

به نام خدا

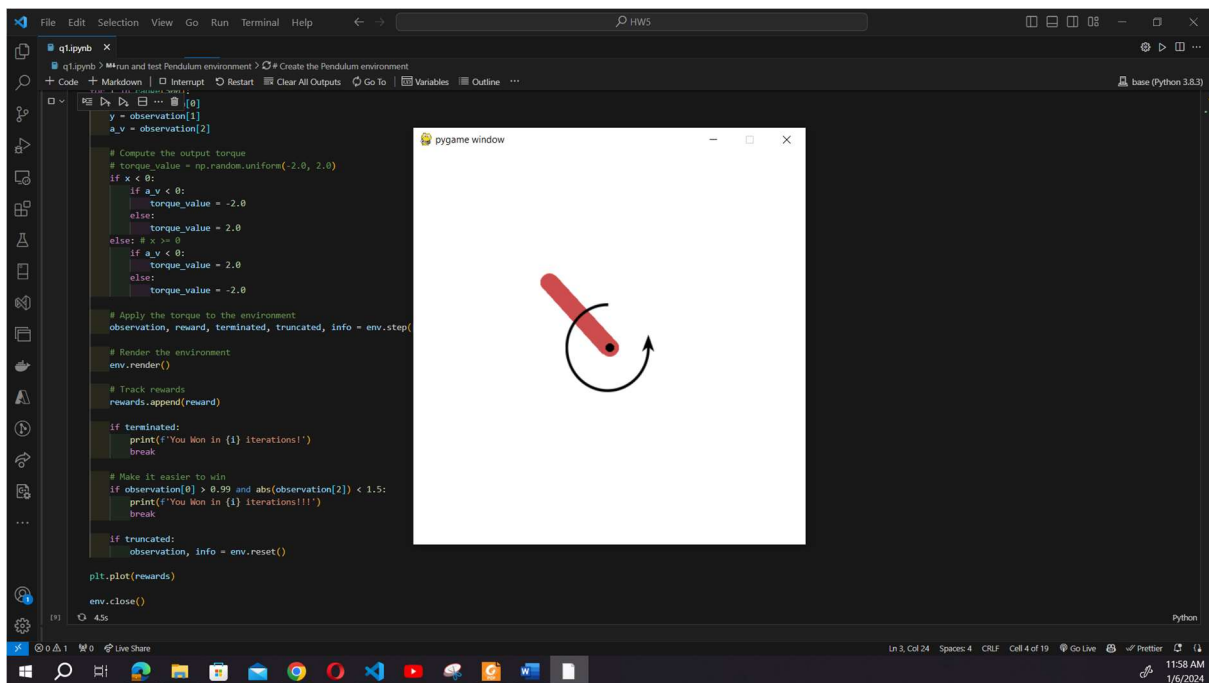
تمرین سری پنجم
درس مبانی هوش محاسباتی
دکتر ناصر مزینی

فرزان رحمانی
۹۹۵۲۱۲۷۱

سوال اول

(مرحله اول)

با استفاده از راهنمایی های دستیاران آموزشی و خواندن مستندات وبسایت gym توانستیم این محیط گرافیکی را اجرا کنیم و پنجره حرکت این بازو را نشان دهیم. در ابتدا action را به صورت عدد تصادفی از -2 تا 2 در نظر گرفتیم. برای ران کردن اگر بکچ های مطلوب را نداشته باشید نیاز به این دستور زیر دارید: `pip install gym numpy scikit-fuzzy matplotlib`



همچنین در هنگام تست اولیه حالت بردن بازی را ساده تر در نظر گرفتیم (با توجه به بحث های مطرح شده داخل گروه درسی) و با اعمال چند شرط ساده توانستیم بازی را ببریم و نمودار پاداش را رسم کنیم. این شرط های به شرح زیر هستند:

- در نیمه پایین آونگ ($x < 0$)، گشتاور را طوری اعمال کنیم که موجب افزایش سرعت زاویه ای شود. (جهت گشتاور و سرعت زاویه ای یکی باشد)
- در نیمه بالای آونگ ($x \geq 0$)، گشتاور را طوری اعمال کنیم که موجب کاهش سرعت زاویه ای شود تا به تعادل در حالت عمود برسیم. (جهت گشتاور و سرعت زاویه ای یکی نباشد)

