Lecture Notes on Data Science: Soft k-Means Clustering

Technical Report · October 2015

DOI: 10.13140/RG.2.1.3582.6643

Christian Bauckhage
University of Bonn

Lecture Notes on Data Science: Soft k-Means Clustering

Christian Bauckhage
B-IT, University of Bonn

In this note, we study the idea of soft k-means clustering which yields soft assignments of data points to clusters. We discuss the basic theory, a simple algorithm, and examples for how this algorithm behaves.

Introduction

So far, our discussion of k-means clustering and its properties has been restricted to the paradigm of ***hard k-means clustering***.

In other words, we considered the problem of partitioning a set

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m \tag{1}$$

of data points $\mathbf{x}_j$ into k clusters $C_1, \ldots, C_k$ for which we required that they are proper subsets $C_i \subset X$ such that $C_1 \cup C_2 \cup \ldots \cup C_k = X$ and

$$C_i \cap C_l = \emptyset \tag{2}$$

for $i \neq l$. Given these requirements, we then derived the classical k-means procedure[1] as a method for computing *hard cluster assignments*[2] where

$$\forall\, i \neq l : \mathbf{x}_j \in C_i \Rightarrow \mathbf{x}_j \notin C_l. \tag{3}$$

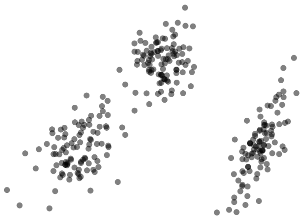


Figure 1: Didactic example of $n = 300$ data points $\mathbf{x}_j \in \mathbb{R}^2$ sampled from three bivariate Gaussian distributions.

Quite often, however, hard cluster assignments may be overzealous. Consider, for example, Figure 1. While most everybody would agree that this data contains three distinct clusters, the two on the left appear to overlap in that there are a few data points for which it is not quite so obvious to which cluster they belong.

In this note, we will therefore relax the assumption in (2) and study the idea of ***soft k-means clustering***.

There are two fundamental innovations with regard to this idea. First of all, soft k-means clustering allows every data point to belong to several clusters simultaneously. Second of all, this is not meant in the simple sense that clusters may overlap. Rather, soft k-means clustering determines to which *degree* each data point belongs to each cluster. This is to say, that the resulting clusters are fuzzy rather than crisp sets so that soft k-means clustering is sometimes also referred to as ***fuzzy k-means clustering***[3].

Soft k-Means Clustering

Recall that, when we studied the relationship between (hard) k-means clustering and Gaussian mixture modeling[4], we saw that clustering is essentially a minimization problem. In particular, we saw that

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

[2] While the logical expression in (3) may look daunting, it is just a fancy way of saying that classical k means clustering assigns each data point to exactly one cluster.

[3] In fact, there seems to be some controversy as to whether or not soft k-means and fuzzy k-means clustering really are the same. We do not partake in this silly debate but consider them equivalent and explain why in our discussion below.

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

k-means clustering can be formalized as the problem of finding k cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$ which minimize the error function

$$E(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k) = \sum_{i=1}^{k} E(\boldsymbol{\mu}_i) = \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 \tag{4}$$

where the z_{ij} are ***indicator variables***

$$z_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{x}_j \in C_i \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

which indicate whether or not $\boldsymbol{x}_j$ belongs to cluster C_i.

Looking at (5), the indicator variables seem not to depend on the cluster centroids $\boldsymbol{\mu}_i$. Yet, if we recall that the classical k-means procedure assigns data point $\boldsymbol{x}_j$ to cluster C_i if and only if $\boldsymbol{\mu}_i$ is the closest centroid, we realize that (5) is equivalent to

$$z_{ij} = \begin{cases} 1, & \text{if } \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2 \leq \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_l \right\|^2 \; \forall \, i \neq l \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

This, however, is to say that the values of the z_{ij} do indeed depend on distances between data points and centroids.

THE FACT THAT the z_{ij} in (6) are binary, i.e. $z_{ij} \in \{0, 1\}$, once again reflects the crisp nature of the clusters produced by hard k-means clustering. But what if we would allow the z_{ij} to vary continuously? That is, what if there were degrees of cluster membership $z_{ij} \in [0, 1]$? And how could we reasonably define these?

