I fold you so! An internal evaluation measure for arbitrary oriented subspace clustering

Daniyal Kazempour, Anna Beer, Peer Kröger, Thomas Seidl
Ludwig-Maximilians-University Munich
{kazempour, beer, kroeger, seidl} @dbs.ifi.lmu.de

2020 International Conference on Data Mining Workshops (ICDMW) | 978-1-7281-9012-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDMW51313.2020.00051

***Abstract*—In this work we propose SRE, the first internal evaluation measure for arbitrary oriented subspace clustering results. For this purpose we present a new perspective on the subspace clustering task: the goal we formalize is to compute a clustering which represents the original dataset by minimizing the reconstruction loss from the obtained subspaces, while at the same time minimizing the dimensionality as well as the number of clusters. A fundamental feature of our approach is that it is model-agnostic, i.e., it is independent of the characteristics of any specific subspace clustering method. It is scale invariant and mathematically founded. The experiments show that the SRE scoring better assesses the quality of an arbitrarily oriented subspace clustering compared to commonly used external evaluation measures.**

I. Introduction

Among the greatest challenges in developing unsupervised machine learning techniques, particularly clustering methods, is the evaluation of the results. In most works, the results of a new algorithm are compared against those of other algorithms based on a ground truth for the datasets they are tested on. This ground truth is not only rarely given for available datasets, but contradicts the goal of unsupervised algorithms. In cases where no ground truth is given or appropriate, internal quality measures are used. However, internal evaluation criteria assume that clusters have certain properties and measure how well these specific properties are fulfilled. Depending on the assumptions behind the measures, there is a strong bias towards certain types of clustering models, which we will elaborate on in Section II.

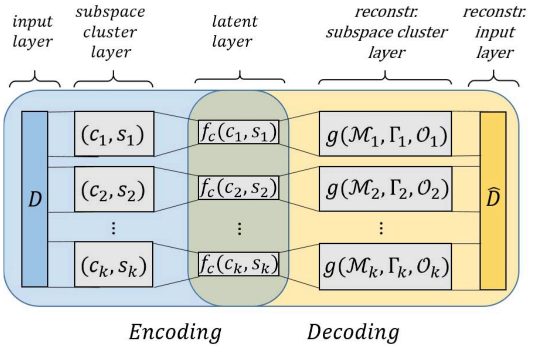

Fig. 1. Architecture of subspace clustering as a reconstruction task.

A very recent class of clustering methods is designed to detect clusters in arbitrarily oriented subspaces of a high-dimensional dataset, also known as correlation clustering methods[1] [1]. The main idea behind this type of clustering methods is to detect groups of objects that have a similar correlation among a given set of features and, thus, are located on a low-dimensional linear subspace E in ambient space. These objects are in fact similar to each other when projected onto the subspace perpendicular to E. While there exists a thorough evaluation of axis-parallel subspace clustering methods (a variant of the general problem described above assuming attribute independence) based on a plethora of different external criteria [2], there is, to the best of our knowledge, no internal quality measure for arbitrarily-oriented subspace clustering methods.

In this work, we introduce SRE (*S*um of subspace *R*econstruction *E*rrors), the first internal evaluation criterion for subspace clustering. It performs well as we show with experiments in Section IV and scores aspects of a clustering which are neglected by external evaluation measures yet important in context of arbitrarily oriented subspace clustering. It is model-agnostic, i.e., it does not have a bias towards a given clustering model (density-based, Hough-based, ICA-based arbitrarily oriented subspace clustering etc.). To achieve this, we propose a different perspective on the subspace clustering problem. Inspired by the concept of autoencoders [3] which have been also proposed for dimensionality reduction [4] we re-define subspace clustering as a task in which the objective is to partition a given dataset in ambient space. This segmentation is performed in such a way that the obtained subspace for each of the clusters allows a reconstruction of the data from this latent space with a minimized loss while at the same time reducing the model complexity, i.e. reducing the number of dimensions for the subspace and reducing the number of clusters. Autoencoders learn a lower-dimensional embedding into the latent space, which minimizes the reconstruction error regarding the original ambient space. The latent space of an autoencoder is an arbitrarily shaped manifold, whereas the result of a subspace clustering is a set of subspace clusters, where each is described by its individual arbitrarily oriented linear subspace. Thus, using our new definition of subspace clustering, it delivers a piecewise linear approximation of an arbitrarily shaped manifold within the data. The non-neural autoencoding architecture of our proposed method is illustrated in Figure 1. The approximation loss together with

[1] Not to be confused with correlation clustering in context of graph mining.

2375-9259/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDMW51313.2020.00051

regularization parameters for the number of clusters and the number of subspaces builds the new internal quality measure SRE. At this point one may object that by relying on the assumption that data is located on or around manifolds, we would have some kind of an ”external” criterion. This however is not the case, since the manifold assumption addresses an *internal* property, like the silhouette-coefficient [5] also relies on the internal model assumption that the clusters are convex. The remainder of this paper is structured as follows: In Section II we introduce related work, in Section III we give a formalization for our new perspective on subspace clustering and the consequential internal quality measurement *SRE*, establishing a link to autoencoders. After investigating some properties of *SRE*, we describe the conducted experiments in Section IV and discuss in this context the feasibility of our internal measure. We conclude our work with the lessons learned and future prospects. In summary, our contributions are as follows: **(1)** We propose a model-agnostic internal evaluation measure for subspace clustering **(2)** We re-define the subspace clustering problem as an optimization task with the objective to minimize the reconstruction error from a piecewise linear approximation of a non-linear manifold, while at the same time reducing the model complexity.

