استخراج قوانین فازی و خوشه بندی فازی

جواد سلیمی

پاییز ۱۳۹۸

فهرست مطالب

- سیستم فازی مبتنی بر قانون
- اداره کردن مسائل با ابعاد بالا
- انتخاب قوانین
- راهکار Michigan
- راهکار Pittsburgh
- الگوریتم خوشه‌بندیK-Means
- الگوریتم خوشه‌بندی Fuzzy C-Means

مفهوم فازی

- **سوال (مفهوم جوانی)؟**
  - برای سن‌های ۱۵ ساله (ابتدای جوانی)، ۲۵ ساله (کاملا جوان) و ۳۵ ساله (انتهای جوانی)

- **ابهام و عدم شفافیت در متغیرهای زبانی و مسائل دنیای واقعی**
  - سرد، کم، پایین، خوب، زیبا، ...

- **فازی**
  - طرز تفکر / دستگاه استنتاجی جدید (چند مقداری)
  - برای برطرف ساختن ناتوانی منطق دومقداری و ریاضیات بسیار دقیق در برخورد با دنیای واقعی و نادقیق

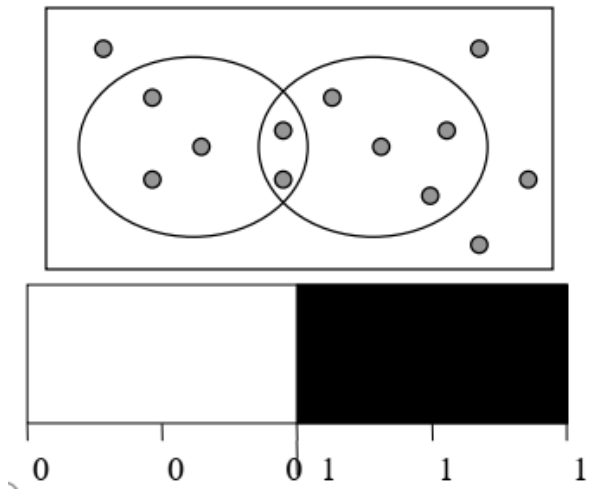
مجموعه ترد

- مجموعه تَرد (Crisp)
  - دو حالته
  - هر عضو مانند a،یا عضو مجموعه A است یا نیست (صفر یا یک)
  - تعریف مجموعه
    - به صورت لیست
    - به صورت قانون

$$A = \{a_1, a_2, \ldots, a_n\}$$

$$A = \{x \mid P(x)\}$$

xهایی که هر x دارای مشخصه P است



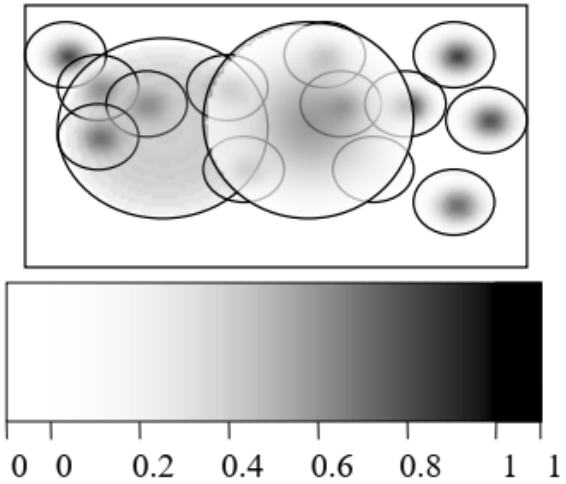
- تابع مشخصه مجموعه (characteristic function)

$$\chi_A(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases}$$

$$\chi_A : X \rightarrow \{0,1\}$$

مجموعه- فازی

- ایده
  - به هر عضو مجموعه (مانند a) میزان تعلق آن به مجموعه A را (بین صفر و یک) در نظر بگیریم



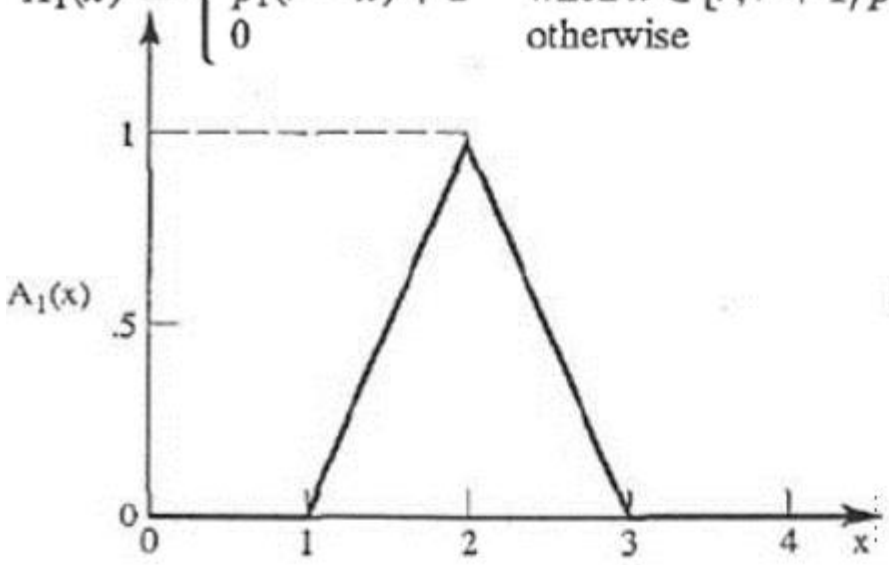
- تابع عضویت (membership function)
  - یک تابع مشخصه (characteristic function)
  - مقادیر در بازه‌ای از اعداد (معمولاً بین صفر و یک)
  - بیانگر میزان درجه/میزان عضویت هر عضو به مجموعه
- مجموعه فازی: مجموعه‌ای که با یک تابع عضویت تعریف می‌شود

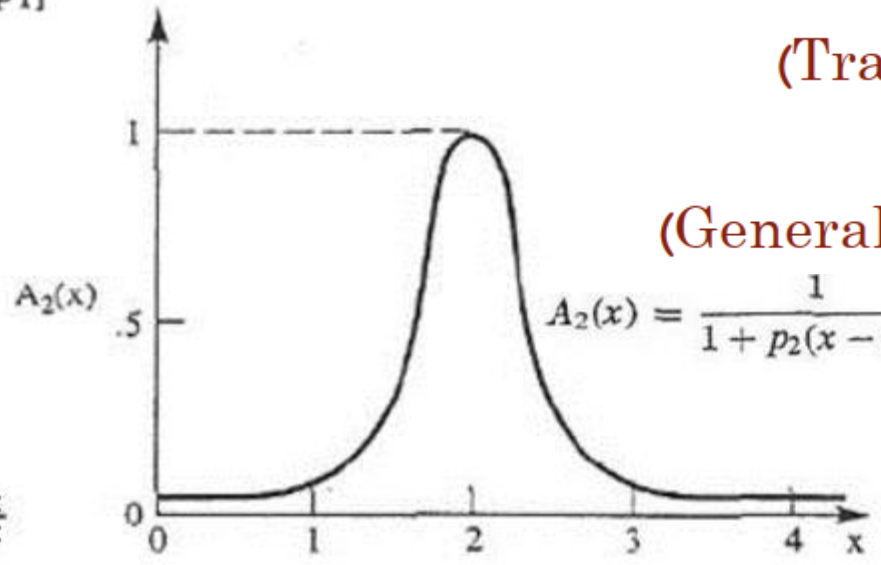
$$\mu_A : X \rightarrow [0,1]$$

توابع عضویت

- مثلثی (Triangular)
- ذوزنقه‌ای (Trapezoidal)
- گاوسی (Gaussian)
- زنگوله‌ای (Generalized bell)

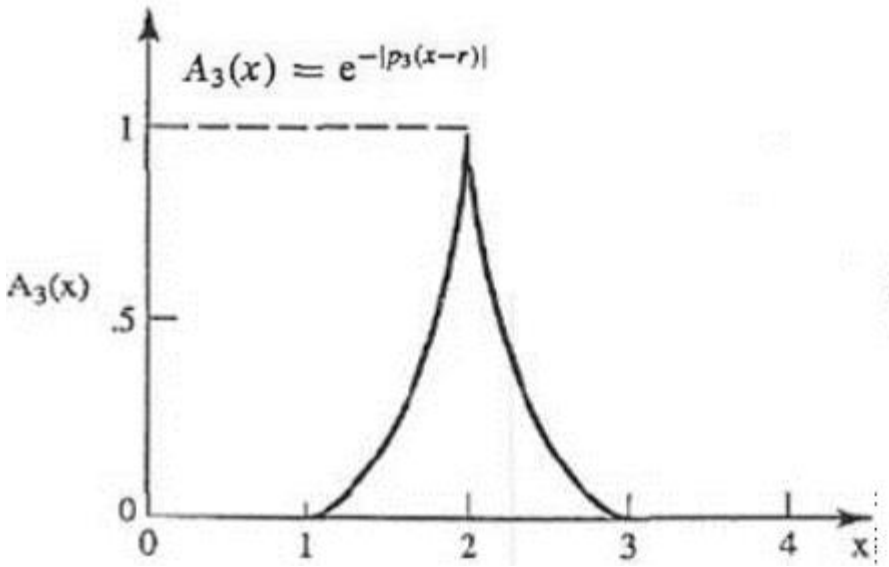
$$A_1(x) = \begin{cases} p_1(x-r)+1 & \text{when } x \in [r-1/p_1, r] \\ p_1(r-x)+1 & \text{when } x \in [r, r+1/p_1] \\ 0 & \text{otherwise} \end{cases}$$



