The handle http://hdl.handle.net/1887/74690 holds various files of this Leiden University dissertation.

**Author:** Kampert, M.M.D.
**Title:** Improved strategies for distance based clustering of objects on subsets of attributes in high-dimensional data
**Issue Date:** 2019-07-03
Chapter 4

COSA Nearest Neighbors: COSA-$K$NN and COSA-$\lambda$NN

In this chapter we will propose further improvements for COSA-$K$NN. With a change in notation, we can generalize the clustering problem from Chapter 2, into settings where the assumption of mutually exclusive clusters does not need to hold. Furthermore, with this changed notation we will introduce a COSA distance that is better in separating pairs of objects from different clusters. This new non-strict COSA-distance will render COSA-$K$NN more powerful. Lastly, we propose a $\lambda$-tuned strategy in which attribute weights are allowed to become equal to zero, and where the non-zero attribute weights are the result of a Kullback-Leibler divergence regularization.

Despite these improvements, a disadvantage remains: in COSA-$K$NN each object is assigned an equal number of $K$ nearest neighbors. Of the $K$ nearest neighbors belonging to an object in a small cluster, we may have some neighbors that are ‘living far apart’. Similarly, for objects belonging to the large clusters, we may have more objects ‘living close together’ than the $K$ positions available. For these situations, we may expect better results if COSA would allow for a varying number of nearest neighbors per object. This is accomplished in COSA-$\lambda$NN. Instead of a fixed $K$, COSA-$\lambda$NN sets a restriction on the space of the neighborhood around each object. The space of the neighborhood is bounded by a pre-set $\lambda$. Only those objects that live within the space of the neighborhood are entitled to be called a nearest neighbor.

To summarize, the chapter starts with the introduction of the improvements for COSA-$K$NN. When all these improvements are implemented in COSA-$K$NN, we will refer to its algorithm as COSA-$K$NN$_1$. What follows is a comparison between COSA-$K$NN$_1$ and the robust version of COSA-$K$NN from Chapter 3, referred to as COSA-$K$NN$_0$. The comparison consists of a simulation study and an application to a real data set. Then, in Section 4.3 we will introduce COSA-$\lambda$NN, which will be compared with both COSA-$K$NN$_1$ and COSA-$K$NN$_0$.
4.1 Improvements of COSA-\(K\)NN

4.1.1 A Notation for Overlapping Clusters

It would be a misunderstanding to think that COSA-\(K\)NN only copes with data where we assume that the underlying clustering structure consists of mutually exclusive clusters. The distances of COSA-\(K\)NN can also very well represent a distance of a structure of overlapping clusters. However, the notation with which COSA-\(K\)NN was described thus far, does not allow for such a clustering structure.

From rule (2.5) of Chapter 2 we obtain that the attribute weights for each object \(i\), denoted by \(w_i\), represents the attribute weight vector of the cluster that object \(i\) belongs to. However, when we wish to allow for overlapping clusters, such that object \(i\) could belong to multiple clusters, then the interpretation of \(w_i\) becomes more complicated. The object’s attribute weights vector becomes a mixture of multiple weight vectors of multiple clusters.

To allow for a clustering structure where each object can belong to multiple clusters, we simplify the notation by associating the attribute weights vectors solely with a group of objects \(C_l\), and not with an object on its own. Instead of \(M\) mutually exclusive true clusters in \(G\), we now define a grouping structure \(C\) that consists of \(L\) groups that may overlap, denoted by \(\{C_l\}_{l=1}^L\). Each group \(C_l\) has its own set of attribute weights. While in Chapter 2 the attribute weights vector for a (mutual-exclusive) true cluster \(G_l\) was denoted by \(w_l^*\), from here on we will denote the attribute vector by \(w_l\) for a group of objects that can represent a true cluster, and have an overlap with other clusters. In the columns of the \(P\) times \(L\) matrix \(W\) we collect each attribute weights vector \(w_l\). An element of \(W\) is denoted by \(w_{kl}\), and represents the weight for attribute \(k\) \(1 \leq k \leq P\) for group \(C_l\) \(1 \leq l \leq L\). A similar change is proposed for the attribute dispersions. We will denote by \(S_{kl}\) the average distance on attribute \(k\) within the \(l\)th cluster of objects. The reader who is familiar with algorithm 1 for COSA in Friedman and Meulman (2004) will see that we are using the notation of that algorithm. Note, however, that the notation here allows for more flexibility, e.g., overlapping clusters.

In previous chapters, we used the \(N \times N\) indicator matrix \(C\). For each object \(i\) (in the rows) it was indicated which objects \(j\) (in the columns) belonged to the group of the \(K\) nearest neighbors of object \(i\). From now on, we will change the definition of this indicator matrix as follows: the indicator matrix \(C\) is of size \(N \times L\) with elements \(\{c_{il}\}\). Here, \(c_{il} = 1\) indicates that object \(i\) belongs to the \(l\)th cluster of objects, wge holds for \(c_{il} = 0\).

4.1.2 COSA-\(K\)NN Reparametrized

In COSA-\(K\)NN, the clustering structure is approximated by \(\mathcal{C}\), consisting of \(L = N\) groups of objects, i.e. the number of groups is equal to the number of objects in the data. Particularly for COSA-\(K\)NN, each group of objects, \(C_l \in \mathcal{C}\), corresponds to a neighborhood of \(K\) objects and these neighborhoods are indexed by \(l\) \((1 \leq l \leq L = N)\). For each unique object \(j\) \((1 \leq j \leq N)\) there is a unique neighborhood \(C_l\) for which
the distance to all other objects \( i \in C_l \) is smaller than, or equal to, any other object \( i \notin C_l \). Thus, \( C_l \) can represent a true cluster, but it can also be a group of noise objects.

With the reparametrized \( N \times L \) matrix for \( C \), the non-robust version of the COSA-KNN criterion is

\[
Q(W, C) = \sum_{i=1}^{L} \left\{ \frac{1}{K} \sum_{i=1}^{N} c_{il} D_{ijl}[w_l] + \lambda \left( \sum_{k=1}^{P} w_{kl} \log (w_{kl}) + \log (P) \right) \right\},
\]

(4.1)

where the within-cluster distance between object \( i \) and \( j \) in group \( C_l \) is

\[
D_{ij}[w_l] = \sum_{k=1}^{P} w_{kl} d_{ijk},
\]

(4.2)

and \( j_l \) is defined as the object \( j \) for which the average distance to all other objects \( i \in C_l \) is smaller than the average distance of object \( j \) to the objects in any other group \( l' \), \( i \in C_{l'} \), i.e.,

\[
\left\{ K^{-1} \sum_{i=1}^{N} c_{il} D^{(\eta)}_{ijl}[W] \right\} < \left\{ K^{-1} \sum_{i=1}^{N} c_{il'} D^{(\eta)}_{ijl'}[W] \right\},
\]

(4.3)

given \( l \neq l' \), where

\[
D^{(\eta)}_{ijl}[W] = -\eta v_{ij}, \log \left\{ v_{ij}^{-1} \sum_{k=1}^{P} v_{ijk} \exp \left( -\frac{d_{ijk}}{\eta} \right) \right\}.
\]

(4.4)

This specific non-strict distance for COSA in equation (4.4) is based on the inverse exponential COSA distance, defined in (2.44) of Chapter 2. However, in this reparametrization, \( v_{ijk} \) is defined as

\[
v_{ijk} := \max \left\{ w_{kl}, w_{kl'} \right\},
\]

(4.5)

the maximum taken over the weight of attribute \( k \) in the neighborhood \( l_i \) for object \( i \), and the weight for attribute \( k \) in the neighborhood \( l_j \) of object \( j \). The neighborhood \( l_j \) is defined as

\[
l_j = \arg\min_l K^{-1} \left\{ \sum_{i=1}^{N} c_{il} D^{(\eta)}_{ijl}[W] \right\},
\]

(4.6)

corresponding to the unique neighborhood \( C_{l_j} \) for which the average distance of object \( j \) to all other objects \( i \in C_{l_j} \) is smaller than the average distance of object \( j \) to the objects of any other group with \( l \neq l' \).

