Day 18 - 10 November - K-means and Clustering

Agenda:

- Course Project
- Clustering
- K-Means
- Proof of Convergence for K-Means

CS 6140: Machine Learning — Fall 2021— Paul Hand

Project Planning
Due: Wednesday December 15, 2021 at 11:59 PM Eastern time via Gradescope.

Names: [Put Your Name(s) Here]

For your final project, you will obtain a dataset, select multiple machine learning models, train the models, and evaluate the performance of the models. You may elect to reproduce some of the results from a scientific paper, but you must code up some aspect of dataset, model, or training yourself. You may use standard Deep Learning frameworks (e.g. PyTorch, TensorFlow, etc.). You may use code that is available on the internet as building blocks. You may run your algorithm in a slightly different context. You must train more than one machine learning model and compare the performance of those models. You are encouraged (but are not required to) train models that we have not discussed in class.

You will write up a short (at most 3 pages) report detailing: the dataset you are using and any data processing you have done, the models you are studying, the details of training the models, and the results of the evaluation. Please use the NeurIPS Style files for your report.

You may work in groups of up to 3 people. You may work alone.

If you want some ideas of projects, here are some ideas. You do not need to select one of these papers.

- Train several handwritten digit classifiers from the table at this website.
- Implement one of the chapters of the Mattmann book.
- Find a Kaggle dataset that you find interesting and train multiple models for it.
- Train a neural network to remove additive noise from images. You can construct a dataset consisting of clean images and noisy images that you construct.
- Create a synthetic dataset and evaluate the k-means and k-means++ algorithms
- Create a synthetic high dimensional dataset and show that k nearest neighbors fails while another classification method succeeds.
- Create a synthetic dataset and evaluate how successful cross-validation is at estimating test error.
- Reproduce aspects of Figure 1 of Understanding Deep Learning Requires Rethinking Generalization

Question 1. *Project Planning*

1. Provide a summary of the goal of your project. If you are replicating part of a paper, include a link to the paper here.

   Response:

2. What dataset will you use?

   Response:

3. What models will you train? You need to have more than one.

   Response:

4. What do you think will be most difficult about training the models?

   Response:

5. How will you evaluate the models?

   Response:

Clustering & Unsupervised Learning

Clustering is the task of partitioning a set of examples into different meaningful groups.

Example: Energy Use

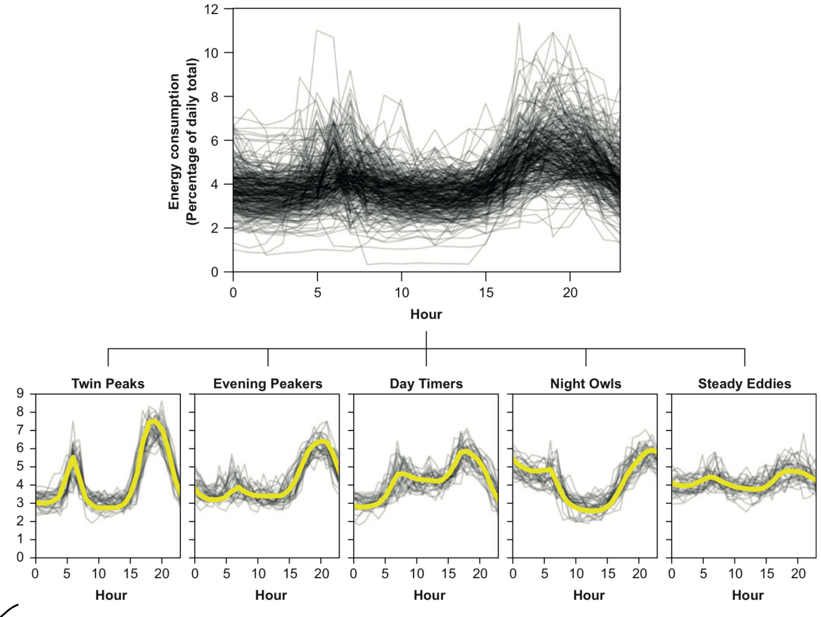
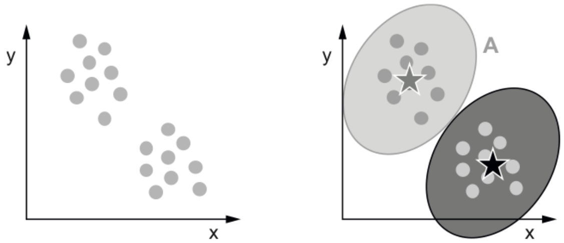


