Discovering Non-Redundant K-means Clusterings in Optimal Subspaces

Dominik Mautz[†], Wei Ye[†], Claudia Plant[§], Christian Böhm[†]
[†]MCML, Ludwig-Maximilians-Universität München, Munich, Germany
{mautz,ye,boehm}@dbs.ifi.lmu.de
[§]ds:UniVie, University of Vienna, Vienna, Austria
claudia.plant@univie.ac.at

ABSTRACT

A huge object collection in high-dimensional space can often be clustered in more than one way, for instance, objects could be clustered by their shape or alternatively by their color. Each grouping represents a different view of the data set. The new research field of *non-redundant clustering* addresses this class of problems. In this paper, we follow the approach that different, non-redundant k-means-like clusterings may exist in different, arbitrarily oriented subspaces of the high-dimensional space. We assume that these subspaces (and optionally a further *noise space* without any cluster structure) are orthogonal to each other. This assumption enables a particularly rigorous mathematical treatment of the non-redundant clustering problem and thus a particularly efficient algorithm, which we call Nr-Kmeans (for non-redundant k-means). The superiority of our algorithm is demonstrated both theoretically, as well as in extensive experiments.

KEYWORDS

clustering; k-means; subspace; non-redundant

ACM Reference Format:
Dominik Mautz[†], Wei Ye[†], Claudia Plant[§], Christian Böhm[†]. 2018. Discovering Non-Redundant K-means Clusterings in Optimal Subspaces. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219945

1 INTRODUCTION

Clustering or finding a natural grouping of a large set of objects is deeply rooted in human cognition. Our brain constantly clusters sensory stimuli in order to recognize, monitor and interpret them. Experiments from cognitive psychology demonstrate that already children of one year of age are able to reliably discover the clusters in a set of objects [10]. They do not have any problem with the task indicated in Figure 1. Given pictures of objects—from the *Amsterdam Library of Object Images* (ALOI)—taken from different viewing angles and illumination temperatures, we intuitively cluster together the red, the green, the round and the cylindrical objects, respectively.

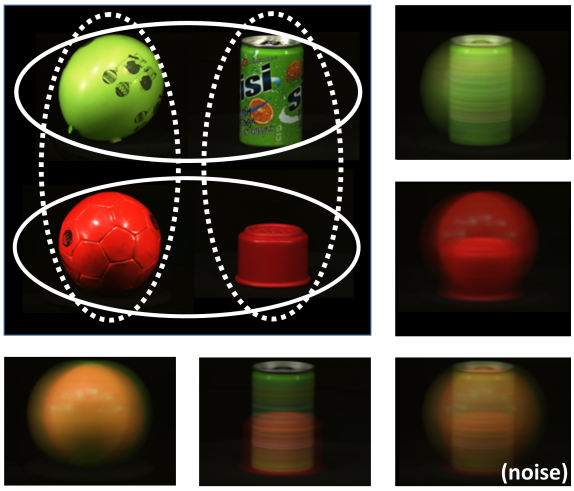


Figure 1: Multiple clustering possibilities of objects according to color and shape.

It is a simple task for toddlers [10], yet it pushes state-of-the-art clustering and dimensionality reduction techniques to their limits for the following two reasons. (1) The images are represented by high-dimensional feature vectors consisting of 611 features in total. Classical clustering algorithms like k-means tend to fail due to the curse of dimensionality. The sparse high-dimensional feature space does not contain any clustering structure. (2) There are two equally meaningful ways to group the objects in Figure 1. Mathematically, there are two different low-dimensional subspaces, each exhibiting an interesting clustering structure. The clusterings in the subspaces are mutually non-redundant, i.e. each object belongs to different clusters in different subspaces.

Motivated by Challenge (1)—the curse of dimensionality—many sophisticated algorithms that integrate feature selection or dimensionality reduction, have been proposed. For a survey see e.g. [22]. These techniques aim at identifying for each cluster the subspace, in which it is best represented in. An individual subspace for each cluster helps against the curse of dimensionality, but makes the interpretation of the result more difficult. There is no common subspace in which all the clusters can be visualized. Therefore, it is

KDD 2018, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219945

difficult to say which clusters are, for instance, most similar to each other or whether outliers or a cluster hierarchy exist. Motivated by this drawback, [17] recently introduced a technique finding the best subspace for a single k-means clustering. Some other recent approaches focus on Challenge (2), i.e. on identifying multiple non-redundant clusters in different subspaces, e.g. [6, 20, 25]. Allowing clusters to exist in arbitrary subspaces and allowing objects to be assigned to different clusters in different subspaces establishes an even larger solution space. Existing approaches to non-redundant subspace clustering suffer from one or more of the following drawbacks: The algorithms require many input parameters, cause massive runtime and/or produce massive amounts of results which are difficult to interpret, see Sections 3 and 4.

We investigate the challenge of discovering multiple interesting k-means clusterings in different subspaces. The basic idea is to find multiple mutually orthogonal subspaces, such that the objective function of classical k-means is optimized in all of them. In addition, our technique introduces a noise subspace—orthogonal to the other subspaces—where the data distribution is assumed to be unimodal. Figure 1 displays the result of Nr-Kmeans (for Non-redundant K-means). Our algorithm discovers relevant subspaces and the corresponding clusterings and outperforms comparison methods on this and other tasks, see Section 3. To summarize our contributions are as follows:

- **Multiple interesting k-means Clusterings in Optimal Subspaces.** Nr-Kmeans discovers multiple, non-redundant k-means clusterings in orthogonal subspaces. The approach finds for each clustering the subspace that optimizes the cluster separation according to the objective function of k-means. For each of these clustered subspaces, Nr-Kmeans constructs the most relevant basis vectors. The orthogonality between subspaces ensures that the discovered clusterings represent different views on the data providing mutually non-redundant information. The subspaces are suitable for visualization and further analysis, as they reveal the relationships between the individual clusters of a clustering. Inheriting from k-means, the result of Nr-Kmeans includes interpretable cluster centers, as displayed in Figure 1.
- **Efficiency.** The proposed optimization algorithm, Nr-Kmeans, is easy to implement and compatible with many proposed extensions of k-means. It is fast, even without sophisticated performance optimizations.
- **Lightweight Parameterization.** The sole input parameter of Nr-Kmeans is the number of clusters for each subspace. The dimensionality of each subspace is determined automatically.
- **Noise Handling.** In contrast to existing approaches to non-redundant subspace clustering, the Nr-Kmeans model includes the idea of a noise subspace. The noise subspace captures all the unimodal variance in the data, which is not interesting for clustering. This property helps Nr-Kmeans to outperform existing methods, especially on high-dimensional data.

