گزارش کار تمرین سوم

درس یادگیری ماشین

استاد درس: دکتر کمندی

دانشجویان گروه:

محیا معتمدی ۸۱۰۸۹۷۰۵۳

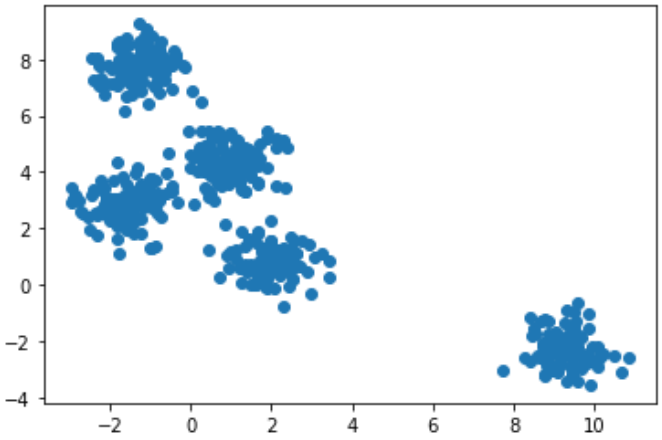
مینو احمدی ۸۱۰۸۹۷۰۳۲

دانشکده علوم مهندسی، دانشگاه تهران

بهار ۱۴۰۱

پیاده سازیK-means

برای پیاده سازی این الگوریتم ابتدا داده ها را خواندیم سپس به کمک scatter دیتا ها را رسم کردیم.



همان طور که دیده میشود از ابتدا تا حدودی مشخص است که دیتا شامل پنج دسته مختلف هستند که ما میخواهیم به کمک این الگوریتم در نهایت به طور کامل ان ها را از هم جدا کنیم و به دسته بندی مشخص تقسیم کنیم.

ابتدا به کمک dataframe.sample() داده ها شافل میکنیم.

سپس تابع **find_clusters()** را پیاده سازی کردیم برای این کار ابتدا دیکشنری ای به نام clusters ساخته ایم سپس داخل ان را با پنج لیست پر کرده ایم.
سپس پنج تا از نقطه ها را به صورت رندوم به عنوان center ها انتخاب کرده ایم و فاصله ی هر نقطه (هر دیتا) را از هر یک از center ها محاسبه کرده ایم (پنج center داریم) و با توجه به ان در خوشه مربوط به ان center قرار داده ایم.
پیاده سازی این تابع در زیر امده است :

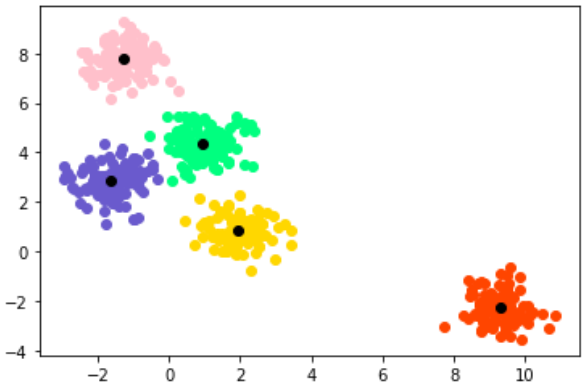
```
def find_clusters(X, centers, k):
    clusters = {}
    for i in range(k):
        clusters[i] = []
    for data in X:
        distance = []
        for j in range(k):
            distance.append(np.linalg.norm(data - centers[j]))
        clusters[distance.index(min(distance))].append(data)
    return clusters
```

در مرحله بعد تابع **find_centers()** را پیاده سازی کرده ایم که در ان میانگین داده های هر دسته را محاسبه می کنیم و این بار میانگین را به عنوان center های جدید انتخاب میکنیم.
پیاده سازی این تابع در زیر امده است :

```
def find_centers(centers, clusters, k):
    for i in range(k):
        centers[i] = np.average(clusters[i], axis=0)
    return centers
```

تابع بعدی **k_means()** است در ان یک سری center تعیین کرده ایم که این center ها داده ی اول تا پنجم ماست ولی چون از ابتدا داده ها را شافل کرده ایم و سپس پنج تای اول را انتخاب کردیم پس انگار که پنج داده ی رندوم برای مرکز ها انتخاب شده اند.
در تابع تا زمانی که سنتر های جدید ایجاد شده در هر بار، با سنتر های قبلی یکسان نباشند(که این یکسان بودن با تابع check_notequal() بررسی شده

است)، طبق الگوریتم دوباره سنتر ها و کلاستر ها را محاسبه میکنیم و سپس نتیجه نهایی را چاپ میکنیم که نتیجه نهایی به شکل زیر میباشد:
(نتیجه نهایی درپنج itrations رخ داد)
داده های مربوط به هر دسته با رنگ به خصوصی مشخص شده است.



در اینجا میبینیم که داده ها به پنج دسته تقسیم شده اند .
پیاده سازی این تابع در زیر امده است :

```
def k_means(X, k):
    centers = {}
    new_centers = {}
    c = 0
    for i in range(k):
        new_centers[i] = X[i]
        centers[i] = np.array([])
    while check_notequal(centers, new_centers):
        c +=1
        centers = copy.deepcopy(new_centers)
        clusters = find_clusters(X, new_centers, k)
        new_centers = find_centers(new_centers, clusters, k)
    return(centers, clusters, c)
```