Pattern Recognition 104 (2020) 107328

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Clustering quality metrics for subspace clustering

John Lipor[a,*], Laura Balzano[b]

[a] Department of Electrical & Computer Engineering, Portland State University, 1900 SW 4th Avenue, Suite 160-11, Portland, OR 97201, United States
[b] Department of Electrical & Computer Engineering, University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109, United States

ARTICLE INFO

Article history:
Received 22 February 2019
Revised 18 December 2019
Accepted 8 March 2020
Available online 13 March 2020

Keywords:
Subspace clustering
Clustering validation
Union of subspaces

ABSTRACT

We study the problem of clustering validation, i.e., clustering evaluation *without* knowledge of ground-truth labels, for the increasingly-popular framework known as subspace clustering. Existing clustering quality metrics (CQMs) rely heavily on a notion of distance between points, but common metrics fail to capture the geometry of subspace clustering. We propose a novel point-to-point pseudometric for points lying on a union of subspaces and show how this allows for the application of existing CQMs to the subspace clustering problem. We provide theoretical and empirical justification for the proposed point-to-point distance, and then demonstrate on a number of common benchmark datasets that our proposed methods generally outperform existing graph-based CQMs in terms of choosing the best clustering and the number of clusters.

1. Introduction

Clustering has long been one of the most fundamental tools for data exploration, and from the start researchers have studied how to determine the quality of a clustering output in order to choose parameters and compare algorithms. In contrast to the supervised learning setting, clustering problems do not provide any labeled data that can be used as a "hold-out" set for cross-validation. The problem of clustering quality has been widely studied for the general clustering problem [1–4]. However, existing methods are not applicable to the *subspace clustering* problem [5], a more modern and widely applicable clustering framework in which the clusters also have low-dimensional structure.

The key ideas in the clustering quality literature are those of intra-cluster *cohesion* and inter-cluster *dispersion*. These notions are defined fundamentally based on some distance metric chosen appropriately for the application. This distance metric is applied between points in the dataset or between points and cluster centers, where the centers are of the same dimension as the data points.

The subspace clustering problem can be formulated as a generalization of PCA, where we seek a collection of low-dimensional subspaces that best fits our data; this is known as the Union of Subspaces (UoS) model. We may think of these subspaces as the cluster centers, in which case there is a natural notion of point-to-center and center-to-center distances. However, quantifying point-to-point distance becomes problematic. Intuitively, we wish to define a metric $d(\cdot, \cdot)$ such that the distance between points in the same subspace is small, whereas points on orthogonal subspaces should have maximum distance. For example, antipodal points always lie in a one-dimensional subspace, and we therefore desire $d(x, -x) = 0$. However, this property cannot be achieved by existing (pseudo) metrics such as the Mahalanobis distance.

In this work, we present what is, to the best of our knowledge, the first approach to internal clustering validation for the UoS model. We propose a novel pseudometric for points lying on a union of subspaces, as well as several clustering quality metrics so that the output of subspace clustering algorithms can be tuned and fairly compared on unsupervised datasets.

2. Problem formulation & related work

Consider a collection of N unit-norm points $\mathcal{X} = \{x_1, \dots, x_N\}$ in ambient space $\mathbb{R}^D$, and let $X \in \mathbb{R}^{D\times N}$ denote the matrix whose columns are the elements of $\mathcal{X}$. We define a *K-clustering* of $\mathcal{X}$ to be a partition of $\mathcal{X}$ into K disjoint sets $\mathcal{C} = \{c_1, \dots, c_K\}$, where we assume $1 < K < N$ to avoid trivial clustering. Let $U_1, \dots, U_K$ denote orthonormal bases for K subspaces $\mathcal{S}_1, \dots, \mathcal{S}_K$ obtained by performing PCA on the points in clusters $c_1, \dots, c_K$, and let $\mathcal{D} = \{d_1, \dots, d_K\}$ be the set of dimensions of these subspaces. An example of data lying in two 2-dimensional subspaces is shown in Fig. 1, where the points from subjects 5 and 23 of the Extended Yale Face Database B [6] are shown after projecting onto their first three principal components via robust PCA [7].