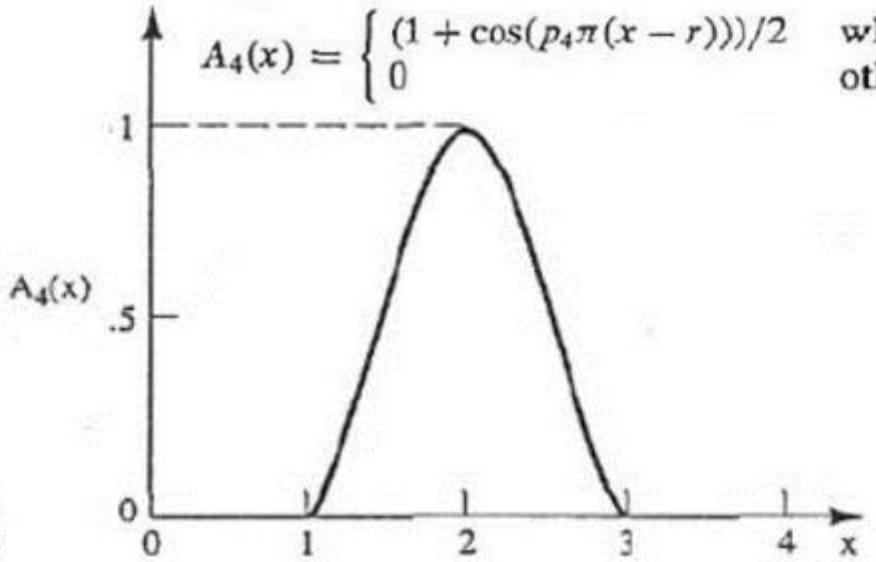


$$A_2(x) = \frac{1}{1+p_2(x-r)^2}$$

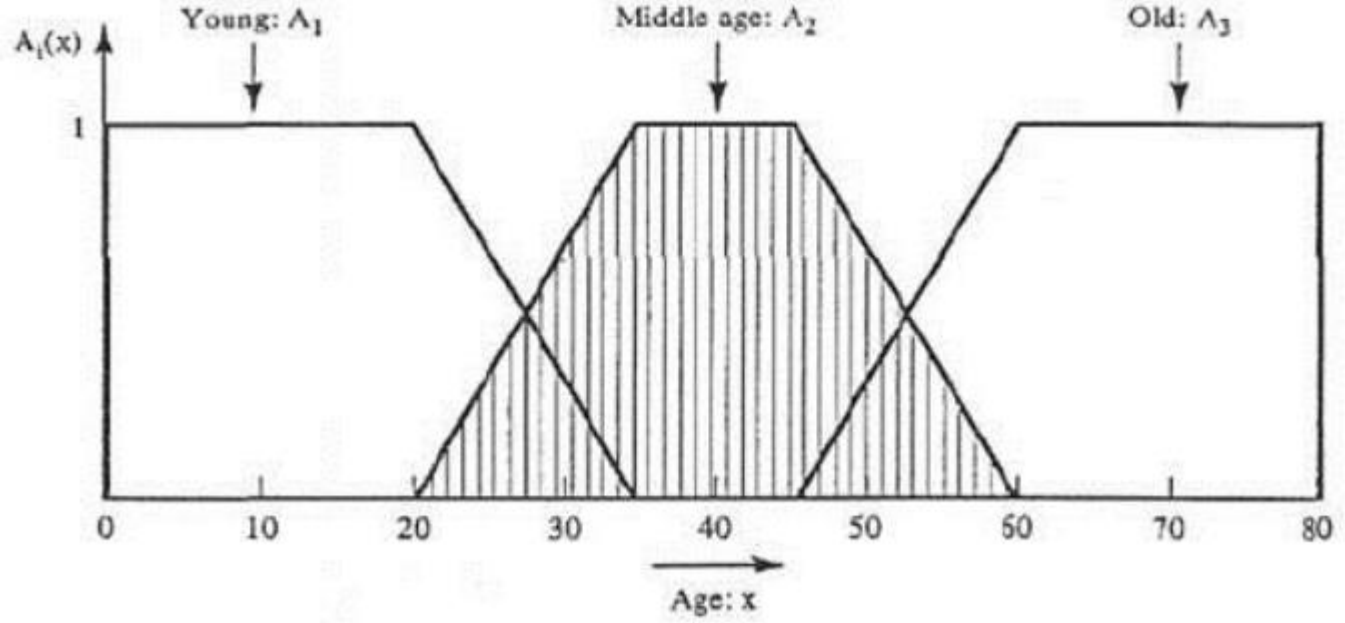
$$A_3(x) = e^{-|p_3(x-r)|}$$

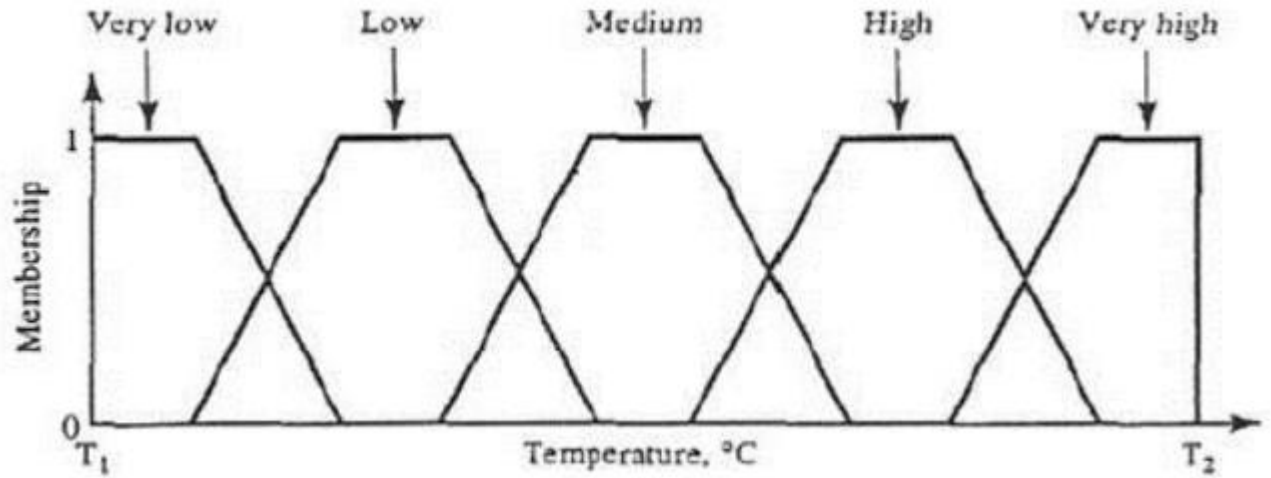


$$A_4(x) = \begin{cases} (1+\cos(p_4\pi(x-r)))/2 & \text{when } x \in [r-1/p_4, r+1/p_4] \\ 0 & \text{otherwise} \end{cases}$$



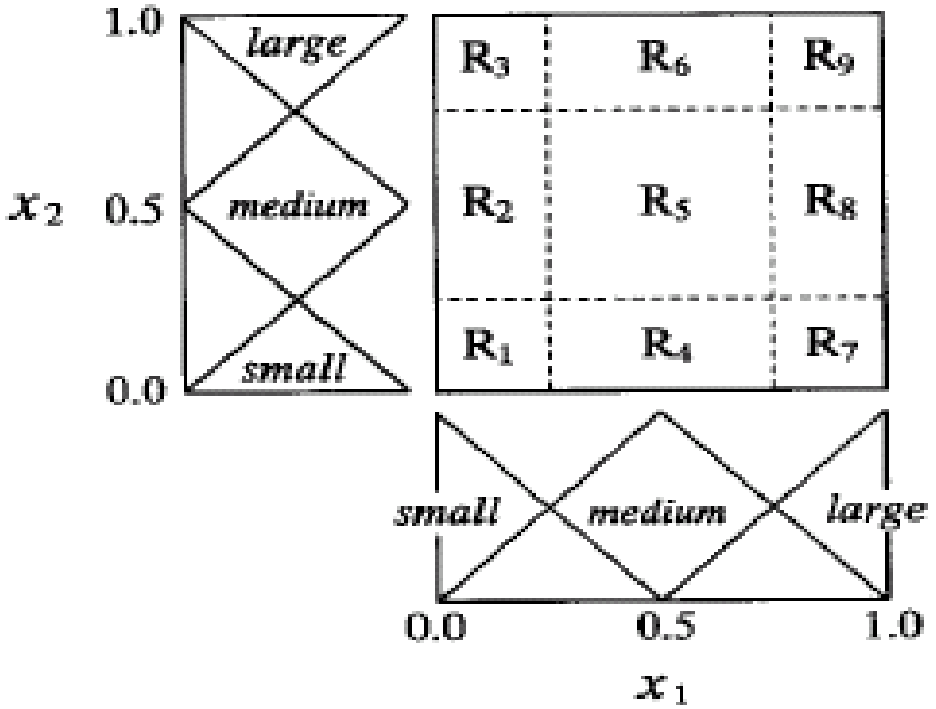
تابع عضویت ذوزنقه ای





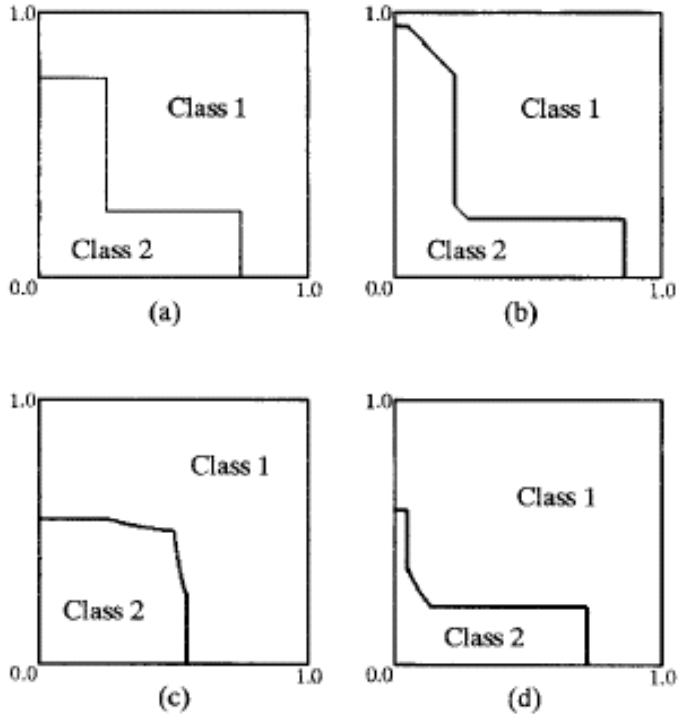
سیستم فازی مبتنی بر قانون

- قوانین فازی به صورت
  - **If linguistic condition THEN a consequent class WITH a certainty grade**
- فرض کنید فضای الگوی ۲ بعدی [۱و۰] *[۱و۰] به کمک ۳ مقدار زبانی {small, medium large} به ۹ زیر فضای فازی تقسیم شود



درجه قطعیت

- اگر ما از درجه قطعیت برای هر قانون فازی استفاده نکنیم (یعنی همه قوانین فازی if-then یک درجه قطعیت داشته باشند) مرز کلاس‌بندی ایجاد شده بوسیله ۹ قانون موجود در شکل قبل هیچ قابلیت انعطافی نخواهد داشت و همیشه به شبکه فازی نشان داده شده بستگی خواهد داشت. از طرف دیگر وقتی هر قانون فازی درجه قاطعیت متفاوتی داشته باشد حتی بدون تغییر توابع عضویت هر مقدار زبانی هم می‌توان مرزهای تصمیم‌گیری متفاوتی از ۹ قانون فازی شکل قبل به دست آورد.

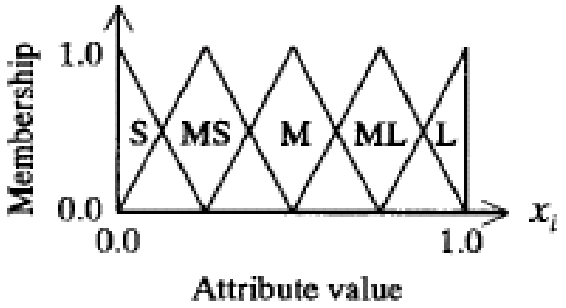


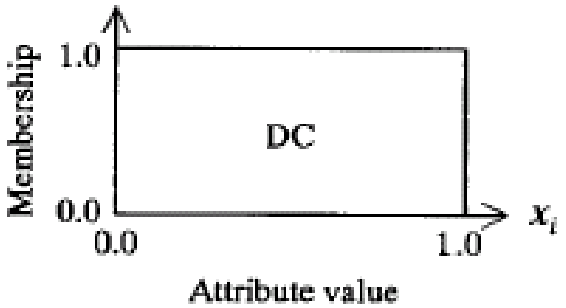
کلاس‌بندی فازی مبتنی بر قانون

- برای مساله کلاس‌بندی n بعدی ما، قوانین if-then به شکل زیر خواهند بود.

Rule R_j: If x_1 is A_{j1} and $\cdots$ and x_n is A_{jn}, then C_j with $CF = CF_j$,

- $x = (x_1, x_2,\ldots,x_n)$ یک بردار ویژگی n بعدی ، Aji یک مجموعه فازی (یک مقدار زبانی مانند small یا large) ، C_j کلاس نتیجه و CF درجه قطعیت قانون است.





- S, small; MS, medium small; M, medium;ML, medium large; L, large; DC, don’t Care.

ایجاد قانون از طریق هیوریستیک

- مرحله ۱: درجه سازگاری هر الگوی آموزش x_p با قانون فازی R_j را بوسیله عملگر ضرب به صورت زیر به دست آورید:

$$\mu_{R_j}(\mathbf{x}_p) = \mu_{A_{j1}}(x_{p1}) \times \cdots \times \mu_{A_{jn}}(x_{pn}), \qquad p = 1, 2, \ldots, m,$$

- مرحله ۲: مجموع درجات سازگاری هر کلاس را بصورت زیر بدست آورید:

$$\beta_{\text{Class } h}(R_j) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{R_j}(\mathbf{x}_p), \qquad h = 1, 2, \ldots, c.$$

- مرحله ۳: کلاس نتیجه C_j که بیشتر مقدار در بین C کلاس را دارد پیدا کنید:

$$\beta_{\text{Class } C_j}(R_j) = \max\{\beta_{\text{Class } 1}(R_j), \ldots, \beta_{\text{Class } c}(R_j)\}.$$

- کلاس بالا به عنوان نتیجه قانون فازی R_j بیان می‌شود.
- اگر دو یا چند کلاس در یک مقدار ماکزیمم مشترک باشند با کلاس نتیجه تهی و درجه قطعیت صفر یک قانون ساختگی ایجاد می‌شود.