2 NON-REDUNDANT K-MEANS

In this section, we describe our proposed method Nr-Kmeans. An implementation of Nr-Kmeans and supplementary material is available on our website.[1]

[1] http://dmm.dbs.ifi.lmu.de/downloads

Symbol	Interpretation
$d \in \mathbb{N}$	Dimensionality of original space
$S \in \mathbb{N}$	Number of subspaces
$k_j \in \mathbb{N}$	Number of Clusters in the j'th subspace
$m_j \in \mathbb{N}$	Dimensionality of the j'th subspace
$\mathcal{D} \subseteq \mathbb{R}^d$	Set of all objects
$C_{j,i}$	Objects of cluster i in subspace j
$\mathbf{x} \in \mathcal{D}$	A data point or object of the dataset
$\boldsymbol{\mu}_{j,i} \in \mathbb{R}^m$	Original space mean of cluster i in subspace j
$P_j \in \mathbb{R}^{d \times m_j}$	Projection onto the j'th subspace
$V \in \mathbb{R}^{d \times d}$	Orthogonal matrix of a rigid transformation
$\Sigma_j \in \mathbb{R}^{d \times d}$	Sum of scatter matrices of clustering j—Eq. 4
$\mathbf{I}_l$	$l \times l$ identity matrix
$\mathbf{0}_{l,r}$	$l \times r$ zero matrix

Table 1: Symbols and Definitions

We describe our algorithm as a simple extension of the well-known Lloyd's algorithm, with its alternating assignment and update steps. Yet, it is possible to extend Nr-Kmeans with many other proposed k-means extensions in a straightforward manner, e.g. exploit the triangle inequality [13] to speed up the assignment step, initialize cluster centers within the subspaces using k-means++ [2] or account for outliers with k-means-- [5].

Table 1 shows essential symbols and definitions used in the following.

2.1 Cost Function

In the classic version of the k-means algorithm, we want to find a set of k clusters C_i such that the sum of the squared Euclidean distances is minimized. We extend this basic idea in Nr-Kmeans by the assumption that the dataset can be partitioned in S different ways. Each clustering j contains k_j clusters and resides in an arbitrarily oriented subspace that is orthogonal to the subspaces assigned the other $S-1$ clusterings.

Further, we assume that there exists an orthogonal transformation matrix V, which rotates (and reflects) the data space such that the subspaces are all axis-parallel in the transformed space. Further, we can use masking matrices P_j to project the data onto the respective axis-parallel subspace. Since the subspaces do not overlap, each dimension of the rotated data space is exclusively mapped onto a single subspace. A data point $\mathbf{x}$ can then be projected onto the j'th subspace by $P_j^\mathsf{T} V^\mathsf{T} \mathbf{x}$. Combining these assumptions, we get the following cost function, which we want to minimize:

$$\mathcal{F} = \sum_{j=1}^{S} \sum_{i=1}^{k_j} \sum_{\mathbf{x} \in C_{j,i}} \left\| P_j^\mathsf{T} V^\mathsf{T} \mathbf{x} - P_j^\mathsf{T} V^\mathsf{T} \boldsymbol{\mu}_{j,i} \right\|^2 \quad (1)$$

By optimizing this objective function, we assign each linear combination of features of the original space to the subspace that optimally represents the containing structural information through the

respective clustering. Therefore, we can find the optimal subspace for each of the S k-means-clustering partitions.

The masking matrices $P_j \in \mathbb{R}^{m_j \times d}$ can then be build by setting the entry $P_j[a,b]$ to 1 if dimension a of the rotated data space should be mapped to the subspace dimension b. Otherwise, we set the entry to 0. For clarity and ease of explanation, we assume that the subspace dimensions may not necessarily be adjacent within the feature vector of the rotated space. Yet, if really needed, one can align the subspaces in the rotated space using a permutation matrix.

We justify this cost function and our claim that our algorithm finds non-redundant clusterings from a theoretical perspective by its relationship to Gaussian mixture models with statistically independent subspaces. It is easy to show that the standard k-means cost function is a limit to a Gaussian mixture model with the probability distribution $p(\mathbf{x}) = \Pi_i^k \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \sigma\mathbf{I})$, where one lets σ go to zero [16].

Following the same approach, one can readily show that the cost function for Nr-Kmeans can be viewed as the limit of the log-likelihood function (w.r.t. $\sigma \to 0$) of the following mixture model with S statistically independent subspaces:

$$p(\mathbf{x}) = \prod_j^S \sum_i^{k_j} \pi_{j,i} \mathcal{N}(P_j^\mathsf{T} V^\mathsf{T} \mathbf{x} | P_j^\mathsf{T} V^\mathsf{T} \boldsymbol{\mu}_{j,i}, \sigma\mathbf{I}).$$

The statistically independent components of this mixture model become in the limit the orthogonal, non-redundant subspaces of the Nr-Kmeans model.

2.2 Optimization Algorithm

We optimize this objective function with a modified version of Lloyd's algorithm—shown in Algorithm 1. It applies update and assignment steps described below until convergence.

As a first step, we have to initialize V with a (random) orthogonal matrix. Further, we need the initial dimensionalities m_j of each subspace such that $m_1 + \cdots + m_S = d$. For simplicity, we distribute the dimensions in our implementation equally among the subspaces. The optimal values for each subspace are subsequently found during the optimization. Last but not least, the initial cluster centers could, for example, be randomly picked or set using k-means++.

The **assignment step** is almost equivalent to the classic k-means algorithm. We keep the parameters V, m_j, and $\boldsymbol{\mu}_{j,i}$ fixed and assign each data point to the cluster for which the squared distance to the mean in the respective subspace is minimized: $\left\| P_j^\mathsf{T} V^\mathsf{T} \mathbf{x} - P_j^\mathsf{T} V^\mathsf{T} \boldsymbol{\mu}_i \right\|^2$.

For the **update step**, we keep the data point assignments fixed and determine the optimal parameter values. The following sections discuss in detail the optimization steps needed and the correctness of each step.

2.2.1 Estimation of the cluster centers $\boldsymbol{\mu}_{j,i}$. We can determine the estimator for the cluster centers by setting the partial derivative with respect to $\boldsymbol{\mu}_{j,i}$ equal to zero. The result of this calculation shows what one would intuitively expect: the cluster mean $P_j^\mathsf{T} V^\mathsf{T} \boldsymbol{\mu}_{j,i}$ in the subspace is simply the transformed mean value of all data points assigned to this cluster in the original space. It can be calculated by the well-known formula: $\boldsymbol{\mu}_{j,i} = \frac{1}{|C_{j,i}|} \sum_{\mathbf{x} \in C_{j,i}} \mathbf{x}$.