In Chapter 2 we have described the algorithm that is used as an heuristic to find the estimates for the optimal \( C \) and \( W \), by minimizing the COSA-KNN criterion in (4.1). Given an estimate for \( W \), we find the minimum for \( C \) by applying a \( K \) nearest
neighbors method on the inverse exponential distance. Then, given \( C \), the optimal solution for each \( w_{kl} \) is achieved when \( W \) is the minimum of \( Q(W, C) \), this is when

\[
\hat{w}_{kl} = \exp\left(-\frac{S_{kl}}{\lambda}\right) \left/ \sum_{k'=1}^{P} \exp\left(-\frac{S_{k'l}}{\lambda}\right) \right.,
\]

where

\[
S_{kl} = \frac{\sum_{i=1}^{N} c_{il}d_{ijl}k}{K},
\]

the mean-based attribute dispersion for attribute \( k \) in the neighborhood \( C_l \). In the previous chapter we have seen that the COSA-KNN is improved when the median-based attribute dispersion is used. In this chapter we will propose two robust versions of the COSA-KNN algorithm, where the definition of the attribute dispersion is based on the median. Thus we re-define the attribute dispersion as

\[
S_{kl} = \text{median}\left(\{d_{ijl}k\}_{i \in C_l}\right),
\]

for the COSA-KNN algorithms.

Setting \( L = N \), the algorithm for COSA-KNN remains the same at this point, and is called COSA-KNN\(_0\), the benchmark COSA algorithm. The result is

\[
\text{COSA-KNN}\_0
\]

\begin{enumerate}
  \item Set: \( \lambda; K; \alpha \lesssim 0.1; L = N \)
  \item Initialize: \( \eta = \lambda; \ W = \{1/P\} \in \mathbb{R}^{P \times L} \)
  \item loop 
    \begin{enumerate}
      \item Compute distances \( D_{ij}^{(\eta)}[W] \) as in (4.4)
      \item \( \hat{C} \leftarrow K \) nearest neighbors on \( D_{ij}^{(\eta)}[W] \)
      \item \( \hat{W} \leftarrow \) Compute attribute weights \( W \) as in (4.7), using (4.9)
      \item Increase \( \eta : \eta + \alpha \star \lambda \)
    \end{enumerate}
  \item } until \( W \) stabilizes
  \item Output: \( D_{ij}[W] \).
\end{enumerate}

4.1.3 Improved Attribute Weights

The main motivation of COSA-KNN was to consider the analysis of high-dimensional data, as obtained in genomics, proteomics, and metabolomics. As compared to 2004, the typical kind of data sets that come from those areas of research have nowadays many more attributes, i.e. the \( P \) / \( N \) ratio is much larger. In particular, there are more attributes in the data set that do not play any role in any of the groups, rendering the signal-to-noise ratio to become smaller. Here, the behavior of COSA-KNN may be suboptimal. So far, COSA-KNN drives the attribute weights to have a small negative entropy, i.e., staying close to equal-valued attribute weights. Moreover, up till now, COSA-KNN does not allow for zero-valued attribute weights. We propose two
improvements for the attribute weights. First, we generalize the criterion of COSA-KNN. Instead of penalizing the divergence from equal attribute weights, we can also penalize the divergence from another type of pre-specified attribute weights. The second improvement is a modification of this generalized criterion, such that COSA-KNN assures that a pre-specified proportion of the attributes within each group, receives a zero-value weight.

Regularization of the Kullback-Leibler divergence

We can rewrite the criterion in equation (4.1) as

$$Q(W, C) = \sum_{l=1}^{L} \left\{ \frac{1}{K} \sum_{i=1}^{N} c_{il} D_{ij} \left[ W_l \right] + \lambda \sum_{k=1}^{P} w_{kl} \log \left( \frac{w_{kl}}{1/P} \right) \right\}, \quad (4.10)$$

from which we can obtain that the criterion of COSA-KNN consists of a regularized Kullback-Leibler divergence term (Kullback & Leibler, 1951) for each neighborhood, the divergence between $w_l$ and a set of $P$ uniform pre-specified attribute weights with the value $1/P$.

We can generalize the criterion by replacing each $1/P$ with a pre-specified attribute weight $u_{kl}$, for which it holds that for each group $C_l$ the attribute weights $\{u_{kl}\}_{k=1}^{P}$ belong to the probability simplex, i.e. $\sum_{k=1}^{P} u_{kl} = 1$. This gives

$$Q(W, C) = \sum_{l=1}^{L} \left\{ \frac{1}{K} \sum_{i=1}^{N} c_{il} D_{ij} \left[ W_l \right] + \lambda D_{KL}(W_l | U_l) \right\}, \quad (4.11)$$

where the Kullback-Leibler divergence is

$$D_{KL}(w_l | u_l) = \sum_{k=1}^{P} w_{kl} \log \left( \frac{w_{kl}}{u_{kl}} \right). \quad (4.12)$$

All pre-specifications for $u_{kl}$ that are different from $u_{kl} = 1/P$ for all $l$ and $k$, will result in more selective attribute weights for $w_l$. Given $C$, the optimal solution for the attribute weight $w_{kl}$ becomes

$$\hat{w}_{kl} = \frac{u_{kl} \exp \left( -S_{kl}/\lambda \right)}{\sum_{k'=1}^{P} u_{k'l} \exp \left( -S_{k'l}/\lambda \right)}. \quad (4.13)$$

By introducing $u_{kl}$, we can add prior knowledge about the attribute weights in the criterion (4.11). For example, when certain attributes for a certain group of objects are expected to be important for the clustering, then we can give these attributes higher pre-specified weight values, compared to the other attributes.

Prior information can also be incorporated based on information in the data. When the attributes are normalized to represent the same scale, our empirical evidence so far suggests to use

$$u_{kl} = \frac{\exp(-s_k/\lambda)}{\sum_{k'}^{P} \exp(-s_{k'}/\lambda)}, \quad (4.14)$$
where the lowercase $s_k$ is the interquartile range of attribute $k$ on all observations divided by 1.35. In this particular case, the pre-specified attribute weights are the same for each group $C_l$. Based on the concept that the interquartile range on a normalized attribute is smaller when clustering is present, given that clustering occurs within the range, lower values of the pre-specified attribute weights are to be expected for the high interquartile attribute ranges, and vice versa.

