**DATA130030.01: Natural Language Processing** **05/24/2023**

# Lecture 12: Clustering — $k$-means

*Lecturer: Baojian Zhou* *The School of Data Science, Fudan University*

In general, clustering means that given a set of data points (embeddings of documents, paragraphs, or words in our case), partition them into groups containing very similar data points [1]. These clustered groups are meaningful, useful, or both [2]. $k$-means clustering is one of the methods to cluster a set of data points into $k$ different groups. It only uses the similarity function (characterized by $\ell_2$-norm) between data examples. In this notes, we discuss $k$-means in more details by answering the following questions:

- **What is the mathematical formulation of $k$-means ?**
- **Why $k$-means can stop ?**
- **Does $k$-means find global optimum ?**
- **How could we choose $k$?**
- **What is the run time of $k$-means ?**

Before introducing the $k$-means algorithm, we give some definitions first.

**Definition 12.1 ($\ell_2$-norm)** *Given any two data points* $\mathbf{x} := [x_1, x_2, \ldots, x_d]^\top$ *and* $\mathbf{y} = [y_1, y_2, \ldots, y_d]^\top$ *in* $\mathbb{R}^d$*, we define the distance function (i.e.* $\ell_2$*-norm)*

$$\|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}.$$

**Definition 12.2 (Centroid)** *Given a set of data points* $\mathcal{D} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ *where* $\mathbf{x}_i \in \mathbb{R}^d$*, the centroid of* $\mathcal{D}$ *is defined as*

$$\mu = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n}{n} = \frac{\sum_{i=1}^n \mathbf{x}_i}{n}.$$

*The centroid of a single cluster is the same as the sample mean of a cluster.*

**Definition 12.3 (Sample Mean/Variance)** *The sample mean* $\bar{\mathbf{x}}$ *of a set of observations* $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *is defined as*

$$\bar{\mathbf{x}} := \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n}{n} = \frac{\sum_{i=1}^n \mathbf{x}_i}{n}.$$

*The sample variance of a set of independent observations* $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *is*

$$Var(\mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2 = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}\|_2^2.$$

*Notice that the above definition of variance assumes the data observations are independent.*

The $k$-means clustering algorithm is as follows:

**$k$-means clustering**

```
1: select $k$ points as initial centroids, i.e., $\mu_1, \mu_2, \dots, \mu_k$
2: **Repeat:**
3:     For $j$, assign $C_j$ be empty. Then for each $i$, find $j := \arg\min_j \|\mathbf{x}_i - \mu_j\|_2^2$ and $C_j = C_j \cup \{i\}$
4:     update the centroid of each cluster $\mu_j = \frac{\sum_{i \in C_j} \mathbf{x}_i}{|C_j|}$
5: **until** centroids do not change
```

**Line 1:** There are two main methods to choose the initial centroids. The first one is randomly choose $k$ data samples from dataset $\mathcal{D}$ as the centroids. The other one is randomly to initialize $k$ points from $\mathbb{R}^d$. Of course, there are other popular methods.

**Line 3:** This is an assignment step. It assigns each $\mathbf{x}_i$ to the cluster whose centroid has the least squared Euclidean distance ($\ell_2$-norm). This is intuitively the "nearest" mean. Therefore, after this step, each cluster $C_j$ should be

$$C_j = \Big\{ i : \|\mathbf{x}_i - \mu_j\|_2^2 \le \min_{\tau : 1 \le \tau \le k} \|\mathbf{x}_i - \mu_\tau\|_2^2, \mathbf{x}_i \in \mathcal{D} \Big\}.$$

**Line 4:** This step updates the centroids until the algorithm converges.

**What is the mathematical formulation of $k$-means ?** Given a set of data points (samples) $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and an integer $k$. The goal of $k$-means clustering is to group these data samples into $k$ clusters. In other words, our goal is to predict $k$ centroids and a label $c_i$ for each data point $\mathbf{x}_i$. The set of labels is a partition $C = \{C_1, C_2, \dots, C_k\}$. Mathematically, the goal of the $k$-means is to find a partition such that the following objective function is minimized:

$$\arg\min_C \sum_{j=1}^{k} \sum_{i \in C_j} \|\mathbf{x}_i - \mu_j\|_2^2 \Longleftrightarrow \arg\min_C \sum_{j=1}^{k} |C_j| \cdot Var(C_j). \tag{12.1}$$

In addition to the above formulation, we also have another version. Let $r_{ij} : \mathbb{R}^d \to \{0, 1\}$ be the indicator denoting whether point $\mathbf{x}_i$ belongs to cluster $k$. $r_{ij} = 1$ means that the data point $\mathbf{x}_i$ belongs to the cluster $j$; 0 otherwise. Another explanation is that $k$-means objective minimizes the total distortion (sum of distances of points from their cluster centers)

$$J(\mathbf{u}, \mathbf{r}) = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{i,j} \|\mathbf{x}_i - \mu_j\|_2^2.$$

Exact optimization of the K-means objective is NP-hard. In the next two questions, you will see that the $k$-means algorithm is a heuristic that converges to (i.e. stops at ) a local optimum.

**Why $k$-means can stop ?** This answer adopts from [3]. First, there are at most $k^n$ ways (actually it is exactly a Stirling numbers of the second kind. Therefore, the number of effective ways will be much less than $k^n$) to partition $n$ data points into $k$ clusters; each such partition can be called "clustering". This is a large but finite number. For each iteration of the algorithm, we produce a new clustering based only on the old clustering. Notice that

- 1) if the old clustering is the same as the new, it terminates.
- 2) if the new clustering is different from the old then the newer one has a lower cost.

