} \\ 0, & \text{برای داده‌هایی که مربوط به خوشه دوم و سایر خوشه‌ها هستند} \\ (0, 1), & \text{برای داده‌هایی که در ناحیه مرزی بین خوشه‌ها قرار دارند} \end{cases} \quad (13)$$

که در آن:

- $u_{i1} \approx 1$: حفظ کامل اثر نمونه‌های متعلق به خوشه اول در محاسبه خطای μ_1
 - $u_{i1} \approx 0$: حذف خودکار اثر نمونه‌های متعلق به سایر خوشه‌ها در جمله خوشه اول
- جدول ۱: مقایسه ساختاری خوشه‌بندی قطعی و خوشه‌بندی فازی بزدک

ویژگی	فرمول تابع هدف و قید	نحوه برخورد با تعلق نمونه‌ها	مجهولات مسئله
خوشه‌بندی قطعی (Hard)	$\sum_{j=1}^c \sum_{x_i \in C_j} d(x_i, \mu_j)$	تعلق گسسته و باینری $(\{0, 1\})$ نیازمند دانستن از پیش مجموعه C_j	مجموعه‌های نامعلوم μ_j شاخص $c + C_j$
خوشه‌بندی فازی بزدک (Fuzzy)	$\sum_{j=1}^c \sum_{i=1}^n u_{ij}^m d(x_i, \mu_j)$ به شرط $\sum_{j=1}^c u_{ij} = 1$	تعلق پیوسته $(u_{ij} \in [0, 1])$ سیگما روی کل نمونه‌ها با پیچ تنظیم m	$n \times c$ درایه ماتریس μ_j شاخص $c + U$

۴ پیاده‌سازی عملی و شبیه‌سازی در پایتون

در این بخش، برای مشاهده عینی و دقیق مفاهیم مطرح‌شده، یک شبیه‌سازی در زبان پایتون^{۱۸} طراحی کرده‌ایم که موارد زیر را پیاده‌سازی و مصورسازی می‌کند:

۱. نمایش ساختار ماتریس درجات تعلق $U_{n \times c}$ و بررسی برقراری قید $\sum_{j=1}^c u_{ij} = 1$ برای نمونه‌های واقع در هسته خوشه اول، مرز میانی و هسته خوشه دوم.

۲. مقایسه بصری افزایش افراز قطعی (صفر و یک) با درجات تعلق پیوسته بزدک (u_{ij}) .

۳. مصورسازی اینکه خوشه اول از دریچه ضریب u_{ij}^m چگونه نمونه‌های خودی را با وزن $1 \approx$ می‌بیند و نمونه‌های خوشه دوم را با وزن $0 \approx$ نادیده می‌گیرد، به همراه نمودار فیلتر تضعیف غیرخطی $(w = u_{ij}^m)$.

۴. بررسی جامع اثر «پیچ تنظیم کاربر (m) » به ازای مقادیر $\{1/1, 2, 3, \dots, 20\}$ و همچنین

¹⁸Python

نمایش منحنی $m = 150$ به عنوان تقریب عددی حالت حدی $m \rightarrow \infty$ جهت مشاهده میل کردن درجات تعلق به $\frac{1}{c} = 0.5$.