**Zero-Valued Attribute Weights**

To allow for zero-valued attribute weights we modify the pre-specified attribute weights into

$$u_{kl} = \frac{a_{kl} \exp(-s_k/\lambda)}{\sum_{k'} a_{k'l} \exp(-s_{k'}/\lambda)},$$

(4.15)

where $a_{kl}$ is the result of an indicator function (attribute activation function), which indicates whether attribute $k$ obtains a non-zero weight in group $C_l$, or not:

$$a_{kl} = \begin{cases} 1 & \text{if } S_{kl} \leq \tilde{S}_l(P_{\lambda}); \\ 0 & \text{if } S_{kl} > \tilde{S}_l(P_{\lambda}). \end{cases}$$

(4.16)

Here, $\tilde{S}_l(P_{\lambda})$ is the value of the $P_{\lambda}$th smallest attribute dispersion in group $C_l$, where

$$P_{\lambda} = \text{ceiling } (P \times \min(1, \lambda)).$$

(4.17)

Here, $P_{\lambda}$ is a $\lambda$ related number of attributes. Knowing $P_{\lambda}$, we obtain $\tilde{S}_l(P_{\lambda})$ as the statistic of order $P_{\lambda}$ in the set $\{S_{kl}\}_k^P$, where the elements are sorted by descending value. Thus, we assign a zero-value attribute weight, $w_{kl} = 0$, when the average distance on that attribute within group $C_l$ belongs to the higher averages, i.e. $S_{kl} > \tilde{S}_l(P_{\lambda})$.

**4.1.4 Stronger Between-Cluster COSA Distances**

To find the grouping structure $C$, we use the $K$ nearest neighbors method on COSA distances. So far, this was the inverse exponential distance $D_{ij}^{(\eta)}[W]$ in (4.4) (and from (2.56) in Chapter 2). As in Chapter 2 that when $\eta \to \infty$, this particular distance evolves into the ‘majorizing’ COSA distance $D_{ij}[W]$ in equation (2.11), i.e. a weighted sum of the attribute distances:

$$D_{ij}[W] = \sum_{k=1}^P v_{ijk} d_{ijk},$$

(4.18)

where now, the definition of $v_{ijk}$ is the maximum of $w_{kl_i}$ and $w_{kl_j}$, as provided in equation (4.5). Although the maximum assures that between-cluster COSA distances will be larger than within-cluster COSA distances, it does not strongly take into account the differences in the attribute weights of the two clusters. Whether the
attribute weights are \( \{w_{kl_i} = 0.1, w_{kl_j} = 0.9\} \) or \( \{w_{kl_i} = 0.8, w_{kl_j} = 0.9\} \), the influence of attribute \( k \) in the separation of objects \( i \) and \( j \) remains the same.

We can modify the definition of \( v_{ijk} \), as long as the between-cluster COSA distance obeys the following necessary condition: when \( w_{kl_i} = w_{kl_j} \) for each attribute \( k \), we have

\[
D_{ij}[W] = v_{ijk}d_{ijk} = \sum_{k=1}^{P} w_{kl_i}d_{ijk} = \sum_{k=1}^{P} w_{kl_j}d_{ijk}. \tag{4.19}
\]

We propose the following modification of the definition for \( v_{ijk} \). Let \( |w_{kl_i} - w_{kl_j}| \) be the absolute difference of the weights in groups \( C_{li} \) and \( C_{lj} \) for attribute \( k \), and let \( 1 - |w_{kl_i} - w_{kl_j}| \) be the overlap. Then, when we add the \( \lambda^{-1} \)-scaled odds of the difference versus the overlap of the two attribute weights to the prior definition of \( v_{ij} \) in equation (4.5), we obtain

\[
v_{ijk} := \max (w_{kl_i}, w_{kl_j}) + \lambda^{-1} \frac{|w_{kl_i} - w_{kl_j}|}{(1 - |w_{kl_i} - w_{kl_j}|)}. \tag{4.20}
\]

Given that the different clusters each have a different subset of attributes, the \( \lambda \)-scaled odds term in equation (4.20), assures a stronger separation between the objects from different clusters in the COSA distances. The more different the subsets of attributes, or the smaller \( \lambda \), the larger the between-cluster distance.

### 4.1.5 The Algorithm of COSA-KNN\(_1\)

At this point we have all the ingredients to write down the improved COSA-KNN algorithm, called COSA-KNN\(_1\). Similar to the COSA algorithms from the previous chapters, all these improvements have a corresponding approximate criterion that needs to be minimized over \( C \) for a fixed \( W \) (see Chapter 2). The approximate criterion is defined as

\[
\tilde{Q}^{(\eta)}(C, W) = \sum_{l=1}^{L} \left\{ \frac{1}{K} \sum_{i=1}^{N} c_{il} D_{ij}^{(\eta)}[W] \right\}. \tag{4.21}
\]

We sketch the algorithm as follows:

**COSA-KNN\(_1\)**

0: Set: \( \lambda; K; L = N; \{u_{kl}\}; P_\lambda = P \times \min(1, \lambda) \)
1: Initialize: \( \eta = \lambda; w_{kl} = u_{kl} \)
2: loop 
3: Compute distances \( D_{ij}^{(\eta)}[W] \) as in (4.4), using (4.20)
4: \( \hat{C} \leftarrow K \) nearest neighbors on \( D_{ij}^{(\eta)}[W] \)
5: \( \hat{W} \leftarrow \) Compute attribute weights for \( W \) as in (4.13), using (4.15)
6: Increase \( \eta : \eta + \alpha * \lambda \)
7: } until \( W \) stabilizes
8: Output: \( D_{ij}[W] \).
Thus, COSA-$K_{NN}$ incorporates the sparser attribute weights and the new COSA distances based on the pairwise weights in equation (4.20). Note that the homotopy path over $\eta$ consists of two steps. We have a start value of $\eta = \lambda$, and have set $\alpha = \infty$ in this chapter, instead of $\alpha = 0.1$ (the value used in the previous chapters). Using $\alpha = 0.1$ or $\alpha = \infty$ would not have changed any of the conclusions in this chapter. Thus, we use the inverse exponential distance in (4.4) as a ‘start’ distance and use the distance in equation (4.19) for the rest of the iterations in the algorithm. Last, we say that the solution for $W$ stabilizes when the average change in the attribute weights is smaller than $10^{-5}$.

### 4.2 Comparing COSA-$K_{NN}$ to COSA-$K_{NN}_0$

While using the median-based (robust) attribute disperions (see Chapter 3), we compare COSA-$K_{NN}$ with COSA-$K_{NN}_0$ on two simulated data sets and a real data set. The first simulated data set is generated from the COSA prototype model, whose structure is summarized in Chapter 1 (Figure 1.2), and Chapter 3 (Figure 3.1). The other simulated data set and the real data also have been used in Witten and Tibshirani (2010) to show that COSA-$K_{NN}_0$ was not able to cope with such data. Therefore, the second simulated data set is generated from a model that we call a ‘weak spot model’, since it shows the weak spot of the original COSA. The weak spot model was already displayed in Chapter 1 (Figure 1.1.), but with smaller variances, and it was shown for a different purpose. In this Chapter we describe the weak spot model in Section 4.2.2. The real data set contains gene expression in breast cancer tumors cells, obtained from a study by Perou et. al (2001). Except for the analysis of the real data set, the tuning parameters $\lambda$ and $K$ were set equal to 0.2 and $\sqrt{N}$ for the old and improved COSA-$K_{NN}$. For the real data set the only difference was that $\lambda$ was set equal to the value that made sure 496 attributes were selected out of 1753 for each neighborhood (cf. Witten and Tibshirani, 2010). In COSA-$K_{NN}_1$ we used the pre-specified attribute weights in (4.15).

#### 4.2.1 COSA’s Prototype Data

In Figure 4.1 we display the COSA distances for both COSA-$K_{NN}_0$ and COSA-$K_{NN}_1$ in average linkage dendrograms. It is clear that the clustering is sharper for the COSA-$K_{NN}_1$; the distances between the objects from different clusters are large, and the distances within the clusters are small. The distances between the noise objects have a higher variability, but there is no spurious clustering visible.
Figure 4.1: Average linkage dendrograms for the distances of COSA-KNN\(_0\) (left panel) and COSA-KNN\(_1\) (right panel).