Soft Cluster Assignments

Given our discussion so far, it appears reasonable to decide cluster membership based on distances to cluster centroids. Hence, given a set of centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$, the problem of computing degrees of cluster membership can be understood as the problem of turning distances $\left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2$ into numbers $z_{ij} \in [0, 1]$.

A POPULAR WAY OF ACHIEVING SUCH A TRANSFORMATION is to consider softmax functions[5].

A corresponding, rather common choice in the context of soft k-means clustering is

$$z_{ij} = \frac{e^{-\beta \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_i \right\|^2}}{\sum\limits_{l=1}^{k} e^{-\beta \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_l \right\|^2}} \tag{7}$$

where the parameter $\beta > 0$ is called the ***stiffness parameter***.

⚠ **softmax function**

⚠ **stiffness parameter**

GIVEN THIS PARTICULAR DEFINITION of the cluster membership degrees z_{ij}, we observe that the numerator in (7) guarantees that $0 < z_{ij} \leq 1$ and the denominator in (7) guarantees that $\sum_i z_{ij} = 1$. But this is to say that the z_{ij} can be interpreted as probabilities $p(\boldsymbol{\mu}_i \mid \boldsymbol{x}_j)$ which may be desirable in practice.

⚠ **probabilistic interpretation**

[5] Softmax functions are a staple of machine learning and frequently found in neural computing algorithms.

Centroid Updates

Having devised a way of computing soft rather than hard cluster membership indicators, we next need to address the question of how to adapt the updating of cluster centroids $\boldsymbol{\mu}_i$ to the situation of soft k-means clustering?

In order to answer this question, let us single out a specific cluster, say, C_l with corresponding centroid $\boldsymbol{\mu}_l$. Looking at (4), we note that this cluster contributes a term of

$$E(\boldsymbol{\mu}_l) = \sum_{j=1}^{n} z_{lj} \left\| \boldsymbol{x}_j - \boldsymbol{\mu}_l \right\|^2 \tag{8}$$

to the overall error we want to minimize. Hence, if we could find a value for $\boldsymbol{\mu}_l$ such that this contribution was as small as possible, the overall error would decrease. In other words, we are dealing with the task of having to solve the following optimization problem

$$\boldsymbol{\mu}_l = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \sum_{j=1}^{n} z_{lj} \left\| \boldsymbol{x}_j - \boldsymbol{\mu} \right\|^2. \tag{9}$$

Luckily, this is a simple problem because the sum on the right hand side of (9) is a convex function in $\boldsymbol{\mu}$. It therefore has a unique minimizer which corresponds to the root of the corresponding gradient.

So let us proceed and find that root. To begin with, we note that

$$\begin{aligned} E(\boldsymbol{\mu}) &= \sum_{j=1}^{n} z_{lj} \left\| \boldsymbol{x}_j - \boldsymbol{\mu} \right\|^2 && (10) \\ &= \sum_{j=1}^{n} z_{lj} \left(\boldsymbol{x}_j - \boldsymbol{\mu} \right)^T \left(\boldsymbol{x}_j - \boldsymbol{\mu} \right) && (11) \\ &= \sum_{j=1}^{n} z_{lj} \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{x}_j^T \boldsymbol{\mu} + \boldsymbol{\mu}^T \boldsymbol{\mu} \right). && (12) \end{aligned}$$

Hence, if we derive[6] $E(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$, we find

$$\frac{\partial}{\partial \boldsymbol{\mu}} E(\boldsymbol{\mu}) = \sum_{j=1}^{n} z_{lj} \frac{\partial}{\partial \boldsymbol{\mu}} \left(\boldsymbol{x}_j^T \boldsymbol{x}_j - 2\boldsymbol{x}_j^T \boldsymbol{\mu} + \boldsymbol{\mu}^T \boldsymbol{\mu} \right) = \sum_{j=1}^{n} z_{lj} \left(2\boldsymbol{\mu} - 2\boldsymbol{x}_j \right) \tag{13}$$

so that equating (13) to zero provides

$$\begin{aligned} && \sum_{j=1}^{n} z_{lj}\, 2\, \boldsymbol{\mu} &= \sum_{j=1}^{n} z_{lj}\, 2\, \boldsymbol{x}_j && (14) \\ \Leftrightarrow && \boldsymbol{\mu} &= \frac{\sum_{j=1}^{n} z_{lj}\, \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{lj}} && (15) \end{aligned}$$

as the solution to (9). In other words, setting $\boldsymbol{\mu}_l$ to the weighted mean