Algorithm 1: Nr-Kmeans

Input: dataset $\mathcal{D}$; nr of clusters in each subspace $k_1, \ldots, k_S$
Output:
  Clusters for each subspace $\{C_{1,1}, \ldots, C_{S,k_S}\}$;
  Rotation matrix V; Projections: $P_1, \ldots, P_S$
  Dimensionalities: $m_1, \ldots, m_S$
// Initialization:
$V \leftarrow$ random orthogonal matrix
$\forall j \in [1, S]$:
  $m_j \leftarrow$ some initial value, e.g. $\frac{d}{S}$
  $P_j \leftarrow$ exclusive mapping to m_j features
  $\forall i \in [1, k_j] : \boldsymbol{\mu}_{j,i} \leftarrow$ random data point of $\mathcal{D}$
// Optimization:
repeat
  // Assignment step
  $\forall j \in [1, S]$:
    $\forall$ clusters i: $C_{j,i} \leftarrow \emptyset$
    $\forall \mathbf{x} \in \mathcal{D}$:
      $i \leftarrow \arg\min_{i \in [1,k_j]} \left\| P_j^\mathsf{T} V^\mathsf{T} \mathbf{x} - P_j^\mathsf{T} V^\mathsf{T} \boldsymbol{\mu}_{j,i} \right\|^2$
      $C_{j,i} \leftarrow C_{j,i} \cup \{\mathbf{x}\}$
  // Update step
  $\forall j \in [1, S], i \in [1, k_j]$:
    Update $\boldsymbol{\mu}_{j,i}$ and Σ_j
  // Updating V---see Section 2.2.3
  foreach *pair* (s,t) *of subspaces* **do**
    create combined projection $P_{s,t}$
    $V_{s,t}^{\langle c \rangle}, \mathcal{E} \leftarrow \text{eig}\left(P_{s,t}^\mathsf{T} V^\mathsf{T} (\Sigma_s - \Sigma_t) V P_{s,t} \right)$
    $m_s \leftarrow |\{e | e \in \mathcal{E} \wedge e < 0\}|$
    $m_t \leftarrow |\mathcal{E}| - m_s$
    $V \leftarrow V \times \text{toFull}(V_{s,t}^{\langle c \rangle})$ // Eq. 5
    Update P_s and P_t
  end
until *convergence*;

2.2.2 Estimation of V and m_j in the case of two subspaces $S = 2$. Next, we need to estimate the optimal value of V. Before we show our proposed strategy to optimize V in the general case, we first need to consider the special case of two subspaces $S = 2$. These results will subsequently help us to optimize the general case $S > 2$.

The case for $S = 2$ shares some similarities with the optimization strategy proposed for [17]. Let us assume for now that m_1 and m_2 are fixed but arbitrary positive integers, such that $d = m_1 + m_2$. Further, let us assume that the projection matrix P_1 projects the first m_1 features of a data vector to the first subspace and P_2 the m_2 latter features to the second subspace:

$$P_1 = \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{0}_{m_1,m_2} \end{bmatrix}^\mathsf{T} \quad \text{and} \quad P_2 = \begin{bmatrix} \mathbf{0}_{m_2,m_1} & \mathbf{I}_{m_2} \end{bmatrix}^\mathsf{T},$$

where $\mathbf{I}_i$ is the i-dimensional identity matrix and $\mathbf{0}_{i,j}$ is the $i \times j$ zero matrix.

The cost function for this special case is then:

$$\mathcal{F} = \left[\sum_{i=1}^{k_1}\sum_{x\in C_{1,i}}\left\|P_1^\mathsf{T}V^\mathsf{T}\mathbf{x} - P_1^\mathsf{T}V^\mathsf{T}\boldsymbol{\mu}_{1,i}\right\|^2\right] + \left[\sum_{i=1}^{k_2}\sum_{x\in C_{2,i}}\left\|P_2^\mathsf{T}V^\mathsf{T}\mathbf{x} - P_2^\mathsf{T}V^\mathsf{T}\boldsymbol{\mu}_{2,i}\right\|^2\right] \tag{2}$$

We can transform this cost function into a trace minimization problem, by utilizing that a scalar is a 1×1-matrix and exploiting the fact that $\mathrm{Tr}(P_2P_2^\mathsf{T}A) = \mathrm{Tr}(A) - \mathrm{Tr}(P_1P_1^\mathsf{T}A)$, for $A\in\mathbb{R}^{d\times d}$.

The transformed objective function then is:

$$\mathcal{F} = \mathrm{Tr}\left(P_1P_1^\mathsf{T}V^\mathsf{T}\left[\Sigma_1-\Sigma_2\right]V\right) + \underbrace{\mathrm{Tr}\left(V^\mathsf{T}\Sigma_2V\right)}_{\text{const. w.r.t. } V}, \tag{3}$$

where Σ_j is the sum of all the clusters' scatter matrices of subspace j:

$$\Sigma_j := \sum_{i=1}^{k_j}\sum_{\mathbf{x}\in C_{j,i}}\left(\mathbf{x}-\boldsymbol{\mu}_{j,i}\right)\left(\mathbf{x}-\boldsymbol{\mu}_{j,i}\right)^\mathsf{T}. \tag{4}$$

The second term of the sum in Eq. 3 is constant for any V, because of the cyclic properties of the trace function and because V is defined as an orthogonal matrix and therefore $\mathrm{Tr}\left(V^\mathsf{T}\Sigma_2V\right) = \mathrm{Tr}\left(\Sigma_2\right)$.

Next, we should note that $P_1P_1^\mathsf{T}$ is a diagonal matrix, where the first m_1 diagonal entries are ones and the remaining m_2 diagonal entries are zeros. Thus, if we multiply it from the right with $V^\mathsf{T}\left[\Sigma_1-\Sigma_2\right]V$, it leaves the upper left $m_1\times m_1$ entries of $V^\mathsf{T}\left[\Sigma_1-\Sigma_2\right]V$ untouched and sets all other entries to zero. Further, we should note that the trace function yields the sum of the eigenvalues. Thus, it is possible to minimize the function (2), for fixed but arbitrary m_1 and m_2, by putting the eigenvectors of $\left[\Sigma_1-\Sigma_2\right]$ into V's columns such that the m_1-eigenvectors corresponding to the m_1-smallest eigenvalues project the data onto the first m_1 dimensions—the first subspace—and the remaining $m_2 = (d-m_1)$ eigenvectors project the data onto the second subspace. Therefore, we perform an eigenvalue decomposition of $\left[\Sigma_1-\Sigma_2\right]$ and use the eigenvectors—sorted in ascending order to their corresponding eigenvalue—as columns in V. We should note that $\left[\Sigma_1-\Sigma_2\right]$ is symmetric and therefore is orthogonal diagonalizable and all its eigenvalues are real.

Since we sort the eigenvectors in V w.r.t to the corresponding eigenvalues in ascending order and the constant term in Eq. 3 does not depend on m_1 or m_2, the optimal value of V is independent of the actual dimensionality of each subspace. This property gives us the added ability to optimize the costs w.r.t. to m_1 and m_2 within each update step. The cost function in Eq. 2 depends on m_1 and m_2 only through the projections P_1 and P_2. Because the trace is the sum of all eigenvalues and we want to minimize this sum, we can only minimize it, if we sum up all negative eigenvalues of $\left[\Sigma_1-\Sigma_2\right]$. Thus, we assign eigenvectors corresponding to a negative eigenvalue to the first subspace and eigenvectors corresponding to eigenvectors greater than zero to the second subspace. If the eigenvalue is zero we are indifferent with respect to the cost function and we can put those features in either subspace.

Consequently, we can optimize the cost function for a given V by setting m_1 to the number of negative eigenvalues of $\left[\Sigma_1-\Sigma_2\right]$ and $m_2 = d - m_1$.