۱.۴ کد کامل شبیه‌سازی و تولید نمودارها

```
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path

# 1. Setup output directory (images folder in project root)
IMAGE_DIR = Path("images")
IMAGE_DIR.mkdir(parents=True, exist_ok=True)
np.random.seed(42)

# =====
# PART 1: Generate Synthetic 2-Cluster Dataset + Boundary Sample
# =====
cluster1 = np.random.normal(loc=[2.0, 3.0], scale=0.55, size=(35, 2))
cluster2 = np.random.normal(loc=[6.0, 3.0], scale=0.55, size=(35, 2))
special_points = np.array([
    [2.0, 3.0], # Sample A: Core of Cluster 1
    [4.0, 3.0], # Sample B: Boundary Point Between Clusters
    [6.0, 3.0] # Sample C: Core of Cluster 2
])
X = np.vstack([cluster1, cluster2, special_points])

# =====
# PART 2: Bezdek's Fuzzy Objective Optimization
# =====
def optimize_bezdek_fcm(X, c=2, m=2.0, max_iter=100, tol=1e-6):
    n = X.shape[0]
    # Dirichlet initialization guarantees  $\sum_{j=1}^c u_{ij} = 1$ 
    U = np.random.dirichlet(np.ones(c), size=n)
    for _ in range(max_iter):
        um = U ** m
        centers = (um.T @ X) / np.sum(um, axis=0)[:, None]
        dists = np.linalg.norm(X[:, None, :] - centers[None, :, :], axis=2)
        dists = np.fmax(dists, 1e-10)
        power = 2.0 / (m - 1.0)
        U_new = 1.0 / np.sum((dists[:, :, None] / dists[:, None, :]) ** power, axis=2)
        if np.linalg.norm(U_new - U) < tol:
```

```

        break
    U = U_new

    if centers[0, 0] > centers[1, 0]:
        centers = centers[::-1]
        U = U[:, ::-1]

    return centers, U

centers, U = optimize_bezdek_fcm(X, c=2, m=2.0)

# Print Bezdek's unknown variables count & sample rows of U_(n x c)
n_samples, d_features = X.shape
c_clusters = 2
print(f"Total Samples (n): {n_samples}, Clusters (c): {c_clusters}, Dimension (d):
      {d_features}")
print(f"Unknowns in Matrix U (n x c): {n_samples * c_clusters}")
print(f"Unknowns in Prototypes mu:      {c_clusters * d_features}\n")

print("=== SAMPLE ROWS OF BEZDEK PARTITION MATRIX U_(n x c) ===")
sample_names = ["Sample A (Core C1 [2,3])", "Sample B (Boundary [4,3])", "Sample C
                (Core C2 [6,3])"]
for idx, name in zip([-3, -2, -1], sample_names):
    u1, u2 = U[idx]
    print(f"{name:26s} -> u_i1 = {u1:.4f}, u_i2 = {u2:.4f}, Sum = {u1+u2:.2f} |
          Weight u_i1^m = {u1**2:.4f}")

# =====
# FIGURE 1: Hard Partitioning vs. Bezdek Fuzzy Membership & Matrix U
# =====
fig, axes = plt.subplots(1, 3, figsize=(15, 4.4))

# Panel 1: Visual Heatmap of Representative Rows in U_(n x c)
u_subset = U[[-3, -2, -1], :]
im = axes[0].imshow(u_subset, cmap="YlGnBu", vmin=0, vmax=1, aspect="auto")
axes[0].set_xticks([0, 1])
axes[0].set_xticklabels(["Cluster 1 (u_i1)", "Cluster 2 (u_i2)"], fontweight="bold"
                        )
axes[0].set_yticks([0, 1, 2])
axes[0].set_yticklabels(["Core C1 [2,3]", "Boundary [4,3]", "Core C2 [6,3]"],
                        fontweight="bold")
for r in range(3):
    for col in range(2):

```

```

        val = u_subset[r, col]
        axes[0].text(col, r, f"{val:.2f}", ha="center", va="center",
                    color="white" if val > 0.5 else "black", fontsize=11,
                    fontweight="bold")
axes[0].set_title("1. Entries of Matrix U(n x c) [Row Sum = 1]")

# Panel 2: Hard Partitioning (Binary 0 or 1)
hard_labels = np.argmax(U, axis=1)
axes[1].scatter(X[:, 0], X[:, 1], c=hard_labels, cmap="coolwarm", s=50, edgecolors=
                "k")
axes[1].scatter(centers[:, 0], centers[:, 1], c="yellow", marker="D", s=130,
                edgecolors="black")
axes[1].annotate("Forced Binary (0 or 1)!", xy=(4.0, 3.0), xytext=(2.6, 4.3),
                arrowprops=dict(arrowstyle="->", lw=1.5, color="black"),
                fontweight="bold")
axes[1].set_title("2. Hard Partitioning (Binary 0/1)")
axes[1].grid(True, alpha=0.3)

# Panel 3: Bezdek's Fuzzy Membership (Continuous uij)
sc = axes[2].scatter(X[:, 0], X[:, 1], c=U[:, 0], cmap="coolwarm", s=55, edgecolors
                    ="k", vmin=0, vmax=1)
axes[2].scatter(centers[:, 0], centers[:, 1], c="yellow", marker="D", s=130,
                edgecolors="black")
axes[2].annotate(f"ui1 = {U[-2,0]:.2f}, ui2 = {U[-2,1]:.2f}\n(ui1 + ui2 = 1.00)",
                xy=(4.0, 3.0), xytext=(2.5, 4.15),
                arrowprops=dict(arrowstyle="->", lw=1.5, color="darkblue"),
                fontweight="bold", fontsize=9)
axes[2].set_title("3. Bezdek Fuzzy Membership (uij)")
axes[2].grid(True, alpha=0.3)
plt.colorbar(sc, ax=axes[2], label="Membership Degree ui1")

plt.tight_layout()
plt.savefig(IMAGE_DIR / "comparison_three_methods.png", dpi=180)
plt.close()

# =====
# FIGURE 2: Cluster Perception Filter & Non-Linear Damping Filter (2 Panels)
# =====
fig, axes = plt.subplots(1, 2, figsize=(11.5, 4.4))

# Panel 1: How Cluster 1 "Sees" the Data (Weighted by ui1m)
weights_c1 = U[:, 0] ** 2.0

