Improved strategies for distance based clustering of objects on subsets of attributes in high-dimensional data

Kampert, M.M.D.

Chapter 7

General Discussion

The main objective of the COSA framework is to produce distances that can capture an underlying clustering structure from high-dimensional data, where each cluster can have its own important attributes. Since its first formulation by Friedman and Meulman (2004), this monograph is the first extensive study on the properties of COSA. In Chapter 1, the background information for COSA is described. Chapter 2 provides a recapitulation of the COSA-K Nearest Neighbors (COSA-KNN) algorithm. Chapter 3 gives an explanation why median-based attribute weights are more robust than mean-based attribute weights, and in the same chapter a strategy is presented for choosing the tuning parameter values in COSA-KNN of λ and K. In Chapter 4, we reformulate COSA in such a way that only the tuning parameter λ remains necessary, referred to as COSA-λNN. Moreover, we show that with a different initialization of the attribute weights, and with a COSA distance that better separates pairs of objects in different clusters, COSA can become more powerful. To derive L clusters from the distances obtained by either COSA-λNN, or COSA-KNN, we propose in Chapter 5 a partitioning algorithm, referred to as MVPIN. In a first examination of its effectiveness, MVPIN produces promising results in combination with COSA-KNN, and especially with COSA-λNN. We compared COSA with MVPIN to other state-of-the-art L clustering algorithms in Chapter 6. We showed that COSA-λNN, but also in combination with MVPIN, is a compelling option for the clustering of high-dimensional data.

7.1 Limitations

This monograph shows that COSA has good potential for real world application. However, the many compelling examples of applications of COSA in this monograph should be considered as demonstrations. Although all the examples do provide useful insights in the behavior of the original COSA algorithm and its improvements, we should address certain limitations in more detail. These are limitations concerning the simulation studies; the computational costs of COSA, theoretical and technical

details of COSA, and missing data.

7.1.1 The Simulation Examples

The variety of models that have been used to generate high-dimensional data for COSA has been small. The two typical models that have been used as starting points were presented in the Chapter 1 of this monograph: the COSA prototype model, and the COSA weak spot model. The motivation for these two generative models has been to support the general conclusion that COSA is especially powerful in identifying clusters that have their own subset of attributes with locally low dispersion as compared to the dispersion computed for these attributes over all the objects. Therefore, it is of importance that all attributes have a similar scale. Moreover, based on these two models, it is also shown that COSA is less strong in finding clusters that have a large within-cluster variance on their own subset of attribute, when compared to the between-cluster variance on these attributes.

A further limitation is the absence of a description of the 'breakdown' points for COSA. It would be interesting to have a well informed overview of the sensitivity of COSA to changes in each within-cluster attribute dispersion, or the cluster sizes, the number of irrelevant attributes, and the number of masking objects. Moreover, the breakdown points that result from such a sensitivity analysis, would be especially valuable to know about if they could indicate when to use COSA versus other algorithms. For example, both the simple approach to sparse clustering (SAS), and the fully improved version of COSA, perform equally well on data from the prototype model as well as the weak spot model. However, empirical evidence so far suggests that when the variance between the cluster attribute means becomes smaller in the weak spot model, SAS will outperform COSA. Similarly, when in the prototype model the number of overlapping attributes becomes larger, or, when the number of masking objects becomes larger, COSA will outperform SAS.

Apart from the two generative models used as starting point, it would have been interesting to see how COSA's performance would have been effected when other families of probability distributions (e.g., mixtures of gamma distributions), are used for generative models. In Steinley and Brusco (2008), a modified version of COSA still had a competitive performance on other normal-mixture model based clustering algorithms. However, these were results based in lower-dimensional data settings ($N > P$), and data in which the underlying clusters were not allowed to have their own unique subspace of attributes. Still, Steinley and Brusco (2008) presented a well-designed comprehensive Monte Carlo study to compare a number of clustering algorithms.