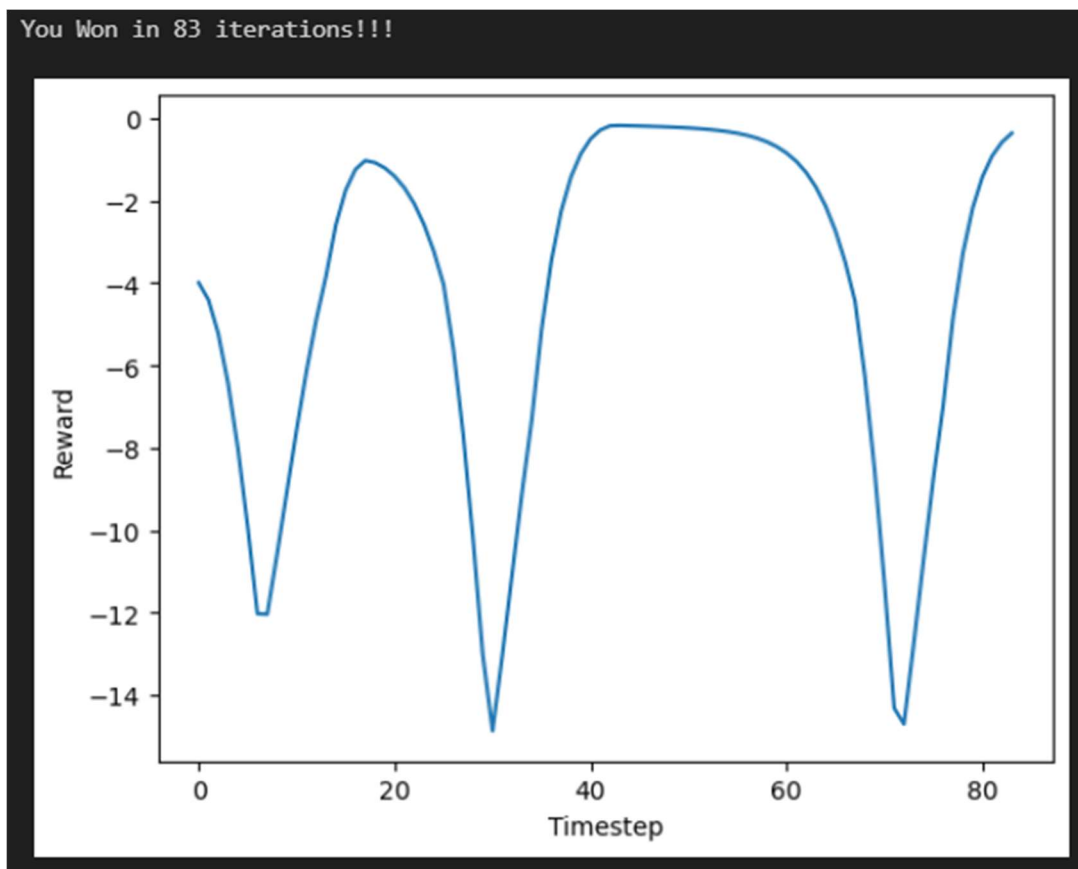
کد حالت ساده برای برزنده شدن (اگر مختصات x رو به بالای 0.99 برسوینید در حالیکه قدر مطلق سرعت بازو کمتر از $1/5$ باشد، مسئله رو حل شده در نظر بگیرید):

```
if observation[0] > 0.99 and abs(observation[2]) < 1.5:
    print(f'You Won in {i} iterations!!!')
    break
```

شرط های اولیه برای بردن بازی (بدون منطق فازی):

```
if x < 0:
    if a_v < 0: # a_v is angular velocity
        torque_value = -2.0
    else:
        torque_value = 2.0
else: # x >= 0
    if a_v < 0:
        torque_value = 2.0
    else:
        torque_value = -2.0
```

نمودار پاداش ها:

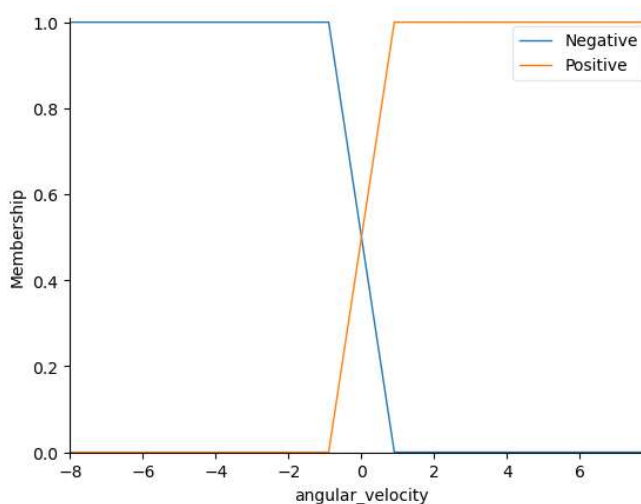
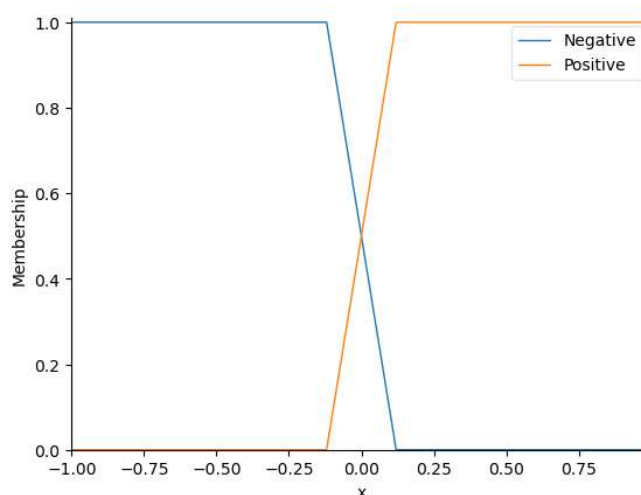