II. Related Work

The so far existing internal evaluation measures such as the silhouette coefficient [5], the density-based validation index [6] or modularity [7] are limited to their respective underlying models, and as such come with their model bias. There exist however further internal evaluation measures such as the Davies-Bouldin score [8]. This score measures the average of the pairwise ratio of compactness and separation. Similarly the Calinski-Harabasz Score [9] and Dunn-Index [10], like many other existing internal evaluation measures, rely on properties such like compactness and separation. One drawback which comes with them, is the fact that for the compactness of the clusters the average distance of objects to their cluster centers is computed. This reliance on cluster centers introduces a certain bias. Further these internal evaluation measures neglect aspects like reconstruction quality, number of clusters or number of dimensions rendering them inadequate for our needs. Our SRE measure differs from the existing methods by integrating the reconstruction loss as well as the model complexity (number of clusters and dimensionality of subspaces). As such, we do not compare in this work SRE against other internal evaluation measures, since they would yield scores which may favor individual algorithms that rely on density or convex clusters, but not favor clusters which are ”tight” around an arbitrarily oriented subspace and which do not penalize increased model complexity. In a very recent work [11], the authors introduce an approach for a holistic assessment of clustering algorithms with respect to their structure discovery capabilities. In that work they rely on criteria, namely stability, structure and consistency. In their work, the notions of stability and consistency rely on a density threshold ϱ. This introduces unfortunately a model bias, since subspace clustering algorithms which rely on density would be in favor. In the work of [2], the authors propose an evaluation method which is founded on three major paradigms, namely (a) cell-based, (b) density-based and (c) clustering-oriented paradigms. All three paradigms are incorporated in the framework of an *external* measure for subspace clustering which relies on ground truth. The method is limited to axis-parallel subspace clusterings. The authors state in the introduction of their work, that it would be beyond the scope of their work to include arbitrarily oriented subspace clustering algorithms.

III. Subspace Clustering as Manifold Learning

Seeking for an algorithm-agnostic internal evaluation measure for subspace clustering leads us to autoencoders, which inspired this work. According to [12][Ch. 14], an autoencoder is a neural network which is aimed at copying its input to its output by encoding the input first to a lower-dimensional latent layer (bottleneck) $h = f(x)$ and decoding it to its full-dimensional representation $g(f(x)) = x$. The limitation of the dimensionality of the latent layer is necessary to learn lower-dimensional representations of the ambient space. The learning process itself is expressed as a minimization of a loss function such as $L(x, g(f(x)))$, where L is a penalty function depending on the dissimilarity between the reconstruction $g(f(x))$ and the ambient space representation of x. Bridging over to manifold learning, according to [12], autoencoders rely on the assumption that data is located on or around lower-dimensional manifolds or subsets of manifolds. A manifold is characterized by its set of tangent planes where each of the tangent planes represents a local euclidean space. Autoencoders learn such manifolds while balancing between two aspects: (1) achieving a good approximation of a manifold (minimizing reconstruction error) and (2) satisfying regularizations such as, e.g., the dimensionality of the latent layer. It is noteworthy at this point that if we use a linear decoder and the mean squared error as a loss function, the autoencoder (if undercomplete) learns the same subspace as a PCA. Under this aspect, a PCA learns one global linear approximation of a potentially non-linear manifold of the ambient space. However, autoencoders with non-linear encoder and decoder are capable of learning non-linear manifolds. If we apply local PCA [13] on the data in ambient space, we obtain a set of linear lower-dimensional subspaces and the objects projected to their respective subspaces. Arbitrarily oriented subspace clustering on the other hand yields a set of lower-dimensional subspaces with their corresponding objects projected to them. These arbitrarily-oriented subspaces can be described by a set of principal components like in ORCLUS [14] or through HNF representation of hyperplanes as in CASH [15]. Subspace clustering algorithms are distinct among themselves. While e.g. ORCLUS minimizes the so called energy of a clustering, CASH maximizes compactness, two terms which not necessarily comply with each other. Further ORCLUS optimizes by the number of partitions k and the dimensionality l while CASH optimizes by the minimum number of objects per hyperplane and maximum allowed deviation from it. At this point we ask: What is the common goal that not only those two, but any subspace clustering algorithm in general can be tailored at to optimize for? The answer is: Expressing data through a set of lower-dimensional subspaces. Just like autoencoders or local PCA aim at encoding the ambient space in such a way that the reconstruction from the latent representation is minimized, we re-phrase the subspace clustering task with the goal in mind to construct an internal evaluation measure:

Definition 1 (Subspace clustering as dimensionality-reduction and manifold learning task)**.** *Given a dataset $\mathcal{D}$ in ambient space, the task of a subspace clustering algorithm is to yield a clustering $\mathcal{C} = \{c_0, c_1, ..., c_k\}$ and their corresponding set of subspaces $\mathcal{S} = \{s_0, s_1, .., s_k\}$ such that the loss $\mathcal{L}(\mathcal{D}, g(f(C, S)))$ is minimized, where $(C, S) := \{(c_i, s_i)\}$, $f(C, S)$ corresponds to an encoding of the dataset and $g(f(C, S))$ corresponds to the decoding of the dataset.*