ایجاد قانون از طریق هیوریستیک

❑مرحله ۴: وقتی کلاس نتیجه C_j تعیین شد درجه قطعیت CF_j به وسیله فرمول زیر تعیین می‌شود.

$$CF_j = \frac{\beta_{\text{Class } C_j}(R_j) - \bar{\beta}}{\sum_{h=1}^{c} \beta_{\text{Class } h}(R_j)},$$

که:

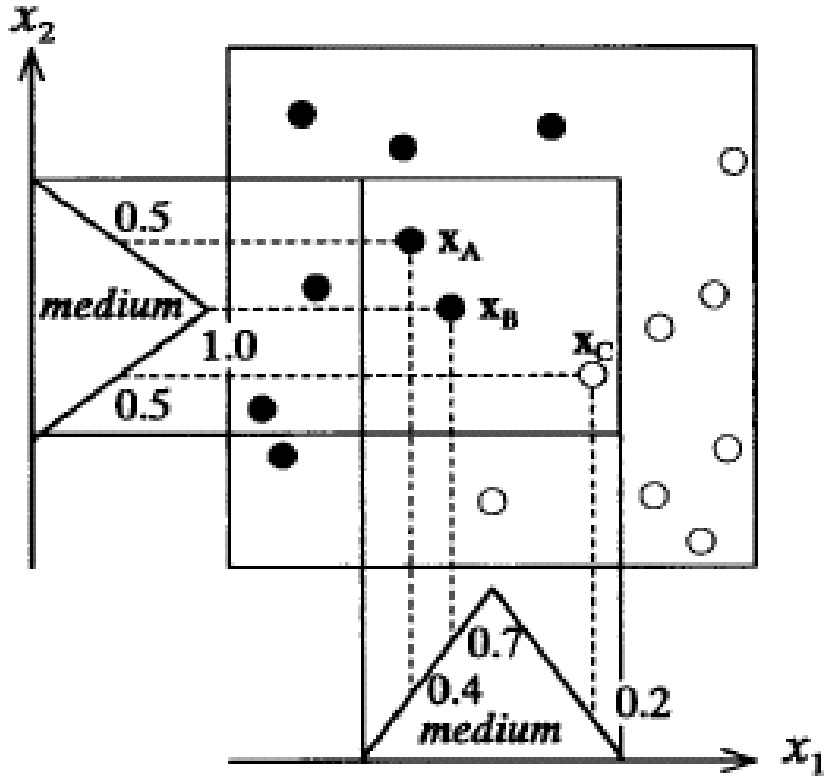
$$\bar{\beta} = \frac{\sum_{h \neq C_j} \beta_{\text{Class } h}(R_j)}{(c-1)}$$

- وقتی یک الگوی جدید X_p به این سیستم کلاس بندی فازی داده شود کلاس X_pبه کمک یک قانون برنده Rj^* که بصورت زیر به دست می‌آید تعیین می‌شود.

$$\mu_{R_{j^*}}(\mathbf{x}_p) \cdot CF_{j^*} = \max\{\mu_{R_j}(\mathbf{x}_p) \cdot CF_j | R_j \in S\}.$$

مثال

- if x_1 is medium and x_2 is medium is, then class1 with Cf = 0.8



$$\mu_{R_j}(\mathbf{x}_A) = 0.4 \times 0.5 = 0.2 \qquad \text{for } \mathbf{x}_A \text{ from Class 1,}$$

$$\mu_{R_j}(\mathbf{x}_B) = 0.7 \times 1.0 = 0.7 \qquad \text{for } \mathbf{x}_B \text{ from Class 1,}$$

$$\mu_{R_j}(\mathbf{x}_C) = 0.2 \times 0.5 = 0.1 \qquad \text{for } \mathbf{x}_C \text{ from Class 2.}$$

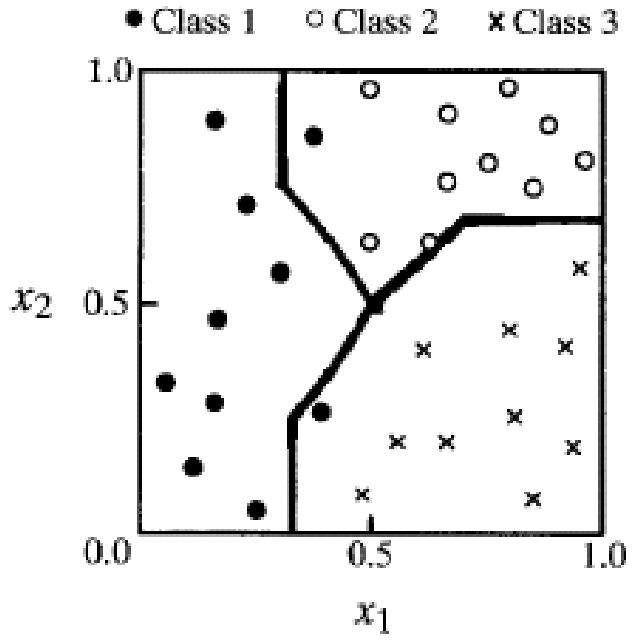
$$\beta_{\text{Class 1}}(R_j) = 0.9 \quad \text{and} \quad \beta_{\text{Class 2}}(R_j) = 0.1.$$

$$CF_j = \frac{0.9 - 0.1}{0.9 + 0.1} = 0.8.$$

مثال

x_2 \ x_1	*small*	*medium*	*large*
large	Class 1 (0.99)	Class 2 (0.63)	Class 2 (0.93)
medium	Class 1 (0.99)	Class 2 (0.01)	Class 3 (0.50)
small	Class 1 (0.97)	Class 3 (0.51)	Class 3 (1.00)

If x_1 is *small*, then Class 1 with $CF = 0.99$.



- ۲ الگو اشتباه کلاس بندی شده است.

- در حالت کلی درصد کلاس بندی الگوهای آموزشی با استفاده از تقسیم بندیهای فازی کوچکتر بهتر می‌شود برای مثال اگر ما از تفکیک فازی ۵*۵ (۲۵ قانون فازی) با مقدار ۵ مقدار زبانی استفاده می‌کنیم همه الگوهای آموزش صحیح کلاس بندی می‌شوند.

اداره کردن مسائل با ابعاد بالا

- برای یک مساله n بعدی تعداد حالات ترکیب مجموعه‌های فازی قسمت مقدم k^n خواهد بود که k تعداد مجموعه‌های فازی برای هر ویژگی است
  - انتخاب تعداد کمی از ویژگیهای مهم
  - استفاده از قوانین if-then عمومی، با تعداد کمی شرط در قسمت مقدم
- If x_1 is small and x_2 is don't care and x_3 is don't care and x_4 is don't care and x_5 is large, then class2 with CF=0.8
- If x_1 is small and x_5 is large, then class 2 with CF=0.8
- **تعداد قوانین فازی با هر طول برای یک مساله ۲۰ بعدی**
- **با سه مقدار زبانی و don't care**
- **تعداد کل قوانین if-then قابل ایجاد برابر $4^{20}=1.1*10^{12}$**
- **تعداد قوانین عمومی (قوانین فازی کوتاه) در مقایسه با تعداد کل قوانین کل عددی بزرگ ($1.1*10^{12}$) زیاد نیست**

TABLE I Number of Fuzzy If–Then Rules of Each Length

Rule length	Number of rules
1	60
2	1,710
3	30,780
4	392,445
5	3,767,472
10	1.1×10^{10}
15	2.2×10^{11}
20	3,486,784,401

مثال

بیمار	وزن	سن	فشار خون	وضعیت
۱	۹۰	۶۲	۱۵	حمله قلبی
۲	۸۰	۳۵	۱۳	بدون حمله قلبی
۳	۶۰	۶۰	۱۴	بدون حمله قلبی
۴	۱۱۰	۴۵	۱۴	حمله قلبی
۵	۹۵	۵۰	۱۳	؟

IF weight= High & Age= High & BP=Medium then

IF weight= Medium & Age= Medium & BP= Low then

الگوریتم تکاملی استخراج قوانین فازی

- مرحله ۱: تعداد زیادی قانون فازی کاندید از الگوهای آموزش ایجاد کن کلاس نتیجه و درجه قطعیت هر قانون توسط روش ایجاد قانون ارائه شده در اسلاید ۱۱ و ۱۲ تعیین می‌شود.
- مرحله ۲: یک جمعیت اولیه از سیستمهای کلاس‌بندی (مجموعه قانون) فازی ایجاد کن هر مجموعه قانون با انتخاب تصادفی قوانین فازی از قوانین کاندید ایجاد می‌شود.
- مرحله ۳: برازندگی هر مجموعه قانون در جمعیت فعلی را ارزیابی کن.
- مرحله ۴: مجموعه‌های قانون جدید از جمعیت فعلی به کمک عملگرهای ژنتیکی مانند انتخاب ترکیب و جهش ایجاد کن. جمعیت فعلی با مجموعه‌های قانون تازه ایجاد شده به روز می‌شود . تکنیک نخبه‌گرایی نیز در حین به روز رسانی اعمال می‌شود.
- مرحله ۵: اگر به شرایط تعریف شده برای توقف نرسیده‌ایم به مرحله ۳ برو.

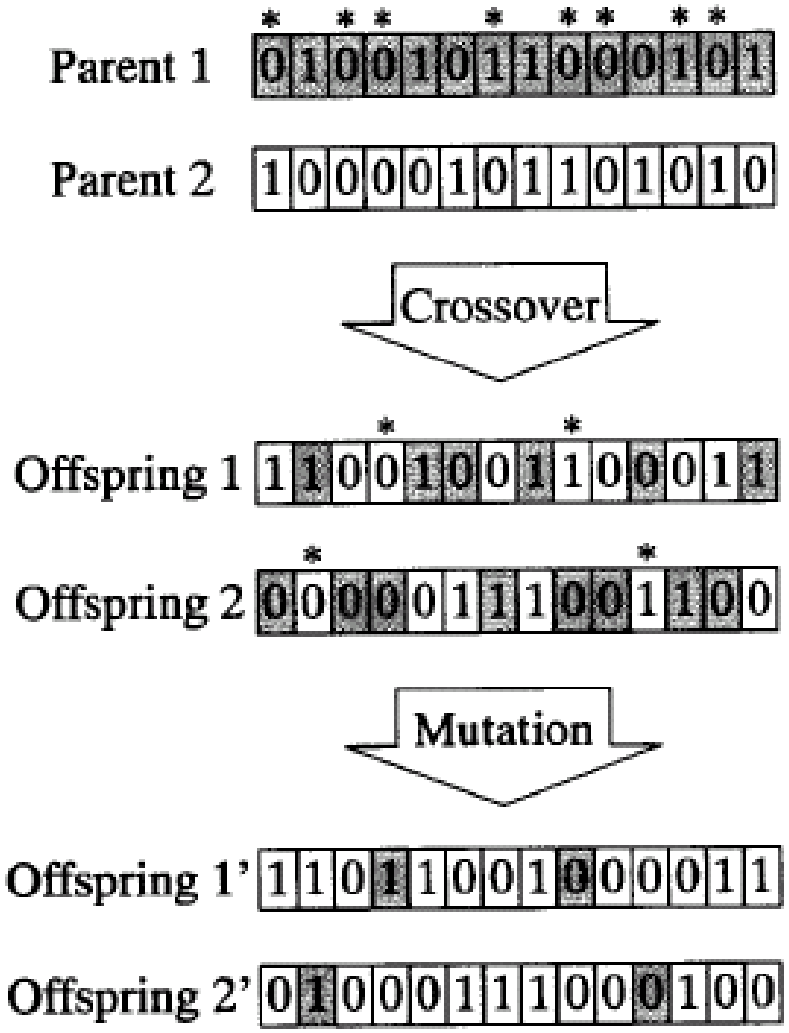
الگوریتم تکاملی استخراج قوانین فازی

- **بازنمایی؟**
- هر سیستم کلاس‌بندی فازی در این الگوریتم با یک عضو نمایش داده می‌شود یعنی یک سیستم S با یک رشته بیتی $s = s_1 s_2 \ldots s_N$ به طول N بیان می‌شود که N تعداد قوانین کاندید است. در رشته بیتی $s = s_1 s_2 \ldots s_N$ ، $sj = 1$ یعنی قانون jام در سیستم فازی هست به طور مشابه $s_j = 0$ یعنی از قانون jام در سیستم استفاده نخواهد شد. از این نمایش می‌توان مشاهده کرد که اندازه فضای جستجو 2^Nاست که برابر تعداد کل رشته بیتی‌های به طول N است.
- **تعریف تابع برازندگی**
- برازندگی هر سیستم کلاس‌بندی فازی بر اساس دقت آن در کلاس‌بندی و اندازه آن تعیین می‌شود

$$fitness(S) = w_{NCP} \cdot NCP(S) - w_S \cdot |S|,$$

- که NCP(S) تعداد الگوهایی است که بوسیله S درست کلاس‌بندی شده‌اند، |S| تعداد قوانین فازی در S و W_{NCP} و W_S وزنهای مثبت هستند مقدار W_{NCP} و W_S به اختیار کاربر انتخاب می‌شوند
  - صحت کلاس‌بندی و فشردگی

انتخاب قوانین



- عملگرهای ژنتیکی

جهش نامتقارن

$$P_m(0 \rightarrow 1) \leq P_m(1 \rightarrow 0)$$

انتخاب قوانین

- **توسعه‌ها**
- جنبه‌های دیگر سیستم کلاس‌بندی نیز می‌تواند در تعریف تابع برازندگی به کار رود برای مثال وقتی قوانین فازی عمومی (قوانین کوتاه) به دلیل قابل درک‌تر بودن نسبت قوانین خاص (قوانین بلند) ارجحیت داشته باشند می‌توان یک عبارت جریمه به تابع برازندگی اضافه کرد.
- می‌توان یک الگوریتم تنظیم درجه قطعیت به هر قانون فازی مجموعه قوانین (یک عضو الگوریتم ژنتیک) اعمال کرد.
  - توجه: درجه قطعیت تنظیم شده به فرزندان به ارث نمی‌رسد چون مقدار مناسب درجه قطعیت هر قانون به قوانین دیگر مجموعه قوانین بستگی دارد.