Figure 3.12 Turning a "hairball" of 1,000 users into five clusters

clusters

Abstractly:



Data points with two features (*x*, *y*) in space

Data points grouped into similiar clusters

Figure 3.7 The effects of clustering on a simple set of points with two features

Example: Customer Segmentation

Table 3.1 Five customer features before being fed to an unsupervised learning algorithm

Customer ID	Age	Gender	Average monthly spending ($)
1	18	M	14.67
2	21	M	15.67
3	28	M	18.02
4	27	F	34.61
5	32	F	30.66

- Customers within the same cluster are similar to each other.
- Customers in different clusters are different from each other.

&

Each cluster has a center/centroid, which can act as a stereotype for the cluster

Table 3.3 A summary of the characteristics of the three clusters spotted by our algorithm

Cluster number	Age	% female	Avg monthly spending ($)	# of customers
1	18.2	20%	15.24	290
2	29.3	90%	28.15	120
3	22	40%	17.89	590

Sign of Successful Clustering (in practice)

1 Are the results *interpretable*? In other words, are the cluster centers interpretable as buyer personas that make logical sense? If the answer is yes, move to question 2.

2 Are the results *actionable*? In other words, are my clusters different enough that I can come up with different strategies to target customers belonging to the different centers?

Supervised vs Unsupervised Learning

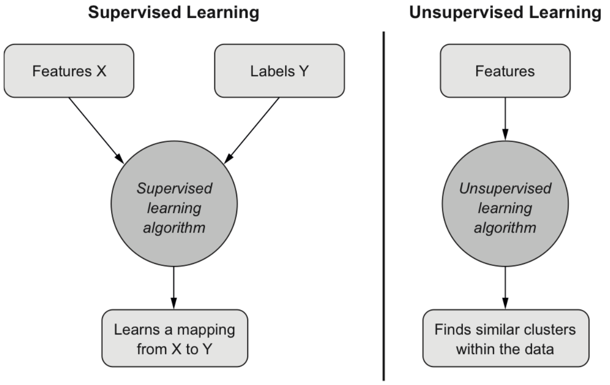

Supervised Learning
Features X
Labels Y
Supervised learning algorithm
Learns a mapping from X to Y
Unsupervised Learning
Features
Unsupervised learning algorithm
Finds similar clusters within the data

Figure 3.6 The differences in input and output of supervised and unsupervised algorithms

K-means Clustering

Given: $\{x_i\}_{i=1 \cdots n}$, k

Find: Cluster assignment for $i=1 \cdots n$
and a centroid of each cluster

Idea:
- Choose a center for each cluster
- Assign each example to nearest center
- Update each center as average of its examples

The k-means algorithm attempts to solve

$$\underset{S_1 \cdots S_k}{\operatorname{argmin}} \sum_{l=1}^{k} \sum_{x \in S_l} \| x - \mu(S_l) \|^2$$

where $S_1 \cdots S_k$ is a partition of $\{1 \cdots n\}$
and $\mu(S) = \frac{1}{|S|} \sum_{x \in S} x$ is the average of all examples in S.

Algorithm: Input: $\{\mu_1 \cdots \mu_k\}$ initialization of k centers

- Assign each example to nearest cluster center

$$l_i = \underset{l=1 \cdots k}{\operatorname{argmin}} \| x_i - \mu_l \|$$

- Update cluster means

$$\mu_l = \operatorname{mean}(\{x_i \mid l_i = l\})$$

- Repeat until convergence

Example w/ $k=2$: (from Bishop)