```

```

axes[0].scatter(X[:, 0], X[:, 1], c="royalblue", alpha=np.clip(weights_c1, 0.06,
    1.0),
                s=30 + 90 * weights_c1, edgecolors="black", linewidth=0.6)
axes[0].scatter(*centers[0], c="gold", marker="*", s=260, edgecolors="black", label
    ="Center mu_1")
axes[0].annotate("u_i1^m ~ 1.0\n(Fully Seen)", xy=(2.0, 3.0), xytext=(0.8, 4.3),
    arrowprops=dict(arrowstyle="->", lw=1.3), fontsize=9, fontweight="
    bold")
axes[0].annotate("u_i1^m ~ 0.0\n(Ignored!)", xy=(6.0, 3.0), xytext=(4.8, 4.3),
    arrowprops=dict(arrowstyle="->", lw=1.3, color="gray"), fontsize
    =9, color="dimgray")
axes[0].set_title("1. Data Seen by Cluster 1 (Weight u_i1^m)")
axes[0].legend(loc="lower left")
axes[0].grid(True, alpha=0.3)

# Panel 2: Non-Linear Damping Filter w = u^m
u_vals = np.linspace(0, 1, 200)
for m_val, col, ls in [(1.0, "gray", "--"), (2.0, "navy", "-"), (4.0, "seagreen", "
    :")]:
    axes[1].plot(u_vals, u_vals ** m_val, color=col, linestyle=ls, lw=2.2, label=f"
        m = {m_val}")
axes[1].scatter([0.2, 0.9], [0.04, 0.81], color="crimson", zorder=5)
axes[1].annotate("0.2^2 = 0.04\n(Suppressed)", xy=(0.2, 0.04), xytext=(0.28, 0.18),
    arrowprops=dict(arrowstyle="->", color="crimson"), fontsize=8.5)
axes[1].set_title("2. Non-Linear Damping Filter (w = u^m)")
axes[1].set_xlabel("Raw Membership (u_ij)")
axes[1].set_ylabel("Effective Weight in Loss (u_ij^m)")
axes[1].legend()
axes[1].grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig(IMAGE_DIR / "cluster_perception_and_m.png", dpi=180)
plt.close()

# =====
# FIGURE 3: Spatial Transition vs. Knob (m) for m in [1.1, ..., 20] & m=150 (Approx
    . inf)
# =====
fig, axes = plt.subplots(1, 2, figsize=(15.0, 5.2))
x_line = np.linspace(0, 8, 500)
c1, c2 = 2.0, 6.0
d1 = np.abs(x_line - c1) + 1e-9
d2 = np.abs(x_line - c2) + 1e-9