We would like to note that the eigenvalues of $\left[\Sigma_1-\Sigma_2\right]$ also give rise to a notion of how important the corresponding eigenvector—and therefore the resulting feature—is for the clustering structure. This can easily be seen when we consider the case were we fix $m_1 = 1$: we can only minimize the cost function by using the eigenvector corresponding to the smallest eigenvalue as the vector projecting onto subspace one. Therefore, this eigenvector is the *most important* one for this clustering. For $m_1 = 2$ we also need the eigenvector with the second smallest eigenvalue and so forth. The same argument holds for the second clustering and the biggest eigenvalues. Thus, sorting the eigenvector columns in V in ascending order by their eigenvalue means that we also sort them by importance—in descending order for the first subspace and in ascending order for the second subspace.

2.2.3 Estimation of V and m_j in the general case $S > 2$. The problem of optimizing V in the general case $S > 2$ is that it does not have a closed form solution like the special case $S = 2$. That is we cannot optimize its value directly in a single eigen-decomposition step. However, we can use the fact that all subspaces are pairwise-orthogonal to each other and rotating the combined space of two subspaces does not affect the costs for the complementing subspaces. Therefore, the trick to optimize the general case is to consider each possible combination of two clusterings in the *rotated space* as the special case $S = 2$ discussed in the last section. The explanation follows the diagram in Figure 2. A complementing hands-on example can be found in the supplementary.

Let us assume that we already have updated all $\boldsymbol{\mu}_{j,i}$, but that the current V and m_j's are not yet optimal w.r.t. already fixed parameters. We consider two clusterings s and t and their corresponding subspaces S_s and S_t in the rotated space (in $i.$: $S_s \mathrel{\widehat{=}} \blacksquare$ and $S_t \mathrel{\widehat{=}} \blacksquare$). We can project the whole dataset in the *rotated space* onto a combined subspace $S_{s,t}$ using a projection matrix $P_{s,t} = [P_s \;\; P_t]$ that maps the dimensions assigned to S_s to the first m_s dimensions of $S_{s,t}$ and the dimensions assigned to S_t onto the latter m_t dimensions ($ii.$). Then we can treat the clusterings s and t and their combined subspace $S_{s,t}$ as the special case $S = 2$, with the initial rotation matrix of this subspace being $V_{s,t}^{\langle c\rangle} = \mathbf{I}_{m_s+m_t}$. Yet, we assume that this rotation is not optimal with respect to the cost function in Eq 2 and the optimal subspaces for s and t may be arbitrarily oriented in $S_{s,t}$ (in $iii.$—optimal: $S_s \mathrel{\widehat{=}} \blacksquare$ and $S_t \mathrel{\widehat{=}} \blacksquare$). But the results discussed in the last section, enable us to find a better value for the rotation matrix $V_{s,t}^{\langle c\rangle}$ and, in addition, allow us to adjust the dimensionality of the two subspaces such that the costs are minimized ($iv.$). Then, we can describe the rotation $V_{s,t}^{\langle c\rangle}$ we found in $S_{s,t}$ as a rotation in the full-space $V_{s,t}^{\langle f\rangle}$:

$$V_{s,t}^{\langle f\rangle}[a,b] = \begin{cases} V_{s,t}^{\langle c\rangle}[n,m], & \text{if } P_{s,t} \text{ maps } a \text{ to } n \text{ and } b \text{ to } m \\ 1, & \text{if } a = b \text{ and not the first case} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The full-space transformation matrix is then updated using $V_{s,t}^{\langle f\rangle}$: $V \leftarrow V \times V_{s,t}^{\langle f\rangle}$ ($v.$). In addition, we have to update P_s and P_t, since

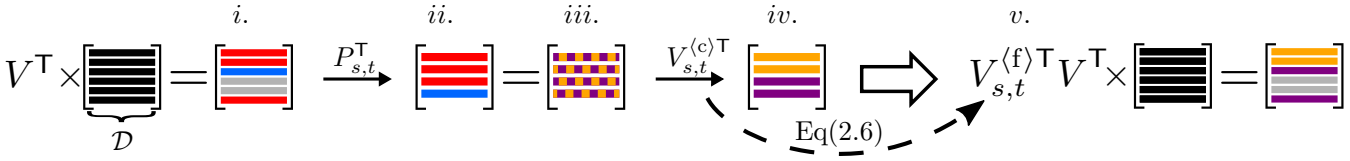


Figure 2: The diagram shows how we can update V w.r.t. two clusterings s and t and their corresponding subspaces (■ → ■ and ■ → ■). The rows in the data matrix represent the feature dimensions and their color represents the assigned subspace corresponding to the different clusterings. We can minimize the cost function w.r.t. the combined s-t-subspace, by finding the update rotation $V^{\langle c\rangle}$. (*Best viewed in color*)

m_s and m_t might have changed and therefore some dimensions of the rotated space may now be assigned to the other subspace. We should keep in mind, that we would like to order the features within each subspace based on their importance for the clustering. That is, in our example the most important feature for the first clustering in the full-space is the one that is mapped to the first dimension (first row) of the combined space $S_{s,t}$ (in $iv.$), but the most important feature for t is the one that corresponds to the last dimension (last row), due to the reasoning explained at the end of the previous section. Performing this procedure for each pair of subspaces, we optimize the transformation matrix V and the dimensionalities m_j w.r.t. each subspace, and hence further minimize our general cost function.

2.2.4 Accelerating the general case $S > 2$. The above-described procedure has the downside that we have to project the whole dataset onto the each of the combined subspaces. This can be quite costly in the face of big datasets.

However, we can circumvent this problem, because we only need for each clustering j the sum of the scatter matrices $\Sigma_j^{\langle c\rangle}$ in the combined subspace. These matrices can be computed from the sum of the scatter matrices $\Sigma_j^{\langle f\rangle}$ in the original data space via V and $P_{s,t}$. We can easily pre-compute these $\Sigma_j^{\langle f\rangle}$ in the original data space once and transform them for each combined-subspace into their counterparts. Thus, $V_{s,t}^{\langle c\rangle}$, m_s and m_t can readily be determined through $\text{eig}\left(P_{s,t}^\mathsf{T}V^\mathsf{T}\left[\Sigma_s^{\langle f\rangle}-\Sigma_t^{\langle f\rangle}\right]VP_{s,t}\right)$.

2.3 Convergence and Complexity

The complexity of our algorithm depends on the complexity of the classic Lloyd's algorithm $O(Idk|\mathcal{D}|)$, where I is the number of iterations. Within each iteration, we need to calculate the scatter matrices $O(Sd^2|\mathcal{D}|)$. Additionally, we have to perform the eigen-decomposition of each subspace pair that has an upper bound of $O(S^2d^3)$. This yields a total asymptotic complexity of $O(I(dk_{\text{total}}|\mathcal{D}|+d^2S|\mathcal{D}|+S^2d^3))$, where k_{total} is the total number of clusters. Therefore, Nr-Kmeans is comparable to most non-redundant clustering algorithms with respect to the cubic runtime in the dimensionality and quadratic runtime in the dataset size.