The two-dimensional multidimensional scaling (MDS) solution of both sets of COSA distances is depicted in Figure 4.2. The two clusters (the red cluster and the blue cluster) are more homogeneous for COSA-KNN\(_1\), while no clear differences are visible between COSA-KNN\(_0\) and COSA-KNN\(_1\) for the noise objects (in grey).

Of the three improvements that define COSA-KNN\(_1\), compared to COSA-KNN\(_0\), it is the improved between-cluster COSA distance, based on the pairwise attribute weights in equation (4.20), that mainly explains the sharper clustering results of COSA-KNN\(_1\). The zero-value attribute weights play a less prominent, but a facilitating role. With the zero-value attribute weights, it is easier for the attribute weights of two clusters to become disjoint, rendering a larger between-cluster distance due to the ‘odds’ term in equation (4.20). For this specific simulation example no differences were found between the use of the pre-specified attribute weights defined in equation
or for $u_{kl} = 1/P$.

### 4.2.2 COSA’s Weak Spot

To obtain a good understanding of COSA, it useful to know the type of data that COSA was not developed for. As was already found by Peter Hoff in the discussion of the paper by Friedman and Meulman (2004), COSA mainly detects differences in the variances on attributes among groups, but not in the means. The clustering found in COSA is predominantly due to the objects having measurements at attributes that are tightly concentrated around common values, and less due to the differences with these common values from objects of another group.

In Figure 4.3, we depict a model for a data set on which COSA-$K_0$NN would not be able find the clustering. We will call this model the ‘weak spot’ model. Based on this model we generate data where the attributes only show differences in the means for each group, but not a difference in the variances. The data set has dimensionality $N = 60$ by $P = 1000$, and consists of three clusters of each 20 objects, clustering on the first 50 attributes coming from an i.i.d. normal distribution with standard deviation 1 and the means set at $\mu_{C_1} = -1.5$, $\mu_{C_2} = 0$, $\mu_{C_3} = 1.5$. All remaining data values are i.i.d. realizations from a standard normal distribution.

After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.

The two version of COSA have been applied with $\lambda = 0.2$ and $K = \text{ceiling}(\sqrt{N})$. The resulting complete linkage dendograms and MDS representations can be found in

---

*Figure 4.3: Data model with which data can be generated that emphasize COSA’s weak spot. After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.*
Figures 4.4 and 4.5, respectively. The dendrograms show that the clustering structure is not captured in the first set of COSA distances, but is captured with the distances from COSA-KNN\(_1\) (right panel).

Comparable results are obtained in Figure 4.5. While the color labels in the Figure hint that the two-dimensional MDS solution for the distances of COSA-KNN\(_0\) almost recovers the clustering structure, a good separation of the clusters is obtained with the distances of COSA-KNN\(_1\) (right panel).

Thus, compared to COSA-KNN\(_0\), COSA-KNN\(_1\) is not only able to result in sharper clustering, it is also better in detecting mean differences on attributes between groups. This improvement is ascribed to the definition of the pre-specified attribute weights, \(\{u_{kl}\}_{k=1}^P\) in equation (4.15), for each cluster. Due to the multi-modal distribution of the attributes that are important for the clustering, a lower inter-quartile
range is obtained on these attributes, rendering higher values for the pre-specified attribute weights.

### 4.2.3 Breast Cancer Data

In Perou et al. (2000), 65 surgical specimens of human breast tumors were profiled on gene expression microarrays. Some of the samples were taken from the same tumor before and after chemotherapy. In the study 62 of the 65 samples were classified into four categories: "basal-like" (eight observations, displayed by red), "Erb-B2" (seven observations, in green), "normal breast-like" (blue, eleven observations), or "ER+" (orange, 46 observations). The three remaining observations did not belong to any of the four classes. These classes were found when 65 samples were hierarchically clustered using 496 out of the available 1753 genes. When all 1753 genes were used, these clusters did not reveal themselves.

With only a few misclassifications, Witten and Tibshirani (2010) did manage to almost perfectly recover the four classes on the 62 samples and 1753 genes with their type of distances: a weighted sum of squared euclidean attribute distances. The attribute weights in this particular sum are restricted by an $\ell_2$ and $\ell_1$ norm, and are the same for all clusters. Witten and Tibshirani (2010) also showed that COSA was not successful on this specific data set. Figures 4.6 and 4.7 show that, indeed, the dendrograms and the two-dimensional MDS solutions of COSA-$K_{NN_0}$ are not successful. Although suboptimal, the results for COSA-$K_{NN_1}$ are much better.

![Figure 4.6: Complete linkage dendrograms for two sets of COSA distances on the Breast Cancer data. The COSA-$K_{NN_0}$ distances are on the left, the COSA-$K_{NN_1}$ distances are on the right. Here, the color labels have the following representation; the basal-like tumors are displayed in red, the Erb-B2 tumors in green, the normal breast-like tumors in blue, the ER+ tumors are displayed in orange.](image)
Figure 4.7: Two-dimensional MDS solutions for the two sets of COSA distances on the Breast cancer data: COSA-KNN\(_0\) (left) vs. COSA-KNN\(_1\) (right). The basal-like tumors are displayed in red, the Erb-B2 tumors in green, the normal breast-like tumors in blue, the ER+ tumors are displayed in orange.

4.3 COSA-λNN

In this section, we propose to circumvent choosing the value of the tuning parameter \(K\), by letting \(λ\), instead of \(K\), indirectly control the size of each group of objects (\(C_l\)). The algorithm that incorporates this modification is called COSA-λNN. A motivation for this improvement will follow from a description of a general form of the COSA criterion for COSA Nearest Neighbors algorithms, and the (mean-based) attribute dispersions.

4.3.1 Motivation based on the Criterion

So far, all versions of the COSA-KNN algorithms fit in the format of the COSA criterion defined as

\[
Q(W, C) = \sum_{l=1}^{L} \sum_{k=1}^{P} \left\{ w_{kl} \left( S_{kl} + \lambda \log \left( \frac{w_{kl}}{u_{kl}} \right) \right) , \right. \\
\left. \quad 0, \quad \text{if} \ u_{kl} \neq 0; \right. \\
\left. \quad \quad \quad \quad \text{if} \ u_{kl} = 0. \right. \\
(4.22)
\]

As was shown in the previous chapter, Section 3.2.1, when we minimize \(Q(W, C)\) in equation (4.22) over the attribute weights (\(W\)), the criterion has a simpler form, i.e.,

\[
Q(C) = \sum_{l=1}^{L} -\lambda \log \left\{ \sum_{k=1}^{P} u_{kl} \exp \left( -\frac{S_{kl}}{\lambda} \right) \right\} . \\
(4.23)
\]

When each \(S_{kl}\) is defined as in equation (4.9), and each \(u_{kl}\) as in equation (4.15), then \(Q(W, C)\) and \(Q(C)\) represent the criteria for COSA-KNN\(_1\). For \(Q(W, C)\) and \(Q(C)\) to be the criteria of COSA-KNN\(_0\) the value for each \(u_{kl}\) is set equal to \(1/P\). For the non-robust version of the COSA-KNN criterion (see Chapter 3), the definition
of each attribute dispersion $S_{kl}$ should be the average of the $K$ attribute distances within the neighborhood $C_l$; also, each $u_{kl} = 1/P$.