راهکار Michigan

- یک قانون if-then به عنوان یک عضو در الگوریتم ژنتیکی
- مراحل
  - مرحله ۱: یک نسل اولیه از قوانین فازی if-then ایجاد کنید. هر قانون فازی با تعیین تصادفی مجموعه‌های فازی مقدم آن ایجاد می‌شود کلاس نتیجه و درجه قاطعیت هر قانون بوسیله روش ایجاد قانون بیان شده تعیین می‌شود.
  - مرحله ۲: هر قانون فازی را در جمعیت فعلی ارزیابی کنید.
  - مرحله ۳: قوانین فازی جدیدی را به کمک عملگرهای ژنتیکی مانند انتخاب، ترکیب و جهش ایجاد کنید.
  - مرحله ۴- قسمتی از جمعیت فعلی را با قوانین فازی که جدیداً ایجاد شده جایگزین کن.
  - مرحله ۵- اگر به شرایط توقف پیش تعریف نرسیده‌ایم به مرحله ۲ برو.

راهکار Michigan

- **کد کردن قوانین فازی**
  - فقط قسمت مقدم هر قانون کد می‌شود. برای مثال سمبول‌های زیر برای کد کردن ۵ متغیر زبانی و don’t care بکار رفته است.

1. small
2. medium small
3. medium
4. medium large
5. larg

don’t care #

  - قانون فازی بعد به صورت " 1 # 4 # " کد می‌شود.

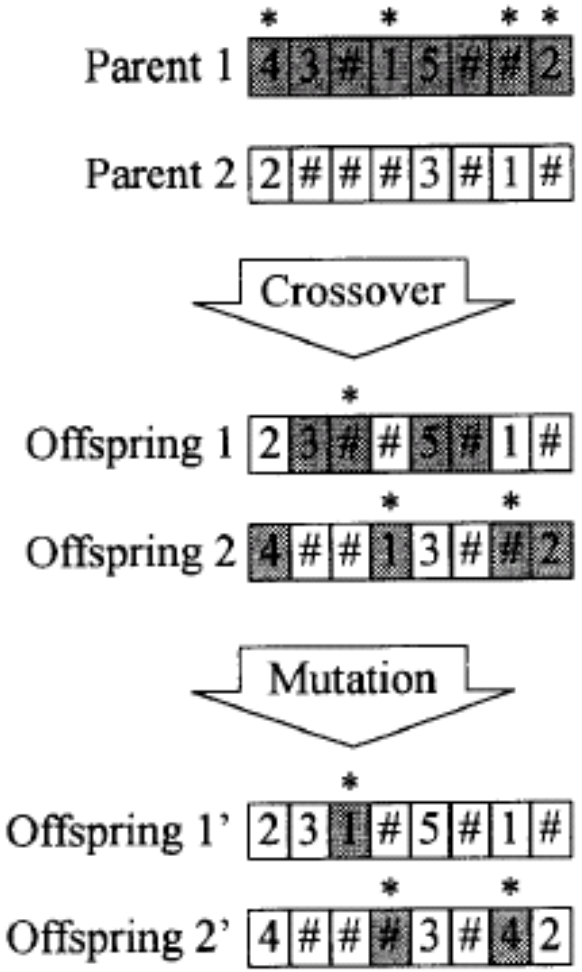
- If x_1 is small and x_2 is don’t care and x_3 is medium large and x_4 is don’t care then class cj with CF = CFj

راهکار Michigan

- **تعریف تابع برازندگی**
  - هر الگو با یک قانون برنده در سیستم کلاس بندی فازی S دسته بندی می‌شود. برازندگی هر قانون فازی if-then بوسیله نتایج سیستم کلاس بندی S متناظر با جمعیت فعلی تعیین می‌شود.
- Fitness (R_i)=NCP(R_i) که NCP(Ri) تعداد الگوهای آموزشی صحیح کلاس بندی بوسیله قانون Ri می‌باشد.

راهکار Michigan

- عملگرهای ژنتیکی



راهکار Michigan

- **توسعه‌ها**
- افزایش احتمال don’t care در هنگام ایجاد اولیه قوانین
  - هر قانون فازی if-then می‌تواند الگوهای آموزش بسیاری را کلاس بندی کند
- ایجاد قوانین فازی اولیه مستقیماً از الگوهای آموزش
- بهبود جمعیت به کمک روش ایجاد مستقیم قانون.
  - در بهبود جمعیت بعضی قوانین فازی مستقیماً از الگوهای بد کلاس بندی شده ساخته می‌شوند قوانین دیگر به کمک عملگرهای ژنتیکی از جمعیت فعلی به دست می‌آیند.
- سایز جمعیت را در طول اجرای سیستم کلاس بندی وفق دهیم:
  - با حذف همه قانون‌هایی که برازندگی آنها کمتر از یک آستانه تعریف شده باشد

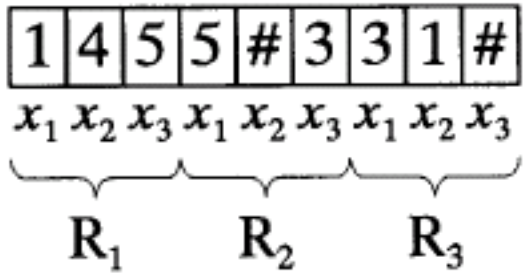
راهکار Pittsburgh

- یک مجموعه از قوانین فازی یک عضو این قسمت را تشکیل میدهد
- مراحل
  - مرحله ۱: یک جمعیت اولیه از سیستمهای کلاس بندی فازی (مجموعه قوانین) ایجاد کن. هر قانون فازی در مجموعه قوانین اولیه به همان روش سیستمهای کلاس بندی ایجاد می‌شود.
  - مرحله ۲: هر مجموعه قانون در جمعیت فعلی را ارزیابی کن.
  - مرحله ۳: مجموعه قوانین جدیدی از جمعیت فعلی به کمک عملگرهای ژنتیکی مانند انتخاب ترکیب و جهش ایجاد کن. جمعیت فعلی با این مجموعه قوانین تازه ایجاد شده به روز می‌شود. روش نخبه‌گرایی هم در ضمن بهبود نسلها اجرا می‌شود.
  - مرحله ۴: اگر به شرایط توقف معین نرسیده‌ایم به مرحله ۲ برو.

راهکار Pittsburgh

- **کد کردن مجموعه قوانین**
- در الگوریتم یادگیری ماشین ژنتیکی یک مجموعه قانون فازی با اتصال تعداد ثابتی زیر رشته نشان داده می‌شود. هر قانون فازی درون مجموعه قوانین به عنوان یک زیر رشته، همانند روش سیستمهای کلاس بندی فازی کد می‌شود.
- "1455 # 331#" سه قانون فازی بعدی را نشان میدهد.

- R_1: if x_1 is small and x_2 is medium large and x_3 is large then class c_1 with CF = CF_1
- R_2: if x_1 is large and x_2 is don't care and x_3 is medium then clss c_2 with CF = CF_2
- R_3: if x_1 is medium and x_2 is small and x_3 is don't care c_3 with CF = CF_3



راهکار Pittsburgh

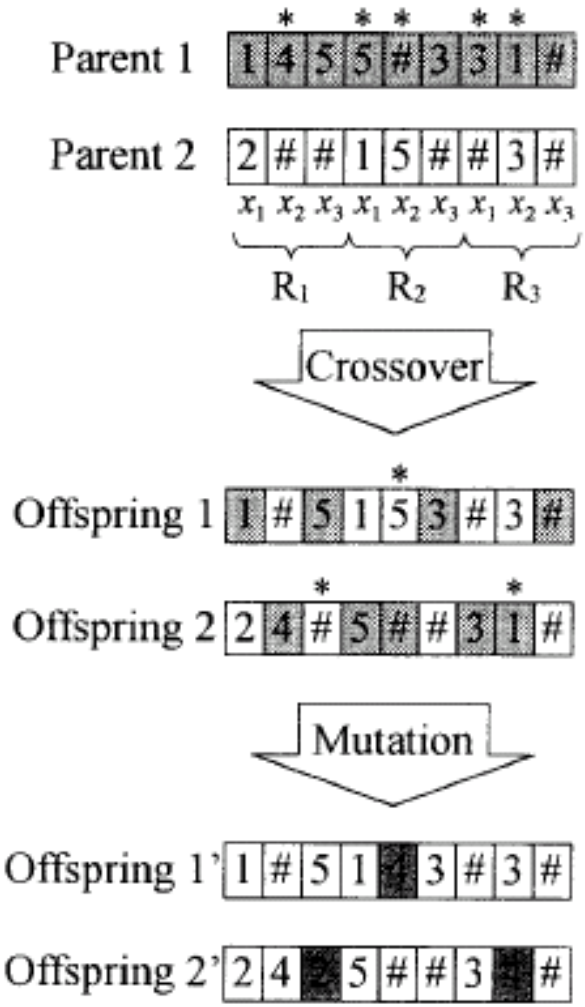
- **تعریف تابع برازندگی**
- در این قسمت چون تعداد قوانین فازی درون هر مجموعه قانون ثابت است قرار دادن تاوان متناسب با طول بی ارزش می‌شود بنابراین برازندگی هر مجموعه قانون را فقط با کارایی آن بررسی می‌کنیم.