⚠ **weighted mean**

$$\boldsymbol{\mu}_l = \frac{\sum_{j=1}^{n} z_{lj}\, \boldsymbol{x}_j}{\sum_{j=1}^{n} z_{lj}} \tag{16}$$

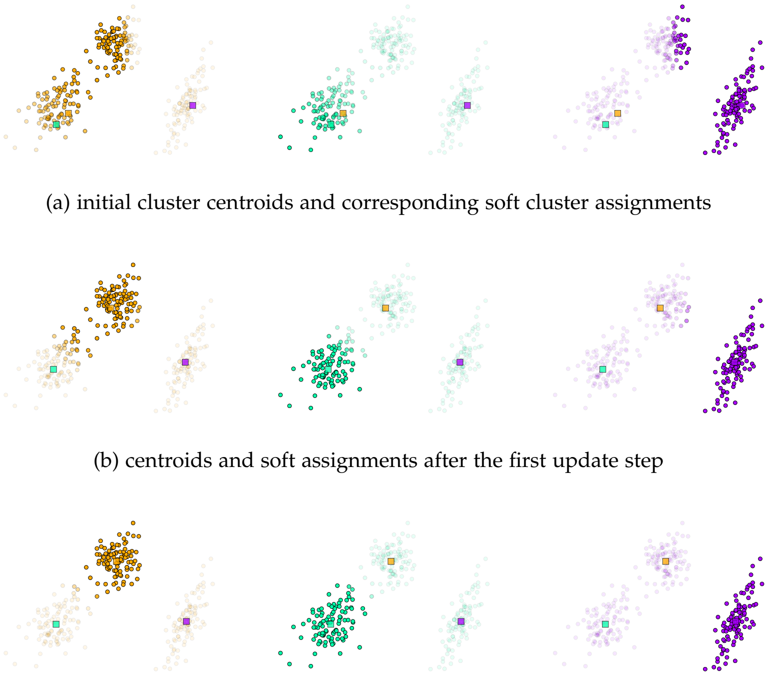
will minimize the contribution of cluster C_l to the overall error in (4).

[6] Recall the following rules from vector calculus

$$\frac{\partial}{\partial \boldsymbol{v}} \boldsymbol{u}^T \boldsymbol{v} = \boldsymbol{u}$$

and

$$\frac{\partial}{\partial \boldsymbol{v}} \boldsymbol{v}^T \boldsymbol{v} = 2\boldsymbol{v}.$$



(a) initial cluster centroids and corresponding soft cluster assignments

(b) centroids and soft assignments after the first update step

(c) centroids and soft assignments upon convergence

Figure 2: Soft k-means clustering for $k = 3$ of the data ($\circ$) in Fig. 1; (a) initial centroids ($\square$) and cluster assignments, (b) situation after the first update, and (c) result upon convergence.

In this example, the procedure in Fig. 3 converged within four iterations to a good solution. However, the caveats as to potentially questionable results we pointed out in our earlier discussions of the k-means procedure generally also apply to soft k-means clustering.

A video that visualizes the process shown in this figure can be found here: www.youtube.com/watch?v=Np9VuEg_aqo

Iterative Algorithm and Examples

Now that we know how to compute cluster assignments and cluster centroids under a soft clustering paradigm, it is easy to guess how the classical procedure (or Lloyd's algorithm) for hard k-means can be adapted to our current setting. Hence, without further ado: Fig. 3 presents pseudo code for soft k-means clustering.

FIGURE 2 ATTEMPTS TO VISUALIZE the behavior of this algorithm. Given the data in Fig. 1, we set the stiffness parameter β in (7) to 1.5, randomly initialized 3 cluster centroids, and ran soft k-means. For each cluster, Fig. 2 shows cluster centroids and cluster memberships during different iterations. Note that membership degrees are indicated in terms of shaded colors, the more opaque a data point is plotted, the more it belongs to the corresponding cluster[7].

