

به نام خدا

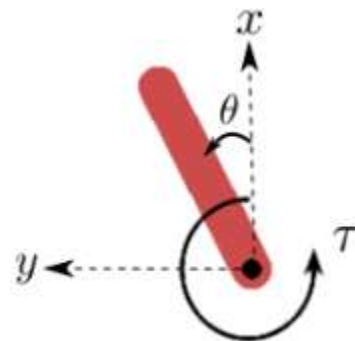
امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

گزارش تمرین سری ۵

(۱)

متغیرهای زبانی را بر اساس observation space تعریف کردیم که شامل x, y و سرعت زاویه ای بودند و بازه آنها طبق داک مشخص بوده است. نحوه بازه بندی را به گونه ای انتخاب کردیم که نقاط اشتراکی آنها کوچکتر باشد تا اگر در موقعیتی بودیم دارای قطعیت بیشتری باشیم البته این مقادیر بیشتر با آزمون و خطا به دست آمد که هر چقدر بازه اشتراکی کوچکتر باشد دارای عملکرد بهتری هم هستیم

البته گشتاور هم از متغیرهای زبانی بود که مربوط به اکشن اسپیس میباشد و آنرا نیز قرار دادیم. نکته ای که در رابطه با متغیرهای زبانی وجود دارد این است که از عکس خود داک استفاده کردیم:



یعنی محور عمودی رو به بالا و محور افقی رو به چپ برای x, y هستند و مقادیر مثبت گرفتند و عکس آنها دارای مقادیر منفی شد.

در رابطه با سرعت زاویه ای و گشتاور همانطور که در شکل مشخص است و در داک هم آورده

شده است ؛ حرکت خلاف عقربه های ساعت مثبت و عکس آن منفی در نظر گرفته شده است و ترکیبات مربوط به قوانین هم بر این اساس نوشته شده اند.

در رابطه با قوانین ابتدا با متدی رفتیم که دارای ۸ قانون هست و تمام ترکیبات از X, Y و سرعت زاویه ای را در آن دخیل کردیم ؛ و با این روش به نتیجه رسیدیم.

با کوتاه کردن روابط قوانین میتوانیم متوجه شویم که تنها با سرعت زاویه ای و پوزیشن X میتوانیم مسئله را به نتیجه برسانیم. به این صورت که اگر در نیمه پایینی حرکت دایره شکل خود باشیم سرعت زاویه ای و گشتاور هم راستا میشوند و مطابق هم باید عمل کنیم تا به سمت بالا حرکت کنیم اما هنگامی که در نیمه بالایی هستیم یعنی به جایی که به نقطه اوج نزدیک یا همان $x=0.99$ هستیم باید سرعت زاویه ای و گشتاور خلاف هم عمل کنند. و در نهایت با ۴ قانون توانستیم به مطلوب مسئله برسیم.

نمودار پاداش در قسمت آخر کد آورده شده است.

در رابطه با تحلیل نمودار ریوارد همانطور که در داک توضیح داده شده است یک فرمول درجه دو برحسب تتا میباشد که علت منحنی بودن آن این است و علت منفی بودن آن طبق فرمول مشخص است چون در ابتدای آن یک منفی داریم. که در بهترین حالت مقدار ریوارد برابر صفر میشود یعنی جایی که ما به نقطه مطلوب رسیدیم و سرعت و گشتاور برابر صفر میشوند. البته چون شرط مسئله را کمی آسانتر کردیم مقدار ریوارد به صفر نمیرسد و به نزدیک آن میرسد.

علت اینکه میبینیم بازه ها به صورت تناوبی در هر منحنی بزرگتر شده اند این است که هر بار که اهرم پاندول را تاب میدهیم تا قسمتی بالا میرود و اگر به مطلوب نرسد همان حرکت را به صورت دایره ای برمیگردد و متوجه میشویم که هر بار مسافت بیشتر را طی میکند تا از سمت دیگر بالا برود و سرعت و گشتاور آن بیشتر است که همین باعث بزرگتر شدن هر منحنی میشود.

الف : به طور کلی clustering به پروسه گروه بندی یا دسته بندی دیتا هایی هستند که پترن های مشابه دارند اما لیبیل گذاری نشده اند یعنی در ابتدا مشخص نیست که دیتا ها با الگوی مشابه باید در یک گروه قرار بگیرند و ما با این عمل، به نوعی دسته بندی را برای دیتا ها با پترن های مشابه انجام میدهیم و هر دسته را یک cluster مینامیم.