To our knowledge there is, as of yet, no comprehensive (Monte Carlo) study available where distance functions for high-dimensional data are compared. There are some comprehensive studies for distance functions such as France, Carroll, and Hiong (2012), Pekalska and Duin (2005), and Aggarwal (2001). None of these studies, however, provide a comparative study between distance functions that apply a weighting strategy to the attributes in high-dimensional data.

7.1.2 Computational Costs

In a prescription by Kriegel et al. (2016) it is stated that

> 'Any paper proposing a new algorithm should come with an evaluation of efficiency and scalability (particularly when we are designing methods for "big data")'.

So far we did not provide any information on the computational costs of COSA-KNN, COSA-λNN, and any of the other algorithms. In the same study by Kriegel et al. (2016), rules of thumb are given that show that the comparison of computational costs of the algorithms (or implementations) is far from trivial.

However, the computational complexity of the original COSA-KNN algorithm is easily determined. In each iteration a distance matrix is computed on N objects and P attributes which results in $P \times N(N-1)/2$ operations, then to obtain the $N \times P$ attribute weights, we need to find the K nearest neighbors for each object i, by sorting the distances for each object using a sort method. The worst case time complexity of the sort method is $\mathcal{O}(N\log(N))$. Then, for each iteration in COSA-KNN, the worst case computing time for COSA-KNN is

$$\mathcal{O}\left(PKN^2\log(N)\right). \tag{7.1}$$

The computational complexity of COSA-λNN is more difficult to formulate. COSA-λNN has an extra merge sorting algorithm for the attributes with complexity $\mathcal{O}(P\log(P))$. Moreover, instead of having neighborhoods each being of size K, COSA-λNN allows the sizes of the neighborhoods to be different for each object. We denote the size of each neighborhood by the function of λ, $N_i(\lambda)$, and is defined as

$$N_i(\lambda) = \left|\lambda NN(i)\right|, \tag{7.2}$$

the number of the λ driven nearest neighbors of object i, as defined in equation (4.29) from Chapter 4. Having described the parameters, the COSA-λNN worst case complexity is

$$\mathcal{O}\left(P\log(P)\left[\sum_{i=1}^{N} N_i(\lambda)\right]N\log(N)\right). \tag{7.3}$$

Whether COSA-λNN or COSA-KNN has higher computational costs is dependent on the noise and clustering structure in the data. When

$$K > \left(\log(P)\sum_{i=1}^{N} N_i(\lambda)\right)/N, \tag{7.4}$$

then COSA-KNN will have higher computational costs. This particular setting occurs in a situation when data set consists of a very large proportion of noise objects, each living in small neighborhoods. However, more important is that that the cost for optimizing the tuning parameters for COSA-λNN is (much) lower than for COSA-KNN, since the optimization is over a one-dimensional (versus a two-dimensional) grid of the tuning parameter values.

It may be helpful to report that the computing time (wall clock time) for COSA-λNN and COSA-KNN was fairly equal to each other on all the data examples we used in this monograph. Another remark is that the COSA algorithms were slower than all other algorithms we ran in this monograph. However, we remark that COSA gives ample opportunity for parallelization. Moreover, the implementations of the algorithms are based on a mixture of `Fortran`, `C++`, and `R` code, each having their own compiling perks and quirks. To stay in line with at least some of the recommendations in Kriegel et al. (2016): we did use realistic data sets (Chapter 6), and all code that has been used in this monograph is published online, see `https://www.tinyurl.com/MonographCOSA`.

7.1.3 Optima and Convergence

So far, the convergence properties of COSA have not been extensively discussed in this monograph. Neither proof, nor empirical support has been given to show that the COSA algorithms converge. Empirical evidence so far suggests that the solution for the attribute weights in COSA most likely converges. For some empirical data examples, for example the COSA weak spot model, we see that COSA ends in an oscillation between two solutions for $\mathbf{W}$. This 'back and forth' process between two solutions typically occurs when local minima for $Q(\mathbf{W} \mid \mathbf{C})$ and $\widetilde{Q}(\mathbf{C} \mid \mathbf{W})$ are equally attractive, but not compatible. When this occurs, we advise to use the solution for the attribute weights that has the minimum value for $Q(\mathbf{W} \mid \mathbf{C})$. This oscillating process between the two solutions can be referred to as a second order stationary process, and loosely speaking, could be seen as convergence.