[*] Corresponding author.
E-mail addresses: lipor@pdx.edu (J. Lipor), girasole@umich.edu (L. Balzano).

https://doi.org/10.1016/j.patcog.2020.107328
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

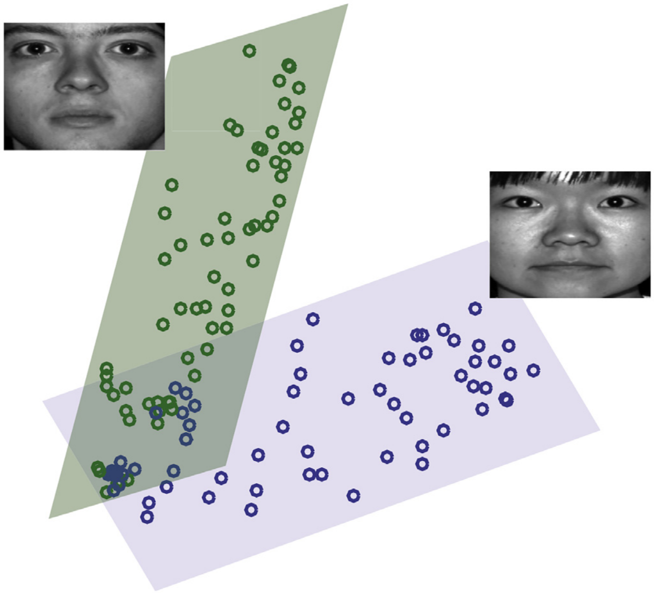

Fig. 1. Data from the Extended Yale Face Database B is known to lie in a union of low-rank subspaces. Images from subjects 5 and 23 projected onto first three principal components are shown.

2.1. Subspace clustering

Subspace clustering algorithms seek to partition $\mathcal{X}$ into K clusters such that the data in each cluster lies near a low-dimensional linear or affine subspace. This is done in an unsupervised manner, *i.e., without* knowledge of the subspaces themselves. This model has applications ranging from structure from motion to image and handwritten character recognition [8–12].

To accomplish this task, researchers leverage a variety of properties of data belonging to a union of subspaces. Perhaps the most popular of these is the *self-expressive* property, which informally states that points can be most efficiently represented as a linear combination of other points lying in the same subspace. Researchers utilize this property by solving sparse regression problems of the form

$$\min_{Z} \|X - XZ\|_F^2 + \lambda \|Z\|$$
$$\text{subject to} \quad \text{diag}(Z) = 0,$$

where $\|Z\|$ is the ℓ_1-norm in Sparse Subspace Clustering (SSC) [8], the nuclear norm in Low-Rank Representation (which omits the constraint on Z) [13,14], and may include a combination of other norms to account for noisy data or outliers. An affinity/similarity matrix is then obtained as $|Z| + |Z|^T$, after which spectral clustering is performed to obtain the clusters. SSC and its variants thus require the selection of at least one hyperparameter λ, as well as a thresholding parameter in the case of the Alternating Direction Method of Multipliers (ADMM) implementation of SSC. In [15], the authors present a range of allowable values for λ to guarantee correct clustering, but this range is based on data parameters such as the inradius of each cluster, which cannot be known a priori, and the result does not apply when a penalty for sparse outliers is included.

An alternative approach to subspace clustering is that of the Thresholded Subspace Clustering (TSC) algorithm [16], which leverages the fact that points within the same subspace have large inner product (on average) relative to points in different subspaces. TSC is the simplest of all subspace clustering algorithms and proceeds by forming the matrix $|X^T X|$ and thresholding each row and column so that all but the top q entries are set to zero. Methods of selecting this threshold are provided in [16,17], but these rely heavily on strict assumptions on the data (*e.g.,* that the data are generated uniformly at random from the intersection of the unit sphere and the subspace). Real-world datasets often violate these assumptions, and in practice, the clustering of lowest error may not result from selecting the threshold within the proposed ranges.