دو روش برای کلاسترینگ داریم یکی روش کلاسیک یا hard clustering و دیگری روش فازی که به آن soft clustering هم میگوییم.

در کلاسترینگ به روش فازی به این صورت است که یک داده (data point) ممکن است لزوما عضو یک کلاستر نباشد و در درجه عضویتی در کلاستر دیگر نیز داشته باشد و منحصرأ عضو یک کلاستر خاص نباشد؛ یعنی مثلا نشانه های یک پترن را در چند کلاستر دیگر ببینیم اما در یک کلاستر بیشتر و در یک کلاستر کمتر در اینجاست که باید درجه عضویت تعریف کنیم و از کلاسترینگ به روش فازی استفاده میکنیم که در دنیای واقعی بسیاری از مسائل به این گونه میباشند.

برای محاسبه کلاسترینگ به روش فازی مراحل زیر را طی میکنیم:
ابتدا تعداد کلاستر ها، مقادیر فازی و مقادیر اولیه مرکزی (centroid) را برای هر کلاستر مشخص میکنیم.

مقادیر موجود در هر کلاستر را محاسبه میکنیم و سپس مقادیر جدیدی برای centroid هر کلاستر محاسبه میکنیم. و اینکار را چند بار تکرار میکنیم.

مقایسه fuzzy c-means vs k-means :

اولین تفاوت همانطور که قبلا گفته شد در k-means یک دیتاپوینت منحصرا مربوط به یک کلاستر است اما در کلاستر فازی درجه عضویت داریم و هر دیتاپوینت ممکن است به چند کلاستر مربوط شود.

مورد بعدی مربوط به سرعت در محاسبات است؛ میدانیم برای اینکه یک دیتاپوینت وارد یک کلاستر شود باید بر اساس فاصله آن تا مرکز آن کلاستر تصمیم گیری انجام دهیم. در الگوریتم k-means فاصله تا مرکز بر اساس فاصله اقلیدسی است اما در روش فازی محاسبات پیچیده تر هستند و باید مقادیر فازی نیز در آنها دخالت داده شده و محاسبه شوند.

$$V_i = \frac{\sum_{j=1}^N x_j * u_{ij}^m}{\sum_{j=1}^N u_{ij}^m} \quad u_{ij} = \frac{1}{\sum_{k=1}^C \frac{\|x_j - v_i\|^{2/(m-1)}}{\|x_j - v_k\|}}$$

فرمول های بالا مربوط به مراحل میانی است که در روش فازی باید طی کنیم. مشخصا سرعت k-means در محاسبات به دلیل پیچیدگی کمتر ، بالاتر خواهد بود.

توضیحات کد:

ابتدا فایل را با استفاده از دستور read_csv میخوانیم سپس از MinMaxScaler برای نرمال سازی آن استفاده میکنیم. در نرمال سازی میانگین برابر صفر و انحراف معیار برابر ۱ میشود و در کل دیتای ما از تنوع و پراکندگی کمتری برخوردار میشود و تمام دیتای موجود در X,Y به بازه ۰ تا ۱ میرسند.

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))  
X_scaled = X_std * (max - min) + min
```

نحوه محاسبه min max scaler طبق داک

سپس تابعی داریم که در آن عملیات فازی کلاسترینگ و پلات آنرا نمایش داده ایم و مقدار fpc را در آخر باز گردانده ایم.

قسمت ۴ : FPC

Fpc معیاری است برای کیفیت کلاسترینگ و اینکه مدل مان به چه میزان و با چه کیفیتی کلاسترینگ را انجام داده است یعنی هر دیتاپوینت با چه دقت و شفافیتی در هر کلاستر قرار گرفته اند؛ و محدوده آن بین ۰ و ۱ میباشد که در آن ۱ بهترین حالت میباشد و هنگامی که کلاسترینگ خود را با چند مقدار متفاوت (مثلا در سوال گفته از ۲ تا ۱۰) انجام میدهم بیشترین fpc نشان دهنده اینست که کدام کلاسترینگ برای دیتاهای دریافتی ما بهترین است.

در فایل data1 بهترین کلاسترینگ مربوط به $c=2$ میباشد. و روند مقادیر fpc همانطور که در نمودار انتهایی نشان داده شده است به صورت نزولی است.

در فایل data2 بهترین کلاسترینگ مربوط به $c=2$ است و مثل قبل حالت نزولی دارد.