For small data sets we could find the global minimum. Consider a data set of $N = 20$ objects, on which we wish to run COSA-KNN with $K = 10$. Then, it is still feasible to find the global minimum for the leading criterion $Q(W \mid C)$.Out of the

$$\binom{N}{K} = 184,756$$

neighborhoods of size K, there are 'only'

$$\binom{N-1}{K} = 92,378$$

neighborhoods for each object for which we need to find the minimum within-neighborhood sum of the distances ($D_{ij}[\mathbf{w}]$). For COSA-λNN a comparable, but feasible, strategy can be created to find the global minimum for $Q(W \mid C)$.

When investigating the convergence properties, the homotopy strategy in COSA also needs to be considered. As was described in Chapter 2, the COSA algorithms have an homotopy strategy implemented to avoid convergence to suboptimal local minima. Whereas in COSA-KNN the suboptimal local minima are avoided due to a linear path for the homotopy parameter η, the COSA-λNN only applies the homotopy strategy at the start of the first iteration and then iterates without a homotopy path, i.e. $\eta = \lambda$ at the start, but after the first itertation η is set equal to ∞. For the

examples in this monograph, however, we would not have obtained differences in the interpretation of the COSA-λNN results, compared to the situation where the linear path was used.

To show empirically that suboptimal local minima are avoided with the homotopy strategy, it could be informative to plot the values for each iteration of the within-neighborhood criterion $Q(\mathbf{W}|\mathbf{C})$, the criterion with the COSA distances $\widetilde{Q}(\mathbf{C}|\mathbf{W})$, and the iteration number. The comparison of a plot for the results of COSA with the (linear) homotopy strategy, on the one hand, and for the results where no homotopy strategy was used, on the other hand, could lead to insights for new homotopy strategies.

It may be easy to come up with a homotopy strategy that would improve the path with which suboptimal local minima are avoided. The homotopy parameter η regularizes the Kullback-Leibler divergence between each of the t_{ijk}'s and v_{ijk}'s, as given in the second term of the inverse exponential distance:

$$D_{ij}^{\eta}[\mathbf{W}] = \min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log \left(\frac{t_{ijk}}{v_{ijk}} \right) \right\} \tag{7.5}$$

(equation 2.47 from Chapter 2). An idea to improve the path over which the distance evolves is to assure that in iteration b with homotopy parameter η_b, the regularized Kullback-Leibler divergence is never larger than the divergence term for any of the object pairs in the next iteration, $b+1$. In this way it can be assured that the role of the solution for the attribute weights ($\mathbf{v}_{ij}$) becomes more important with each iteration. However, so far, such strategies result in considerably higher computational costs.

Another open avenue that could be explored is to avoid local minima by introducing fuzzy membership for the clusters or neighborhoods. Whereas in COSA the membership of object i to cluster (or neighborhood) l could only be true ($c_{il} = 1$) or false ($c_{il} = 0$), a fuzzy version of the criterion, where $0 \leq c_{il} \leq 1$ with

$$\sum_{l=1}^{L} c_{il} = 1, \tag{7.6}$$

could smooth away some of the local optima in the criterion. For a comparable strategy and its properties, see Heiser and Groenen (1997). Thus, apart from smoothing the distances between the objects with a homotopy strategy, the attribute weights could also be smoothed across the objects with the use of fuzzy cluster memberships.

