

## Regularized Kmeans

الگوریتم Kmeans یک روش ساده و قدرتمند برای دسته‌بندی داده‌های بدون برچسب است. در این قسمت می‌خواهیم دسته‌بندی **Kmeans به همراه regularization** داشته باشیم. در این نوع خوشه‌بندی علاوه بر فاصله اقلیدسی نمونه‌ها از مراکز دسته‌ها، فاصله مرکز دسته از مبدا نیز مهم خواهد بود؛ به عبارت دیگر می‌خواهیم مراکز دسته‌ها به مبدا نزدیک باشند. به همین منظور در محاسبه و تخصیص نمونه‌ها به دسته  $k$  ام،  $\|C'_k\|$  را نیز به فاصله نمونه‌ها از مرکز این دسته اضافه می‌کنیم. بقیه مراحل خوشه‌بندی همانند ورژن اصلی Kmeans خواهد بود. تعداد نمونه‌های مسئله برابر ۳۰۰ است و هر نمونه دارای چهار ویژگی است؛ بنابراین در کل ۱۲۰۰ ردیف ورودی برای نشان دادن داده‌ها خواهیم داشت. ورودی دیگر مسئله تعداد کلاس‌ها  $K$  است و پس از آن به تعداد 4K ورودی برای نشان دادن محل اولیه مراکز دسته‌ها خواهیم داشت (هر ۴ ورودی نمایانگر یک مرکز دسته هستند). بدین ترتیب ورودی‌های کد شما به نحو زیر خواهند بود:

- ۱۲۰۰ ردیف اول، اطلاعات نمونه‌ها را در اختیار می‌گذارد؛ ۳۰۰ ردیف اول آن بیانگر مقدار ویژگی نخست برای تمام ۳۰۰ نمونه، ۳۰۰ ردیف بعد مقدار دومین ویژگی برای ۳۰۰ نمونه و ... خواهند بود. به طور مثال اگر ورودی‌ها را با  $I_n$  نشان دهیم، نمونه پنجم به شکل  $[I_5, I_{305}, I_{605}, I_{905}]$  قابل نمایش است.
- ردیف ۱۲۰۱ بیانگر تعداد دسته‌های مسئله  $K$  است.
- از ردیف ۱۲۰۲ به تعداد 4K ورودی خواهیم داشت که بیانگر مرکز اولیه دسته‌ها (برای initialization) هستند و هر ۴ ردیف متوالی مرکز یک دسته را نشان می‌دهد (مثلاً ورودی ۱۲۰۶ تا ۱۲۰۹ مرکز اولیه برای دسته دوم را نشان می‌دهد).

برنامه شما باید الگوریتم Kmeans را در ۱۰۰ تکرار اجرا کرده و به عنوان خروجی، مراکز دسته‌ها را در 4K (هر ۴ ردیف متوالی بیانگر مرکز یک دسته است) ردیف (تا دو رقم اعشار) بدهد.

در ادامه یک نمونه ورودی و خروجی برای فهم بیشتر آورده شده است. دقت کنید که در این سوال ورودی و خروجی از نوع استاندارد است.

### ورودی

می‌توانید یک نمونه از ورودی استاندارد را در اینجا ببینید.

## خروجی

می‌توانید یک نمونه از خروجی استاندارد را در [اینجا](#) ببینید.

در این مسئله می‌توانید از کتابخانه‌های *numpy* و *math* استفاده کنید.