Definition 2 (Encoding of a subspace cluster)**.** *Given is a subspace cluster c_i as part of a $n \times d$-dimensional data matrix in a subspace s_i with dimensionality l, where d denotes the full dimensionality of the ambient space and k denotes the number of clusters. Further the index $i \in [0, k]$ denotes the corresponding cluster id. First, the data is centered by subtracting $\mathcal{O}_i$, the mean of all points in c_i. Then the PCA, which is based on the covariance matrix of all points in c_i, provides the ordered eigenvalues $\lambda_0, \ldots, \lambda_d$ and respective eigenvectors $\gamma_0, \ldots, \gamma_d$, last of which are known as the principal components of c_i. The subspace cluster is projected onto the first (i.e., those belonging to the largest eigenvalues) l principal components $\gamma_0, \ldots, \gamma_l$ by multiplying it with the $l \times d$-dimensional part of the eigenvector matrix $\Gamma_i = (\gamma_0, \ldots, \gamma_l)^T$: $\mathcal{M}_i = c_i \times \Gamma_i^T$. The complete encoding of a subspace cluster is then this l-dimensional projection of the points $\mathcal{M}_i$ together with Γ_i and the origin in the ambient space $\mathcal{O}$. Thus,*

$$f_c(c_i, s_i) = (\mathcal{M}_i, \Gamma_i, \mathcal{O}_i)$$

Definition 3 (Piecewise Linear Approximation (PLA))**.** *The PLA of a dataset $\mathcal{D}$ using results of a subspace clustering $(C, S) = \{(c_i, s_i)\}$ corresponds to the encoding $f(C, S)$ of the dataset, which is the combination of the encodings of all subspace clusters $f_c(c_i, s_i)$.*

Definition 4 (Decoding of a subspace cluster)**.** *Given the matrices $(\mathcal{M}_i, \Gamma_i, \mathcal{O}_i)$ of the encoding function as described in Definition 2 the decoding $\hat{\mathcal{D}}$ is obtained by transforming the encoded points back into the original data space: $\hat{\mathcal{D}} = \mathcal{M}_i \times \Gamma_i + \mathcal{O}_i$. In other words, the decoding function of a cluster is*

$$g_c(\mathcal{M}_i, \Gamma_i, \mathcal{O}_i) = \mathcal{M}_i \times \Gamma_i + \mathcal{O}_i$$

Based on these new definitions we want to investigate some connections between the piecewise linear approximation based on subspace clusterings and autoencoders. For autoencoders or neural networks in general, the so called universal approximation theorem [16] guarantees that a feed-forward neural network with at least one hidden layer is capable to approximate any function φ with a non-linear manifold F so that $\forall x, \forall \varepsilon > 0 : |F(x) - \varphi(x)| < \varepsilon$, if sufficient hidden units are provided. That is useful for the case that all points of a dataset were created by a common function φ, or in other terms: that all points were generated by a single process which are located on or around a non-linear manifold. But that is not necessarily the case in real world data and that is when subspace clustering is advantageous. We can formulate the corresponding theorem for subspace clustering, which is based on the points in the dataset and not on the function creating them:

Theorem 1 (Adapted Approximation Theorem)**.** *Every dataset $\mathcal{D}$ on a potentially non-linear manifold can be described by a piecewise linear approximation based on a subspace clustering (C, S) so that: $\forall \varepsilon > 0 : \mathcal{L}(\mathcal{D}, g(f(C, S))) < \varepsilon$, if sufficient subspace clusters of sufficient dimensionality are allowed.*

As a sketch the theorem can be proved using the original universal approximation theorem: for every function φ generating a dataset $\mathcal{D}$ there is a manifold F approximating φ as described above. Since a manifold is local Euclidean, there exists a neighbourhood for every data point which is homeomorph to an open subset of $\mathbb{R}^n$. These subsets correspond to the eigenspaces of the subspace clusters as described in Definition 2, and thus φ resp. $\mathcal{D}$ can also be approximated with the PLA as described in Definition 3. A *trivial solution* to approximate $\mathcal{D}$ could put every point in a single, full-dimensional cluster and thus maximize the number of clusters $|(C, S)|$ as well as their dimensionality $dim(c_i)$. The number of clusters $|(C, S)|$ and their dimensionality $dim(c_i)$ plays a fundamental role in Theorem 1, as a trivial solution could maximize both variables and put every point in a single, full-dimensional cluster. The arbitrariness of those two variables corresponds to the arbitrary number of hidden units of the neural network in the universal approximation theorem. This leads to a diversification of this one abstract variable ”number of hidden units” to the two variables ”number of groups in the dataset” and ”number of relevant attributes for each group”. Nevertheless, both variables should be taken into account when using it as optimization criterion or as evaluation measure, see Section III-A.

A. Manifold learning as internal measure

Just as in [12] it is stated that autoencoders need a balancing between minimizing the reconstruction error and satisfying regularizations such as the dimensionality of the latent layer in order to learn a meaningful lower-dimensional representation of the data. We introduce here constraints which are indispensable for the subspace clustering task.