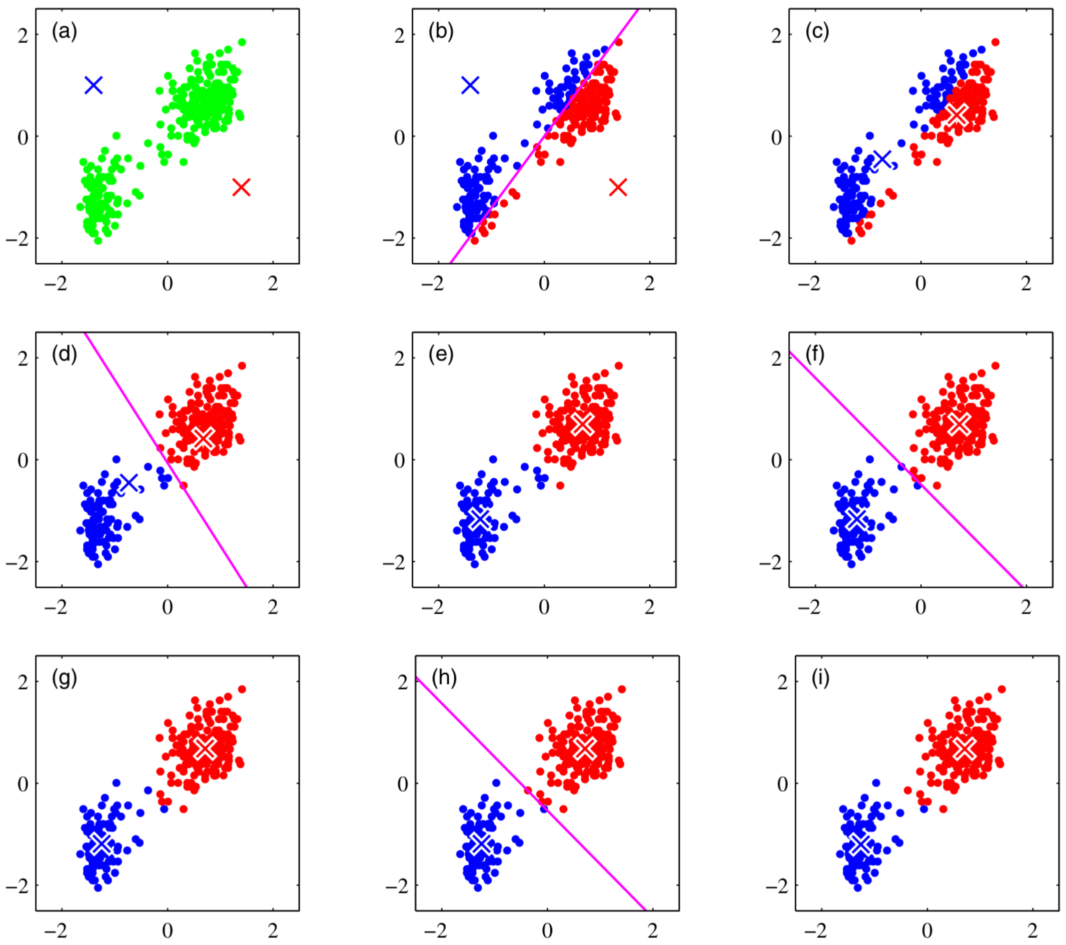


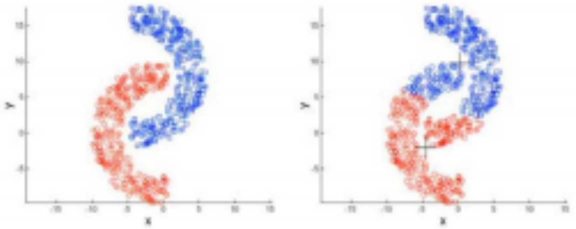
Figure 9.1 Illustration of the K-means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres μ_1 and μ_2 are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of the algorithm.

Can k-means fail?