All versions of COSA-KNN have two tuning parameters, $\lambda$ and $K$. In general, for a larger $K$, the COSA-KNN criterion obtains a higher value. For an example see Figure 4.8. In the two previous chapter we also described how this behavior can be modified by $\lambda$: it could be that a certain larger value for $K$ assures that for a certain fixed value of $\lambda$ the attribute weights can be defined more sharply, such that the value of the criterion becomes lower. However, when the value for $K$ becomes too large, the role of $\lambda$ in the attribute selection becomes smaller, since for $K \rightarrow N$ we have $S_{kl} = S_{kl}'$ due to the standardized attribute distances (see Chapter 2, equation (2.1)), and hence, all neighborhoods will have the same attribute weights. Therefore, according to Friedman and Meulman (2004), the value for $K$ should not be larger than the cluster sizes in the data, but should also not be too small, to avoid unstable attribute dispersions. However, for some situations it will be difficult to choose a $K$ when the data consists of clusters varying highly in size.

4.3.2 COSA-\(\lambda\)NN: Different Sizes for $C_l$

In COSA-\(\lambda\)NN, the size of each neighborhood may be different, while in COSA-KNN the size for each neighborhood of objects is set equal to $K$. We will denote the size of each $C_l$ in COSA-\(\lambda\)NN by $N_l$. To obtain the value of each $N_l$, we exploit the behavior of the attribute dispersions, as was explained for the COSA-KNN criterion in the previous section. We have seen that for $Kl \rightarrow N$, all attribute dispersions will become equal, and we fail to detect a clustering structure since we cannot distinguish the signal
attributes from the noise. Relying on the assumption that of the \( P \) attributes, there are many that just represent noise values for a cluster, it could be useful to detect a sharp distinction between large values of the attribute dispersions (mostly noise), and small values of the attribute dispersion (mostly signal).

In COSA-\( \lambda \)-NN, the idea is to set the value of each \( N_l \) to create a sharp distinction between noise and signal attribute dispersions. In particular, we base the size of each neighborhood on a type of variance of \( \{S_{kl}\} \) in neighborhood \( C_l \) and the Kullback-Leibler divergence \( D_{KL}(\mathbf{w}_l | \mathbf{u}_l) \) in equation [4.12]. Compared to the median-based attribute dispersions, the value of the variance of the mean-based attribute dispersions in \( C_l \) is more sensitive to different values of \( N_l \). Similarly, compared to median-based attribute dispersions, the value of \( D_{KL}(\mathbf{w}_l | \mathbf{u}_l) \) is more sensitive to different values of \( N_l \) when based on the mean-based attribute dispersions. Therefore, in COSA-\( \lambda \)-NN, the definition of the attribute dispersion is based on the mean, i.e.

\[
S_{kl} = \frac{\sum_{i=1}^{N} c_{il} d_{ij_k}}{N_l}.
\]  

(4.24)

Here, \( c_{il} = 1 \) for the objects \( i \) that belong to the \( N_l \) nearest neighbors of \( j_l \), and \( c_{il} = 0 \) otherwise. Thus,

\[
N_l = \sum_{i=1}^{N} c_{il}.
\]  

(4.25)

Remember that \( j_l \) in equation [4.3] denotes the object \( j \) for which all objects in the neighborhood \( C_l \) are the nearest \( N_l \) neighbors. The nearest neighbors are found based on \( D^{(\eta)}(\mathbf{W}) \) in equation [4.4], in which equation [4.20] is used for the definition of \( v_{ijk} \).

We find \( N_l \) for each \( C_l \), by adding the next closest object until we reach a ‘certain’ (sub)optimal value for a particular ratio between \( D_{KL}(\hat{\mathbf{w}}_l | \mathbf{u}_l) \) and the weighted variance of \( \{S_{kl}/\lambda\} \). Here, the variance of the attribute dispersions, \( \{S_{kl}\}_{k=1}^{P} \), in neighborhood \( C_l \) is weighted by \( \hat{\mathbf{w}}_l \), i.e.

\[
\text{VAR} (S_{kl}) = \sum_{k=1}^{P} \hat{w}_{kl} S_{kl}^2 - \left( \sum_{k=1}^{P} \hat{w}_{kl} S_{kl} \right)^2
\]  

(4.26)

Since both \( D_{KL}(\hat{\mathbf{w}}_l | \mathbf{u}_l) \) and \( \text{VAR} (S_{kl}) \) are functions of \( \lambda \), the sizes of the neighborhoods will also be a function of \( \lambda \) – hence the name: COSA-\( \lambda \)-NN.

### 4.3.3 Selection of \( N_l \) based on \( D_{KL}(\hat{\mathbf{w}}_l | \mathbf{u}_l) \) and \( \text{VAR} (S_{kl}) \)

COSA-\( \lambda \)-NN exploits two relations to obtain the size of each neighborhood. The relation between \( D_{KL}(\mathbf{w}_l | \mathbf{u}_l) \) in [4.12] and \( N_l \), as well as the relation between \( \text{VAR} (S_{kl}) \) in [4.26] and \( N_l \). Within a neighborhood of objects that cluster on the same subset of attributes, we expect a relatively small subset of attributes each with a low \( S_{kl} \), and a high attribute weight \( \hat{w}_{kl} \). For the remaining attributes we expect large attribute dispersions, and low attribute weights. In particular, the higher the number of objects
that cluster on the same subset of attributes, the sharper this distinction. Thus, as long as we increase $N_l$ with objects that cluster on the same subset of the attributes, we obtain more or less a higher variance of the weighted attribute dispersions.

At a certain point, this particular variance reaches either an inflection point (from concave to convex) or a local maximum for $N_l$. When approaching such a point for $\text{VAR}(S_{kl})$, each next nearest object that is being added to the neighborhood does not give a large increase in the variance anymore. For example, it could be that for a neighborhood the subsets of attributes that contribute to the clustering have been discovered. Then, adding an extra object that clusters on about the same subset of attributes does not contribute much to a sharper distinction between the relevant and the non-relevant subset of attributes. By increasing $N_l$ after such an inflection point has been reached, we are probably adding objects that (also) cluster on other non-overlapping subsets of attributes. Therefore, the variance of $S_{kl}$ will decrease, since the attribute dispersions will become more similar on more of the attributes for $k$. Note that the attribute dispersions that are based on larger $N_l$, will become more similar to each other since each $d_{ijk}$ is scaled for each attribute (see Chapter 2, equation (2.1)).

As is depicted for an example in Figure 4.9, it may happen that $\text{VAR}(S_{kl})$ approaches an inflection point first, and that the variance starts to increase again for larger $N_l$. An explanation is that after such an inflection point, we may be adding objects to the neighborhood that do not cluster on the same subset of attributes but on a larger overlapping subset of attributes.
Figure 4.9: $D_{KL}(\hat{w}_l|u_l)$ and $\text{VAR}(S_{kl})$ plotted against $N_l$ for the neighborhood of an object $j_l$ (left panel) that belongs to the ‘blue’ cluster of the prototype model. The blue vertical line is at $N_l = 15$ and the purple vertical line is at $N_l = 30$. In the right panel a similar plot is shown for a noise object $j'_l$ of neighborhood $l'$ from the same data set of the prototype COSA model. The nearest neighbors were based on the COSA-\lambda NN distances that resulted from COSA-\lambda NN with \lambda = 0.2.