Definition 5 (Reconstruction Loss)**.** *The reconstruction loss of a single cluster is the sum of all squared distances between the original points x_j and the reconstructed points $\hat{x}_j$ of the cluster. Normalizing by the number of elements in the cluster leads to invariance regarding the number of points in the* *cluster (and thus in the dataset)*

$$\mathcal{L}_c(c_i, g(f_c(c_i, s_i))) = \frac{1}{|c_i|}\sum_{j=1}^{|c_i|} dist(x_j, \hat{x}_j)^2$$

As distance function we use the Euclidean distance divided by the square root of the number of dimensions to stay scale invariant (see Section III-B) w.r.t. the number of dimensions: $dist(x, \hat{x})^2 = \frac{1}{d}\sum_{i=1}^{d}(x^i - \hat{x}^i)^2$*, where* x^i *denotes the value of* x *in the* i*-th attribute. The reconstruction loss of the dataset is then*

$$\mathcal{L}(\mathcal{D}, g(f(C,S))) = \sum_{i=1}^{|(C,S)|} \mathcal{L}_c(c_i, g(f_c(c_i, s_i)))$$

As we want to prevent that the subspace clustering simply learns the identity function as described after Theorem 1 we introduce the possibility to penalize high dimensionalities of subspace clusters as well as a high number of clusters. For the former, the hyperparameter α is multiplied with the subspace dimensionality[2] l and the product is added to the cluster loss $\mathcal{L}$. Here, $\alpha = 0$ means that we do not put any sparsity constraint on the dimensionality of the latent layer, while the larger α the higher we penalize the dimensionality. For the latter, the hyperparameter β is multiplied with the number of clusters $|(C,S)|$ and also added to $\mathcal{L}$, leading to the total reconstruction loss:

$$\mathcal{L}_{total}(\mathcal{D}, g(f(C,S))) = \mathcal{L}(\mathcal{D}, g(f(C,S))) + \alpha \cdot l + \beta \cdot |(C,S)|$$

Figure 1 points out the connection between our new definition of subspace clustering and classic autoencoders. Equivalent to the encoding step of an autoencoder we perform the subspace clustering and with that a PLA of the original data. In the latent layer each cluster is projected onto its own low dimensional eigenspace according to the principal components and the dimensionality as given by the clustering. The decoding corresponds to the re-transformation to the original space. At this point there are certain questions which may arise, regarding the hyperparameters α and β: (1) Couldn't both hyperparameters be regarded as an "external" penalty? Here the answer is: it depends. If both are set equal ($\alpha = \beta$) then both influences (number of clusters and dimensionality of subspaces) are treated equally. However, the external character of α and β do not make this approach an external evaluation measure, since it still relies on the internal assumption that non-linear manifolds are piecewise-linearly approximated in such a way that the deviation from the linear approximations is minimized. (2) Does a hyperparameter not introduce uncertainty in the evaluation? It depends in one aspect on the sensitivity of the hyperparameters. The authors in [11] also introduce a hyperparameter ϱ in their internal measure and claim that it is insensitive. In this work we conducted experiments (Experiment 4) with the purpose to investigate the sensitivity. Further, this "uncertainty" can, depending on the use-case of the scientists, be a degree of control. In autoencoders the users can set the dimensionality of the latent layer (aka "the bottleneck). In our architecture, the scientists can also, if they have a certain background knowledge explicitly set and therefore control the weights of latent space dimensionality and cluster cardinality, emphasizing besides the reconstruction loss either the importance of the number of clusters or the importance of a low-dimensional subspace.

B. Properties of the internal evaluation measure

In the following we investigate the SRE w.r.t. some desirable properties for internal evaluation measurements.

a) Scale invariance regarding number of dimensions: If the quality of the reconstruction per dimension is fixed, i.e., if the mean and the variance of the dimension-wise distance between original point and reconstructed point is fixed, then the number of dimensions of the dataset does not influence the SRE $\mathcal{L}$. The main idea of the proof, which is here omitted for brevity, is that the distance between original and reconstructed points we defined in Definition 5 is independent of their dimensionality in contrast to, e.g., the Euclidean distance. In practice, the premises are fulfilled if the ratio between noise dimenions and important dimensions per cluster stays the same.

b) Scale invariance regarding number of points : SRE is independent of the number of points in a dataset for a fixed number of clusters, since the cluster loss $\mathcal{L}_c$ is not the accumulated, but the average loss of all points in that cluster.

c) Scale invariance regarding number of clusters: As the loss can be minimized by increasing the number of clusters for describing the dataset, $\mathcal{L}$ is dependent on the number of clusters. Since it depends on the structure of the dataset to what extent additional clusters can reduce the reconstruction loss, there is no universal normalization we could improve the SRE with regarding this aspect, which is why β is chosen according to the dataset. Additionally, the optimal number of clusters depends on the usecase, so here β allows the user to tailor the quality function to the respective goal of the clustering.

d) Comparability: Basically, there are three types of experiments when scientists need an internal evaluation measurement. They want to compare (1) different algorithms on the same dataset, (2) different parameter settings of one algorithm on a particular dataset, or (3) the applicability of an algorithm on different datasets. SRE is most suitable for type 1 and through α and β it is also useful for type 2. Type 3, the comparison of the performance of an algorithm on different datasets, can be difficult since the reconstructability of a dataset depends on its structure. However, we should compare these results on different datasets with other algorithms' results anyway.

IV. Experiments

In this section we describe and provide the results of several experiments, conducted on generated data as well as on popular real world datasets Iris, Wine, Breast Cancer, and Digits

[2] If there are different dimensionalities the median is used in the experiments.