```

```

# Panel 1: Low-to-Moderate Fuzziness Regime (m = 1.1, 2, 3, ..., 10)
m_regime1 = [1.1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
colors1 = plt.cm.tab10(np.linspace(0, 1, len(m_regime1)))

for idx, m_val in enumerate(m_regime1):
    u1 = 1.0 / (1.0 + (d1 / d2) ** (2.0 / (m_val - 1.0)))
    ls = "--" if m_val == 1.1 else "-" if m_val in [2, 3] else "."
    lw = 2.4 if m_val in [1.1, 2, 3] else 1.5
    axes[0].plot(x_line, u1, linestyle=ls, color=colors1[idx], lw=lw, label=f"m = {
        m_val}")

axes[0].axvline(c1, color="gray", alpha=0.45, linestyle=":")
axes[0].axvline(c2, color="gray", alpha=0.45, linestyle=":")
axes[0].axhline(0.5, color="black", alpha=0.35, linestyle="--", lw=1.0)
axes[0].scatter([4.0], [0.5], color="black", s=45, zorder=6)
axes[0].annotate("Pivot (4.0, 0.5)", xy=(4.0, 0.5), xytext=(4.35, 0.58),
    arrowprops=dict(arrowstyle=">", lw=1.1, color="black"), fontsize
    =8.5, fontweight="bold")
axes[0].set_title("1. Structural Transition Regime (m = 1.1 to 10)")
axes[0].set_xlabel("Sample Position (x)")
axes[0].set_ylabel("Membership Degree ui1(x)")
axes[0].set_ylim(-0.03, 1.05)
axes[0].legend(ncol=2, fontsize=8.5, loc="upper right", framealpha=0.92)
axes[0].grid(True, alpha=0.3)

# Panel 2: High Fuzziness Regime (m = 11, ..., 20) + Numerical Approximation of m
# -> inf (m = 150)
m_regime2 = [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
colors2 = plt.cm.viridis(np.linspace(0.05, 0.90, len(m_regime2)))

for idx, m_val in enumerate(m_regime2):
    u1 = 1.0 / (1.0 + (d1 / d2) ** (2.0 / (m_val - 1.0)))
    ls = "-" if m_val in [11, 15, 20] else "--"
    lw = 2.1 if m_val in [11, 20] else 1.35
    axes[1].plot(x_line, u1, linestyle=ls, color=colors2[idx], lw=lw, label=f"m = {
        m_val}")

# Approximate curve for m -> infinity using m = 150 (clearly visible without
# overlapping horizontal line)
m_approx_inf = 150.0
u1_approx_inf = 1.0 / (1.0 + (d1 / d2) ** (2.0 / (m_approx_inf - 1.0)))

```

```

axes[1].plot(x_line, u1_approx_inf, linestyle="-.", color="crimson", lw=2.5, zorder
            =6,
            label="m = 150 (Approx. m -> inf)")

axes[1].axvline(c1, color="gray", alpha=0.45, linestyle=":")
axes[1].axvline(c2, color="gray", alpha=0.45, linestyle=":")
axes[1].scatter([4.0], [0.5], color="crimson", s=50, zorder=7)
axes[1].annotate("Approx. m -> inf (m=150)\n Approaches uil = 1/c = 0.5",
                xy=(1.1, float(1.0 / (1.0 + (np.abs(1.1 - c1) / np.abs(1.1 - c2))
                ** (2.0 / (m_approx_inf - 1.0))))),
                xytext=(0.25, 0.18),
                arrowprops=dict(arrowstyle="->", lw=1.4, color="crimson"),
                fontsize=8.5, color="crimson", fontweight="bold")

axes[1].set_title("2. Saturation Regime (m = 11 to 20) & Approx. Limit (m = 150)")
axes[1].set_xlabel("Sample Position (x)")
axes[1].set_ylabel("Membership Degree uil(x)")
axes[1].set_ylim(-0.03, 1.05)
axes[1].legend(ncol=2, fontsize=8.0, loc="upper right", framealpha=0.94)
axes[1].grid(True, alpha=0.3)

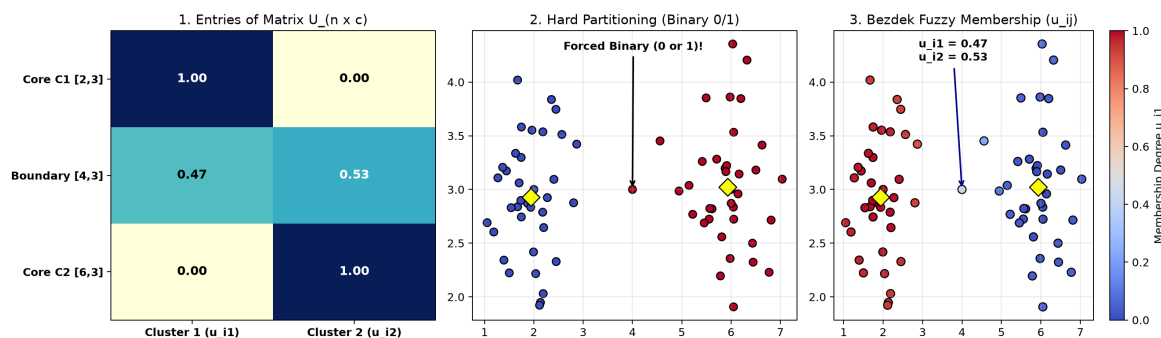
plt.tight_layout()
plt.savefig(IMAGE_DIR / "spatial_transition_m_1_to_20.png", dpi=180)
plt.close()