Figure 4.9 shows the relation between $\text{VAR}(S_{kl})$ and $N_l$ for a neighborhood $l$ around an object $i_l$ that belongs to the blue cluster of the prototype COSA model (left panel), and for a neighborhood $l'$ around a noise object $i'_l$ (right panel). These two objects come from the same data set that was generated from prototype model in Section 4.2.1. For an increasing number of nearest neighbors of $j_l$, we see an increase in $\text{VAR}(S_{kl})$ until we reach a point of diminishing increases around $N_l = 15$ (the blue vertical bar). Then, we reach a second point of diminishing increases for $\text{VAR}(S_{kl})$ at $N_l = 30$. At this point the neighborhoods consist of the blue and red clusters of the prototype model. After this point, we see that the variance of the attribute dispersions decreases rapidly. Figure 4.9 also shows the relation between $D_{KL}(\hat{w}_l|u_l)$, in (4.12), and $N_l$. Although smoother, its pattern is quite similar to that of $\text{VAR}(S_{kl})$.

In the right panel of Figure 4.9 we show a similar plot, but then for one of the noise objects (‘grey’). Note that for each $N_l'$ in the neighborhood of this noise object, both the Kullback-Leibler divergence and the variance of the attribute dispersions are smaller. Moreover, the $D_{KL}(\hat{w}_{l'}|u_{l'})$ starts decreasing already after a much smaller value of $N_l'$. The same holds for $\text{VAR}(S_{kl})$, however it shows a less stable pattern. Thus both $D_{KL}(\hat{w}_l|u_l)$ and $\text{VAR}(S_{kl})$ give us information about the size of the neighborhood for each object.

The ratio of $\text{VAR}(S_{kl})$ to $D_{KL}(\hat{w}_l|u_l)$ is even more powerful in revealing the
Improved Strategies for COSA

inflection point of interest. We denote this ratio by \( \lambda^*_N \), and it is defined as

\[
\lambda^*_N = \sqrt{\frac{\text{VAR}(S_{kl})}{D_{KL}(\hat{w}_l | u_l)}}.
\]  

(4.27)

Compared to the left panel of Figure 4.9, in Figure 4.10 the stationarity point for the signal object in the region of \( N_l \in \{8, 9, \ldots, 15\} \) is better visible. For the noise object (in grey), the resulting line Figure 4.10 consists of more fluctuations when compared with the results in the right panel of Figure 4.9.

\[ \text{Figure 4.10: } \lambda^*_N \text{ plotted against } N_l \text{ for the neighborhood of an object belonging to the cluster and for } N_{l'} \text{ of the neighborhood of a noise object.} \]

Based on the value of \( \lambda^*_N \), COSA-\( \lambda \)-ANN finds each \( N_l \). The value for \( N_l \) is defined as the first neighborhood size for which

\[
\lambda^*_N \geq \lambda^*_N + 1.
\]  

(4.28)

For each neighborhood \( l \), COSA-\( \lambda \)-ANN starts with \( N_l = 2 \) and computes \( \lambda^*_N \). Then, COSA-\( \lambda \)-ANN increases \( N_l \) until it observes (4.28). Thus,

\[
C_l = i \in \lambda_{NN}(j) = \{i \mid \lambda^*_N \geq \lambda^*_N + 1\}.
\]  

(4.29)

With Figures 4.9 and 4.10, we have illustrated the conceptual idea how COSA-\( \lambda \)-ANN finds \( N_l \) using \( \lambda^*_N \). Note that the definition for \( \lambda^*_N \) is not just arbitrary. This definition in equation (4.28) can also be seen as a solution for the minimization problem of a particular constrained version of the COSA criterion (Appendix Section 4.6.1), i.e., it is a solution that satisfies the Complementary Slackness condition of the Karush-Kuhn-Tucker (KKT) conditions (see, e.g., Boyd & Vandenberghe 2004).
4.3.4 The COSA-\(\lambda\)NN Algorithm

We sketch the algorithm COSA-\(\lambda\)NN as follows:

\[
\text{COSA-}\lambda\text{NN} \\
0: \text{Set: } \lambda; \ L = N; \ \{u_{kl}\}; \ P_\lambda = P \times \min(1, \lambda) \\
1: \text{Initialize: } \eta = \lambda; \ w_{kl} = u_{kl} \\
2: \text{loop} \{ \\
3: \text{Compute distances } D^{(\eta)}_{ij}[W] \text{ as in } (4.4), \text{ based on } (4.20) \\
4: \text{For each } l_j, \text{ set } N_{l_j} = 1 \\
5: \text{For each } l_j, \text{ loop} \{ \\
6: \ N_{l_j} \ = N_{l_j} + 1 \\
7: \ \{\hat{c}_{i(l_j)}\}_{i=1}^{N_{l_j}} \leftarrow N_l \text{ nearest neighbors on } D^{(\eta)}_{ij}[W] \\
8: \ \hat{w}_{l_j} \leftarrow \text{Compute attribute weights for } w_{l_j} \text{ as in } (4.13) \\
9: \} \text{ until each } N_l \text{ satisfies condition } (4.28) \\
10: \text{Increase } \eta : \eta + \alpha \lambda \\
11: \} \text{ until } W \text{ stabilizes} \\
12: \text{Output: } D_{ij}[W].
\]

4.4 Comparing COSA-\(\lambda\)NN with COSA-\(K\)NN

In this section we will add the results of the COSA-\(\lambda\)NN to the results of COSA-\(K\)NN\(_0\) and COSA-\(K\)NN\(_1\) obtained in Section 4.2. In particular, the same data set has been analyzed with COSA-\(\lambda\)NN: the data sets generated from the prototype model, the data set generated from the weak spot model (Figure 4.3), and the breast cancer data (Perou et. al., 2001).

4.4.1 Simulated Data

For the data set that was generated from the COSA prototype model, it is difficult to say whether the results obtained with COSA-\(\lambda\)NN show an improvement over COSA-\(K\)NN\(_1\). Figure 4.11 shows that the distances between the clusters and the group of noise objects remain the same. There is a noteworthy difference between the dendrogram of the COSA-\(\lambda\)NN (right panel) and that of COSA-\(K\)NN\(_1\) (middle panel). For the results of COSA-\(\lambda\)NN the distances between the noise object pairs are more variable. These varying distances may have the disadvantage that some noise object pairs, trio’s, or even some quartets may appear closer to each other.
Figure 4.11: The resulting dendrograms of the distances from the Old COSA-KNN₀ (left), COSA-KNN₁ (middle), COSA-ANN (right) obtained from a data set of the prototype model.

Figure 4.12 shows the Multidimensional Scaling (MDS) results for COSA-λNN in the right panel. Here, the MDS solution of the COSA-λNN results in a sharp and similar clustering structure as that of the COSA-KNN₁ results. Note that the distances between the noise objects do not appear to be smaller, as was the case in the dendrograms. Consistent with the dendrogram for COSA-λNN, though less clearly visible, there are fewer noise objects that are closer-by to the clustered objects (red and green) as compared to the MDS solution for COSA-KNN.

Figure 4.12: The resulting MDS solutions of the distances from the Old COSA-KNN (left), COSA-KNN₁ (middle), COSA-λNN (right) obtained from a data set of the prototype model.

Very similar dendrograms and MDS solutions are obtained from the results of COSA-λNN and the results of COSA-KNN₁ on the data that was generated from the COSA weak spot model, see Figure 4.13. It may seem that the dendrograms indicate that the cophenetic distance between the blue cluster and the green cluster is larger for the COSA-KNN distances, when compared to those obtained by COSA-λNN distances. However, these results do not show in the MDS solutions.
4.4.2 Breast Cancer Data

Although COSA does not ‘perfectly’ replicate the clustering structure in the breast cancer data set, the visualizations of the COSA-\(\lambda\)NN results capture the structure better than the COSA-KNN results, see Figure 4.14. The dendrogram of the COSA-\(\lambda\)NN distances seem to show the fewest missclassifications.
Figure 4.14: The complete linkage dendrograms and MDS solutions of the distances from the results of COSA-\textit{KNN}_0 (left), COSA-\textit{KNN}_1 (middle), and COSA-\textit{\lambdaNN} (right) that were obtained from the Breast cancer data set.