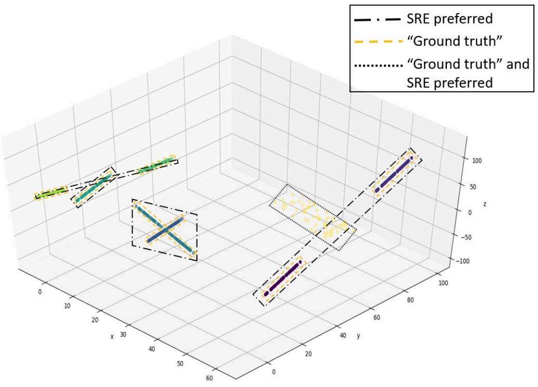

Fig. 2. Synthetic dataset for comparing SRE against a set of external evaluation measures. Best viewed in color.

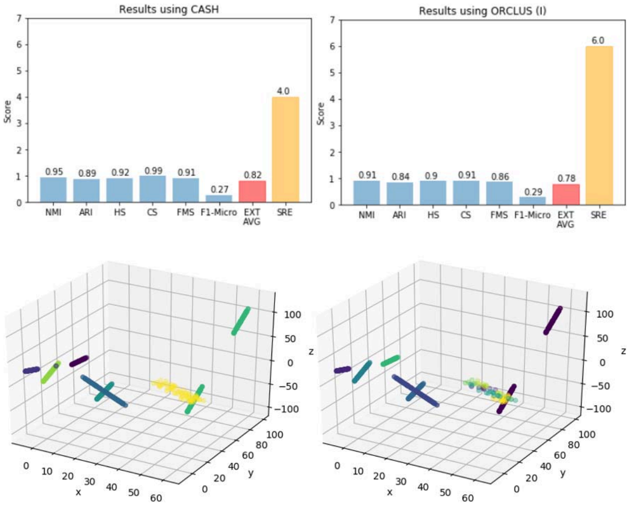

Fig. 3. Synthetic dataset for comparing SRE against a set of external evaluation measures. Best viewed in color.

as provided by the sklearn[3] framework. The dimensionality, number of objects per dataset and number of classes can be seen on the sklearn dataset webpage[3].

A. Experiment 1: SRE vs. External Evaluation Measures

We introduce this first experiment by asking: What makes an evaluation measure actually convincing? The reader may ask also in this context: why should one bother with SRE, if we already have a ground truth with our labeled data? Why not just relying on external evaluation measures such as NMI, ARI etc. like it is done in the majority of previous research so far? In order to approach these questions we have constructed an artificial dataset (3D), with the purpose to make the clustering results as comprehensive as possible. The dataset consists of eight clusters according to our defined "ground truth" indicated by dashed lines and consists of five clusters according to our evaluation model indicated by alternating dashed and dotted lines as shown in Figure 2. Each of the ground truth clusters contain between 60 and 80 objects.

We compare SRE ($\alpha = \beta = 0.5$) against external evaluation measures such as the normalized mutual information (NMI), the adjusted rand index (ARI), the homogenity score (HS), the completeness score (CS), the Fowlkes Mallows score (FMS) and the micro F1 score (F1S). For all mentioned external evaluation measures holds that a score of 1 corresponds to the best possible result, while a score of 0 corresponds to a poor clustering. For the SRE it holds that the lower the score, the better. The experiments were conducted on the artificial dataset, running ORCLUS [14] and CASH [15]. Both algorithms were executed on a hyperparameter grid. We have chosen the best clustering result based on the criterion of obtaining the highest average of the external quality measures: $\frac{NMI+ARI+HS+CS+FMS+F1S}{5}$. The results can be seen in Figure 3 and 4.

In Figure 3 the top part shows bar charts with the external measure results and the SRE. In the bottom we have scatter plots where one can see the clustering results. The left bar chart and scatter plot shows the results for CASH, the right scatter plot shows the results for ORCLUS. For CASH we can observe that the average of external quality measures yielded 0.82. Why didn't we obtain a value of 1.0 or at least close to 1.0? One reason for that observation is the poor value of the F1 score. Besides that, there is one major aspect which prevents the other external measures to achieve a score of 1: the two rightmost linear segments are detected as one line despite the fact that according to the ground truth they should be two separate clusters. According to the model behind SRE, assigning both distant lines to the same cluster is favorable, since it results in fewer number of clusters while having no impact on the loss, since both clusters are perfectly located on the same line. At this point one drawback of using external measures becomes visible: neglecting properties of subspace clustering. The results of ORCLUS (Figure 3 right) yield an average external score of 0.78 and therefore just by 0.04 worse compared to the CASH result. Comparing both clustering results (CASH and ORCLUS) in the bottom plots it can be observed that ORCLUS fragmented the planar cluster into four smaller planar clusters. This fragmentation does not seem to affect the external scores significantly, which is however from the perspective of the underlying model of SRE a massive difference. The hyperparameter settings of ORCLUS which led to this clustering are $k = 10, l = 2$. The number of clusters has a high impact on the SRE score, as well as the fact that all clusters are of dimensionality $l = 2$. By purely observing the average scores (0.82 for CASH and 0.78 for ORCLUS) one may think that both clustering results are almost equally good. By looking at the SRE scores (4.0 for CASH and 6.0 for ORCLUS) one can see that the result of CASH is by 33% better compared to the ORCLUS clustering. Through this case we get a first impression that relying on external evaluation measures only, may lead us to draw either wrong conclusions, or at least conclusions which neglect the