```

شبیه‌سازی ماتریس تعلق بزدک، قید نرمال‌سازی و تحلیل پیچ تنظیم کاربر از $m=1,1$ تا تقریب بی‌نهایت ($m=150$)

۵ تحلیل خروجی‌ها و نتیجه‌گیری

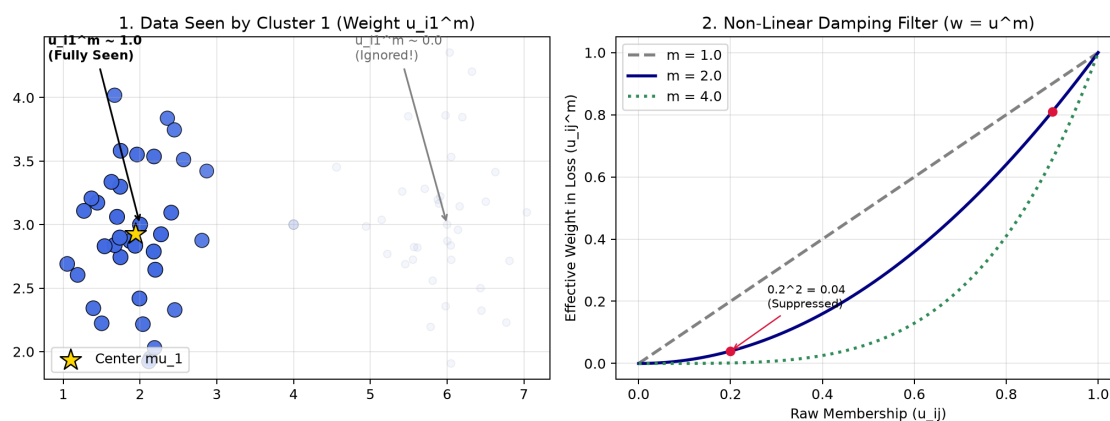
با اجرای کد شبیه‌سازی، خروجی‌های عددی و بصری زیر حاصل می‌شوند که مکانیزم دقیق فرمول‌بندی بزدک را به تصویر می‌کشند:



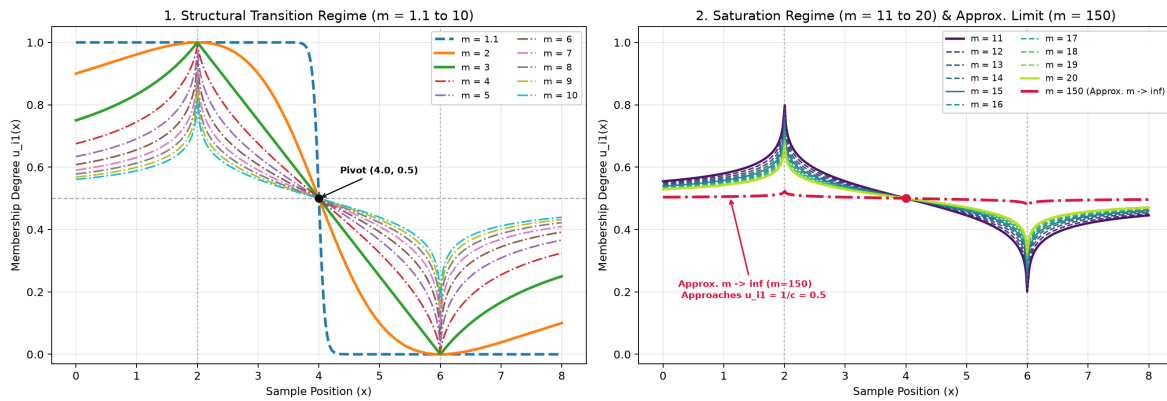
شکل ۳: مقایسه خوشه‌بندی قطعی (منطق صفر و یک) با عضویت فازی پیوسته بزرگ (u_{ij}) تحت قید $\sum_{j=1}^c u_{ij} = 1$