Fitness (S) =NCP (S)

- که NCP(S) تعداد الگوهای آموزشی است که بوسیله مجموعه قانون S درست کلاس بندی شده است. این تابع برازندگی را، بر اساس برازندگی هر یک از قوانین سیستم کلاس بندی فازی می‌توان به صورت زیر نوشت.

$$fitness(S) = \sum_{R_j \in S} fitness(R_j) = \sum_{R_j \in S} NCP(R_j)$$

راهکار Pittsburgh



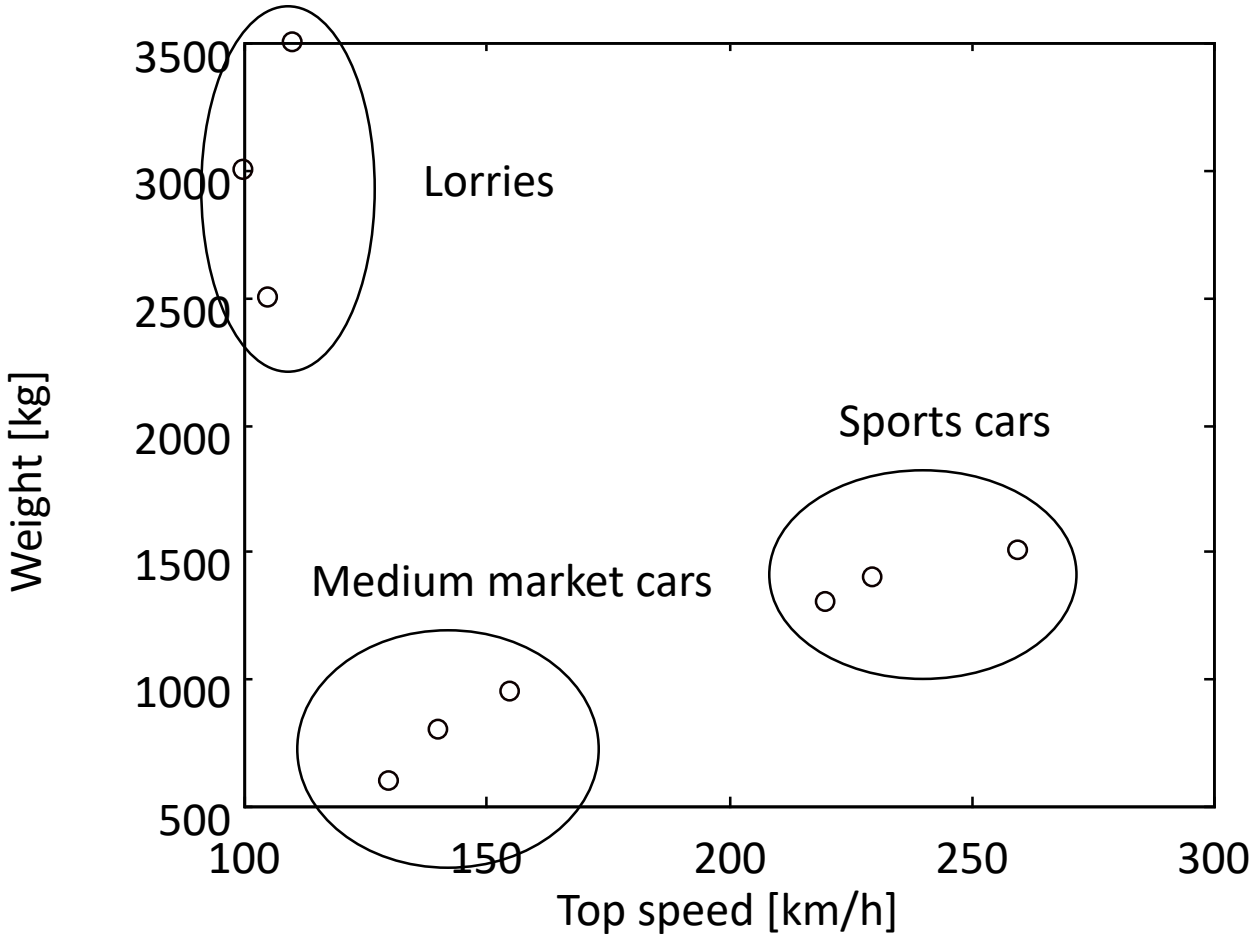
- عملگرهای ژنتیکی

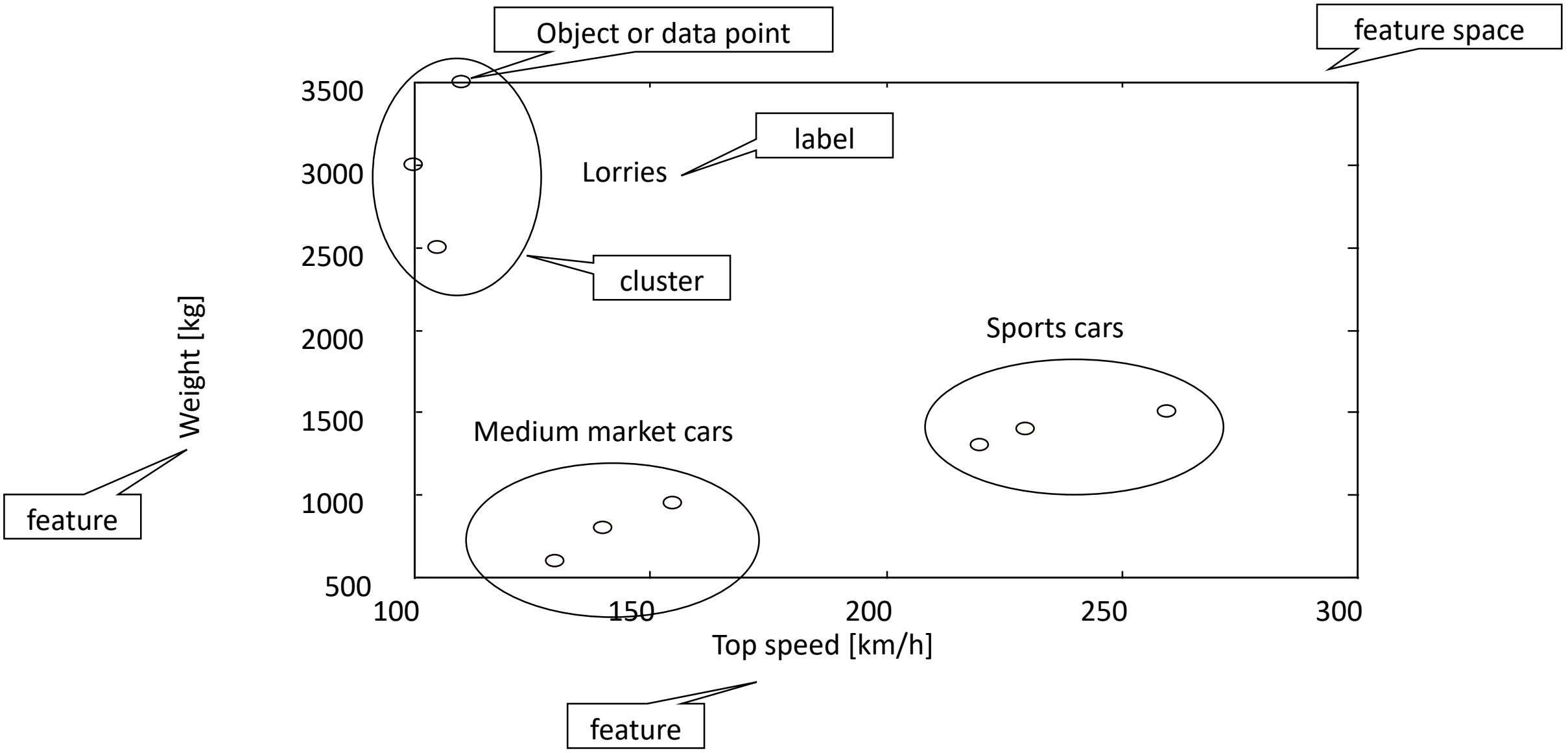
خوشه‌بندی فازی

Vehicle Example

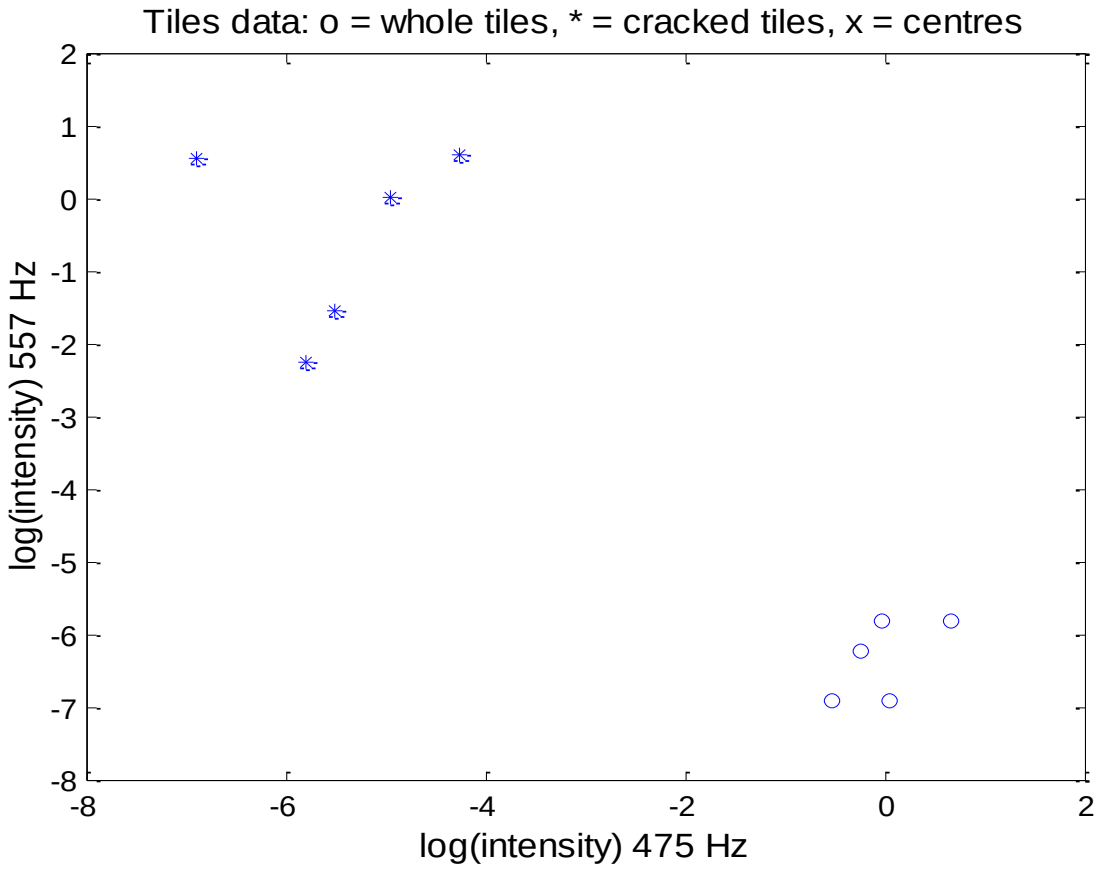
Vehicle	Top speed km/h	Colour	Air resistance	Weight Kg
V1	220	red	0.30	1300
V2	230	black	0.32	1400
V3	260	red	0.29	1500
V4	140	gray	0.35	800
V5	155	blue	0.33	950
V6	130	white	0.40	600
V7	100	black	0.50	3000
V8	105	red	0.60	2500
V9	110	gray	0.55	3500