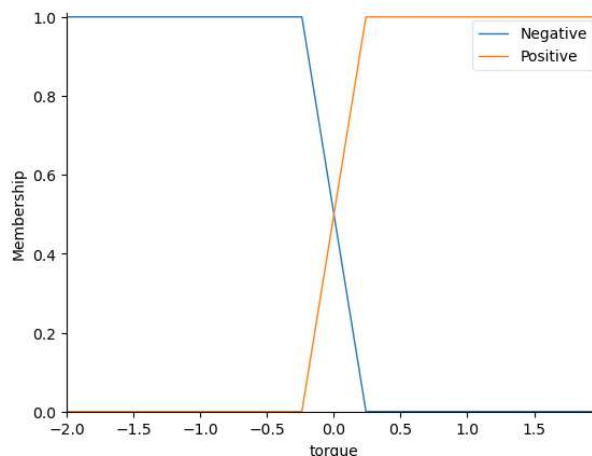
مرحله دوم)

این مرحله که شامل پیاده سازی هست با کمک کدی که از طرف دستیاران آموزشی در اختیار ما قرار داده شد پیاده سازی شد. با مراجعه به نوتبوک q1.ipynb میتوانید با خواندن کامنت ها و markdown های گذاشته شده مشاهده کنید که هر قسمت کد چه کاری را انجام می دهد. در نوتبوک به طور کامل تشریح داده شده است ولی ما اینجا به طور خلاصه توضیح می دهیم.

تعریف متغیر های زبانی، توابع عضویت و نحوه بازه بندی آنها

با توجه به توضیحاتی که در ابتدای سوال دادم و تحلیل سوال به صورت فیزیکی دریافتیم که برای حل این مسئله نیاز است تا آونگ را از نیمه پایینی ($x < 0$) به نیمه بالایی ($x > 0$) ببریم سپس سرعت آن را در نیمه بالایی کم کنیم تا به حالت تعادل در حالت عمود برسد. پس برای حل این مسئله نیازی به تعریف متغیر زبانی y نداریم. لذا برای ترم های پیشین در قوانین (Antecedent) فقط نیاز به تعریف متغیر زبانی مکان x و سرعت زاویه ای (angular_velocity) داریم. همچنین با توجه به اینکه فقط یک action در این محیط وجود دارد لذا برای ترم های پسین در قوانین (Consequent) فقط گشتاور (torque) را تعریف می کنیم. برای MBF ها ابتدا سه حالت مثبت، سفر و منفی در نظر گرفتم و سعی کردم با قوانینی آن را حل کنم ولی به مرور هیچ مقداری در هنگام defuzzification تولید نمی شود خوردم. لذا با توجه به درک مسئله و تجربه حل سوال به صورت ساده در ابتدای سوال تصمیم گرفتم برای هر یک از این سه متغیر زبانی دو حالت مثبت و منفی در نظر بگیرم. همان طور که در ادامه می بینید در نواحی اطراف صفر مقادیر مثبت و منفی به صورت fuzzy در هم آمیخته می شوند ولی با فاصله گرفتن به خلوص در مثبت و منفی بودن می رسم. برای بازه بندی، ابتدا بازه عدم قطعیت بین مثبت و منفی را خیلی بزرگ گرفتم که نتیجه خوبی حاصل نشد لذا تصمیم بازه را کوچک تر کردم تا مسئله حل شود. در زیر نمودار های متغیر های زبانی و توابع عضویت را می توانید ببینید:





نحوه تعریف قوانین

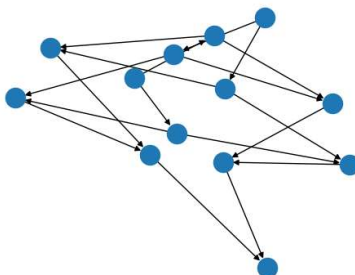
برای تعریف قوانین با بررسی مسئله به صورت فیزیکی و تحلیل آن متوجه شدیم که برای بردن پاندول به بالا نیاز به حرکت شتاب زاویه ای تند شونده در نیمه پایین ($x < 0$) و برای نگه داشتن و حفظ تعادل در حالت عمود نیاز به حرکت شتاب زاویه کند شونده در نیمه بالا ($x > 0$) داریم. همچنین برای حالت حدی $x=0$ فرقی ندارد کدام حالت را در نظر بگیریم (چون یک اپسیلون در حالت پیوسته است) لذا ما حرکت شتاب زاویه کند شونده را برای آن در نظر گرفتیم. به طور خلاصه:

- در نیمه پایین آونگ ($x < 0$)، گشتاور را طوری اعمال کنیم که موجب افزایش سرعت زاویه ای شود. (جهت گشتاور (شتاب زاویه ای) و سرعت زاویه ای یکی باشد)
- در نیمه بالای آونگ ($x \geq 0$)، گشتاور را طوری اعمال کنیم که موجب کاهش سرعت زاویه ای شود تا به تعادل در حالت عمود برسیم. (جهت گشتاور (شتاب زاویه ای) و سرعت زاویه ای خلاف هم باشد)

کد قوانین به صورت زیر است:

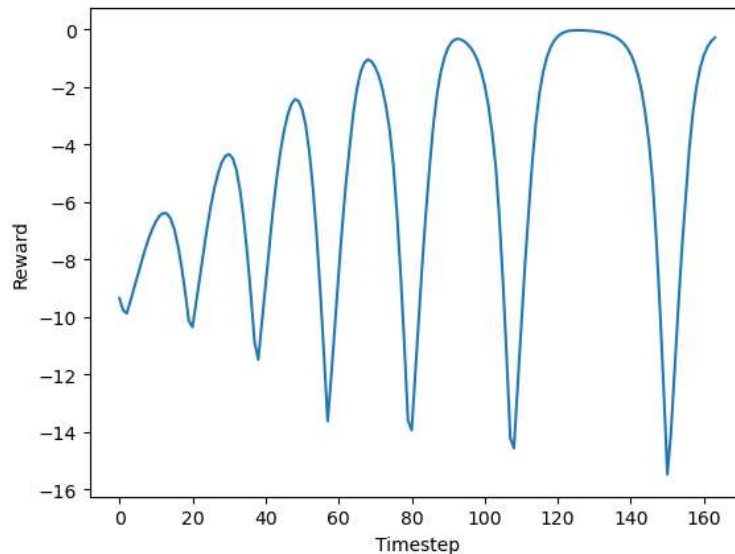
```
# define rules for the fuzzy system
rules = [
    # if x > 0 then decelerate (angular_velocity and torque have the opposite sign)
    ctrl.Rule(x['Positive'] & angular_velocity['Positive'], torque['Negative']),
    ctrl.Rule(x['Positive'] & angular_velocity['Negative'], torque['Positive']),
    # if x < 0 then accelerate (angular_velocity and torque have the same sign)
    ctrl.Rule(x['Negative'] & angular_velocity['Negative'], torque['Negative']),
    ctrl.Rule(x['Negative'] & angular_velocity['Positive'], torque['Positive']),
]
```

همچنین خروجی دستور `control_system.view()` به صورت زیر است:



نمودار پاداش های دریافتی

در هنگام اجرای شبیه سازی در محیط gym یک آرایه خالی با نام rewards طراحی میکنیم. بار هر action ای که در هر step زمانی برمیداریم، محیط چندین خروجی از جمله reward به ما می دهد و آن را در rewards ذخیره میکنیم. در نهایت نمودار آن را رسم میکنیم. بسته به حالت تصادفی اولی که انتخاب می شود معمولا از ۳۰ تا ۲۰۰ تکرار بردن بازی طول می کشد. مثلا در نمودار زیر با ۱۲۷ تکرار بازی را بردیم.



تحلیل نمودار پاداش های دریافتی

ابتدا تعریف پاداش های دریافتی که در سایت gym آمده است را با هم بررسی کنیم:

The reward function is defined as:

$$r = -(theta^2 + 0.1 * theta_dt^2 + 0.001 * torque^2)$$

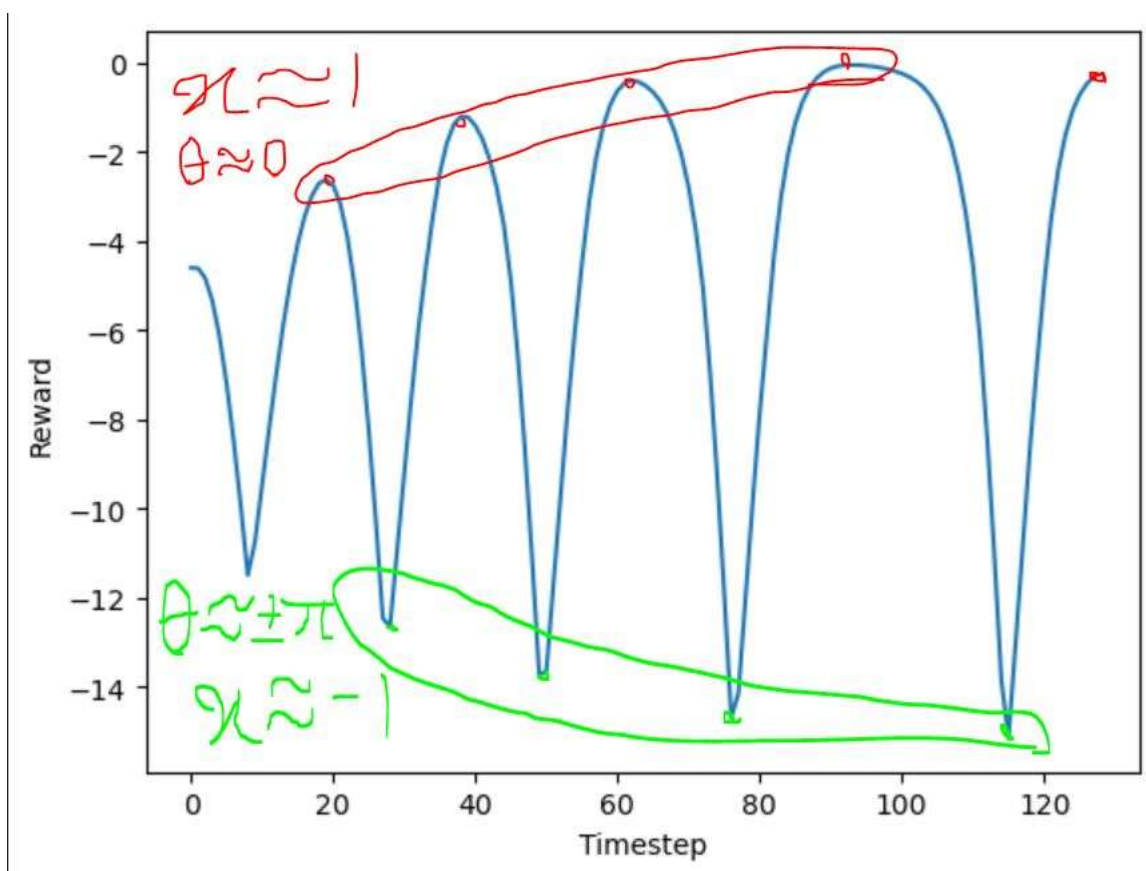
where θ is the pendulum's angle normalized between $[-\pi, \pi]$ (with 0 being in the upright position). Based on the above equation, the minimum reward that can be obtained is $-(\pi^2 + 0.1 * 8^2 + 0.001 * 2^2) = -16.2736044$, while the maximum reward is zero (pendulum is upright with zero velocity and no torque applied).