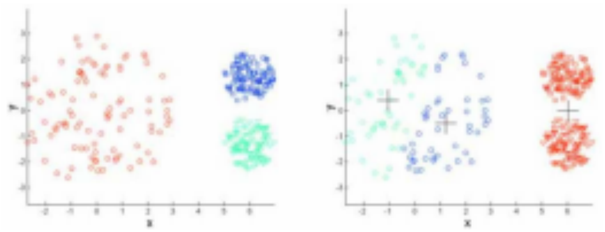
Yes, for multiple reasons

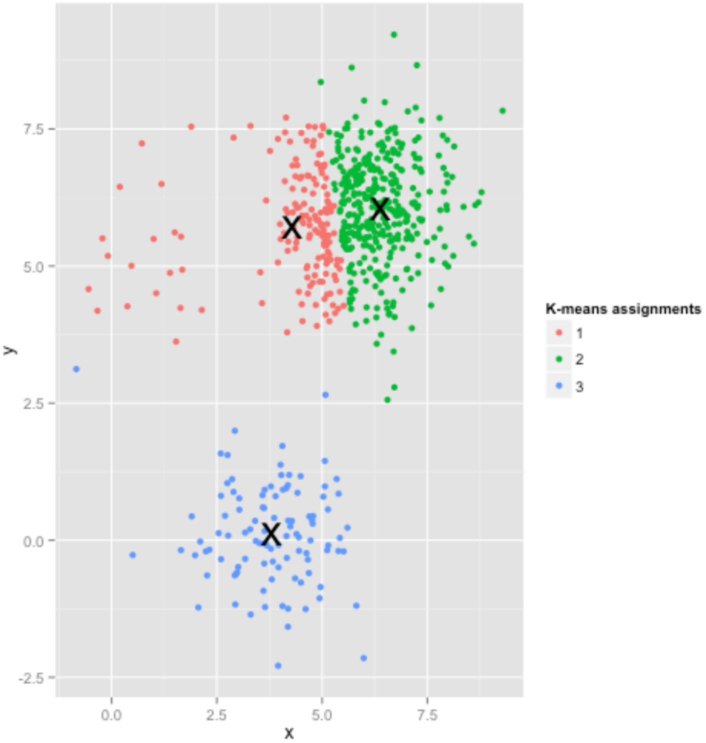
- Outliers
- Non convex shapes
- Differing variances of clusters
- Differing densities of clusters

Non-convex/non-round-shaped clusters: Standard *K*-means fails!



Clusters with different densities




7.5
5.0
2.5
0.0
-2.5
0.0
2.5
5.0
7.5
x
y
K-means assignments
1
2
3

Convergence of k-means

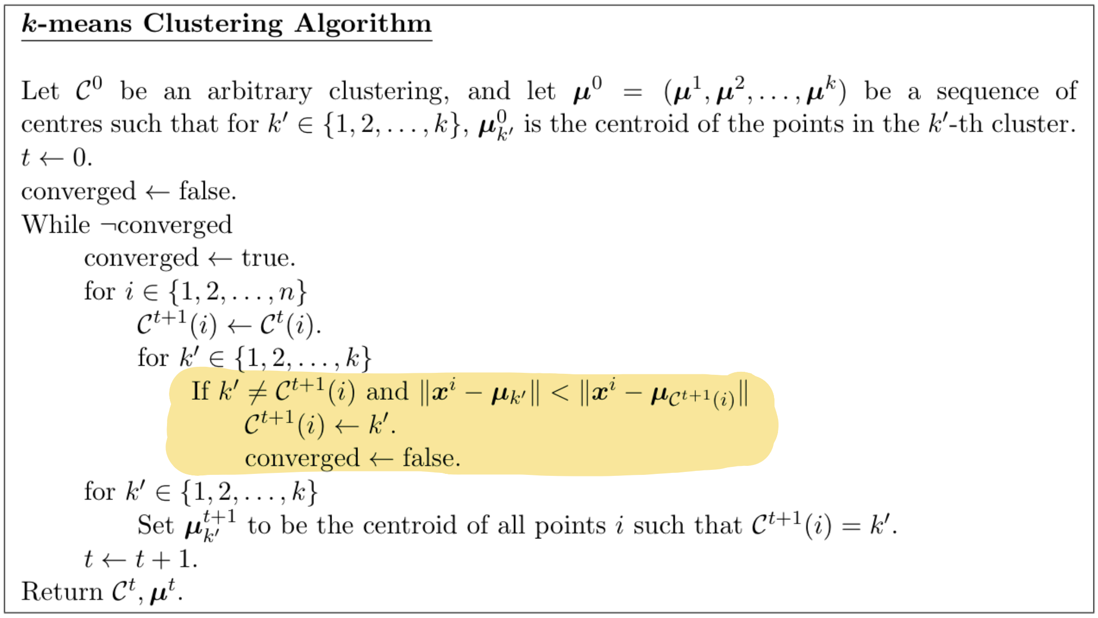
Theorem: k-means applied to any dataset with any initialization will converge