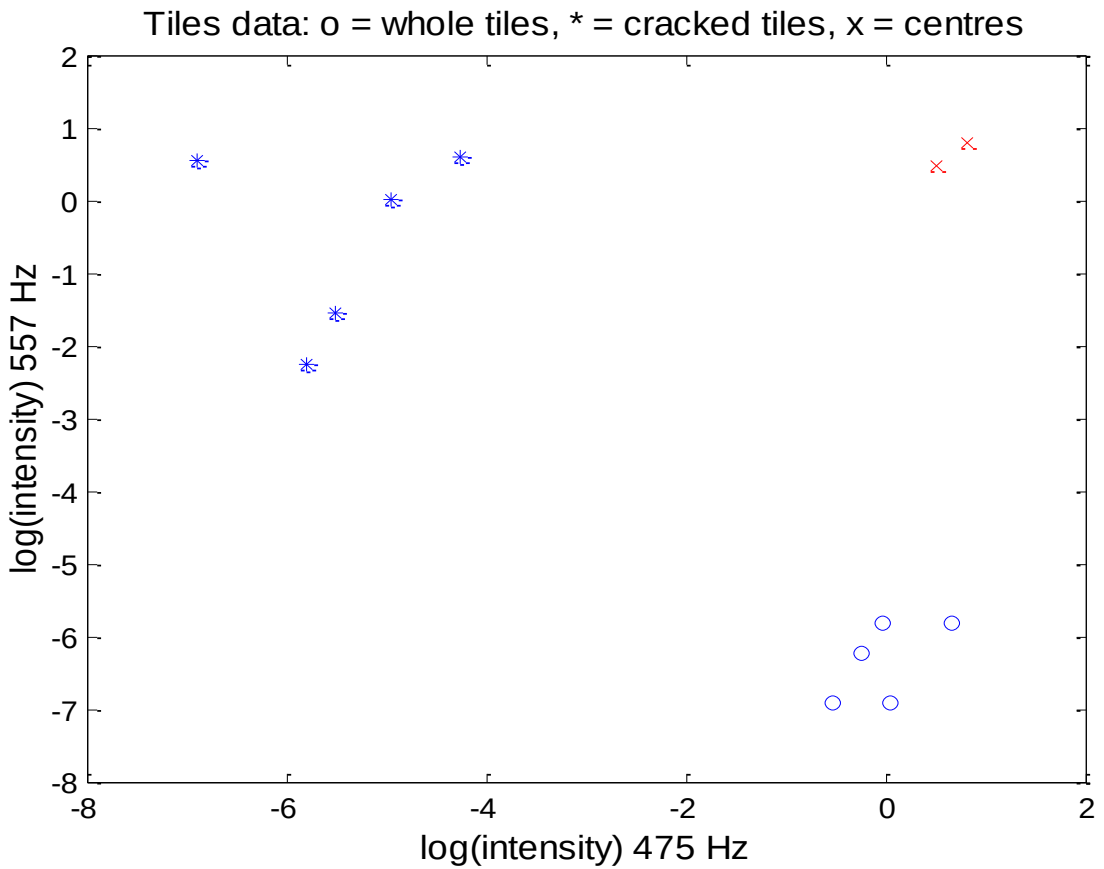
Vehicle Clusters



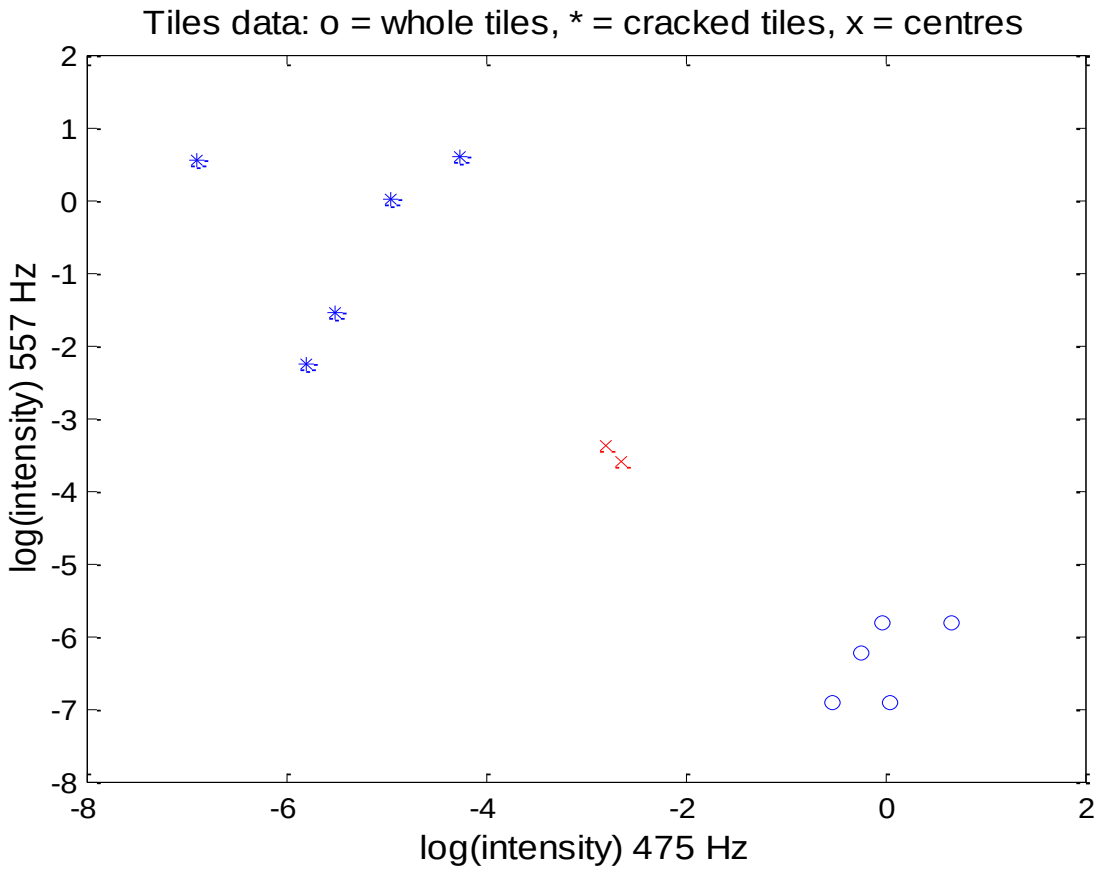

Object or data point
feature space
label
Lorries
cluster
Sports cars
Medium market cars
Weight [kg]
feature
3500
3000
2500
2000
1500
1000
500
100
150
200
250
300
Top speed [km/h]
feature

خوشه‌بندی K-Means

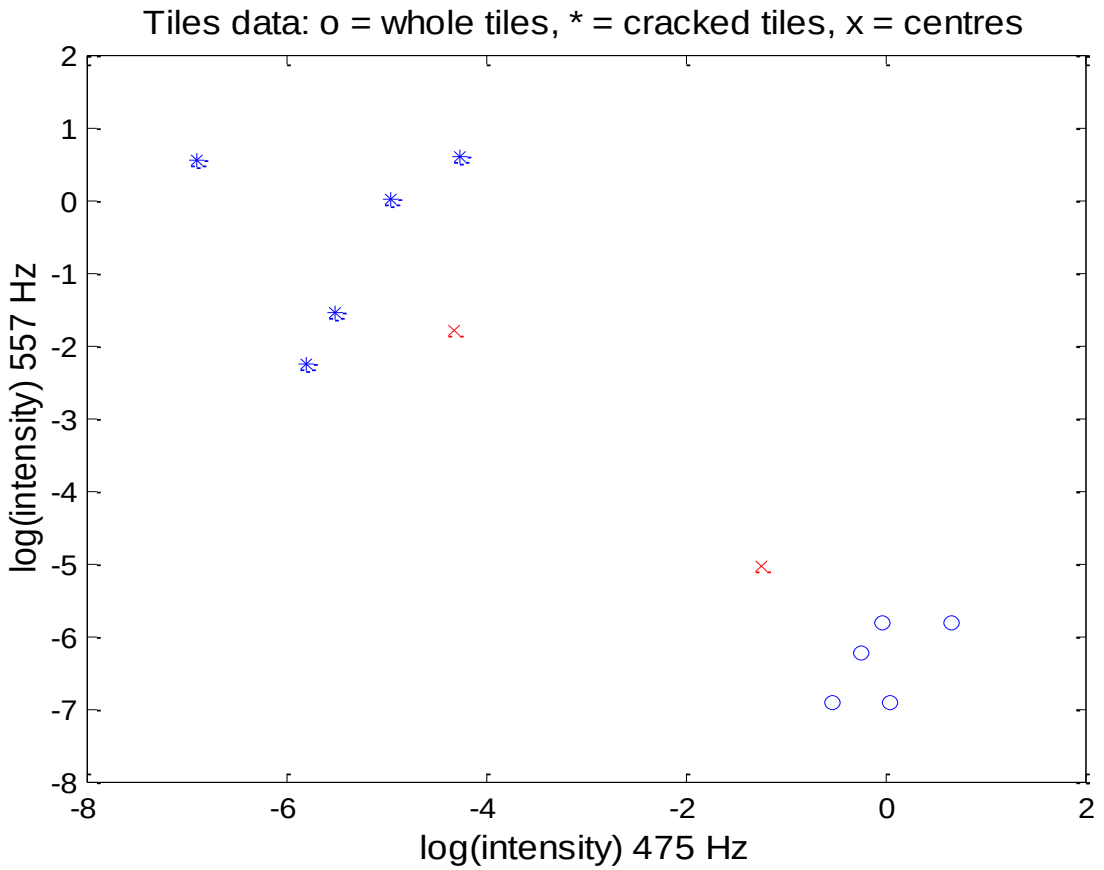

Tiles data: o = whole tiles, * = cracked tiles, x = centres
log(intensity) 557 Hz
log(intensity) 475 Hz
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-8
-6
-4
-2
0
2



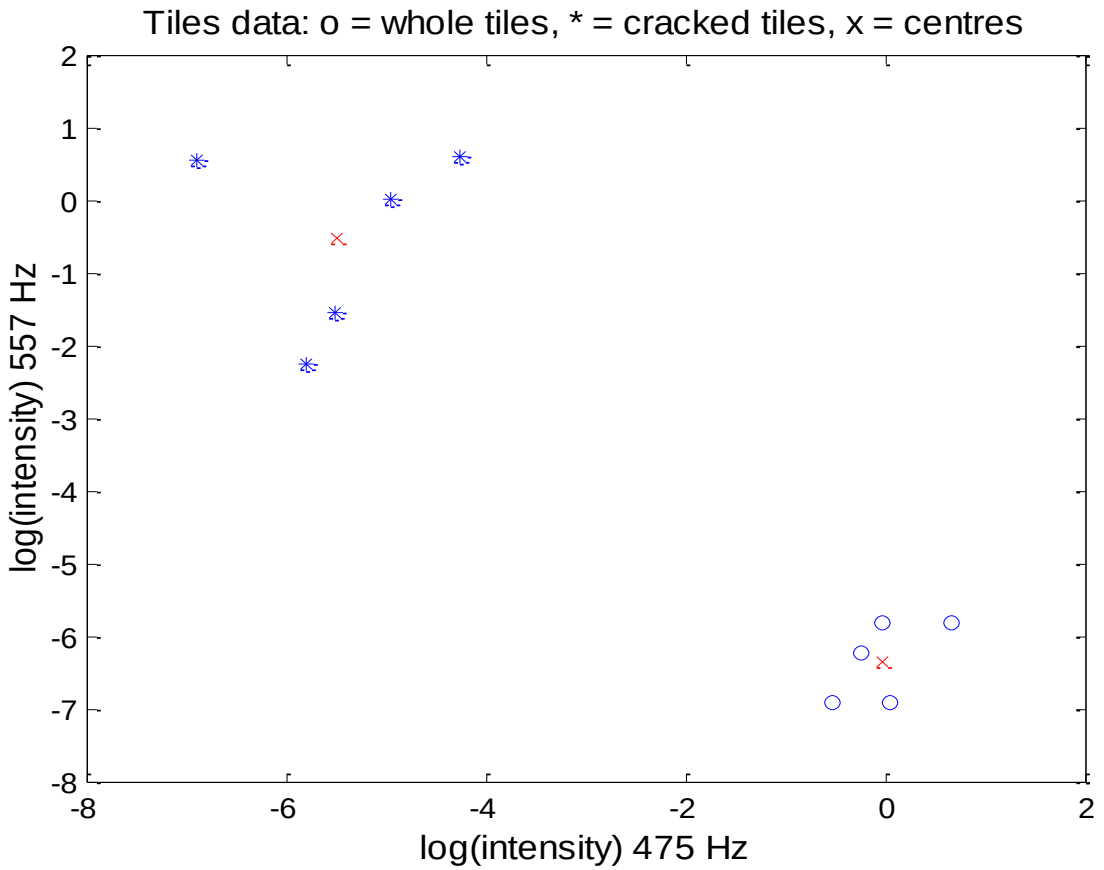
1. Place two cluster centres (x) at random.
2. Assign each data point (* and o) to the nearest cluster centre (x)