4.5 Discussion

In this chapter we introduced a new notation and proposed to implement prior attribute weights, sparser attribute weights and a new COSA distance. We have seen that these implementations render COSA-\textit{KNN} to reveal a sharper clustering both for simulated data examples and a real data set. Without these implementations COSA-\textit{KNN} would not be able to separate clusters that mainly differ in their mean on attributes where the within-cluster variances are equal to those of the noise objects.

Furthermore, we have also shown that we can do without the tuning parameter \textit{K}. Instead of fixing the size of each neighborhood to \textit{K}, we proposed COSA-\textit{\lambdaNN} to restrict the neighborhood space by \(\lambda_{NI}^*\), whose value is found based on \(\lambda\). We have shown that COSA-\textit{\lambdaNN} does equally well as the improved version of COSA-\textit{KNN}, if not better. Instead of setting the size of each neighborhood to a fixed number \textit{K}, COSA-\textit{\lambdaNN} restricts the neighborhood space by \(\lambda_{NI}^*\), which allows for neighborhoods of different sizes. This is something that is shown to be useful for the Breast cancer data, where the ground truth clusters were of unequal sizes. The advantage of this improvement is not only that each neighborhood can be of a different size, but also comes with a large reduction in computing costs since the value for \textit{K} does not need to be tuned anymore. While COSA-\textit{KNN} needs tuning for all combinations of \textit{K} and
\( \lambda \), with COSA-\( \lambda \)NN only the value for \( \lambda \) needs to be tuned.

Although the scale parameter \( \lambda \), as well as the homotopy parameter \( \eta \) have default values that are set at \( \lambda = \eta = 0.2 \), successful results will be shown in the rest of this monograph where the value \( \lambda \) in COSA-\( \lambda \)NN is tuned with the Gap statistic procedure (Tibshirani et al., 2001). Concerning the relaxation of the \( \eta \) parameter, so far, a good choice of its value remains dependent on empirical experience. However, in this chapter and the rest of the Monograph we have shortened the homotopy path for \( \eta \), in all versions of COSA-KNN and COSA-\( \lambda \)NN. While at the start we remain with the initialization of \( \eta = \lambda \), after our first estimate of \( \mathbf{C} \), we directly set \( \eta = \infty \). A theoretical framework for the relaxation of \( \eta \) may need further investigation.

What we did not show is whether the COSA-KNN\(_1\) and COSA-\( \lambda \)NN are more prone to Type-I errors. Empirical evidence so far suggests that this is not the case. For example, even after the application of the Gap statistic procedure, no clustering structures were found in a data set of 100 objects, and \( P = 1000 \) or \( P = 10,000 \) i.i.d. attribute values from a standard normal distribution. In addition, the standard error on the non-smooth fluctuations of the Gap statistics were also a good indication that the solutions obtained with COSA-KNN\(_1\) and COSA-\( \lambda \)NN may have been based on noise-only data examples.
4.6 Appendix

4.6.1 Minimizing the A Constrained COSA Criterion

Suppose we have a constrained COSA criterion we wish to minimize.

\[
\widehat{W} = \arg\min_{W \in \mathcal{W}} \left\{ \sum_{l=1}^{L} \sum_{k=1}^{P} w_{kl} S_{kl} \right\}, \tag{4.30}
\]

where for each column \( l \) of \( W \) in the space \( \mathcal{W} \) we have

\[
\sum_{k=1}^{P} w_{kl} \log \left( \frac{w_{kl}}{u_{kl}} \right) \leq \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2},
\]

and

\[
\sum_{k=1}^{P} w_{kl} - 1 = 0, \text{ for all } l.
\]

Here, each \( u_{kl}, S_{kl} \) and \( \lambda^*_l \) is pre-specified. The values for each \( u_{kl} \) are in the real interval \([0, 1]\), and have the property:

\[
\sum_{k=1}^{P} u_{kl} = 1. \tag{4.31}
\]

The value for each \( \lambda^*_l \) is greater than 0. The value for \( \text{VAR}(S_{kl}) \) follows from \( u_{kl}, S_{kl} \) and \( \lambda^*_l \). Its definition is

\[
\sum_{k=1}^{P} w^*_k S^2_{kl} - \left( \sum_{k=1}^{P} w^*_k S_{kl} \right)^2,
\]

where

\[
w^*_{kl} = \frac{u_{kl} \exp \left( \frac{-S_{kl}}{\lambda^*_l} \right)}{\sum_{k'=1}^{P} u_{k'\ell} \exp \left( \frac{-S_{k'\ell}}{\lambda^*_l} \right)}.
\]

(4.33)
The KKT Conditions

The solution for $W$ can be obtained by the Karush-Kuhn-Tucker conditions, these are

(C1) Stationarity:

\[ 0 \in \partial_w \sum_{l=1}^L \sum_{k=1}^P a_{kl} w_{kl} S_{kl} \]

\[ + \partial_w \lambda_l \left\{ \sum_{l=1}^L \sum_{k=1}^P w_{kl} \log \left( \frac{w_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda_l^*)^2} \right\} \]

\[ + \partial_w \sum_{l=1}^L \delta_l \left( \sum_{k=1}^P w_{kl} - 1 \right); \quad (4.34) \]

(C2) Complementary slackness: for each $l$, we have

\[ \lambda_l \left\{ \sum_{k=1}^P w_{kl} \log \left( \frac{w_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda_l^*)^2} \right\} = 0; \quad (4.35) \]

(C3) Primal feasibility:

\[ \sum_{l=1}^L \sum_{k=1}^P w_{kl} \log \left( \frac{w_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda_l^*)^2} \leq 0, \quad (4.36) \]

and

\[ \sum_{k=1}^P w_{kl} - 1 = 0, \text{ for all } l. \quad (4.37) \]

(C4) Dual feasibility:

\[ \lambda_l \geq 0, \text{ for all } l. \quad (4.38) \]

Stationarity and primal solution

Let us start with the stationarity condition (C1). Since the right-hand side of (4.34) is a sum over $L$ neighborhoods, we can rewrite the derivative with respect to $w_{kl}$ for each neighborhood separately. Each such function for $l$ can be written as

\[ \hat{w}_{kl} = \frac{u_{kl} \exp\left(-\frac{S_{kl}}{\lambda_l^*}\right)}{\sum_{k'=1}^P u_{k'l} \exp\left(-\frac{S_{k'l}}{\lambda_l^*}\right)}. \quad (4.40) \]
Note that the derivations for the solution \( \hat{w}_{kl} \) are similar to those given in the Appendix of Chapter 2 (Section 2.7.1). However, there is a difference. Here, \( \lambda_l \) is not specified beforehand, it needs to be found such that the solution for \( \hat{w}_{kl} \) obeys the primal feasibility condition.

**(C2) Complementary slackness and dual solution**

Here, the complementary slackness condition represents the solution we need to search for, if we wish to obtain strong duality. After we have obtained the solution \([4.40]\) that solves the primal problem, we are left with the Lagrange dual function for each \( l \):

\[
g(\lambda_l) = \sum_{k=1}^{P} \hat{w}_{kl} S_{kl} + \lambda_l \left( \sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2} \right) 
- \lambda_l \log \left( \sum_{k=1}^{P} u_{kl} \exp \left( -\frac{S_{kl}}{\lambda_l} \right) \right) - \lambda_l \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2}.
\]

\[ (4.41) \]

Here, \( g(\lambda_l) \) is a first lower bound for our restricted minimization problem in \([4.30]\) for any \( \lambda_l > 0 \) and \( \delta_l \).