که در آن θ زاویه پاندول، θ_dt مشتق زاویه پاندول نسبت به زمان یا همان سرعت زاویه ای است و torque گشتاور اعمال شده به سر آزاد پاندول می باشد. همان طور که میبینید توان دو باعث می شود که علامت + و - تاثیری نداشته باشد و فقط قدر مطلق مقادیر در محاسبه پاداش (negative of cost) نقش دارند. سه عامل با ضرایب مختلف در محاسبه پاداش دخیل هستند:

- زاویه (θ) یا همان مکان x که در بازه $[-\pi, \pi]$ هست. با فاصله گرفتن از حالت عمود ($\theta = 0$ یا $x = +1$) پاداش منفی (ضرر) کسب میکنیم.
- مشتق زاویه پاندول نسبت به زمان (θ_dt) یا همان سرعت زاویه ای که در بازه $[-8, +8]$ هست. با فاصله گرفتن از سرعت زاویه ای صفر (حالت پایدار و تعادل) حرکت به سمت $+8, -8$ پاداش منفی (ضرر) بیشتری کسب میکنیم.
- گشتاور اعمال شده به سر آزاد (torque) که در بازه $[-2, +2]$ هست. با فاصله گرفتن از گشتاور صفر (حالت پایدار و تعادل) حرکت به سمت $+2, -2$ پاداش منفی (ضرر) بیشتری کسب میکنیم.

حال به تحلیل نمودار می پردازیم:

این نمودار، پاداش‌ها را در مراحل مختلف برای حل مسئله آونگ (پاندول) با استفاده از کنترل‌کننده منطق فازی نشان می‌دهد. محور x نشان دهنده مراحل زمانی (timesteps) و محور y نشان دهنده پاداش‌ها است. نمودار نشان می‌دهد که پاداش‌ها در تمام مراحل زمانی منفی هستند (همان‌طور که در بالا با توجه به تعریف توضیح داده شد). در ابتدا، مقادیر پاداش تقریباً بین -1 و -14 تا حدود گام زمانی ۸۵ در حال نوسان هستند (به مرور زمان بازه مقادیر نوسان بیشتر می‌شود). تا مرحله ۸۵ که نوسان انجام می‌شود پاندول تلاش میکند تا با این چرخش نوسانی نیرو ذخیره کند تا بتواند به بالا (حالت عمود برسد). سپس تقریباً از مرحله زمانی ۸۵ تا ۱۰۵، افزایش قابل توجهی در مقدار پاداش وجود دارد (پاندول در نزدیکی حالت عمود در حالت تقریباً تعادل است). از زمان ۱۰۵ تا ۱۲۰ پاندول یک بار دیگر به پایین می‌آید و با ذخیره کردن انرژی و شتاب دوباره این بار به حالت عمود می‌رسد و تعادل خود را حفظ می‌کند. در حدود مرحله زمانی ۱۲۵ پاداش به نزدیک ۰ (بیشترین مقدار ممکن) می‌رسد. این نشان دهنده حل شدن مسئله است و نشان می‌دهد کنترل‌کننده منطق فازی در طول زمان بهتر عمل میکند تا مسئله حل شود. در جاهایی که با قرمز در شکل زیر مشخص شدند (پاداش بیشتر) x نزدیک به $+1$ و قدر مطلق سرعت زاویه ای کم است. در جاهایی که با سبز در شکل زیر مشخص شدند (پاداش کمتر) x نزدیک به -1 و قدر مطلق سرعت زاویه ای زیاد است.



مراجع:

<https://chat.openai.com/>
<https://bard.google.com/>
<https://claude.ai/chats>

سوال دوم

Fuzzy C-Means (FCM) یک الگوریتم خوشه‌بندی بدون نظارت است که به نقاط داده اجازه می‌دهد به خوشه‌های متعدد با درجات مختلف عضویت تعلق داشته باشند. این یک ورژن توسعه یافته از الگوریتم خوشه‌بندی k-means است که از منطق فازی برای تخصیص این مقادیر عضویت به جای تخصیص دقیق آنها به یک خوشه یا خوشه سخت (hard clustering) استفاده می‌کند. این الگوریتم فازی (FCM) یک روش خوشه‌بندی iterative است که در داده کاوی و تشخیص الگو هم استفاده می‌شود. هدف آن تقسیم یک مجموعه داده به خوشه‌ها بر اساس شباهت نقاط داده است.

