Lecture Notes on Machine Learning
Kernel k-Means Clustering (Part 1)

Christian Bauckhage
B-IT, University of Bonn

In this note, we show that the objective function for k-means clustering can be cast in a form which allows for invoking the kernel trick.

Setting the Stage

Previously[1], we said the *kernel trick* is *1) to rewrite an algorithm for data analysis in such a way that input data only appear in form of inner products with other input data, and 2) to replace any occurrence of such inner products by kernel evaluations*.

In the following, we demonstrate how to apply this this trick to the problem of k-means clustering.

Recall[2] that k-means clustering aims at partitioning a given set of n data points $\boldsymbol{x}_j \in \mathbb{R}^m$ into k distinct clusters C_i which are defined in terms of prototypes $\boldsymbol{\mu}_i$.

The basic problem, therefore, is finding optimal cluster prototypes and most k-means algorithms try to accomplish this via minimizing

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 \tag{1}$$

with respect to the $\boldsymbol{\mu}_i$ where the z_{ij} are binary *indicator variables*[3] defined as

$$z_{ij} = \begin{cases} 1, \text{ if } \boldsymbol{x}_j \in C_i \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

Regarding our goal of kernelizing k-means clustering, all of this is to say that "all we have to do" is to kernelize the minimization objective in (1). Next, we walk through the steps this involves.

[1] C. Bauckhage. Lecture Notes on Machine Learning: The Kernel Trick. B-IT, University of Bonn, 2019

[2] C. Bauckhage and O. Cremers. Lecture Notes on Machine Learning: k-Means Clustering. B-IT, University of Bonn, 2019

k-means objective

[3] C. Bauckhage and D. Speicher. Lecture Notes on Machine Learning: Rewriting the k-Means Objective. B-IT, University of Bonn, 2019

Kernelizing the k-Means Objective Function

The data that enter k-means clustering are the n points $\boldsymbol{x}_j \in \mathbb{R}^m$ we want to cluster. We therefore have to rewrite the objective function in (1) such that the $\boldsymbol{x}_j$ only occur in form of inner products.

To begin with, we recall the following elementary identity for the Euclidean norm

$$\left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 = (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^\mathsf{T} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) = \left(\boldsymbol{x}_j^\mathsf{T} \boldsymbol{x}_j - 2\, \boldsymbol{x}_j^\mathsf{T} \boldsymbol{\mu}_i + \boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{\mu}_i \right). \tag{3}$$

This immediately allows us to cast the k-means objective in (1) as[4]

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left(\boldsymbol{x}_j^\mathsf{T} \boldsymbol{x}_j - 2\, \boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{x}_j + \boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{\mu}_i \right) \tag{4}$$

where all the $\boldsymbol{x}_j$ now appear as factors of inner products.

[4] Observe that we use the symmetry of the inner product to write $\boldsymbol{x}_j^\mathsf{T} \boldsymbol{\mu}_i$ as $\boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{x}_j$.

HOWEVER, we are not quite there yet. This is because not all inner products involving data points are inner products of data points only. Some of them involve the cluster means

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j \tag{5}$$

where $n_i = |C_i|$ denotes the size of cluster C_i.

We therefore recall that, using the binary indicator variables z_{ij}, we may also write

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n} z_{ij}\, \boldsymbol{x}_j \tag{6}$$

mean $\boldsymbol{\mu}_i$ and size n_i of cluster C_i can be expressed in terms of the indicator variables $z_{ij} \in \{0, 1\}$ defined in (2)

as well as

$$n_i = \sum_{j=1}^{n} z_{ij} \tag{7}$$

Hence, those inner products in (4) that involve cluster means $\boldsymbol{\mu}_i$ can also be written as

$$\boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{x}_j = \frac{1}{n_i} \sum_{p=1}^{n} z_{ip}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_j \tag{8}$$

as well as

$$\boldsymbol{\mu}_i^\mathsf{T} \boldsymbol{\mu}_i = \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip}\, z_{iq}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_q. \tag{9}$$

where we had to introduce additional summation indices p and q to correctly expand the inner products into (double) sums.

PUTTING TOGETHER (4), (8), and (9), we therefore find that (1) can equivalently be expressed as

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left(\boldsymbol{x}_j^\mathsf{T} \boldsymbol{x}_j - \frac{2}{n_i} \sum_{p=1}^{n} z_{ip}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_j + \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip}\, z_{iq}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_q \right). \tag{10}$$

the k-means objective function in (1) can be expressed exclusively in terms of inner products between input data

AT THIS POINT, we could conclude our discussion. Looking at (10), we recognize that the minimization objective for k-means clustering can be written entirely in terms of inner products between data. This immediately allows us to invoke the second step of the kernel trick, where we replace these inner products by kernel functions. However, before we do so, we will further simplify the result in (10).