The highest lower bound is given when we can find the value for \( \lambda_l \) that solves \( \frac{\partial g(\lambda_l)}{\partial \lambda_l} = 0 \), where

\[
\frac{\partial g(\lambda_l)}{\partial \lambda_l} = \sum_{k=1}^{P} \frac{u_{kl} \exp \left( -\frac{S_{kl}}{\lambda_l} \right)}{\sum_{k'=1}^{P} u_{k'l} \exp \left( -\frac{S_{k'l}}{\lambda_l} \right)} \left( -\frac{S_{kl}}{\lambda_l} \right) 
- \log \left( \sum_{k=1}^{P} u_{kl} \exp \left( -\frac{S_{kl}}{\lambda_l} \right) \right) - \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2}
= \sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2}.
\]

\[ (4.42) \]

However, setting \([4.42]\) equal to zero and solving for \( \lambda_l \) is quite difficult. In accordance with the complementary slackness condition, we need to find the value for \( \lambda_l \) by numerical optimization such that

\[
\sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) = \frac{\text{VAR}(S_{kl})}{(\lambda^*_l)^2}.
\]
4.6.2 When $\lambda_{N_l}^*$ and $\lambda_l$ Satisfy Complementary Slackness

Suppose $\lambda_l^* = \lambda_l$, and we have

$$\sum_{k=1}^P \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) - \frac{\text{VAR}(S_{kl})}{(\lambda_l^*)^2} = 0, \quad (4.43)$$

then, we have

$$\lambda_l^* = \sqrt{\frac{\sum_{k=1}^P \hat{w}_{kl} S_{kl}^2 - \left( \sum_{k=1}^P \hat{w}_{kl} S_{kl} \right)^2}{\sum_{k=1}^P \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right)}}. \quad (4.44)$$

Noteworthy is that the condition in equation (4.43) can be achieved for $\lambda_{N_l}^* = \lambda_l$, when

$$\frac{\partial}{\partial \lambda_l} \left\{ \lambda_l \sum_{k=1}^P \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) \right\} = \sum_{k=1}^P \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) - \text{VAR}(S_{kl}) \left( \frac{\lambda_l^*}{\lambda_l} \right)^2 = 0. \quad (4.45)$$

When the Kullback-Leibler divergence is multiplied with $\lambda_l$, it becomes a concave function for the value of $\lambda_l$, as can be seen in the right panel of Figure 4.15. The maximum of this function is achieved when Complementary Slackness holds given that $\lambda_{N_l}^* = \lambda_l$ (equation 4.43).

![Figure 4.15: In the left panel the Kullback-Leibler divergence (black) and the variance of the dispersions (red) are plotted against $\lambda_l$. In the right panel the KL-divergence, as well as the variance term, is multiplied by $\lambda_l$.](image-url)
Derivation

Given that \( u_{kl} \) is independent from \( \lambda_l \), then, the value of \( \lambda_l \) that obeys (4.45), also maximizes

\[
\lambda_l \sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right).
\]  
(4.46)

Since the Kullback-Leibler divergence is concave when multiplied by \( \lambda_l \), we can find the maximum when we solve

\[
\frac{\partial}{\partial \lambda_l} \left\{ \lambda_l \sum_{l=1}^{L} \sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right) \right\} = 0,
\]  
(4.47)

with respect to \( \lambda_l \). We can also express this equation as,

\[
\frac{\partial}{\partial \lambda_l} \left( -\lambda_l \log \left( \sum_{k=1}^{P} u_{kl} \exp \left( -\frac{S_{kl}}{\lambda_l} \right) \right) \right) - \frac{\partial}{\partial \lambda_l} \sum_{k=1}^{P} \hat{w}_{kl} S_{kl} = 0.
\]  
(4.48)

The derivative w.r.t. \( \lambda_l \) for the first term, the \( \lambda_l \)-inverse exponential mean of the attribute dispersions, is as follows:

\[
\frac{\partial}{\partial \lambda_l} - \lambda_l \log \left( \sum_{k=1}^{P} u_{kl} \exp \left( -\frac{S_{kl}}{\lambda_l} \right) \right) = \sum_{k=1}^{P} \hat{w}_k \log \left( \frac{\hat{w}_k}{u_k} \right).
\]  
(4.49)

The second part, the derivative of the weighted mean of the attribute dispersions, will be shown based on more elaborate derivations. First we can decompose the weighted mean, defined as

\[
\hat{w}_k S_k = \frac{u_k S_k \exp \left( -\frac{S_k}{\lambda_l} \right)}{\sum_{k'=1}^{P} u_{k'} \exp \left( -\frac{S_{k'}}{\lambda_l} \right)},
\]  
(4.50)

into the functions \( g(\lambda_l) \) and \( h(\lambda_l) \), where

\[
g(\lambda_l) = u_k S_k \exp \left( -\frac{S_k}{\lambda_l} \right),
\]  
(4.51)

with

\[
g'(\lambda_l) = \frac{u_k S_k^2 \exp \left( -\frac{S_k}{\lambda_l} \right)}{\lambda_l^2},
\]  
(4.52)

and

\[
h(\lambda_l) = \sum_{k'=1}^{P} u_{k'} \exp \left( -\frac{S_{k'}}{\lambda_l} \right),
\]  
(4.53)

with

\[
h'(\lambda_l) = \frac{1}{\lambda_l^2} \sum_{k'=1}^{P} u_{k'} S_{k'} \exp \left( -\frac{S_{k'}}{\lambda_l} \right).
\]  
(4.54)
Based on the quotient rule we know that
\[
\frac{\partial}{\partial \lambda_l} \hat{w}_k S_k = \frac{g'(\lambda_l) h(\lambda_l) - g(\lambda_l) h'(\lambda_l)}{[h(\lambda_l)]^2}.
\] (4.55)

Thus,
\[
\frac{\partial}{\partial \lambda_l} \sum_{k=1}^{P} \hat{w}_k S_k = \sum_{k=1}^{P} \frac{\partial}{\partial \lambda_l} \hat{w}_k S_k = \frac{1}{\lambda_l^2} \left\{ \sum \hat{w}_k S_k^2 - \left( \sum \hat{w}_k S_k \right)^2 \right\}.
\] (4.56)

When we plug this into (4.56) and (4.49) back into the equation of (4.47), we obtain
\[
\frac{\partial}{\partial \lambda_l} \lambda_l D_{KL} = \sum_{k=1}^{P} \hat{w}_k \log \left( \frac{\hat{w}_k}{u_k} \right) - \frac{1}{\lambda_l^2} \left\{ \sum \hat{w}_k S_k^2 - \left( \sum \hat{w}_k S_k \right)^2 \right\}.
\] (4.57)

Thus, given that \( u_{kl} \) is independent from \( \lambda_l \), the value for \( \lambda_l \) that maximizes
\[
\lambda_l \sum_{k=1}^{P} \hat{w}_k \log \left( \frac{\hat{w}_k}{u_k} \right)
\] (4.58)
is found when
\[
\lambda_l = \sqrt{\frac{\sum_{k=1}^{P} \hat{w}_{kl} S_{kl}^2 - \left( \sum_{k=1}^{P} \hat{w}_{kl} S_{kl} \right)^2}{\sum_{k=1}^{P} \hat{w}_{kl} \log \left( \frac{\hat{w}_{kl}}{u_{kl}} \right)}}.
\] (4.59)