7.1.4 Missing Data

A problem of practical importance is how well COSA deals with missing data. In Chapter 4 we did apply COSA to a data set that contained missing values, but how COSA copes with incomplete data has not been discussed. When either object i or object j have a missing value on attribute k, then the attribute weight v_{ijk} is modified

with the following rule:

$$v_{ijk} \leftarrow I(x_{ik} \neq \text{missing})I(x_{jk} \neq \text{missing})v_{ijk}. \tag{7.7}$$

For each object i' that has missing values on attributes, the attribute weights on the non-missing values are re-normalized as

$$w_{kl^*_{i'}} \leftarrow w_{kl^*_{i'}} \Big/ \sum_{k=1}^{P} I(x_{i'k} \neq \text{missing}) w_{kl^*_{i'}} \ . \tag{7.8}$$

Thus, for those object pairs where at least one of the objects has a missing value on attribute k, we set v_{ijk} equal to a value of zero, and the attribute weights for the non-missing attributes $w_{kl^*_{i'}}$ are renormalized to sum to 1 for each object i'. If for two objects there are no overlapping non-missing attribute values, then these two objects are assigned an infinite COSA distance.

When applying COSA to data with missing values, one should be aware of resulting effects from the above strategy. Suppose objects i and j have no missing attributes, have exactly the same clustering pattern, and let object i' be a version of object i where at least one of the attribute values x_{ijk} is missing. Then, with COSA's strategy on dealing with missing values we could have the following inequality property for the attribute weights:

$$\sum_{k=1}^{P} v_{i'jk} > \sum_{k=1}^{P} v_{ijk}. \tag{7.9}$$

Due to the re-normalization of the attribute weights, this inequality property (7.9) (7.9) would strongly hold when object i has missing values on the attributes that would have received attribute weights above average. The consequence of this property is that the COSA distance between object i' and j becomes larger than it should have been, risking that objects i' and j will not end up in the same cluster.

The reverse effect occurs when object i' has missing values on the attributes that would have received weight values below average, when the attributes are not important for the clustering. Then, due to the re-normalization of the attribute weights we obtain

$$\sum_{k=1}^{P} v_{i'jk} < \sum_{k=1}^{P} v_{ijk}. \tag{7.10}$$

This inequality property (7.10) results in a smaller COSA distance for objects i' and j, and is less problematic.

In Chapter 4 we applied COSA-KNN and COSA-λNN on a benchmark gene expression data set for breast cancer tumors (Perou, 2000) that contained missing values, and COSA seems to cope well. A very likely reason is that there were missing values on those attributes that were irrelevant for the clustering of the involved objects, while these objects did have values on the attributes important for clustering. For such a specific distribution of missing values, re-normalization of the attribute weights will not have a detrimental effect on the COSA distances.

The renormalization strategy in equation (7.8) can have harmful effects for object pairs where all attributes have equal importance. Let us create the following three scenarios. In the first scenario, scenario **i.**, we have a complete data set of $N = 100$ objects by $P = 1,000$ non-missing attributes that consists of noise values only (i.i.d $\sim N(0, 1)$). In the second and third scenario we have for 20 out of the $N = 100$ objects that either have **ii.** 200 missing values, completely at random, out of the $P = 1,000$ attributes for each object; or **iii.** 200 missing values on exactly the same attributes for each object. Figure 7.1 displays the typical average linkage dendrogams for our COSA-KNN distances.

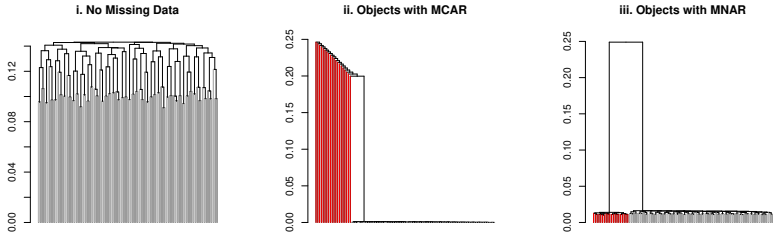

Figure 7.1: Results of COSA-KNN distances of 'noise only' data, displayed in average linkage dendrograms. The left dendrogram gives the results based on a data set with no missing values, the middle dendrogram is based on the data set with 20 objects (in red) that have missing values completely at random (MCAR) on 200 attributes (out of the 1,000 attributes), the right dendrogram is based on a data set where 20 objects (in red) have 200 missing not at random (MNAR) values on exactly the same attributes. The COSA distances are normalized to have a sum of squares equal to N.