UPON CLOSER INSPECTION, we realize that (10) is a sum over three terms, namely

$$T_1 = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij}\, \boldsymbol{x}_j^\mathsf{T} \boldsymbol{x}_j \tag{11}$$

$$T_2 = -2 \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \frac{1}{n_i} \sum_{p=1}^{n} z_{ip}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_j \tag{12}$$

$$T_3 = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip}\, z_{iq}\, \boldsymbol{x}_p^\mathsf{T} \boldsymbol{x}_q. \tag{13}$$

With respect to the first term T_1, we note that we can rearrange it as follows

$$T_1 = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \, \boldsymbol{x}_j^\intercal \boldsymbol{x}_j \tag{14}$$

$$= \sum_{j=1}^{n} \boldsymbol{x}_j^\intercal \boldsymbol{x}_j \sum_{i=1}^{k} z_{ij} \tag{15}$$

$$= \sum_{j=1}^{n} \boldsymbol{x}_j^\intercal \boldsymbol{x}_j \tag{16}$$

where, in the step from (15) to (16), we made use of a crucial property of (hard) k-means clustering. Since each data point $\boldsymbol{x}_j$ is assigned to exactly one cluster C_i, the indicator variables $z_{ij} \in \{0, 1\}$ obey

$$\sum_{i=1}^{k} z_{ij} = 1. \tag{17}$$

For the second term T_2, it will come in handy to slightly rearrange it so that it reads

$$T_2 = -2 \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \frac{1}{n_i} \sum_{p=1}^{n} z_{ip} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_j \tag{18}$$

$$= -2 \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n} \sum_{p=1}^{n} z_{ij} \, z_{ip} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_j. \tag{19}$$

For the third term T_3, we observe that it can also be written as

$$T_3 = \sum_{i=1}^{k} \left(\sum_{j=1}^{n} z_{ij} \right) \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip} \, z_{iq} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_q \tag{20}$$

$$= \sum_{i=1}^{k} n_i \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip} \, z_{iq} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_q \tag{21}$$

$$= \sum_{i=1}^{k} \frac{1}{n_i} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip} \, z_{iq} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_q \tag{22}$$

where the step from (20) to (21) made use of (7). And, comparing the result in (22) to our previous one in (19), we find $T_3 = -\frac{1}{2} T_2$.

Summing back together (16), (19), and (22), we therefore obtain the objective in (10) as

the objective in (10) can be written in a much shorter form

$$E = \sum_{j=1}^{n} \boldsymbol{x}_j^\intercal \boldsymbol{x}_j - \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n} \sum_{p=1}^{n} z_{ij} \, z_{ip} \, \boldsymbol{x}_p^\intercal \boldsymbol{x}_j. \tag{23}$$

Given this compact form of the rewritten k-means objective, it is finally worth our while to proceed to step two of the kernel trick, namely to replace inner products by kernel functions. This way, we obtain the ***minimization objective for kernel k-means clustering***

kernel k-means objective

$$E_K = \sum_{j=1}^{n} K(\boldsymbol{x}_j, \boldsymbol{x}_j) - \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n} \sum_{p=1}^{n} z_{ij} \, z_{ip} \, K(\boldsymbol{x}_p, \boldsymbol{x}_j). \tag{24}$$

Summary and Outlook

In this note, we saw that k-means clustering allows for invoking the kernel trick. In particular, we demonstrated that the minimization objective

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 \tag{25}$$

considered in conventional k-means clustering can be kernelized to become

$$E_K = \sum_{j=1}^{n} K(\boldsymbol{x}_j, \boldsymbol{x}_j) - \sum_{i=1}^{k} \frac{1}{n_i} \sum_{j=1}^{n} \sum_{p=1}^{n} z_{ij} \, z_{ip} \, K(\boldsymbol{x}_p, \boldsymbol{x}_j). \tag{26}$$

However, while conventional k-means clustering is (typically taken to be) tantamount to the problem of minimizing E with respect to the cluster means $\boldsymbol{\mu}_i$, we must point out that there are no means in E_K anymore. This is a consequence of invoking the kernel trick and begs the following question

> Given the kernelized objective function in (26), which minimization problem do we have to solve in kernel k-means clustering?

This crucial question as well as the equally crucial question of how to practically solve the kernel k-means problem will be answered in later notes.

Acknowledgments

This material was prepared within project P3ML which is funded by the Ministry of Education and Research of Germany (BMBF) under grant number 01/S17064. The authors gratefully acknowledge this support.

References

C. Bauckhage. Lecture Notes on Machine Learning: The Kernel Trick. B-IT, University of Bonn, 2019.

C. Bauckhage and O. Cremers. Lecture Notes on Machine Learning: k-Means Clustering. B-IT, University of Bonn, 2019.

C. Bauckhage and D. Speicher. Lecture Notes on Machine Learning: Rewriting the k-Means Objective. B-IT, University of Bonn, 2019.