Since the cost function is decreasing (or rather non-increasing) with each update and assignment step and it is bounded from below, it converges towards a (local) minimum. It can be run multiple times—with different initializations—for a satisfactory clustering solution.

2.4 Noise-Space as a Special Subspace

We observed that the clustering result could often be improved by adding an additional subspace that only contains a single cluster. We call this subspace the *noise* space and the other subspaces consequently *clustered* spaces. The features associated with the *noise* space do not exhibit the structures of any clusters in the *clustered* spaces and all data point values within this subspace are drawn from the same unimodal, or uniform distribution. Therefore, in accordance with the k-means cluster assumption, we describe this space by a single cluster.

3 EXPERIMENTS

3.1 Quantitative Experiments

We compare Nr-Kmeans to several state-of-the-art algorithms. STATPC [19], INSCY [3] and RESCU [20] assign each cluster its individual axis-parallel subspace. ISAAC [25], mSC [21] and Orth1&2 (both in [6]) aim to find multiple clusterings in multiple, arbitrarily oriented subspaces. We selected these algorithms because of their common goal to reduce the redundancy between clusters. A more detailed discussion about differences and similarities follows in Section 4.

Table 2 shows the experimental results of our quantitative analysis for six synthetic and real-world datasets. Table 3 shows essential statistics of these datasets. The first four datasets—ALOI-2Sub,[2] Stickfigures [11] and Fruits [14], Syn3Sub[3]—are special non-redundant datasets and contain more than one set of class labels. Syn3Sub is a very simple dataset that contains three clustered subspaces with three, four and five Gaussian clusters, respectively. An additional subspace only contains noise drawn from a single Gaussian distribution. The remaining two datasets were taken from the UCI[4] repository and only contain a single set of labels each. The datasets ALOI-2Sub and Stickfigures are quite high-dimensional and some of the comparison methods are not able to handle the dimensionality without preprocessing.[5] Only Nr-Kmeans and Orth1&2 are fast enough to deal with both full-dimensional datasets and Table 3 shows the results for the full-dimensional dataset in parenthesis. In order to compare with all methods, we applied PCA to both datasets, keeping 90% of the total variance.

Since we have to account for multiple sets of class labels on the ground-truth side, as well as for multiple sets of clusterings as

[2] http://aloi.science.uva.nl/
[3] Available in the supplementary.
[4] https://archive.ics.uci.edu/ml/datasets.html
[5] We canceled each process after 24 hours.

	Pair-Counting F1 (pc-F1)						Average Variation of Information (∅VI)					
	ALOI-2Sub	Stickfigures	Fruits	Syn3Sub	Spam	Shuttle	ALOI-2Sub	Stickfigures	Fruits	Syn3Sub	Spam	Shuttle
Nr-Kmeans	(**0.77**) **1.00**	(**1.00**) **1.00**	0.74	**1.00**	**0.70**	**0.80**	(**0.95**) **1.39**	(**2.20**) **2.20**	**1.82**	**2.72**	0.85	0.97
ISAAC	0.46	0.86	0.71	0.63	0.69	0.78	*n/a*	2.06	1.57	*n/a*	1.04	0.87
mSC	0.81	0.71	0.59	0.72	0.62	†	1.38	1.26	*n/a*	1.39	**1.36**	†
ORTH1	(**0.77**) 0.82	(0.71) 0.89	0.72	0.70	0.68	0.74	(**0.95**) 1.37	(1.73) 1.96	1.79	2.71	0.20	1.93
ORTH2	(0.67) 0.82	(0.79) 0.71	**0.75**	0.77	0.69	0.69	(0.82) 0.60	(1.31) 1.52	1.37	2.70	0.22	**2.55**
STATPC	0.52	0.60	†	0.10	0.68	0.19	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*
INSCY	†	0.62	†	†	0.01	†	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*
RESCU	0.33	0.58	0.00	0.00	†	†	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*	*n/a*

Table 2: The left part of this table show the pair-counting F1 measures of several datasets. The right part shows the average Variation of Information, which measures the non-redundancy of the different clusterings of each result. Higher values are better for both scores. The values in parenthesis depict the full-dimensional result for high-dimensional dataset for which some of the comparing methods needed PCA-preprocessing. Results marked with † failed either due to memory constraints (>32GB), runtime demands (> 24 hours).

Name	$\lvert\mathcal{D}\rvert$	#Dims	#Clusters	Labels ∅VI
ALOI-2Sub	288	Org: 611, PCA: 8	2, 2	1.39
Stickfigures	900	Org: 400, PCA: 5	3, 3	2.20
Fruits	105	6	3, 3	1.73
Syn3Sub	2000	15	5, 4, 3	2.72
Spam	4601	56	2	*n/a*
Shuttle	43500	9	7	*n/a*

Table 3: The table shows some statistics of datasets used for the experiments.

a result of the algorithms, we use the pair-counting F1-measure (pc-F1) [1], which is—in contrast to classic performance measures like NMI or AMI—able to account for multiple label sets on both sides. Like the traditional F1-measure the best achievable value is 1, which is achieved if the clustering conforms to the class labels. Further, we measure the redundancy among the found subspaces in each outcome using the average of the Variation of Information metric [18] (∅VI). It measures the distance between two different clusterings, where higher values are better. The maximal value of this metric depends on the dataset size and it can therefore only be used to compare different values of the same dataset. This measure is only applicable if the clustering outcome contains more than one set of labels. Table 3 shows the ∅VI of the ground-truth labeling.

We set parameters concerning the number of subspaces—and the number of clusters within each subspace—depending on the dataset (needed for Nr-Kmeans, mSC, Orth1&2). For the first four datasets, we know the exact number of subspaces and clusters and we set the parameters accordingly. For the latter two UCI datasets, we estimated the number of subspaces and clusters using a greedy version of the elbow method: for each step, we created several candidate configurations based on the current configuration: (a) introduce a new subspace with two clusters, (b) increment the number of clusters in a subspace. For example, the configuration $(3, 2)$ yields the following candidates: $\{(4, 2), (3, 3), (3, 2, 2)\}$. Then, we selected among these configurations the one which yields the least costs for the next step. Finally, we selected among these steps the final configuration using the elbow method. Further, we configured Nr-Kmeans to also assume the existence of a *noise* space, as defined in Section 2.4.

Figure 3: The nine basic objects of the stickfigures dataset.

ISAAC uses the Minimum Description Length Principle and is therefore essentially parameter-free. All other parameters of each method were set according to the values suggested in the respective original paper.

3.2 Qualitative Experiments

3.2.1 Stickfigures Dataset. We now focus on a more detailed, qualitative interpretation of the clustering result found by Nr-Kmeans for the full-dimensional Dancing Stickfigures datasets. This analysis shows an additional strength of our approach that we have neglected so far—interpretable visualizations of the clustering outcome. The dataset is based on nine stickfigures in different poses shown as in Figure 3. From these nine images we can readily identify two different subspaces corresponding to the upper and lower body of the stickfigures and three different body poses for each subspace. Each data object represents one of these nine basic stickfigures with some additional, apparently random, clutter (noise) added to it.

