Lecture Notes on Data Science: Kernel k-Means Clustering (Part 1)

Christian Bauckhage
B-IT, University of Bonn

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

Introduction

Previously[1], we saw that the problem of k-means clustering of a data set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^m$ into k clusters $C_1, \ldots, C_k$ is, at its heart, equivalent to the problem of finding k appropriate cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$.

In a later note[2], we then saw that the problem of seraching for appropriate centroids can be cast as the problem of minimizing the the following objective function

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

over all possible choices of the $\boldsymbol{\mu}_i$ where the z_{ij} are so called latent- or *indicator variables*. They indicate for any data point $\boldsymbol{x}_j$ whether or not it belongs to cluster C_i. In other words,

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

indicator variables

Regarding our topic in this note, it is interesting to observe that these indicator variables provide us with alternative expressions for each cluster centroid $\boldsymbol{\mu}_i$. Up until now, we always considered

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j \tag{3}$$

where $n_i = |C_i|$ denotes the size of cluster C_i. However, using the z_{ij}, we may just as well write

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n} z_{ij} \, \boldsymbol{x}_j. \tag{4}$$

Exercise: convince yourself that (3) and (4) are indeed equivalent.

This result will play a key role in the following. In particular, we will draw on it to show that the k-means objective function can be kernelized.

Kernelizing the k-Means Objective Function

So far, our study of the k-means algorithm and its properties was (more or less implicitly) confined to setting involving Euclidean data vectors. Of course, these are very common in our daily practice, but, sometimes, we need to cluster data which are not additive so that

[1] C. Bauckhage. Lecture Notes on Data Science: k-Means Clustering, 2015b. DOI: 10.13140/RG.2.1.2829.4886

[2] C. Bauckhage. Lecture Notes on Data Science: k-Means Clustering Is Gaussian Mixture Modeling, 2015a. DOI: 10.13140/RG.2.1.3033.2646

the notion of a mean is ill defined[3]. While it may seem, that k-means clustering does not apply to situations like these, it is indeed possible to generalize the approach to basically any kind of data using kernel k-means clustering. In this section, we will have a first brief look at what this means.

[3] Consider, e.g., categorical data, textual data, or relational data.

To begin with, we recall the following elementary identity

$$\|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2 = (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T (\boldsymbol{x}_j - \boldsymbol{\mu}_i) = \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{\mu}_i^T \boldsymbol{x}_j + \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i\right). \tag{5}$$

which allows us to cast the k-means objective function in (1) as

$$E(k) = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{\mu}_i^T \boldsymbol{x}_j + \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i\right). \tag{6}$$

Given what we worked out in (4), we next note that

$$\boldsymbol{\mu}_i^T \boldsymbol{x}_j = \frac{1}{n_i} \sum_{p=1}^{n} z_{ip}\, \boldsymbol{x}_p^T \boldsymbol{x}_j \tag{7}$$

as well as

$$\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i = \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip}\, z_{iq}\, \boldsymbol{x}_p^T \boldsymbol{x}_q \tag{8}$$

so that the k-means objective function can also be expressed as

$$E(k) = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\frac{1}{n_i} \sum_{p=1}^{n} z_{ip}\, \boldsymbol{x}_p^T \boldsymbol{x}_j + \frac{1}{n_i^2} \sum_{p=1}^{n} \sum_{q=1}^{n} z_{ip}\, z_{iq}\, \boldsymbol{x}_p^T \boldsymbol{x}_q\right). \tag{9}$$

expanded k-means objective function

At this point, we can basically conclude our discussion. Looking at (9), we recognize that the k-means objective function in (1) can be written entirely in terms of inner products between data vectors. This allows for invoking the *kernel trick* where we replace inner products $\boldsymbol{x}_p^T \boldsymbol{x}_q$ by non-linear kernel functions $k(\boldsymbol{x}_p, \boldsymbol{x}_q)$.

kernel trick

This trick has become a staple in areas such as data mining or pattern recognition because it allows for applying linear techniques in order to tackle nonlinear problems [4,5,6].

[4] C. Bauckhage. Lecture Notes on the Kernel Trick (I), 2015c. DOI: 10.13140/2.1.4524.8806

[5] J. Shaw-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004

[6] B. Schölkopf and A. Smola. *Learning with Kernels – Support Vector Machines, Optimizatyion and Beyond*. MIT Press, 2002

In the context of k-means clustering, the kernel trick may help us to obtain reasonable clusters even for highly non-Gaussian data. Furthermore, kernel functions may be defined for a wide range of data types so that k-means clustering is no longer confined to Euclidean vectors. Looking at (9), we realize that kernel k-means clustering is basically tantamount to determining suitable values of the indicator variable z_{ij}.

However, the latter is usually a rather daunting problem and applying the kernel trick typically increases computation times. It also requires experience as to appropriate kernel functions and necessitates especially careful initializations of the algorithm[7]. For the time being, we therefore postpone solution strategies for all of these problems to later notes.

[7] Note that these aspects of kernel k-means clustering are often passed over in the literature.

References

C. Bauckhage. Lecture Notes on Data Science: k-Means Clustering Is Gaussian Mixture Modeling, 2015a. DOI: 10.13140/RG.2.1.3033.2646.

C. Bauckhage. Lecture Notes on Data Science: k-Means Clustering, 2015b. DOI: 10.13140/RG.2.1.2829.4886.

C. Bauckhage. Lecture Notes on the Kernel Trick (I), 2015c. DOI: 10.13140/2.1.4524.8806.

B. Schölkopf and A. Smola. *Learning with Kernels – Support Vector Machines, Optimizatyion and Beyond*. MIT Press, 2002.

J. Shaw-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.