Notation: Data $\{X^1, X^2, \cdots, X^n\}$ $X^i \in \mathbb{R}^d$

A clustering $C: \{1,2,\cdots,n\} \to \{1,2,\cdots,k\}$ assigns each data point to a cluster id

$\mu = (\mu_1, \cdots, \mu_k)$ is a set of cluster centers

$$SSE(C, \mu) = \sum_{i=1}^{n} \| X^i - \mu_{C(i)} \|^2$$



k-means Clustering Algorithm

Let $\mathcal{C}^0$ be an arbitrary clustering, and let $\boldsymbol{\mu}^0 = (\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \ldots, \boldsymbol{\mu}^k)$ be a sequence of centres such that for $k' \in \{1, 2, \ldots, k\}$, $\boldsymbol{\mu}^0_{k'}$ is the centroid of the points in the k'-th cluster.

```
$t \leftarrow 0$.
converged $\leftarrow$ false.
While $\neg$converged
    converged $\leftarrow$ true.
    for $i \in \{1, 2, \ldots, n\}$
        $\mathcal{C}^{t+1}(i) \leftarrow \mathcal{C}^t(i)$.
        for $k' \in \{1, 2, \ldots, k\}$
            If $k' \neq \mathcal{C}^{t+1}(i)$ and $\|\boldsymbol{x}^i - \boldsymbol{\mu}_{k'}\| < \|\boldsymbol{x}^i - \boldsymbol{\mu}_{\mathcal{C}^{t+1}(i)}\|$
                $\mathcal{C}^{t+1}(i) \leftarrow k'$.
                converged $\leftarrow$ false.
    for $k' \in \{1, 2, \ldots, k\}$
        Set $\boldsymbol{\mu}^{t+1}_{k'}$ to be the centroid of all points $i$ such that $\mathcal{C}^{t+1}(i) = k'$.
    $t \leftarrow t + 1$.
Return $\mathcal{C}^t, \boldsymbol{\mu}^t$.
```

Lemma 1. *Consider the points $\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^m$, where $m \geq 1$, and for $i \in \{1, 2, \ldots, m\}$, $\boldsymbol{z}^i \in \mathbb{R}^d$. Let $\bar{\boldsymbol{z}} = \frac{1}{m}\sum_{i=1}^m \boldsymbol{z}^i$ be the mean of these points, and let $\boldsymbol{z} \in \mathbb{R}^d$ be an arbitrary point in the same (d-dimensional) space. Then*

$$\sum_{i=1}^{m} \|\boldsymbol{z}^i - \boldsymbol{z}\|^2 \geq \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2.$$

Proof: We show $\bar{z} = \underset{z}{\operatorname{argmin}} \sum_{i=1}^{m} \|z^i - z\|^2$

$$0 = \nabla \sum_{i=1}^{m} \|z^i - z\|^2 = \sum_{i=1}^{m} 2(z - z^i) = 2mz - 2\sum_{i=1}^{m} z^i$$

$$\Rightarrow z = \frac{1}{m} \sum_{i=1}^{m} z^i \quad \blacksquare$$

Theorem 2. *The k-means clustering algorithm converges.*

Proof. Suppose that the algorithm proceeds from iteration t to iteration $t+1$. It suffices to show that $\mathrm{SSE}(\mathcal{C}^{t+1},\, \boldsymbol{\mu}^{t+1}) < \mathrm{SSE}(\mathcal{C}^{t},\, \boldsymbol{\mu}^{t})$. To see why, consider that if that was true, no clustering can be visited twice; since the number of possible clusterings is finite (k^n), the algorithm must necessarily terminate. By the construction of the algorithm, we know that it terminates when no point has a cluster centre closer than the centre of its current cluster: in other words, the current clustering is *locally* optimal.