However, while experimenting with different parameters, we experienced a major drop in the cost function, if we assume, next to the clusterings for the upper and lower body poses, an additional third subspace with three clusters. The first two clusterings recover the ground truth for the upper and lower body poses perfectly, each in two-dimensional subspaces. Yet, the third

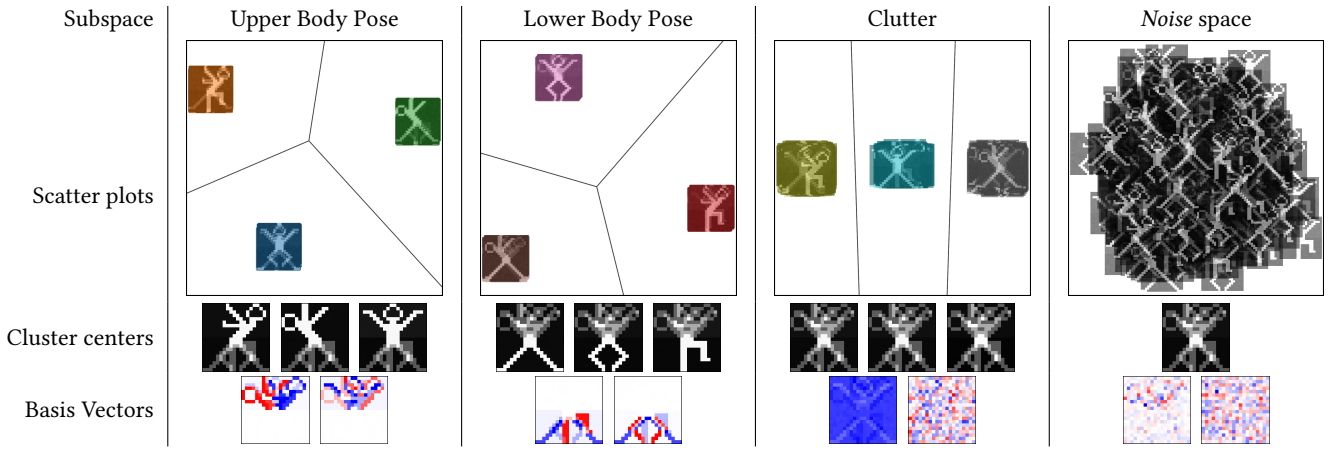


Figure 4: The table shows the different subspaces of the stickfigures dataset found by Nr-Kmeans. The first three columns represent the three two-dimensional *clustered* spaces. The last column represents the first two dimensions of the *noise* space. The scatter plots show the respective subspaces with cluster boundaries indicated by lines. The colors encode the labels of the closest matching ground truth. The next row shows the cluster centers $\mu_{j,i}$ in the original full-dimensional space, reshaped as matrix to match the stickfigure images. The last row shows the reshaped column vectors of V corresponding to the subspace. Each of these vectors is a basis vector for the respective subspace. The color saturation of each pixel represents the magnitude of the vector's entry with white being zero. The color encodes the sign with blue being negative and red being positive.

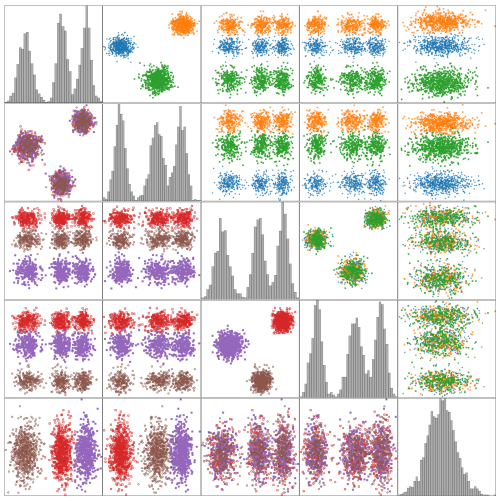
clustering—also two-dimensional—did not seem meaningful at first sight. Through further investigation and utilizing the visualization aspects of Nr-Kmeans, we found out that this third clustering actually expresses information about the type of clutter added to each image: The dataset creators used different ways to generate the clutter of each image.

Figure 4 shows several aspects of the found subspaces, as well as the *noise* space. The scatter plots reveal the important cluster structures of each *clustered* space, whereas for the *noise* space, we can see that it captures the unimodal structural information contained in the features. The cluster centers for the upper and lower body clearly show the respective pose. Yet, the cluster centers of the clutter subspace do not reveal the hidden concept. However, since we cluster the raw pixel of the images directly, we have the opportunity to reshape the column vectors of the rotation matrix V back into an image and interpret them much like the 'eigenfaces' of a PCA transformation. Because these vectors build the basis of the respective subspace, they reveal which linear mixture of features of the original space is essential for the clustering structure in the respective subspace. The vectors concerned with the upper body put almost no weight on features (pixels) of the lower image part. The features with high absolute values (high color saturation) correspond to the three different stickfigure poses. A similar behavior can be seen in the feature-components of the lower-body subspace. The first basis vector of the cluttered subspace finally indicates towards the nature of this subspace. We can see that this vector puts almost equal negative weight on all features and does not use positive weights for counterbalance—as the basis vectors of the other two *clustered* spaces do. This means that the structural information yielding the tri-section of the subspace is equally distributed among all features—just like the clutter.

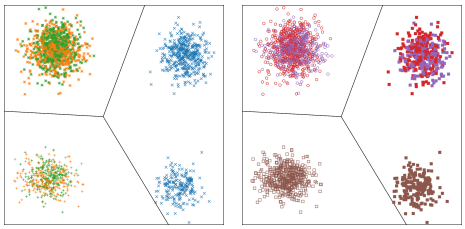
We were able to verify our findings of the clutter subspace, by extracting the ground truth independently from our clustering result. The clutter of each image can be extracted by subtracting its uncluttered basic counterpart. We established the ground truth by analyzing the distribution of this clutter. It confirms that the clutter was produced roughly in three different ways: (i) no clutter or uniformly distributed clutter within $[-3; 3]$ or a subinterval, (ii) uniformly distributed within $[-12; -1]$ or (iii) uniformly distributed within $[1; 12]$. We presume that the dataset creators did not anticipate that this procedure resulted in a third meaningful partitioning of the data and—to the best of our knowledge—this partitioning was never identified before. However, it felt somewhat presumptuous to just add these found clutter labels to the ground truth of our quantitative experiments.

In conclusion, the ability to visualize different aspects of the clustering outcome in a straightforward manner enabled us to find a third, previously unknown, but meaningful partitioning of the dancing stickfigures dataset.