In the three scenario's all objects should be approximately equidistant to each other. However, as we can see in Figure 7.1, for scenario's **ii.** and **iii.** the distances seem to be systematically different for the objects with missing values (colored red) and the objects that have no missing values (in black). In scenario **ii** we see that the objects with no missing values are somehow very similar to each other, and the objects with missing values are very distant to the other objects. Moreover, note that the objects with missing values have the largest distances between each other. In scenario **iii.** all objects that have the exact same MNAR pattern for the attribute values seem to be equidistant to each other. Similarly, all objects without missing attribute values also seem to be equidistant to each other. While the MCAR scenario in practice can easily be detected and used as an advice to discard the objects that have missings, the MNAR may lead to misleading clustering conclusions. However, also the MNAR disturbance can be easily detected when re-running COSA on the attributes that do not contain any missing values. Apart from the need of being aware of such results, it may be of interest to further study the behavior of COSA to recognize results that are systematically influenced by missing values. Still, it is a strength that COSA can cope with missing values. Especially since the typical K-means, or fuzzy C-means,

algorithms are not able to cope with missing values at all.

7.2 Future Avenues

In the previous section we stipulated how some limitations of this monograph could be further studied as research problems in future investigations. In this section we give a small overview on topics that we did not deal with in this monograph, but are still deemed noteworthy in relation to the COSA framework. Some of these topics have already been proposed or studied in Friedman and Meulman (2004), Kampert, Meulman and Friedman (2017). Others are open avenues for further study that have not (yet) received any attention at all.

7.2.1 Different regularization strategies for the attributes

While the COSA algorithms uses Kullback-Leibler divergence regularization, we conjecture that a family of COSA algorithms with closed-form solutions for the attribute weights can be created from other classes of divergences as well. A canonical example of a regularization based on the Bregman divergence (Bregman, 1967) is COSA criterion where, instead of the Kullback-Leibler divergence, the Squared Euclidean distance between the attribute weights in $\mathbf{W}$ and the initial attribute weights $\{\mathbf{u}_l\}_{l=1}^{L}$ is regularized. Especially in the light of the new COSA-λNN algorithm, where the number of zero-value attribute weights is also steered by λ, these different regularization forms could be interesting directions for future research.

Another interesting direction is to simplify the COSA framework. Instead of attribute weights, a hard crisp subset of equally weighted attributes may lead to better and simpler results. The number of attributes that are selected for each neighborhood could be selected based on the largest gap between the sum of the within-neighborhood attribute dispersions and a reference sum of attribute dispersion from random neighborhoods. Note that this strategy could also be seen as a modification of SAS towards a COSA approach, resulting in different COSA distances.

7.2.2 COSA Distances

We have defined a new COSA distance in Chapter 4 to create a stronger separation between objects from different clusters. We have seen that v_{ijk} could be defined differently from being the maximum of $w_{kl_i^*}$ and $w_{kl_j^*}$. So far, we have only one restriction for any definition of a COSA distance, i.e. the COSA distance should reduce to a COSA within-cluster distance when $v_{ijk} = w_{kl_i^*} = w_{kl_j^*}$ for all atributes k, expressed as

$$D[\mathbf{W}] = D[\mathbf{w}_i] = D[\mathbf{w}_j]. \tag{7.11}$$

Thus, we could study new COSA distances with the purpose to incorporate between-cluster distances based on the centroids of clusters, i.e. the within-cluster average value for each attribute. Let $\overline{d}_{kl_{ij}}$ be the distance between the average value on attribute k

for cluster l_i, on the one hand, and the average value on attribute k for cluster l_j, on the other hand. Then, we could consider a following definition for v_{ijk}:

$$v_{ijk} = \frac{w_{kl_i^*} + w_{kl_j^*}}{2} + \left(\frac{|w_{kl_i^*} - w_{kl_j^*}|}{1 - |w_{kl_i^*} - w_{kl_j^*}|} \right) \left(\frac{1 + \overline{d}_{kl_{ij}}}{\eta} \right). \tag{7.12}$$

Here, η is the homotopy parameter, and with this definition of v_{ijk}, we obtain one of the many possible definitions in the family of COSA distances. Note that when $w_{kl_i^*} = w_{kl_j^*}$ holds for all k, then we also have $v_{ijk} = w_{kl_i^*} = w_{kl_j^*}$ for each k.