[3] https://scikit-learn.org/stable/datasets/index.html

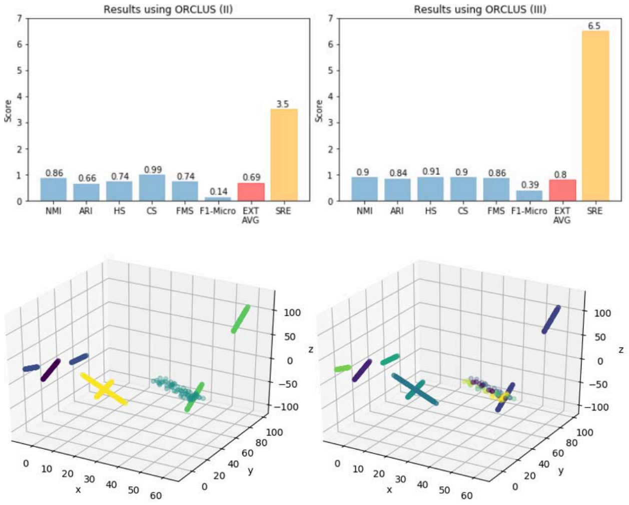

Fig. 4. Synthetic dataset for comparing SRE against a set of external evaluation measures on ORCLUS clustering results. Best viewed in color.

principles of arbitrarily oriented subspace clustering.

So far we have looked at the external evaluation scores and the corresponding SRE score of a clustering. We investigate now the question: Does SRE correspond to the external evaluation scores? For this purpose we run ORCLUS again on our artificial dataset with the same parameter settings as before. After 20 runs with the hyperparameters $k = 10$, $l = 2$ we obtain a clustering as seen in Figure 4 (left). Here the average external score is 0.69 and thus by 0.09 worse compared to the previous ORCLUS run (ORCLUS (I) Figure 4) and by 0.13 worse compared to CASH. Solely based on the external scores one would state that this clustering is of clearly worse quality than the others. Albeit the external scores are worse, the SRE is with 3.5 much lower, i.e., better, compared to the first ORCLUS run with an SRE of 6.0 (42%) and even by 12.5% better compared to the CASH result. How can we actually explain these massive discrepancies? Regarding the SRE, the discrepancies can be understood by taking a closer look at the resulting clustering (Figure 4 left, bottom). ORCLUS has detected five clusters of dimensionality $l = 2$. Compared to the first ORCLUS run, which detected 10 clusters by fragmenting the bottom planar cluster, we have a reduction in number of clusters by 50%. Further the two intersecting lines have been detected as a single planar cluster. For the underlying model of SRE, this clustering (1 cluster, dimensionality of 2) is equally scored as detecting both separately (2 clusters, dimensionality of 1), since both lines are located on a plane, the reconstruction loss in both cases is 0. External evaluation measures, however, are incapable to handle such cases of equivalence, leading to a drop of their respective scores. This issue comes additionally to the previously elaborated fact that both distant linear clusters on the right are detected as one, despite the ”ground truth” expects them both to be regarded as separate clusters. Finally in our last run (ORCLUS (III)) with $k = 20$, $l = 2$ we obtain a clustering with an average external score of 0.8, thus being almost en par with the CASH result (0.82). Here again, by solely looking at the external average scores one may be tempted to state that the ORCLUS (III) clustering would be almost as good as the CASH clustering. A close look at the clustering itself (Figure 4, right, bottom) reveals that while the average external score is high, it again has fragmented the planar cluster in many smaller ones. While the external measures fail to capture this fragmentation the SRE score of 6.5 reveals that the result is by about 39% worse compared to the result of CASH (4.0). Referring to the initial question of this experiment of why one should bother using SRE when we could as well just use external measures, one answer to that is: Even if we have labels, and even if we want to trust a “ground truth”, external quality measures fail to capture fundamental concepts of an arbitrarily oriented subspace clustering: taking into account the dimensionality and the number of clusters besides the loss.

B. Experiment 2: Pure Reconstruction Quality

How do the algorithms behave on different datasets if we do not apply any regularization at all, which means $\alpha = \beta = 0$? Do we observe common patterns among different datasets regarding loss, number of dimensions or number of clusters? And do the algorithms exploit automatically a higher number of dimensions and higher number of clusters if no regularization is given? The experiment was conducted using the following algorithms: ORCLUS [14], 4C [17], COPAC [18] and LMCLUS [19]. We used the implementations from the data mining framework ELKI [20] for all algorithms. The best hyperparameters for every algorithm was found by iterating over their respective hyperparameter grid using the same step size[4] for each algorithm per experiment. The settings yielding the lowest measured loss per algorithm can be seen in the sourcecode under the following link[5].