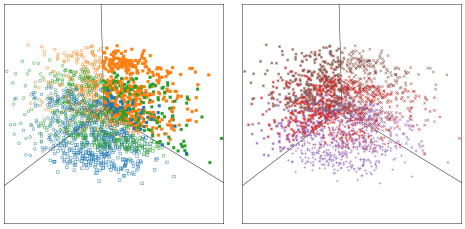
3.2.2 Incremental vs. Simultaneous Subspace Extraction. Another advantage of Nr-Kmeans is its ability to extract all clusterings and their associated subspaces simultaneously. To see why this is an important feature, we compare Nr-Kmeans to an incremental application of the special case $S = 2$ as discussed in Section 2.2.2 with only a single *clustered* space and a complementing *noise* space. The idea for this variation is that we extract subspaces incrementally from the *noise* space of the previous iteration. Figure 5 shows the result of this approach for a dataset with two non-redundant clustering structures, where we try to extract both clusterings. We can see that the first iteration, in Figure 5 (b), actually yields a result that is a mixture of both clustering structures contained in the data. However, as we can see in Figure 5 (c), this structural information



(a) Ground truth



(b) 1. Iteration of the incremental subspace extraction



(c) 2. Iteration of the incremental subspace extraction

Figure 5: The diagram shows a simple case for which NR-KMEANS recovers the ground truth (pc-F1=1.0), but incremental subspace extraction—related to what is applied by some comparison method—fails and yields an unsatisfactory result (pc-F1=0.73). The scatter matrix in (a) shows the dataset with two *clustered* spaces with three clusters each and a single noise feature. The colors of the scatter plot above the diagonal encode the cluster labels of the first subspace (orange, blue, green), the colors (red, purple, brown) in plots below encode the second clustering. The figures (b) and (c) show the clustering result of the first and second iteration of the incremental version, with color-coded ground truth for both sets of labels.

of both clusterings is subsequently lost in for the succeeding iterations in the complementing space and the quality of the result suffers severely from this. Furthermore, the scatter plot of the first iteration actually suggests a structure of four clusters, whereas the second iteration indicates no further clustering structure. The reason for this is that both clusterings are almost equally strong. We experienced a similar behavior for other incremental approaches such as Orth 1&2. On the other hand, a simultaneous extraction procedure of all *clustered* spaces, forces all clusterings to compete for the structural information contained in the data. As a consequence of this behavior, NR-KMEANS is able to recover the ground truth of this dataset with ease.

3.3 Runtime Experiments

Furthermore, we also performed two runtime experiments. In the first experiment, we progressively increased the number of objects, while keeping the number of features fixed. In the second experiment, we fixed the number of objects and successively increased the number of subspaces and dimensions. Figure 6 shows the results of these experiments.[6] For the first experiment, we assumed two two-dimensional subspaces, containing three normal distributed clusters each. We increased the dataset size by sampling uniformly from these distributions. For the second experiment, we fixed the dataset size at $|\mathcal{D}| = 1000$ and subsequently added a ten-dimensional subspaces containing two clusters with each iteration. This setup is similar to the one in [25]. For the comparison methods we used either their original implementation (if available) or the version

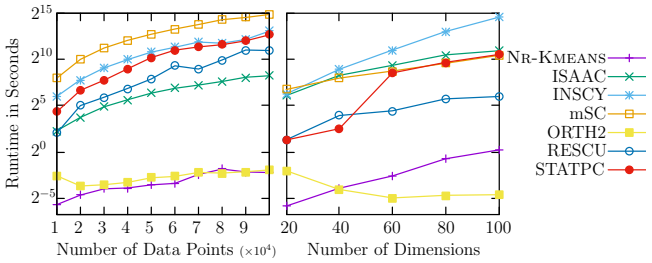


Figure 6: The diagrams show the scalability of NR-KMEANS compared to its competitors w.r.t. to $|\mathcal{D}|$ and d.

provided by OpenSubspace[7]. All experiments were conducted on a computer with an Intel Core i7 3.40GHz, 32GB RAM.

3.4 Discussion

Our experiments show that NR-KMEANS is a fast algorithm that, at the same time, yields results of a very high clustering quality and with a high non-redundancy. It achieves in all experiments the highest F1-scores and outperforms its competitors. ISAAC is the closest competitor in terms of scoring performance. Yet, this algorithm is in higher dimensional datasets—like most competitors—at least two orders of magnitude slower than NR-KMEANS. The only algorithms that are comparable in terms of runtime performance and non-redundancy are ORTH1&2, however, these methods find in most cases only a clustering of average accuracy. This is because

[6] Since ORTH2 is faster, we omitted the results of ORTH1.

[7] http://dme.rwth-aachen.de/de/opensubspace

ORTH1&2 buy their speed by performing PCA as an initial preprocessing step. Yet, if the first extracted subspace does not—or only partially—exhibit a single clustering structure but mixtures of different structures they cannot recover from it.

4 RELATED WORK

Many different methods have been proposed in the field of clustering that cover different aspects of the general idea. We concentrate our attention on work related to Nr-Kmeans.

Subspace clustering algorithms aim to find clusters in projections of the original data space. Related to the approach of Nr-Kmeans are redundancy-reducing subspace clustering algorithms, like INSCY [3], STATPC [19], RESCU [20] and NORD [15]. In contrast to Nr-Kmeans, these methods provide only a single set of clusters and assign each cluster to its individual axis-parallel subspace. The algorithms allow that an object can be assigned to multiple clusters and for this reason, they apply different techniques to reduce the redundancy between clusters. INSCY aims to reduce the redundancy via a depth-first processing with in-process-removal of redundant clusters, supported by a novel index structure. STATPC approximately extracts a suitable reduced, non-redundant set of statistically significant regions to detect clusters. RESCU involves a global optimization that detects the most interesting non-redundant subspace clusters by inspecting overlapping clusters and reducing the results to a manageable size. All methods have several input parameters that would be difficult to set in real-world applications.

Slightly different to the above-described methods is the recently proposed algorithm SubKmeans [17]. It aims to combine the k-means algorithm with a simultaneous dimensionality reduction step to find one arbitrarily oriented subspace. It can be seen as a special instance of Nr-Kmeans, in which we only assume a single *clustered* space with a single clustering and a complementing *noise* space.

The idea of **multiple, alternative clusterings** is also related to our method. They can be divided into semi-supervised and unsupervised approaches.

Unlike Nr-Kmeans, semi-supervised alternative clusterings methods, like COALA [4], ADFT [9], NACI [8] seek to find an alternative to one or more given reference clustering(s). The new clustering solution should thereby be dissimilar to the given ones and of high quality. COALA and ADFT generate a new clustering based on instance level 'cannot-link' constraints, extracted from an existing clustering. NACI utilizes a mutual information criterion. SMVC [11] combines instance-level constraints with variational Bayesian methods and additionally assigns axis-parallel subspaces. NMF [24] is a semi-supervised alternative clustering method that transforms the data into different subspaces by non-negative matrix factorization.