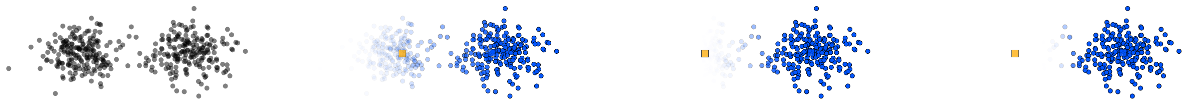
In this simple example, it took the algorithm only four iterations to move the cluster centroids to positions which most human observers would agree make sense. Generally, soft k-means clustering will be more stable than hard k-means clustering since the idea of degrees of cluster memberships causes the algorithm to optimize more globally rather than just locally. Yet, the performance of soft k-means, too, will depend on the initialization of the centroids and there are still no guarantees for the algorithm to find a globally optimal solution. Moreover, result obtained from soft k-means clustering will depend on the choice of the stiffness paramtere β.

FIGURE 4 PROVIDES AN IMPRESSION for how the choice of β may

```
$t \leftarrow 0$
randomly initialize centroids $\mu_1^t, \mu_2^t, \ldots, \mu_k^t$
repeat
    // compute all indicator variables
    $z_{ij} = \frac{\exp\left[-\beta \left\| x_j - \mu_i^t \right\|^2\right]}{\sum_l \exp\left[-\beta \left\| x_j - \mu_l^t \right\|^2\right]}$
    // update all centroids
    $\mu_i^{t+1} = \frac{\sum_j z_{ij} x_j}{\sum_j z_{ij}}$
    $t \leftarrow t + 1$
until centroids stabilize
```

Figure 3: Pseudo code for soft k-means.

[7] Also note that we cheated a bit when creating these plots; colors for low membership values are exaggerated for better visibility.



(a) data

(b) $\beta = \frac{1}{2}$

(c) $\beta = 1$

(d) $\beta = 2$

Figure 4: Examples of how the choice of the stiffness parameter β in (7) may impact soft k-means clustering.

influence the overall result of soft k-means clustering. In order to make sense of these results, we note that, w.r.t. each cluster C_i, the z_{ij} in (7) essentially follow a Gaussian distribution so that β can be seen as an inverse variance. Hence, larger values of β will cause the Gaussian to be narrower so that degrees of membership of data points $\boldsymbol{x}_j$ far from the centroid $\boldsymbol{\mu}_i$ will become small[8].

Notes and References

Before we conclude, we feel it is warranted to address the issue of "soft vs. fuzzy k-means" which seems to cause confusion in blogs and forums on the Web. To be specific, the debate revolves around the question of whether or not soft k-means clustering as discussed in this note and fuzzy k-means clustering are the same?

Note that the defining feature of fuzzy computing paradigms is that they are concerned with fuzzy sets. A fuzzy set S is a set whose elements s have degrees of membership often written as $m_S(s)$ where m_S is a function $m_S : S \to [0,1]$.

Looking at (7), we observe once again that the degrees of cluster membership defined there are values $z_{ij} \in [0,1]$. In fact, each z_{ij} is determined w.r.t. a cluster (a set) C_i and an element $\boldsymbol{x}_j$. Except for notational clutter, there is therefore no reason that would prevent us from writing $z_{C_i}(\boldsymbol{x}_j)$ instead of z_{ij}.

In other words, the probabilistic cluster membership indicators in (7) constitute but a specific choice of a fuzzy membership function. We could have chosen other functions such as triangle functions or trapezoids that are common in fuzzy logic. We could have also chosen a different function for each cluster. But, in the end, (7) can be understood as fuzzy membership function. Hence, soft k-means clustering as discussed above is but a specific instance of fuzzy k-means clustering. It is neither "better" or "worse". Any differences between the two "paradigms" boil down to parametrizations.

To finally conclude this note, we would like to point readers interested in textbooks dealing with soft k-means clustering to the excellent texts by MacKay[9] or Duda, Hart, and Stork[10].

[8] The obvious question then is how to choose β in practice? When in doubt, $\beta = 1$ is always a pragmatic choice. A generally better idea would be to set it as a multiple of the average distance between the given data points.

[9] D.J.C. MacKay. *Information Theory, Inference, & Learning Algorithms*. Cambridge University Press, 2003

[10] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001

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.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

D.J.C. MacKay. *Information Theory, Inference, & Learning Algorithms*. Cambridge University Press, 2003.