We show that $\mathrm{SSE}(\mathcal{C}^{t+1},\, \boldsymbol{\mu}^{t+1}) < \mathrm{SSE}(\mathcal{C}^{t},\, \boldsymbol{\mu}^{t})$ in two steps. First, we show that

$$\mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}) < \mathrm{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t}), \tag{1}$$

and next, we show that

$$\mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) \leq \mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}). \tag{2}$$

The first step follows directly from the logic of the algorithm: $\mathcal{C}^{t}$ and $\mathcal{C}^{t+1}$ are different only if there is a point that finds a closer cluster centre in $\boldsymbol{\mu}^{t}$ than the one assigned to it by $\mathcal{C}^{t}$:

$$\mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}) = \sum_{i=1}^{n} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t}_{C^{t+1}(i)}\|^2 < \sum_{i=1}^{n} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t}_{C^{t}(i)}\|^2 = \mathrm{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t}).$$

The second step puts Lemma 1 to use:

$$\begin{aligned}
\mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) &= \sum_{i=1}^{n} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t+1}_{C^{t+1}(i)}\|^2 \\
&= \sum_{k'=1}^{k} \sum_{i\in\{1,2,\ldots,n\},\mathcal{C}^{t+1}(i)=k'} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t+1}_{C^{t+1}(i)}\|^2 \\
&\leq \sum_{k'=1}^{k} \sum_{i\in\{1,2,\ldots,n\},\mathcal{C}^{t+1}(i)=k'} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t}_{C^{t+1}(i)}\|^2 \\
&= \sum_{i=1}^{n} \|\boldsymbol{x}^{i} - \boldsymbol{\mu}^{t}_{C^{t+1}(i)}\|^2 \\
&= \mathrm{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}).
\end{aligned}$$

□

Resource:

k-means Clustering

Shivaram Kalyanakrishnan

February 17, 2017

Abstract

We introduce the k-means clustering problem, describe the k-means clustering algorithm, and provide a proof of convergence for the algorithm.

1 The k-means Clustering Problem

We are given a data set $(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^n)$, where for $i \in \{1, 2, \ldots, n\}$, $\mathbf{x}^i \in \mathbb{R}^d$. Here $d \geq 2$ is the dimension of the data set. We are also specified an integer $k \geq 2$. The objective of k-means clustering is to partition the data set into k clusters, such that each cluster is as "tight" as possible. We define this objective more precisely.

A *clustering* $\mathcal{C} : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, k\}$ assigns one of k clusters to each point in the data set. Each cluster $k' \in \{1, 2, \ldots, k\}$ is also associated with a centre $\boldsymbol{\mu}_{k'} \in \mathbb{R}^d$. If we take a clustering $\mathcal{C}$ along with the sequence $\boldsymbol{\mu}$ representing the centres of its k clusters—$\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k)$—we can define "tightness" in terms of the aggregate distance between the data points and the centres of the clusters to which they are assigned by $\mathcal{C}$. If $\mathcal{C}(i)$ is the cluster in $\{1, 2, \ldots, k\}$ to which $\mathcal{C}$ assigns input point i, the Euclidean distance between the point and its cluster center is $\|\boldsymbol{x}^i - \boldsymbol{\mu}_{\mathcal{C}(i)}\|$. The most common measure of the tightness of a clustering $\mathcal{C}$ (along with cluster centres $\boldsymbol{\mu}$) is the sum squared error (SSE), defined as

$$\sum_{i=1}^{n} \|\boldsymbol{x}^i - \boldsymbol{\mu}_{\mathcal{C}(i)}\|^2.$$

Other definitions of tightness may also be used, but this particular one enjoys nice mathematical properties, as we shall shortly see.

The k-means clustering problem is the problem of finding a clustering among the set of all clusterings, along with a sequence of cluster centres, such that the corresponding SSE is minimal. Unfortunately, even for $k = 2$, this problem is NP-hard for general d and n [2]. If we revise our aim to find a "reasonable", rather than optimal, clustering, it turns out we can do quite nicely by applying the k-means clustering algorithm. This algorithm is an iterative one, which provably converges to a local minimum.