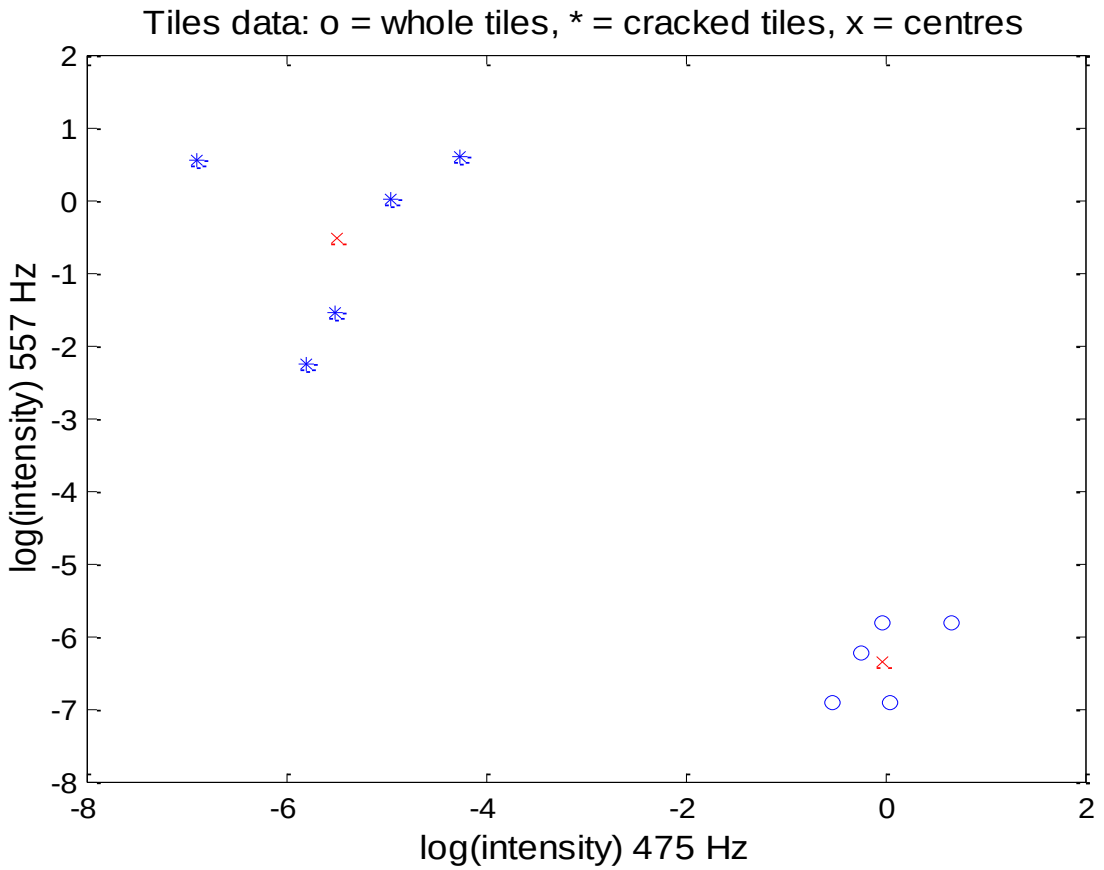
1. Compute the new centre of each class
2. Move the crosses (x)


Tiles data: o = whole tiles, * = cracked tiles, x = centres
log(intensity) 557 Hz
log(intensity) 475 Hz
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-8
-6
-4
-2
0
2

Iteration 2


Tiles data: o = whole tiles, * = cracked tiles, x = centres
log(intensity) 557 Hz
log(intensity) 475 Hz
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-8
-6
-4
-2
0
2

Iteration 3



Iteration 4 (then stop, because no visible change)
Each data point belongs to the cluster defined by the nearest centre

```
M =

    0.0000    1.0000
    0.0000    1.0000
    0.0000    1.0000
    0.0000    1.0000
    0.0000    1.0000
    1.0000    0.0000
    1.0000    0.0000
    1.0000    0.0000
    1.0000    0.0000
    1.0000    0.0000
```

The membership matrix M:

1. The last five data points (rows) belong to the first cluster (column)
2. The first five data points (rows) belong to the second cluster (column)

K-Means الگوریتم

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

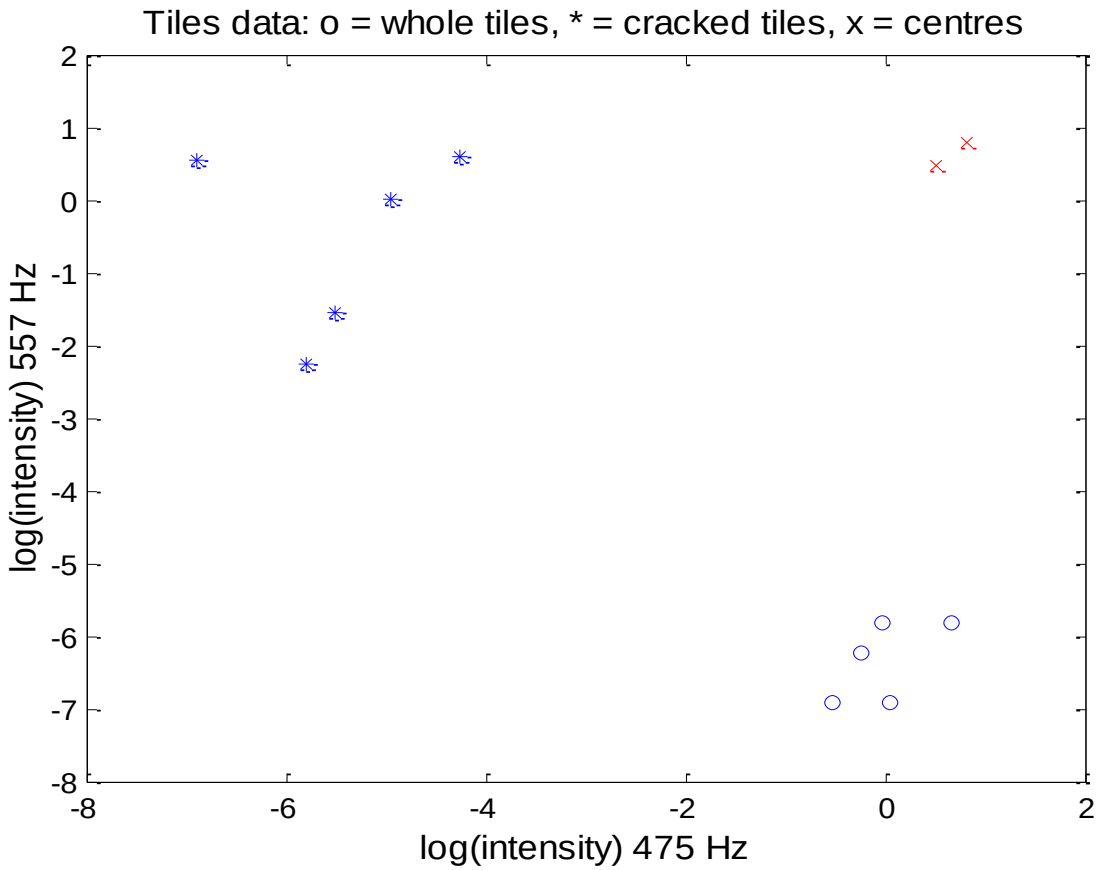
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set

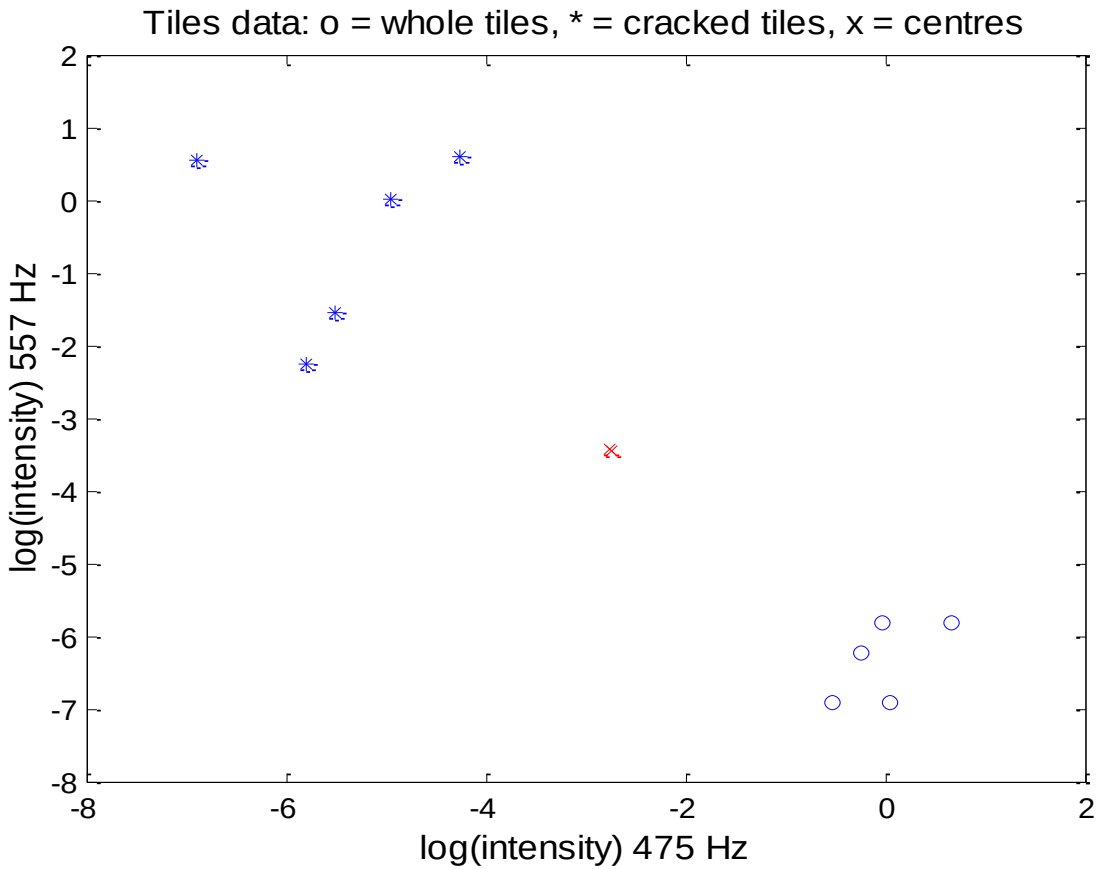
   $$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