این الگوریتم می‌تواند در مقایسه با خوشه‌بندی سخت سنتی (k-means) که در آن به هر نقطه یک برچسب واضح و دقیق اختصاص داده می‌شود بسیار قدرتمندتر باشد. این الگوریتم با اختصاص عضویت به هر نقطه داده مربوط به هر مرکز خوشه بر اساس فاصله بین مرکز خوشه و نقطه داده کار می‌کند. هر چه داده‌ها به مرکز خوشه نزدیک باشند، عضویت آن در مرکز خوشه‌ای خاص بیشتر است. واضح است که مجموع عضویت هر نقطه داده باید برابر با یک باشد.

این یک الگوریتم خوشه‌بندی بدون نظارت است که به ما امکان می‌دهد یک پارتیشن فازی از داده‌ها بسازیم. الگوریتم به پارامتر m بستگی دارد که با درجه مبهم بودن جواب مطابقت دارد. مقادیر بزرگ m کلاس‌ها را محو می‌کند و همه عناصر به همه خوشه‌ها تعلق دارند. راه حل‌های مسئله بهینه‌سازی به پارامتر m بستگی دارد. یعنی انتخاب‌های مختلف m معمولاً منجر به پارتیشن‌های متفاوتی می‌شود.

الف) نحوه عملکرد FCM:

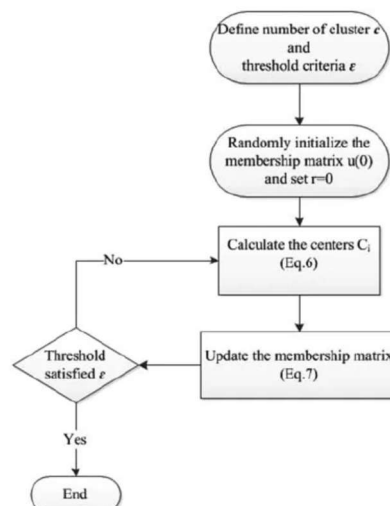
جریان فرآیند fuzzy c-means فازی در زیر برشمرده شده است:

۱. تعداد ثابتی از خوشه‌ها k را فرض کنید.
۲. مقداردهی اولیه (Initialization): به طور تصادفی μ_k k-means مرتبط با خوشه‌ها را مقداردهی کنید و احتمال اینکه هر نقطه داده x_i از یک خوشه معین k باشد (Membership Calculation) را محاسبه کنید. $P(\text{point } x_i \text{ has label } k | x_i, k)$
۳. تکرار (Iteration): مرکز خوشه (Centroid Update) را مجدداً به عنوان مرکز وزنی با توجه به احتمالات عضویت همه نقاط داده x_i محاسبه کنید:

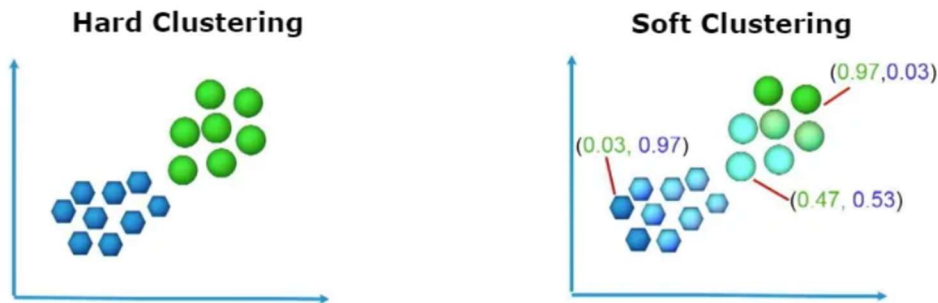
$$\mu_k(n+1) = \frac{\sum_{x_i \in k} x_i * P(\mu_k | x_i)^b}{\sum_{x_i \in k} P(\mu_k | x_i)^b}$$

۴. خاتمه: تکرار تا زمان همگرایی یا تا رسیدن به تعداد تکرار مشخص شده توسط کاربر ادامه دهید (تکرار ممکن است در حداکثر یا حداقل محلی به دام افتاده باشد).

همچنین در ادامه flow-chart این فرایند آمده است:



تفاوت FCM با K-Means:



بیاپید این دو الگوریتم را با هم مقایسه کنیم تا ایده روشنی از جایی که الگوریتم FCM در آن قرار می گیرد به دست آوریم.

۱. انتساب به یک خوشه (Attribution to a cluster): در خوشه بندی فازی، هر نقطه احتمال تعلق به هر خوشه را دارد، نه اینکه کاملاً به یک خوشه تعلق داشته باشد، مانند چیزی که در k-means سنتی وجود دارد. در خوشه بندی Fuzzy-C Means، هر نقطه دارای یک وزن مرتبط با یک خوشه خاص است، بنابراین یک نقطه به اندازه ای که ارتباط ضعیف یا قوی با خوشه دارد، در یک خوشه قرار نمی گیرد، که با معکوس فاصله از با مرکز خوشه تعیین می شود.
۲. سرعت: FCM نسبت به K-Means کندتر عمل می کند، زیرا در واقع کار بیشتری انجام می دهد. هر نقطه با هر خوشه ارزیابی می شود و عملیات بیشتری در هر ارزیابی درگیر می شود. K-Means فقط باید یک محاسبه فاصله انجام دهد، در حالی که FCM باید یک وزن دهی کامل با مقادیر معکوس فاصله ها انجام دهد.
۳. نظر شخصی نویسنده: FCM/Soft-K-Means نسبت به Hard-K-Means "کمتر احمقانه" است وقتی که خوشه های دراز داریم (زمانی که نقاطی که در ابعاد دیگر سازگار هستند تمایل دارند در طول یک یا دو بعد خاص پراکنده شوند).
۴. FCM به انتخاب پارامتر فازی 'm' حساس است.