2 k-means Clustering Algorithm

Before we specify the k-means clustering algorithm, we settle one relevant matter. Recall that a clustering algorithm must return both a clustering and a centre for each cluster. The following lemma shows that for any fixed clustering, the SSE is minimised when the centre associated with each cluster is the mean (or centroid) of the set of points assigned to that cluster.

Lemma 1. *Consider the points* $\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^m$*, where* $m \geq 1$*, and for* $i \in \{1, 2, \ldots, m\}$*,* $\boldsymbol{z}^i \in \mathbb{R}^d$*. Let* $\bar{\boldsymbol{z}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{z}^i$ *be the mean of these points, and let* $\boldsymbol{z} \in \mathbb{R}^d$ *be an arbitrary point in the same (d-dimensional) space. Then*

$$\sum_{i=1}^{m} \|\boldsymbol{z}^i - \boldsymbol{z}\|^2 \geq \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2.$$

Proof.

$$\begin{aligned}
\sum_{i=1}^{m} \|\boldsymbol{z}^i - \boldsymbol{z}\|^2 &= \sum_{i=1}^{m} \|(\boldsymbol{z}^i - \bar{\boldsymbol{z}}) + (\bar{\boldsymbol{z}} - \boldsymbol{z})\|^2 \\
&= \sum_{i=1}^{m} \left(\|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2 + \|\bar{\boldsymbol{z}} - \boldsymbol{z}\|^2 + 2(\boldsymbol{z}^i - \bar{\boldsymbol{z}}) \cdot (\bar{\boldsymbol{z}} - \boldsymbol{z})\right) \\
&= \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2 + \sum_{i=1}^{m} \|\bar{\boldsymbol{z}} - \boldsymbol{z}\|^2 + 2\sum_{i=1}^{m}(\boldsymbol{z}^i \cdot \bar{\boldsymbol{z}} - \boldsymbol{z}^i \cdot \boldsymbol{z} - \bar{\boldsymbol{z}} \cdot \bar{\boldsymbol{z}} + \bar{\boldsymbol{z}} \cdot \boldsymbol{z}) \\
&= \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2 + m\|\bar{\boldsymbol{z}} - \boldsymbol{z}\|^2 + 2(m\bar{\boldsymbol{z}} \cdot \bar{\boldsymbol{z}} - m\bar{\boldsymbol{z}} \cdot \boldsymbol{z} - m\bar{\boldsymbol{z}} \cdot \bar{\boldsymbol{z}} + m\bar{\boldsymbol{z}} \cdot \boldsymbol{z}) \\
&= \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2 + m\|\bar{\boldsymbol{z}} - \boldsymbol{z}\|^2 \\
&\geq \sum_{i=1}^{m} \|\boldsymbol{z}^i - \bar{\boldsymbol{z}}\|^2. \qquad \square
\end{aligned}$$

The k-means clustering algorithm, shown below, is rather straightforward. We begin with an arbitrary clustering, and in line with Lemma 1, set the cluster centres to be the means of the points in each cluster. Thereafter, we examine each point. If it so happens that the closest cluster centre to a point is not the centre of its current cluster, the point is shifted to the cluster to whose centre it is closest. The change in cluster assignments now calls for a corresponding recalculation of the cluster centres; this process iterates until convergence.

***k*-means Clustering Algorithm**

```
Let $\mathcal{C}^0$ be an arbitrary clustering, and let $\boldsymbol{\mu}^0 = (\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \ldots, \boldsymbol{\mu}^k)$ be a sequence of centres such that for $k' \in \{1, 2, \ldots, k\}$, $\boldsymbol{\mu}^0_{k'}$ is the centroid of the points in the $k'$-th cluster.
$t \leftarrow 0$.
converged $\leftarrow$ false.
While $\neg$converged
    converged $\leftarrow$ true.
    for $i \in \{1, 2, \ldots, n\}$
        $\mathcal{C}^{t+1}(i) \leftarrow \mathcal{C}^t(i)$.
        for $k' \in \{1, 2, \ldots, k\}$
            If $k' \neq \mathcal{C}^{t+1}(i)$ and $\|\boldsymbol{x}^i - \boldsymbol{\mu}_{k'}\| < \|\boldsymbol{x}^i - \boldsymbol{\mu}_{\mathcal{C}^{t+1}(i)}\|$
                $\mathcal{C}^{t+1}(i) \leftarrow k'$.
                converged $\leftarrow$ false.
    for $k' \in \{1, 2, \ldots, k\}$
        Set $\boldsymbol{\mu}^{t+1}_{k'}$ to be the centroid of all points $i$ such that $\mathcal{C}^{t+1}(i) = k'$.
    $t \leftarrow t + 1$.
Return $\mathcal{C}^t, \boldsymbol{\mu}^t$.
```