So far, we have shown examples that merely indicate that there is a wide open world to explore for COSA distances. Another direction could be to create proper COSA distances that have a perfect fit with an MDS configuration that is constrained on finding clusters, or even ultrametric COSA distances from which a dendrogram could be formed directly.

7.2.3 Targeting and the Attribute Distances[1]

Not only the COSA distances, but also the attribute distances have potential for further research. In this monograph the COSA clustering could be on any possible joint values on subsets of attributes. However, in Friedman and Meulman (2004) it was also possible to look for clusters that group on particular values. This was referred to as 'targeted clustering' and can actually also be seen as a first step that may bring us closer towards distances based on composite kernel spaces (Wange et al., 2016). Examples of the usefulness of targeting can be found in Friedman and Meulman (2004), as well as in Kampert, Meulman and Friedman (2017); here we just give a short explanation on what we mean by targeted attribute distances.

Suppose that objects only cluster on particular values, say y_k, which are possibly different for each attribute k. Here, the $\{y_k\}$ are chosen to be of special interest; and can be used to reduce the search space of the solutions for the clustering structure; when chosen correctly, these targets render it more likely to recover clusters. Examples are groups of consumers (objects) that spend relatively large amounts on products (attributes), while we wish to ignore consumers who spend relatively small or average amounts (or the other way around). If we focus on one particular value, we call this single targeting. We modify the original distance between objects i and j on attribute k, $d_{ijk} = d(x_{ik}, x_{jk})$, into targeted distances, and require objects i and j to be close to each other *and* to the particular target.

The so-called single target distance is defined as:

$$d_{ijk}(t_k) = \max[d_k(x_{ik}, y_k), d_k(x_{jk}, y_k)], \tag{7.13}$$

where y_k is the target value, e.g., a *high* or *low* or even *average* value. This distance is small only if both objects i and j are close to the target value y_k on attribute k. In addition to single targeting, we can also focus on two different targets, e.g. being naturally either high or low values. An example is in microarray data, where we could

[1] A large part of this subsection is from Kampert, Meulman, and Friedman (2017).

search for clusters of samples with either high or low (but not moderate) expression levels on subsets of genes (attributes). In dual targeting, we define two targets y_{1k} and y_{2k}, and we use the dual target distance

$$d_{ijk}(y_{1k}, y_{2k}) = \min[d_{ijk}(y_{1k}), d_{ijk}(y_{2k})] \tag{7.14}$$

on selected attributes x_k, where $d_{ijk}(\cdot)$ is the corresponding single target distance (7.13). This dual target distance is small whenever x_{ik} and x_{jk} are either both close to y_{1k} *or* both close to y_{2k}. Thus, in gene expression and consumer spending examples, one might set y_{1k} and y_{2k} to values near the maximum and minimum data values of the attributes, respectively, and we will cause COSA to seek clusters based on extreme attribute values, ignoring clusters with (uninteresting) moderate attribute values.

7.2.4 Mixed Types of Attributes

In this monograph we only used 'numeric' attributes. Although, no specific advice was given on the choice for the specific attribute distance functions, COSA was originally designed for mixed type of attributes e.g. numerical and categorical (Friedman & Meulman, 2004), and this is (still) a desired and ongoing topic of research, e.g. see Grané and Romera (2016), Van de Velden et al. (2018); for an overview see Foss et al. (2018).

7.2.5 Different Objectives for COSA

Future avenues for different objectives with COSA may also be considered when relating COSA to techniques as Points of View Analysis (PVA; Tucker & Messick, 1963, Meulman & Verboon, 1993) and the Self-Organizing Map (SOM; Kohonen, 1980). In this last subsection of the monograph we will restate aspects of COSA such that it becomes relatable to techniques as PVA and SOM. We conjecture that these relatable techniques provide useful insights regarding the properties and further improvement of the COSA approach.