One further approach to subspace clustering is based on the K-subspaces (KSS) algorithm [18–20], a generalization of K-means that seeks to minimize the sum of squared distances from points to subspaces through alternating minimization. While KSS is computationally efficient and only requires the selection of a single tuning parameter (the subspace dimension), its performance on benchmark datasets is known to lag behind that of self-expressive methods. Recently, in [21], the authors show that incorporating robust subspace estimation via the Coherence Pursuit algorithm [22] can significantly improve the performance of KSS, though this requires the selection of an additional tuning parameter. Another recent approach to improving KSS is that of the Ensemble K-subspaces (EKSS) algorithm [23], which combines the results of numerous KSS instances via the evidence accumulation framework [24] to achieve superior empirical performance and strong theoretical guarantees. Like TSC, EKSS builds an affinity matrix and then thresholds this matrix before applying spectral clustering. In this case, both the subspace dimension and the threshold parameter have significant impact on performance.

2.2. Internal clustering validation

The trend illustrated above exists for all subspace clustering algorithms; hyperparameters, thresholds, and other variables must be tuned in order to achieve strong performance. Hence, in order to provide a principled, interpretable method for practitioners to utilize these methods, we must define some measure of “goodness of fit” for subspace clustering. The problem of evaluating clustering results in the absence of ground truth has been studied for decades in the general clustering community and is known as *internal clustering validation* [25]. It has applications ranging from image segmentation [26] to community analysis in graphs [2] to clustering acoustic signals [27,28], among many others.

In contrast to *external* clustering validation methods [29,30], internal methods, known as *clustering quality metrics* (CQMs) seek to measure clustering quality without access to ground-truth labels. Such measures are designed to capture the “natural” goals of clustering, the chief being that points within clusters should have high similarity or *cohesion*, while points in different clusters should have low similarity or high *dispersion*.

Early examples of internal CQMs include the Dunn index [31], Davies-Bouldin index [32], and the Silhouette index [33]. The Dunn index is the ratio of dispersion to cohesion, where cohesion is measured using the cluster diameter and dispersion using the minimum distance between points in different clusters. A number of variations on this index are proposed throughout the literature and defined in [1], one of which we consider in this work (see Section 4). The Davies-Bouldin index measures cohesion using the mean distance from points to centroids and dispersion as the distance between centroids. The Silhouette index is based on the (normalized) difference between average intra-cluster pairwise distance and average inter-cluster pairwise distance. These and other more recent CQMs are studied extensively in the surveys [1,25], with the Dunn, Davies-Bouldin, Silhouette, and Calinsky-Harabasz [34] indices being among the top performers. A comprehensive list of CQMs can be found in [3]. One major drawback to these methods for application to subspace clustering is that they often rely on the pairwise distance between points. For points lying on a low-rank subspace, pairwise Euclidean distance is not indicative. For example, the points x and $-x$ clearly lie on the same one-dimensional subspace but may be arbitrarily far apart. Further, the notion of centroids must be revised before these methods can be applied.

The above CQMs are designed for traditional distance-based clustering algorithms such as *K*-means. However, many modern clustering algorithms rely only on the entries of an adjacency matrix, whose (i, j)th entry $A_{ij} \in \{0, 1\}$ denotes whether two items in the set are "connected," or an affinity matrix, whose entries $A_{ij} \geq 0$ denote the strength of that connection. Such algorithms are referred to as *graph-based* methods and include single linkage, other hierarchical methods, and spectral clustering (see [35, Ch. 14] for a description of these methods). Empirical graph clustering quality measures have existed for a number of years, and several comparisons of such metrics exist [2,36,37], with no CQM consistently outperforming others when a large number of datasets are considered. Two of the most widely used CQMs are *coverage* [38] and *modularity* [39]. The former is defined as the ratio of intra-cluster connectivity and total connectivity in the graph, and the latter measures the strength of intra-cluster connectivity compared to the average connectivity of each cluster. Since nearly all subspace clustering algorithms produce an affinity matrix, graph-based CQMs present a reasonable off-the-shelf approach to parameter selection. However, these suffer from known drawbacks such as favoring sparse affinity matrices [36]. Further, they ignore knowledge of the underlying UoS structure in the data, which has been shown to provide significant benefits in other clustering contexts [40].