   }

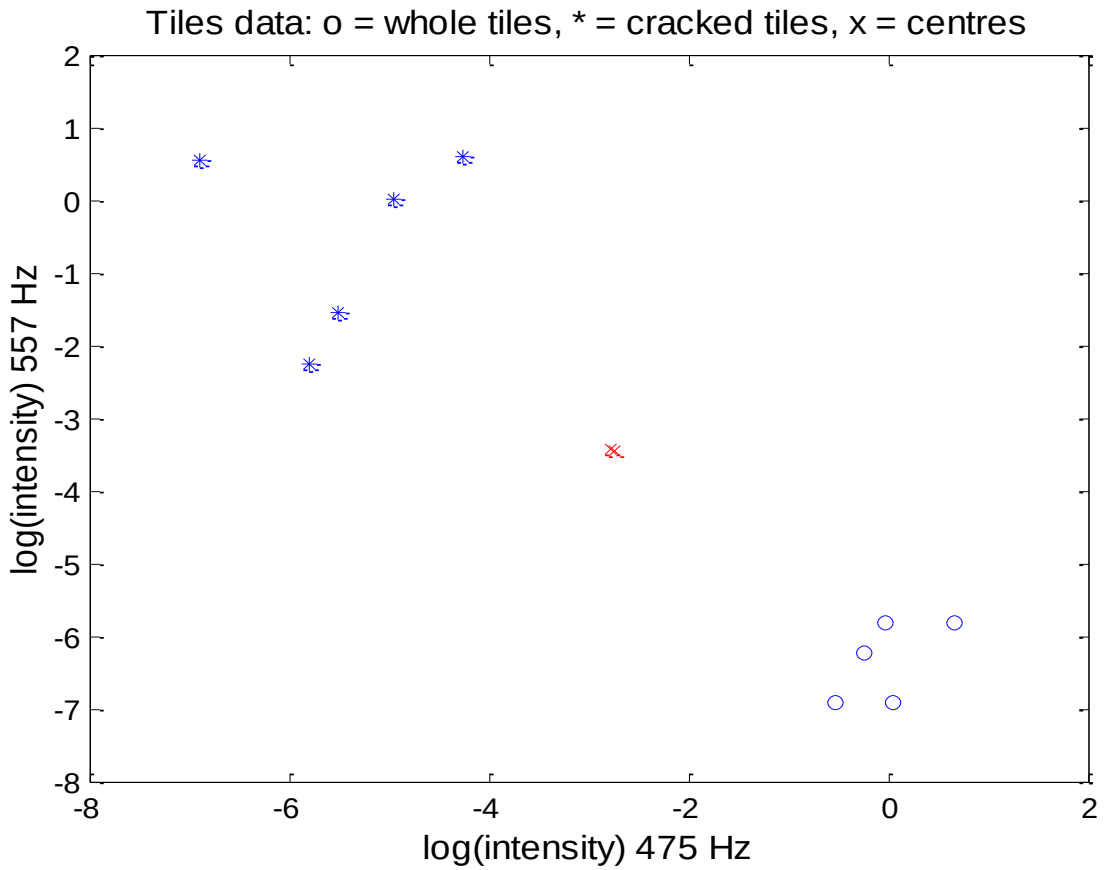
خوشه‌بندی Fuzzy C-Means



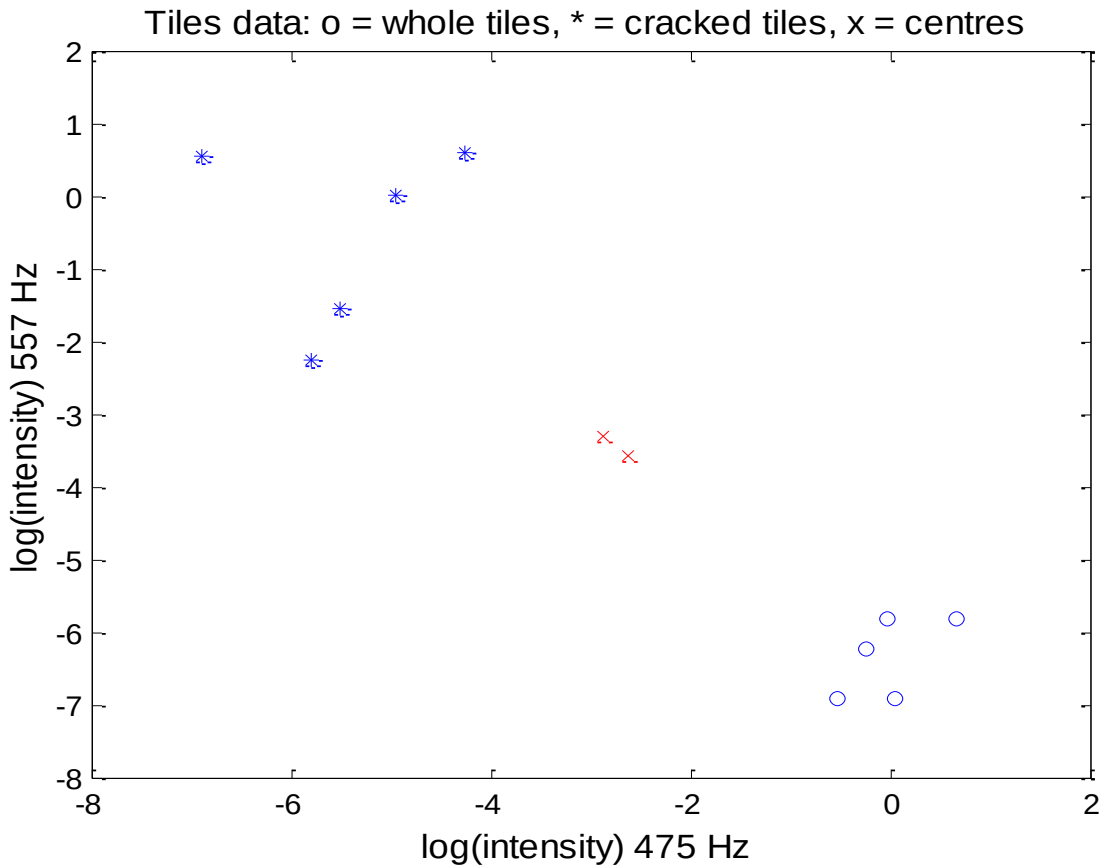
1. Place two cluster centres
2. Assign a fuzzy membership to each data point depending on distance



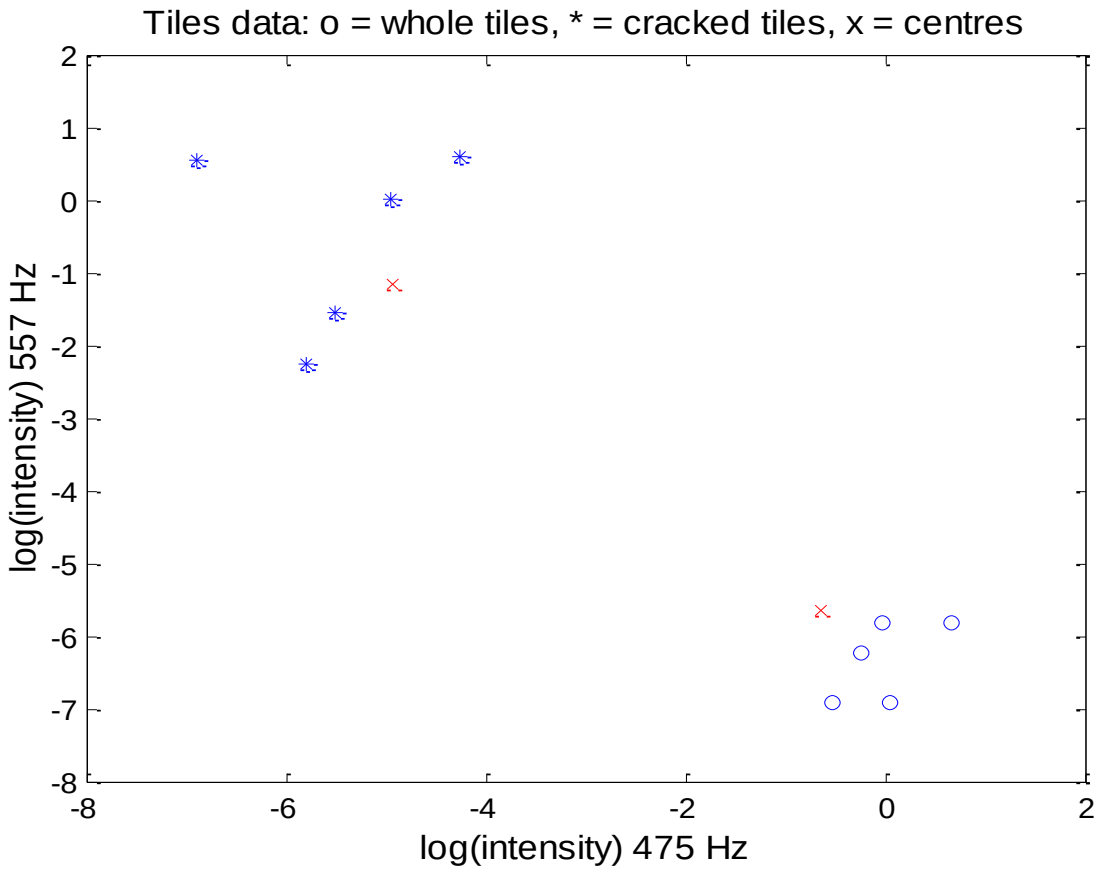
1. Compute the new centre of each class
2. Move the crosses (x)


Tiles data: o = whole tiles, * = cracked tiles, x = centres
log(intensity) 557 Hz
log(intensity) 475 Hz
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-8
-6
-4
-2
0
2

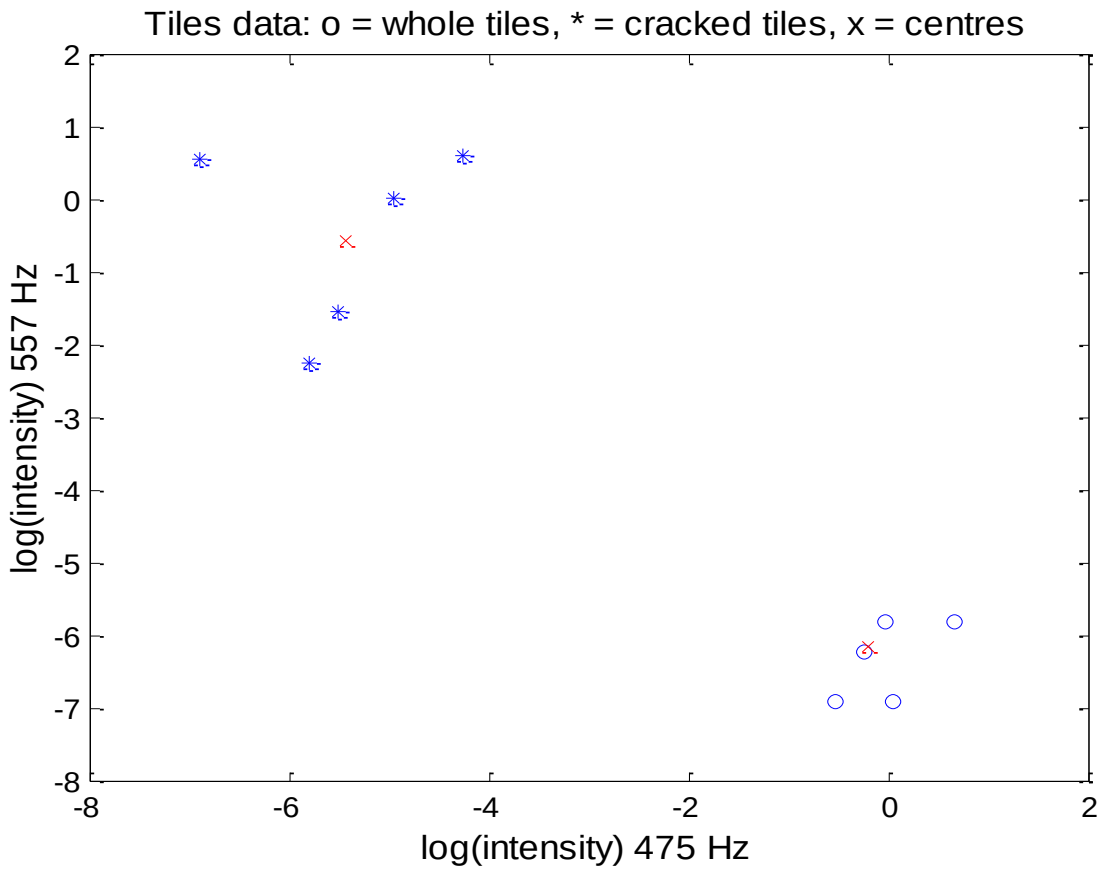
Iteration 2


Tiles data: o = whole tiles, * = cracked tiles, x = centres
log(intensity) 557 Hz
log(intensity) 475 Hz
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-8
-6
-4
-2
0
2

Iteration 5



Iteration 10



Iteration 13 (then stop, because no visible change)
Each data point belongs to the two clusters to a degree

```
M =

    0.0025    0.9975
    0.0091    0.9909
    0.0129    0.9871
    0.0001    0.9999
    0.0107    0.9893
    0.9393    0.0607
    0.9638    0.0362
    0.9574    0.0426
    0.9906    0.0094
    0.9807    0.0193
```

The membership matrix M:

1. The last five data points (rows) belong mostly to the first cluster (column)
2. The first five data points (rows) belong mostly to the second cluster (column)

Fuzzy C-Means Clustering

$$\min_{(\mathbf{U},\mathbf{V})} \left\{ J_m(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} D_{ik}^{2} \right\}$$

(FCM), Objective Function

Constraint

$$\sum_{i=1}^{c} u_{ik} = 1 \,, \forall k$$

Distance

$$D_{ik}^{2} = \left\| \mathbf{x}_k - \mathbf{v}_i \right\|_{\mathbf{A}}^{2}$$

Degree of Fuzzification

$$m \geq 1$$

Fuzzy C-Means Clustering

$$u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{D_{ik}}{D_{jk}} \right)^{\frac{2}{m-1}} \right]^{-1}, \forall i, k$$

$$\mathbf{v}_i = \left(\sum_{k=1}^{n} u_{ik}^{m} \mathbf{x}_k \Big/ \sum_{k=1}^{n} u_{ik}^{m} \right), \forall i$$

procedure FCM-CLUSTERING (**x**) **returns** prototypes and partition matrix
input: data $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$
local: fuzzification parameter: m
threshold: ε
norm: $\|\cdot\|$
INITIALIZE-PARTITION-MATRIX
$t \leftarrow 0$
repeat
for $i = 1 : c$ **do**

$$\mathbf{v}_i(t) \leftarrow \frac{\sum_{k=1}^{N} u_{ik}^m(t)\mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^m(t)} \text{ (compute prototypes)}$$

for $i = 1 : c$ **do**
for $k = 1 : N$ **do**
update partition matrix

$$u_{ik}(t+1) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \mathbf{x}_k - \mathbf{v}_i(t) \|}{\| \mathbf{x}_k - \mathbf{v}_j(t) \|} \right)^{2/(m-1)}} \text{ (update partition) matrix}$$

$t \leftarrow t + 1$
until $\| U(t+1) - U(t) \| \leq \varepsilon$
return U, V