Points of View Analysis

When we consider the neighborhood of an object i to be a cluster l_i, then the COSA within-cluster distance is based the attribute weights of the neighborhood of object i, and is defined as

$$D[\mathbf{w}_{l_i}]_{ij} = \sum_{k=1}^{P} w_{kl_i} d_{ijk}, \tag{7.15}$$

which is a proper metric distance when each attribute distance also satisfies the metric properties. Here, we argue that this specifc COSA within-cluster distance could also be interpreted as a distance from the viewpoint of object i.

Similarly, $D[\mathbf{w}_{l_j}]_{ji}$ can be interpreted as a distance from the viewpoint of object j, and, when $\mathbf{w}_{l_i} \neq= \mathbf{w}_{l_j}$, it is most likely that

$$D[\mathbf{w}_{l_i}]_{ij} \neq D[\mathbf{w}_{l_j}]_{ji}, \tag{7.16}$$

meaning that the viewpoint of the distances from object j is different than that from object i. Suppose the objective would be to find for each of the N possible COSA within-cluster distance matrices a MDS representation for the objects, then we have for each object a visualization of each object's viewpoint. Similarly, viewpoints could be formed, based on the attribute weights from equation (6.17) in Chapter 6, for each COSA-validated cluster. Since each cluster is based on its own unique partitioning of the attributes, these viewpoints are closely related and linked to the objective in the so-called Point of View Analysis (Tucker & Messick, 1963, Meulman & Verboon, 1993).

Self-Organizing Maps

COSA is also relatable to Kohonen's Self-Organizing Maps (SOM), an artifical neural network technique based on competitive learning with which the data are non linearly projected onto a lower-dimensional display (Kohonen, 2001). Consider each object i_l from neighborhood l_i as a neuron that receives input information from the neighboring objects. Comparable with SOM, we find that the neighboring neurons for i_l will gradually specialize to represent similar inputs. In other words, when objects i and j live in closely the same neighborhoods, the attribute weights $\mathbf{w}_{l_j}$ and $\mathbf{w}_{l_j}$, for neurons l_i and l_j, respectively, will become more similar over time, i.e., during the iterations in the COSA algorithm. Although the attribute weights and the objects for each neighborhood become updated with each iteration, in COSA the attribute values of the neuron remain the same, while in SOM the attribute values of the neuron are 'smoothed' based on the nearest neighbor objects, e.g.,

$$x_{l_i k} = \sum_{j}^{N} c_{jl_i} x_{jk} \Big/ \sum_{j}^{N} c_{jl_i} \ . \tag{7.17}$$

Information Retrieval

When relating COSA to a special case of SOM, as was done in the previous section, the possibilities of COSA and information retrieval may also become apparent. E.g., an estimate for the missing value x_{ik} could be retrieved based on equation (7.17), as long as the nearest neighbors do not have missing values on attribute k. In Gabrielsson and Gabrielson (2008) and Purbey et al. (2014), SOM is being used for the purpose of information retrieval in recommender systems.

When we know the behavior of COSA in the presence of missing data, in greater detail, the COSA approach could contribute to research involved with information retrieval. Suppose that object i does not have a value on attribute k, but each of its nearest neighbors ($c_{jl_i} = 1$), has a non-missing value on the specific attribute. Then,

we may be able to compute the attribute weights w_{kl_i} when the attribute dispersion is computed as

$$S_{kl_i} = \frac{1}{N_{l_i}^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} c_{jl_i} c_{j'l_i} d_{jj'k}. \tag{7.18}$$

Here, the definition of the attribute dispersion S_{kl_i} does not involve object i, itself. Thus, even though we do not know the value of object i on attribute k, we can compute to what extent attribute k is important for the clustering of object i, which is valuable information for e.g., recommender systems and imputation strategies for missing data.