For each of the algorithms a triple containing the lowest loss, the number of subspaces and the dimensionality of the subspaces is obtained as shown in Table I. In that table we can observe what we have expected and elaborated on in Section 2: Without any regularization, algorithms like ORCLUS, COPAC and 4C exploit a high number of dimensions as well as a high number of clusters. For the Iris dataset COPAC dominates with the lowest loss and at the same time the lowest dimensionality, but puts almost every point in a single cluster. LMCLUS exploits the dimensionality but interestingly not the number of clusters which yields one single cluster with a reconstruction loss of 0.005. Throughout all datasets LMCLUS dominates the number of clusters by requiring only one cluster, which is interesting since there is no actual parameter of the algorithm directly influencing the number of clusters except $minpoints$. On the Wine dataset ORCLUS dominates by having the smallest loss *and* the smallest number of clusters, while COPAC has also a loss of zero, but has the lowest dimensionality, at the cost of having 176 clusters. In the Breast

[4] 1 for minpts, dim, k, 0.1 for ε-range, Eigenvalue threshold δ, sensitivity threshold

[5] Sourcecode: https://www.dropbox.com/s/1nhln46cxaidael/ORTA.zip?dl=0

TABLE I
Reconstruction loss, dimensionality and number of clusters for the detected subspaces by the algorithms for different real-world datasets ($\alpha = \beta = 0$). Bold entries represent results being dominated by at least one criteria (loss, dim, clus).

	Iris			Wine			Breast			Digits		
Algorithms	Loss	Dim	Clus	Loss	Dim	Clus	Loss	Dim	Clus	Loss	Dim	Clus
ORCLUS	0.000	3	15	**0.000**	**12**	**2**	**0.000**	**25**	**5**	**0.000**	**60**	**5**
COPAC	**0.000**	**2**	**149**	**0.000**	**1**	**176**	**26.747**	**2**	**1**	**7.617**	**5**	**5**
4C	**0.000**	**3**	**144**	0.000	12	178	305.744	29	5	0.000	63	1613
LMCLUS	**0.005**	**3**	**1**	**0.019**	**8**	**1**	**0.010**	**8**	**1**	**6.809**	**11**	**1**

cancer setting, ORCLUS yields the lowest loss, but at the same time with a higher dimensionality, while COPAC has a comparably high loss but at the same time a very low dimensionality and number of clusters. Finally in the Digits dataset ORCLUS comes with a loss of zero, but achieves this by exploiting almost the full dimensionality of the dataset (64) where COPAC has a high loss, but comes again with lower number of dimensions and same number of clusters.

C. Experiment 3: Influence of Dimensionality

In this experiment we ask: How does a fixed dimensionality of the subspaces influence the number of clusters and the pure reconstruction loss $\mathcal{L}$ for the different algorithms? For that we compare $\mathcal{L}$ as well as the number of clusters of ORCLUS, 4C, LMCLUS, and a simple autoencoder AE (with a single layer, ReLU for encoding, linear activation function for decoding, 1000 epochs, and batch size 4) from Keras[6] on the wine dataset for the fixed dimensionaloities $l \in \{1, 2, 3, ..., 12\}$. The runs of the autoencoder have been repeated ten times per dimensionality, selecting the result with the lowest loss. The algorithms CASH and COPAC have been omitted in this experiment, since no maximum dimensionality of subspaces can be enforced. The results in Figure 5 (left) show, as expected, a decreasing loss with increasing dimensionality for almost all algorithms. Only 4C achieves a loss of 0 for all dimensionalities, exploiting the fact, that we have imposed no limitations on the number of clusters and thus creating an own cluster for each object. While LMCLUS provides a lower loss even at dimensionality 1, it is surpassed by ORCLUS at a dimensionality of 8 which is not visible in Figure 5 (left). The exploitation as seen by 4C demonstrates the necessity of α to regulate the number of clusters. The less effective performance of the autoencoder has to be taken with some grains of salt, since it consists of a single-layer without any further optimizations which may not reflect the full potential of this technique, whose further investigation is beyond the scope of this work.

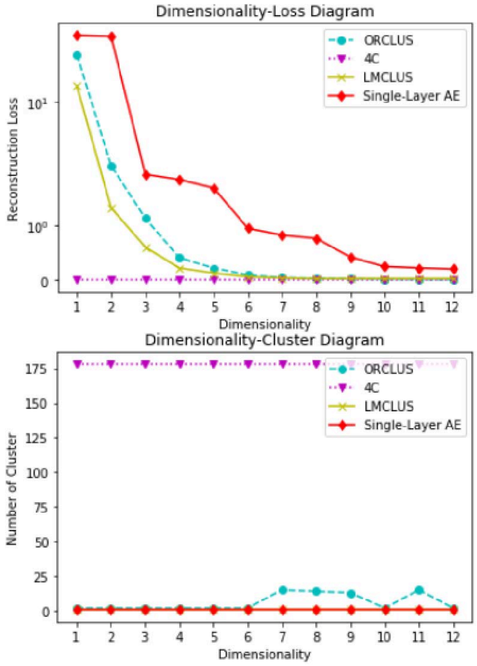

Fig. 5. Top: Loss depending on the dimensionality l on the Wine dataset. Vertical axis is in symmetric log-scale. Bottom: Number of clusters depending on the dimensionality l on the Wine dataset.

While we have obtained the loss from each of the methods considering different number of dimensions, we ask: how many clusters did the methods yield for achieving their minimum loss? In Figure 5 (right) we can see that 4C fully exploited the fact that no limitations were imposed on the number of clusters. ORCLUS in contrast was capable to achieve low reconstruction losses mostly with two clusters. However there are number of dimensions in which ORCLUS exploited the number of clusters as well, achieving low reconstruction errors by yielding 12 to 15 clusters. The autoencoder has been set by default to 1 regarding the number of clusters, since it learns a single manifold. To our surprise LMCLUS yielded throughout all numbers of dimensions one single cluster. Changing the hyperparameter settings of the LMCLUS algorithms did not result in an increased number. Further investigations may be required to elucidate the reasons for this behavior, which is beyond the scope of this work.