Unsupervised alternative clustering methods aim to find multiple clusterings either iteratively or through some dissimilarity constraint. However, unlike Nr-Kmeans, most methods in this category either consider always or at least for the first clustering the full-dimensional space. CAMI [7] aims to find exactly two Gaussian Mixture Models (GMMs) with sharing minimal information using an EM-style optimization technique. minCEntropy [23] utilizes the conditional entropy as an optimization criteria. Orthogonal projection clustering [6] uses two strategies, (1) orthogonal clustering (Orth1), and (2) clustering in orthogonal subspaces (Orth2), to partition the data sequentially into multiple clusterings. Both strategies use PCA as a preprocessing step to reduce the dimensionality and then use k-means to find clusters. However, as our experiments have shown, this sometimes removes valuable structural information. MVGen [12] generates multiple overlapping clusterings in different views of data by using mixture models. MVGen does not focus on non-redundancy of the data. Multiple Stable Clustering [14] detects multiple clusterings using the idea of clustering stability based on the Laplacian Eigengap. Yet, the found multiple stable clusterings cannot guarantee diversity, i.e. some clusterings are redundant and potentially difficult to interpret. mSC [21] integrates the relaxed spectral clustering objective with the Hilbert-Schmidt independence criterion (HSIC) to find multiple non-redundant views and clusterings within each view. A serious disadvantage of mSC compared to Nr-Kmeans is that it does not scale to high-volume data because it needs to perform an eigen-decomposition of the graph Laplacian matrix. ISAAC [25] is a parameter-free, non-redundant subspace clustering method, which combines the MDL principle with Independent Subspace Analysis (ISA). It first detects statistically independent subspaces and then fits GMMs within each subspace. It is similar to Nr-Kmeans in that it learns a linear transformation to decompose the data space. In contrast to Nr-Kmeans, ISAAC is only interested in independent subspaces and therefore cannot search some arbitrarily-oriented subspaces that may be dependent, but in which clusters may really reside.

5 CONCLUSION

In this paper, we proposed Nr-Kmeans, an extension of the classical k-means algorithm that is able to find multiple non-redundant partitions within a dataset. It simultaneously identifies for each partition the optimal, arbitrarily oriented subspace, orthogonal to the other subspaces. The inclusion of a *noise* space enables Nr-Kmeans to additionally remove features that are not well represented by any of the partitions. This can be interpreted as a simultaneous dimensional reduction step. Nr-Kmeans has many desirable properties. The most basic, non-optimized version is easy to implement and only uses standard features provided by all linear algebra frameworks. Even in its most basic implementation, it is very fast. Additionally, Nr-Kmeans can be incorporated with many other extensions proposed for the classic k-means in a straightforward manner.

Future efforts may be directed towards the incorporation of a fully-automated selection procedure for the number of subspaces and clusters within. Other research directions may lie on the development of approximative extensions.

REFERENCES

[1] Elke Achtert, Sascha Goldhofer, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. 2012. Evaluation of Clusterings - Metrics and Visual Support. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012.* 1285–1288. https://doi.org/10.1109/ICDE.2012.128

[2] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

[3] Ira Assent, Ralph Krieger, Emmanuel Müller, and Thomas Seidl. 2008. INSCY: Indexing Subspace Clusters with In-Process-Removal of Redundancy. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008),*

December 15-19, 2008, Pisa, Italy. 719–724. https://doi.org/10.1109/ICDM.2008.46

[4] Eric Bae and James Bailey. 2006. COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*. 53–62. https://doi.org/10.1109/ICDM.2006.37

[5] Sanjay Chawla and Aristides Gionis. 2013. k-means-: A Unified Approach to Clustering and Outlier Detection. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA*. 189–197. https://doi.org/10.1137/1.9781611972832.21

[6] Ying Cui, Xiaoli Z. Fern, and Jennifer G. Dy. 2007. Non-redundant Multi-view Clustering via Orthogonalization. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*. 133–142. https://doi.org/10.1109/ICDM.2007.94

[7] Xuan-Hong Dang and James Bailey. 2010. Generation of Alternative Clusterings Using the CAMI Approach. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*. 118–129. https://doi.org/10.1137/1.9781611972801.11

[8] Xuan Hong Dang and James Bailey. 2010. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*. 573–582. https://doi.org/10.1145/1835804.1835878

[9] Ian Davidson and Zijie Qi. 2008. Finding alternative clusterings using constraints. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 773–778.

[10] Susan A Gelman and Meredith Meyer. 2011. Child categorization. *Wiley Interdisciplinary Reviews: Cognitive Science* 2, 1 (2011), 95–105.

[11] Stephan Günnemann, Ines Färber, Matthias Rüdiger, and Thomas Seidl. 2014. SMVC: semi-supervised multi-view clustering in subspace projections. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. 253–262. https://doi.org/10.1145/2623330.2623734

[12] Stephan Günnemann, Ines Färber, and Thomas Seidl. 2012. Multi-view clustering using mixture models in subspace projections. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*. 132–140. https://doi.org/10.1145/2339530.2339553

[13] Greg Hamerly. 2010. Making k-means even faster. In *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 130–140.

[14] Juhua Hu, Qi Qian, Jian Pei, Rong Jin, and Shenghuo Zhu. 2015. Finding Multiple Stable Clusterings. In *2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015*. 171–180. https://doi.org/10.1109/ICDM.2015.101

[15] Nina Hubig and Claudia Plant. 2017. Information-Theoretic Non-redundant Subspace Clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 198–209.

[16] Brian Kulis and Michael I Jordan. 2012. Revisiting k-means: New algorithms via Bayesian nonparametrics. *In Proceedings of the 23rd International Conference on Ma- chine Learning* (2012).

[17] Dominik Mautz, Wei Ye, Claudia Plant, and Christian Böhm. 2017. Towards an Optimal Subspace for K-Means. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. 365–373. https://doi.org/10.1145/3097983.3097989

[18] Marina Meilă. 2007. Comparing clusterings—an information based distance. *Journal of multivariate analysis* 98, 5 (2007), 873–895.

[19] Gabriela Moise and Jörg Sander. 2008. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*. 533–541. https://doi.org/10.1145/1401890.1401956

[20] Emmanuel Müller, Ira Assent, Stephan Günnemann, Ralph Krieger, and Thomas Seidl. 2009. Relevant Subspace Clustering: Mining the Most Interesting Non-redundant Concepts in High Dimensional Data. In *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*. 377–386. https://doi.org/10.1109/ICDM.2009.10

[21] Donglin Niu, Jennifer G. Dy, and Michael I. Jordan. 2010. Multiple Non-Redundant Spectral Clustering Views. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. 831–838. http://www.icml2010.org/papers/342.pdf

[22] Kelvin Sim, Vivekanand Gopalkrishnan, Arthur Zimek, and Gao Cong. 2013. A survey on enhanced subspace clustering. *Data Min. Knowl. Discov.* 26, 2 (2013), 332–397. https://doi.org/10.1007/s10618-012-0258-x

[23] Nguyen Xuan Vinh and Julien Epps. 2010. minCEntropy: A Novel Information Theoretic Approach for the Generation of Alternative Clusterings. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*. 521–530. https://doi.org/10.1109/ICDM.2010.24

[24] Sen Yang and Lijun Zhang. 2016. Non-redundant multiple clustering by nonnegative matrix factorization. *Machine Learning* (2016), 1–18.

[25] Wei Ye, Samuel Maurus, Nina Hubig, and Claudia Plant. 2016. Generalized Independent Subspace Clustering. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 569–578.