ما باید متوجه باشیم که FCM یک مورد خاص از K-means است وقتی که تابع احتمال مورد استفاده به سادگی ۱ است اگر نقطه داده به یک مرکز نزدیکتر باشد و در غیر این صورت ۰ است.

ب) اجرای Fuzzy C-Means در پایتون با skfuzzy:

در این قسمت با پیاده سازی موارد خواسته شده با نحوه کارکرد این الگوریتم آشنا شدیم. با مراجعه به نوتبوک q2.ipynb میتوانید با جزئیات بیشتر با آن آشنا شوید. در این نوتبوک کامنت و markdown هایی گذاشته ایم که با خواندن آنها متوجه جزئیات پیاده سازی شوید. در اینجا ما به طور خلاصه موارد انجام شده را توضیح می دهیم.

۱. از pandas برای خواندن "data1.csv" استفاده میکنیم.
۲. ویژگی ها را عادی یا استاندارد میکنیم.
۳. از الگوریتم FCM از کتابخانه skfuzzy برای انجام خوشه بندی برای مقادیر 'c' از ۲ تا ۱۰ استفاده میکنیم.
۴. با اختصاص دادن نقاط داده به خوشه ای که به احتمال زیاد به آن تعلق دارند، خوشه ها را برای هر مقدار 'c' مجسم میکنیم.
۵. ضریب پارتیشن فازی (FPC: Fuzzy Partition Coefficient) را برای هر مقدار 'c' محاسبه میکنیم. FPC فشردگی و جدایی خوشه (cluster compactness and separation) را اندازه گیری می کند.
۶. همین کار ها را برای "data2.csv" انجام می دهیم.

معیار FPC به انتخاب تعداد بهینه خوشه ها با در نظر گرفتن مبادله بین انسجام و جداسازی خوشه (cluster cohesion and separation) کمک می کند. مقادیر بالاتر FPC نشان دهنده خوشه های با تعریف بهتر است.

پس از محاسبه FPC برای هر مقدار 'c'، بهترین تعداد خوشه ها را می توان بر اساس بالاترین مقدار FPC انتخاب کرد. سپس، نقاط داده را می توان به خوشه های مربوطه آنها برای بهترین مقدار 'c' اختصاص داد.

توضیح مختصر راجع به FPC: Fuzzy Partition Coefficient

ضریب تقسیم فازی (FPC) معیاری است که برای ارزیابی کیفیت خوشه بندی در الگوریتم های خوشه بندی فازی مانند Fuzzy-C Means (FCM) استفاده می شود. درجه ابهام (degree of fuzziness) یا قدرت پارتیشن بندی خوشه ها در مجموعه داده شده را اندازه گیری می کند.

FPC بر اساس مقادیر عضویت تولید شده توسط الگوریتم خوشه بندی محاسبه می شود. این مقادیر عضویت نشان دهنده میزان تعلق نقاط داده به هر خوشه است. مقادیر عضویت بالاتر دلالت بر ارتباط قوی تر یا قطعیت بالاتر یک نقطه داده متعلق به یک خوشه خاص دارد.

فرمول FPC شامل محاسبه نسبت مقادیر عضویت واقعی نقاط داده به حداکثر مقدار عضویت ممکن است که یک نقطه داده می تواند در هر خوشه ای داشته باشد.

FPC بین ۰ و ۱ است ($0 \leq FPC \leq 1$). هنگامی که FPC به ۱ نزدیک می شود، نشان می دهد که نقاط داده دارای عضویت بالایی در خوشه های اختصاص داده شده خود و عضویت کم در سایر خوشه ها هستند، که نشان دهنده جدایی واضح بین خوشه ها است. برعکس، مقادیر پایین تر FPC حاکی از همپوشانی یا ابهام بیشتر در انتساب های خوشه ای است. هر چه مقدار FPC بیشتر و نزدیک تر به ۱ باشد برای ما مطلوب تر است.

در عمل، ضمن ارزیابی تعداد مختلف خوشه ها (تغییر پارامتر c در FCM)، تعداد خوشه هایی که مقدار FPC را به حداکثر می رساند، اغلب به عنوان تعداد بهینه خوشه ها برای مجموعه داده داده شده در نظر گرفته می شود.