D. Experiment 4: Regularization Sensitivity

So far we have observed the effects of having no regularization at all on the evaluation as well as the effects on the reconstruction loss by imposing limitations on the dimensionality. In this experiment we ask: How sensitive are the results with respect to the regularization terms, specifically with regards to their hyperparameters α and β? Asking what are the aspects which are important for an evaluation measure,

[6] blog.keras.io/building-autoencoders-in-keras.html

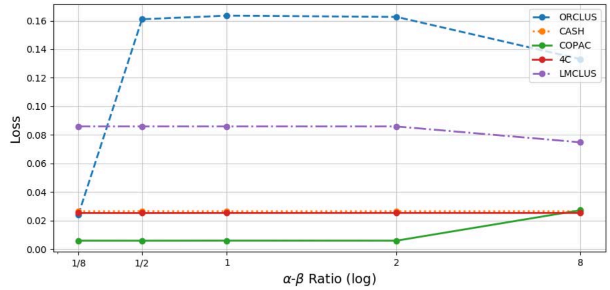

Fig. 6. Sensitivity of the loss w.r.t the hyperparameters α and β of the regularization terms on the iris dataset.

we demand that it should be insensitive towards α and β. Connected to this fact, we ask: Which impacts can we observe if we weight both of them equally? What can be observed if we set α to a high value and β to a low one, and vice-versa? For this purpose in this experiment all five subspace clustering algorithms were tested on the iris dataset with different (α, β)- settings, namely $\{(\frac{1}{9}, \frac{8}{9}), (\frac{1}{3}, \frac{2}{3}), (\frac{1}{2}, \frac{1}{2}), (\frac{2}{3}, \frac{1}{3}), (\frac{8}{9}, \frac{1}{9})\}$. The settings of the ratios of α and β have been chosen in such a way that the sum satisfies $\alpha + \beta = 1$. In Figure 6 we have on the horizontal axis the ratio α/β, and on the vertical axis the computed loss. The results reveal that for different ratios of α and β the loss remains mostly the same. Which confirms that SRE is robust with respect to different hyperparameter settings. An exception to the rule poses ORCLUS which seems sensitive on the extreme cases $\frac{1}{8}$ and 8. This is due to the fact that at $\frac{1}{8}$ more dimensions are permitted, due to a lower α and thus a lower penalty for the number of dimensions. In the following ratios ORCLUS detected optimum settings which have only one dimension but two clusters. Another aspect which this experiment reveals is that overall COPAC yields a lower reconstruction loss, followed by 4C and CASH. LMCLUS performs as the second worst algorithm and ORCLUS as the worst.

V. Conclusion

In conclusion we developed a new internal evaluation measurement for subspace clustering. We translated approaches from autoencoders such as the approximation of non-linear manifolds by a function to the concept of subspace clusters, leading to piecewise linear approximations based on PCAs of single subspace clusters. Our evaluation measure is the first one for arbitrary oriented subspace clustering and fulfills important properties like invariance regarding number of clusters as well as number of dimensions. The experiments show that SRE takes aspects into account which are neglected by external evaluation measures. It also differs from existing internal evaluation measures, since it considers aspects as reconstruction error and model complexity in terms of subspace dimensionality and number of clusters. In future work we want to investigate the two regularization parameters α and β in more detail. Further we want to enhance the autoencoding view to a data compression view, by also utilizing the Minimum Description Length (MDL) for evaluating the reconstruction loss and model complexity. We envision that this work may pave the path for a different understanding and perspective on the arbitrarily oriented subspace clustering problem as well give rise to novel internal evaluation measures.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content.

References

[1] H.-P. Kriegel, P. Kröger, and A. Zimek, "Subspace clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 351–364, 2012.

[2] E. Müller, S. Günnemann, I. Assent, and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 1270–1281, 2009.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Univ San Diego. Inst for Cognitive Science, Tech. Rep., 1985.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[6] D. Moulavi, P. A. Jaskowiak, R. J. Campello, A. Zimek, and J. Sander, "Density-based clustering validation," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 839–847.

[7] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[8] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[9] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[10] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[11] F. Höppner and M. Jahnke, "Holistic assessment of structure discovery capabilities of clustering algorithms," *ECML-PKDD*, 2019.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[13] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural computation*, vol. 9, no. 7, pp. 1493–1516, 1997.

[14] C. C. Aggarwal and P. S. Yu, *Finding generalized projected clusters in high dimensional spaces*. ACM, 2000, vol. 29.

[15] E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek, "Global correlation clustering based on the hough transform," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 1, no. 3, pp. 111–127, 2008.

[16] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[17] C. Böhm, K. Kailing, P. Kröger, and A. Zimek, "Computing clusters of correlation connected objects," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 455–466.

[18] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek, "Robust, complete, and efficient correlation clustering," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 413–418.

[19] R. Haralick and R. Harpaz, "Linear manifold clustering," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2005, pp. 132–141.

[20] E. Schubert and A. Zimek, "ELKI: A large open-source library for data analysis - ELKI release 0.7.5 "heidelberg"," *CoRR*, vol. abs/1902.03616, 2019.