جدول ۲: مقادیر درجات تعلق استخراج‌شده از ماتریس $U_{n \times c}$ و بررسی قید نرمال‌سازی برای سه نمونه شاخص ($m = 2$)

موقعیت نمونه در فضای داده	تعلق به خوشه ۱ (u_{i1})	تعلق به خوشه ۲ (u_{i2})	مجموع سطری ($u_{i1} + u_{i2}$)	وزن مؤثر در خوشه ۱ (u_{i1}^m)
نمونه هسته خوشه اول ([۲/۰, ۳/۰])	$\approx 1/00$	$\approx 0/00$	$1/00$	$1/00 \approx$ کاملاً دیده می‌شود
نمونه مرزی میانی ([۴/۰, ۳/۰])	$0/47$	$0/53$	$1/00$	$0/22$ (نسبی مشارکت)
نمونه هسته خوشه دوم ([۶/۰, ۳/۰])	$\approx 0/00$	$\approx 1/00$	$1/00$	$0/00 \approx$ نادیده گرفته می‌شود



شکل ۴: (چپ) نحوه دیده شدن داده‌ها از دید خوشه اول با وزن u_{i1}^m ، و (راست) فیلتر تضعیف غیرخطی $w = u_{ij}^m$



شکل ۵: تحلیل جامع اثر پیچ تنظیم کاربر (m) بر توابع عضویت فضایی به ازای $m \in \{1/1, 2, 3, \dots, 20\}$ و تقریب حالت حدی $m \rightarrow \infty$: (چپ) رژیم گذار ساختاری از رفتار پله‌ای به قله هموار و قله نوک‌تیز در $1/1 \leq m \leq 10$ ، و (راست) رژیم اشباع حساسیت در $11 \leq m \leq 20$ به همراه منحنی تقریب عددی بی‌نهایت ($m = 150$) که میل کردن تابع عضویت به $1/c = 0.5$ را نشان می‌دهد.

۱. عبور از منطق صفر و یک و رعایت قید نرمال‌سازی در نقاط مرزی (شکل ۳ و جدول ۲): در خوشه‌بندی قطعی، نمونه مرزی واقع در مختصات $[4/0, 3/0]$ به اجبار برچسب صفر یا یک مطلق می‌گیرد؛ در حالی که در رویکرد بزدک، این نمونه دارای درجات تعلق پیوسته $u_{i1} = 0.47$ و $u_{i2} = 0.53$ است که ضمن رعایت دقیق قید سطری $u_{i1} + u_{i2} = 1/0$ ، ابهام طبیعی مرز را بدون نیاز به دانستن از پیش مجموعه‌های C_j مدل‌سازی می‌کند.

۲. نحوه دیده شدن خوشه‌ها و فیلتر میرایی غیرخطی (شکل ۴): در نمودار سمت چپ، با ضرب شدن u_{i1}^m در جمله فاصله خوشه اول، نمونه‌های واقع در خوشه اول با وزن نزدیک به یک ($u_{i1}^m \approx 1/0$)، دایره‌های آبی پررنگ) کاملاً دیده می‌شوند، در حالی که نمونه‌های متعلق به خوشه دوم به دلیل داشتن $u_{i1}^m \approx 0/0$ (دایره‌های محو در سمت راست) عملاً در محاسبه خطای خوشه اول بی‌اثر می‌شوند. نمودار سمت راست نیز نشان می‌دهد که به توان رساندن ضریب تعلق ($w = u_{ij}^m$)، تعلق‌های کوچک را به شدت سرکوب می‌کند (در $m = 2$ ، تعلق 0.2 به وزن ناچیز 0.04 کاهش می‌یابد).