مراجع:

<https://chat.openai.com/>

<https://bard.google.com/>

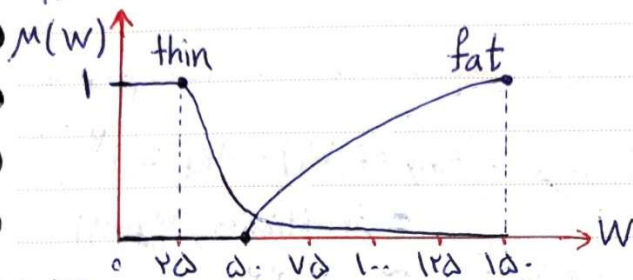
<https://claude.ai/chats>

<https://medium.com/geekculture/fuzzy-c-means-clustering-fcm-algorithm-in-machine-learning-c2e51e586fff>

<https://pythonhosted.org/scikit-fuzzy/api/skfuzzy.cluster.html>

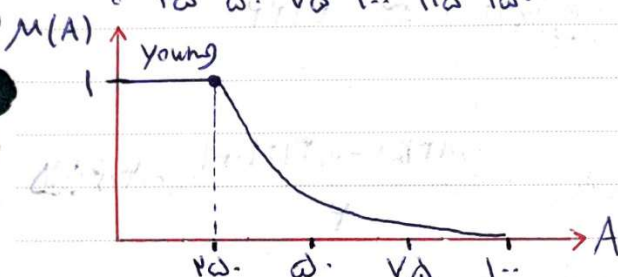
Subject
Date

متغیر زبانی وزن را با \tilde{W} و متغیر زبانی سن را با \tilde{A} نشان می دهیم.



ابترافنودارهای توابع

عضویت را رسم می کنیم.



سپس برای هر شخص میزان

عضویت آن در هر متغیر زبانی

را نشان می دهیم. تفراول را با

اندیس ۱ و نفردوم را با اندیس ۲ نمایش می دهیم.

تفراول: $W_1 = 55, A_1 = 45$

نفردوم: $W_2 = 95, A_2 = 4$

$$\mu_{thin}(W_1) = \left(1 + \left(\frac{55-25}{5}\right)^2\right)^{-1} \approx 0,027$$

$$\mu_{thin}(W_2) = \left(1 + \left(\frac{95-25}{5}\right)^2\right)^{-1} \approx 0,005$$

$$\mu_{fat}(W_1) = \left(1 - \left(\frac{55-150}{100}\right)^2\right) \approx 0,975$$

$$\mu_{fat}(W_2) = 1 - \left(\frac{95-150}{5}\right)^2 \approx 0,9975$$

$$\mu_{young}(A_1) = \left(1 + \left(\frac{45-25}{5}\right)^2\right)^{-1} \approx 0,059$$

$$\mu_{young}(A_2) = \left(1 + \left(\frac{4-25}{5}\right)^2\right)^{-1} \approx 0,02$$

الف) نفردوم نسبتاً چاق تر و جوان تر از تفراول است. $P =$ تفراول > نفردوم
 $u > v$

تابع عضویت صفت تفضیلی "تر" را به صورت روبه رو در نظری بگیریم.

(به صورت خطی بدتر حالت را معروض بهترین حالت را در نظری بگیریم.)

P4PCO

u	v	$\mu_c(u,v) \approx u > v$
1	0	1
0	1	0

گزاره داده شده شامل دو بخش است و بین آنها و (AND) وجود دارد پس

هر کدام را جدا حساب می کنیم و سپس بین آنها min می گیریم.

$$M = \text{فردوم نسبتاً چاقتر از فراول است.} \quad \left. \begin{aligned} \mu_{\text{fairly fat}}(W_1) &= \sqrt{\mu_{\text{fat}}(W_1)} \\ &= \text{dilation}(\mu_{\text{fat}}(W_1)) \\ &\approx 0,3122 \end{aligned} \right\}$$

$$\mu_{\text{fairly fat}}(W_2) = \text{dilation}(\mu_{\text{fat}}(W_2)) = \frac{\mu_{\text{fat}}(W_2)}{\mu_{\text{fat}}(W_2)^2} = \frac{1}{\mu_{\text{fat}}(W_2)}$$

$$= 0,1351$$

$$M = \mu_c(\mu_{\text{fairly fat}}(W_2), \mu_{\text{fairly fat}}(W_1)) = \frac{0,1351 - 0,3122 + 1}{2} \approx 0,4115$$

$N = \text{فردوم جوان تر از فراول است}$

$$N = \mu_c(\mu_{\text{young}}(A_2), \mu_{\text{young}}(A_1)) = \frac{0,02 - 0,059 + 1}{2} \approx 0,4805$$

$$P = M \wedge N = \min(0,4115, 0,4805) = 0,4115$$

\min
(AND)

$$\mu_{\text{truth}}(P) = 0,4115$$

یعنی اگر فراول خیلی لاغر باشد آنگاه فردوم نسبتاً جوان است.

$$\underbrace{\mu_{\text{very thin}}(W_1)}_L \rightarrow \underbrace{\mu_{\text{fairly young}}(A_2)}_K \quad \text{معادل} \quad L \rightarrow K$$

$$L = \mu_{\text{very thin}}(W_1) = \text{concentration}(\mu_{\text{thin}}(W_1)) = (0,027)^2 = 0,000729$$

$$K = \mu_{\text{fairly young}}(A_2) = \text{dilation}(\mu_{\text{young}}(A_2)) = \sqrt{0,02} = 0,1414$$

$$\mu_{\text{implication}}(L, K) = \min(L, K) = \min(0,000729, 0,1414) = 0,000729$$

از روش Mamdani استفاده کردیم

$$\mu_{\text{truth}}(L \rightarrow K) = 0,000729$$

پایان