Since the algorithm iterates a function whose domain is a finite set, the iteration must eventually enter a cycle. The cycle can not have length greater than 1 because otherwise by 2) you would have some clustering which has a lower cost than itself which is impossible. Hence the cycle must have length exactly 1. Hence $k$-means converges in a finite number of iterations. To see 2), notice that give the following optimization problem

$$\min_{\mu} \sum_{i \in C_j} \|\mathbf{x}_i - \mu\|_2^2. \tag{12.2}$$

We can easily see that the minimizer of the above problem is the sample mean $\mu_j = 1/|C_j| \sum_{i \in C_j} \mathbf{x}_i$. Just by taking the gradient of (12.2). Hence, if the centroid has been updated, we must have the following

$$\sum_{i \in C_j} \|\mathbf{x}_i - \mu_j\|_2^2 < \sum_{i \in C_j} \|\mathbf{x}_i - \mu_j\|_2^2.$$

Therefore, at each iteration, the objective must decrease.

**Does $k$-means find global optimum ?** The answer is no. $k$-means clustering algorithm does not guarantee to find a global optimum. Let us recall it again, in general, solving (12.1) is NP-hard problem. $k$-means only converges to a local optimal. We can construct an example to illustrate this. Let us assume we have a set of data points $\mathcal{D} = \{(-0.1, 2), (0.1, 2), (-2, 0.1), (-2, -0.1), (2, 0.1), (2, -0.1)\} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_6\}$ and set $k = 3$. Let the initial centroids be $\mu_1 = (-0.1, 1.9), \mu_2 = (0.1, 1.9), \mu_3 = (0, 0)$. The $k$-means algorithm goes as the follows:

- Iteration 1: assign the clusters $C_1 = \{\mathbf{x}_1\}, C_2 = \{\mathbf{x}_2\}, C_3 = \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ and then update the centroids $\mu_1 = (-0.1, 2.0), \mu_2 = (0.1, 2.0), \mu_3 = (0, 0)$.
- Iteration 2: assign the clusters $C_1 = \{\mathbf{x}_1\}, C_2 = \{\mathbf{x}_2\}, C_3 = \{\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ and then update the centroids $\mu_1 = (-0.1, 2.0), \mu_2 = (0.1, 2.0), \mu_3 = (0, 0)$. These centroids are the same as the previous ones. The algorithm stops.

In above case, after the algorithm terminates, the value of the objective function is

$$\|\mathbf{x}_1 - \mu_1\|_2^2 + \|\mathbf{x}_2 - \mu_2\|_2^2 + \sum_{i \in C_3} \|\mathbf{x}_i - \mu_3\|_2^2 = 0 + 0 + 4 \cdot (4 + 0.01) = 16.04.$$

However, we can clearly see that the optimal objective function is much less than 16.04. Notice that the optimal centroids should be $\mu_1 = (0, 2), \mu_2 = (-2, 0), \mu_3 = (2, 0)$ and the optimal cluster partition should be $C_1 = \{\mathbf{x}_1, \mathbf{x}_2\}, C_2 = \{\mathbf{x}_3, \mathbf{x}_4\}, C_3 = \{\mathbf{x}_5, \mathbf{x}_6\}$. The optimal function value should be

$$\sum_{j=1}^{3} \sum_{i \in C_j} \|\mathbf{x}_i - \mu_j\|_2^2 = 3 \cdot (2 \cdot (0.01 + 0.0)) = 0.06.$$

Therefore, above case is a local minimum. Due to $k$-means is extremely sensitive to cluster center initialization, it could have a bad initialization. A bad initialization can lead to poor convergence speed or bad overall clustering. To find a better local optimal, we should consider the following

- Be careful about where you start. Recall that we have different strategies of initialization. Place first center on top of randomly chosen data point. Place second center on data point that is as far away as possible from first center. Place j-th center on data point that is as far away as possible from the closest of centers 1 through $j - 1$.
- Do many runs of $k$-means, each from a different random start configuration.

**How could we choose $k$?** In general, it is a difficult problem. Most common approach is to try to find the solution that minimizes the Schwarz Criterion or to find the elbow point.

**What is the run time of $k$-means ?** It could have a bad parameter $k$. Running time of the $k$-means clustering algorithm (and most variants) is $\mathcal{O}(nkdT)$, where $n$ is the total number of data samples, $k$ is the number of clusters, and $T$ is the number of iterations needed.

**Others:** Some other important properties we need to know: 1) $k$-means is really just a special case of the EM (Expectation Maximization) algorithm applied to a particular naive Bayes model. See more details in mixture of Gaussian. 2) we shoud notice that the $k$-means algorithm is sensitive to outlier examples.

## References

[1] Aggarwal, Charu C. Data mining: the textbook. Springer, 2015.

[2] Tan, Pang-Ning, Michael Steinbach, Vipin Kumar, and ZhaoHui Tang. Introduction to Data Mining.

[3] https://stats.stackexchange.com/questions/188087/proof-of-convergence-of-k-means

[4] K-means and Hierarchical Clustering, Andrew W. Moore,

https://www.autonlab.org/tutorials/kmeans.html