۳. تحلیل رژیم اول پیچ تنظیم کاربر $1/1 \leq m \leq 10$ — تغییر فاز هندسی در $m = 3$ (شکل ۵ چپ): بر اساس رابطه تحلیلی $u_{i1}(x) = (1 + (|x - c_1|/|x - c_2|)^{\frac{1}{m-1}})^{-1}$ با مراکز $c_1 = 2$ و $c_2 = 6$ ، توان مؤثر نسبت فواصل برابر $p = \frac{1}{m-1}$ است. زمانی که $1 < m < 3$ است $p > 1$ ، مشتق تابع در مرکز خوشه صفر بوده و قله منحنی در $x = 2$ کاملاً هموار و مشتق‌پذیر است. در نقطه بحرانی $m = 3$ ($p = 1$)، قله هموار شکسته و به یک راس مخروطی تبدیل می‌شود و به ازای $m \in \{4, \dots, 10\}$ ($p < 1$)، قله‌ها به فرم تیغه‌های تیز تکین (Singular Cusps) درمی‌آیند.

۴. تحلیل رژیم دوم $11 \leq m \leq 20$ و تقریب رفتار حدی در $m \rightarrow \infty$ (شکل ۵ راست): در بازه $m \in [11, 20]$ ، به دلیل کاهش نرخ تغییرات توان مؤثر $(\frac{dp}{dm} = \frac{1}{(m-1)^2})$ ، منحنی‌ها دچار اشباع شده و به یکدیگر فشرده می‌شوند. در این نمودار، منحنی $m = 150$ (خط چین قرمز) به عنوان تقریب عددی حالت حدی $m \rightarrow \infty$ ترسیم شده است. از نظر تئوری، هنگامی که $m \rightarrow \infty$ میل می‌کند، توان $\frac{1}{m-1}$ به صفر میل کرده و درجه تعلق تمامی نقاط غیرمرکزی به مقدار ثابت و یکنواخت

$\lim_{m \rightarrow \infty} u_{ij} = \frac{1}{c}$ میل می‌کند؛ که در این مسئله با داشتن $c = 2$ خوشه، برابر با $0.5 = \frac{1}{2}$ (محوشدگی یکنواخت کامل) است و منحنی $m = 150$ به خوبی هموار شدن و خوابیدن تابع عضویت به سمت تراز 0.5 را نشان می‌دهد.

جمع‌بندی نهایی: رویکرد جیمز بزدک با جایگزینی شرط نامعلوم تعلق به مجموعه $(x_i \in C_j)$ با ماتریس درجات تعلق پیوسته $(U_{n \times c})$ تحت قید نرمال‌سازی $\sum_{j=1}^c u_{ij} = 1$ و افزودن $n \times c$ متغیر مجهول همراه با پیچ تنظیمی کاربر (m) ، سیگمای تابع هدف را روی تمام نمونه‌ها یکپارچه می‌سازد؛ به طوری که هر خوشه از درجه ضریب وزنی u_{ij}^m ، داده‌های مربوط به خود را با وزن نزدیک به یک و داده‌های سایر خوشه‌ها را با وزن نزدیک به صفر مشاهده می‌کند و کاربر می‌تواند با انتخاب m در بازه متعادل (نظیر $m = 2$)، از دو حالت حدی افراز باینری ($m \rightarrow 1^+$) و محوشدگی یکنواخت $\frac{1}{c}$ ($m \rightarrow \infty$) پرهیز نماید.

۶ منابع

۱. W. Full, & R., Ehrlich, C., J. Bezdek, (۱۹۸۴). **FCM: The fuzzy c-means clustering algorithm**, Computers & Geosciences, ۱۰ (۳-۲), ۱۹۱-۲۰۳.
۲. C. J. Bezdek, (۱۹۸۱). **Pattern Recognition with Fuzzy Objective Function Algorithms**, Plenum Press / Springer.