As per the procedure outlined above, it is entirely possible to achieve clusterings that assign *no* points to some of the k clusters. In such a case, the corresponding cluster centre can be set arbitrarily (since the mean is undefined). In practice, though, it is common to use all k clusters effectively—for instance, one could set the centre of an empty cluster to be one of the points in the data set, which would ensure that the cluster will not be empty in the next iteration.

It should also be noted that the choice of the initial clustering, $\mathcal{C}^0$, can make a significant difference to the SSE of the final clustering obtained. Specialised initialisation strategies (such as **k-means++** [1]) are often used to good effect. It exceeds the scope of this discussion to describe initialisation procedures in detail. Rather, we proceed to prove that regardless of the initialisation, the algorithm will necessarily converge.

Theorem 2. *The k-means clustering algorithm converges.*

Proof. Suppose that the algorithm proceeds from iteration t to iteration $t+1$. It suffices to show that $\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) < \text{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t})$. To see why, consider that if that was true, no clustering can be visited twice; since the number of possible clusterings is finite (k^n), the algorithm must necessarily terminate. By the construction of the algorithm, we know that it terminates when no point has a cluster centre closer than the centre of its current cluster: in other words, the current clustering is *locally* optimal.

We show that $\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) < \text{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t})$ in two steps. First, we show that

$$\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}) < \text{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t}), \tag{1}$$

and next, we show that

$$\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) \leq \text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}). \tag{2}$$

The first step follows directly from the logic of the algorithm: $\mathcal{C}^t$ and $\mathcal{C}^{t+1}$ are different only if there is a point that finds a closer cluster centre in $\boldsymbol{\mu}^t$ than the one assigned to it by $\mathcal{C}^t$:

$$\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}) = \sum_{i=1}^{n} \|\boldsymbol{x}^i - \boldsymbol{\mu}^t_{\mathcal{C}^{t+1}(i)}\|^2 < \sum_{i=1}^{n} \|\boldsymbol{x}^i - \boldsymbol{\mu}^t_{\mathcal{C}^{t}(i)}\|^2 = \text{SSE}(\mathcal{C}^{t}, \boldsymbol{\mu}^{t}).$$

The second step puts Lemma 1 to use:

$$\begin{aligned}
\text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t+1}) &= \sum_{i=1}^{n} \|\boldsymbol{x}^i - \boldsymbol{\mu}^{t+1}_{\mathcal{C}^{t+1}(i)}\|^2 \\
&= \sum_{k'=1}^{k} \sum_{i \in \{1,2,\ldots,n\}, \mathcal{C}^{t+1}(i)=k'} \|\boldsymbol{x}^i - \boldsymbol{\mu}^{t+1}_{\mathcal{C}^{t+1}(i)}\|^2 \\
&\leq \sum_{k'=1}^{k} \sum_{i \in \{1,2,\ldots,n\}, \mathcal{C}^{t+1}(i)=k'} \|\boldsymbol{x}^i - \boldsymbol{\mu}^{t}_{\mathcal{C}^{t+1}(i)}\|^2 \\
&= \sum_{i=1}^{n} \|\boldsymbol{x}^i - \boldsymbol{\mu}^{t}_{\mathcal{C}^{t+1}(i)}\|^2 \\
&= \text{SSE}(\mathcal{C}^{t+1}, \boldsymbol{\mu}^{t}).
\end{aligned}$$

□

References

[1] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, pages 1027–1035. SIAM, 2007.

[2] Sanjoy Dasgupta. The hardness of k-means clustering. Technical Report CS2008-0916, Department of Computer Science and Engineering, University of California, San Diego, 2008. Available at `http://cseweb.ucsd.edu/~dasgupta/papers/kmeans.pdf`.