# CV Lab9 Notes

TA: 郑浩 (RA in SUSTech CV Lab)

## Prerequisites

1. Python packages

    ```
    conda activate YOUR_ENV
    conda install matplotlib seaborn numpy
    ```

2. K-means Visualization: https://stanford.edu/class/engr108/visualizations/kmeans/kmeans.html

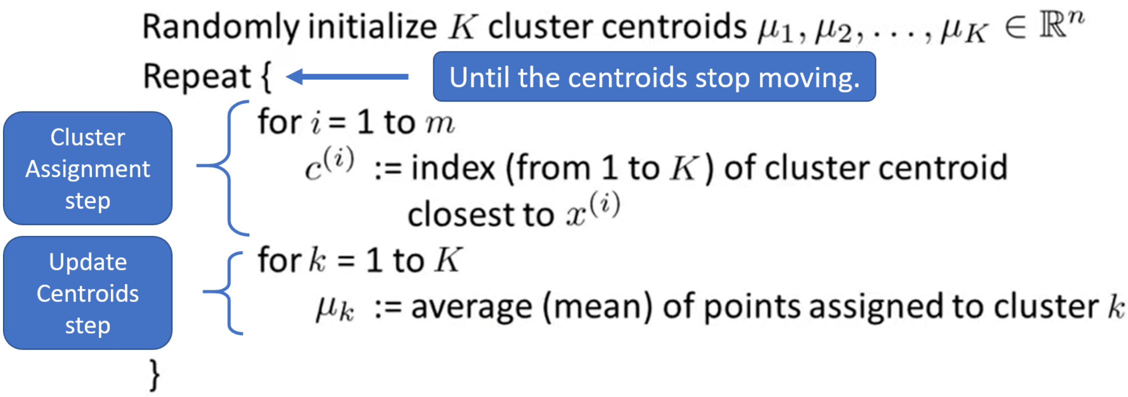
## K-means

1. 牧师-村民模型：

    > 有四个牧师去郊区布道，一开始牧师们随意选了几个布道点，并且把这几个布道点的情况公告给了郊区所有的居民，于是每个居民到离自己家最近的布道点去听课。
    > 听课之后，大家觉得距离太远了，于是每个牧师统计了一下自己的课上所有的居民的地址，搬到了所有地址的中心地带，并且在海报上更新了自己的布道点的位置。
    > 牧师每一次移动不可能离所有人都更近，有的人发现A牧师移动以后自己还不如去B牧师处听课更近，于是每个居民又去了离自己最近的布道点……
    > 就这样，牧师每个礼拜更新自己的位置，居民根据自己的情况选择布道点，最终稳定了下来。

2. Visualization demo

3. Algorithm




4. Shortcoming

    - Highly depends on the initialization of $K$ centroids, which may lead to arbitrarily bad clustering.

# K-means ++

1. Algorithm

### 2.2 The k-means++ algorithm

We propose a specific way of choosing centers for the `k-means` algorithm. In particular, let $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen. Then, we define the following algorithm, which we call `k-means++`.

1a. Take one center $c_1$, chosen uniformly at random from $\mathcal{X}$.
1b. Take a new center $c_i$, choosing $x \in \mathcal{X}$ with probability $\frac{D(x)^2}{\sum_{x\in\mathcal{X}} D(x)^2}$.
1c. Repeat Step 1b. until we have taken $k$ centers altogether.
2-4. Proceed as with the standard `k-means` algorithm.

# Reference


1. https://www.youtube.com/watch?v=hDmNF9JG3lo&ab_channel=MITOpenCourseWare
2. Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, *21*, 768-769.
3. https://en.wikipedia.org/wiki/K-means_clustering
4. https://en.wikipedia.org/wiki/K-means%2B%2B
5. Arthur, David, and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Stanford, 2006.