In [41], the authors argue that lack of interpretability plagues modern clustering algorithms and accounts for the widespread use of *K*-means in spite of its known shortcomings. Subspace clustering falls victim to a similar problem, as relatively few people understand the concept of a union of subspaces, perhaps accounting for its relative anonymity among practitioners.[1] For this paradigm to gain popularity, the ability to select parameters is paramount, and hence the need to compare clusterings resulting from different subspace clustering algorithms is an important contribution that has received no attention to this point.

3. Metrics for unions of subspaces

As stated above, we wish to design internal CQMs that take into account the low-dimensional intrinsic structure of the data, rather than relying solely on the elements of the affinity matrix formed by an algorithm. One approach to leveraging this geometry is to develop analogs to existing measures such as the Davies-Bouldin or Dunn index. These and other CQMs rely on three key distances: (1) point-to-centroid, (2) centroid-to-centroid, and (3) point-to-point. The first two have natural interpretations under the UoS model, which we state in Sections 3.1 and 3.2. In Section 3.3, we propose a novel notion of pairwise distance for points lying on a union of K subspaces and examine its properties. We overload the term $\text{dist}(\,\cdot\,,\;\cdot\,)$ in this and following sections to represent all three distances, with the definition being clear based on type.

3.1. Point-to-subspace distance

A widely-used notion of point-to-centroid distance under the UoS model is that from a point to a subspace, *i.e.,*

$$\text{dist}(x, \mathcal{S}) = \left\| x - UU^T x \right\|_2, \tag{1}$$

where $U \in \mathbb{R}^{D \times d}$ is an orthonormal basis for the subspace $\mathcal{S}$. This notion of distance is used in the KSS algorithm.

3.2. Subspace-to-subspace distance

Recall that under the UoS model, the subspaces take the place of centroids. Hence, it is reasonable to assume that the centroid-to-centroid distance should be replaced by the distance between points on the Grassmannian. Two key problems arise with this approach. First, there are multiple proper metrics on the Grassmannian, including the sine of the maximum principal angle between subspaces (see [42, Section 6.4.3] for a definition of principal angles) and the ℓ_2-norm of principal angles between subspaces, which corresponds to the geodesic distance [43]. While these two distances result in the same topological structure, they capture different properties of the subspaces being considered. More importantly, these distances are only defined for subspaces of the same dimension. Since this assumption is not a requirement of our data model or any recent subspace clustering algorithm, we seek a notion of subspace-to-subspace distance that can handle subspaces of varying dimension. A notion of nearness between subspaces, known as the *subspace affinity*, appears frequently in the analysis of various algorithms [8,16,44]. The subspace affinity is formally defined as

$$\text{aff}(\mathcal{S}_i, \mathcal{S}_j) = \frac{1}{\sqrt{d_i \wedge d_j}} \left\| U_i^T U_j \right\|_F \tag{2}$$

$$= \sqrt{\frac{1}{d_i \wedge d_j} \sum_{l=1}^{d_i \wedge d_j} \cos^2 \theta_l}, \tag{3}$$

where $a \wedge b$ denotes the minimum between a and b, U_i (U_j) is an orthonormal basis for $\mathcal{S}_i$ $(\mathcal{S}_j)$, $\| \cdot \|_F$ denotes the Frobenius norm, and θ_l denotes the lth principal angle between the subspaces. The subspace affinity is between 0 and 1, with $\text{aff}(\mathcal{S}_i, \mathcal{S}_j) = 0$ indicating the subspaces are orthogonal and $\text{aff}(\mathcal{S}_i, \mathcal{S}_j) = 1$ if and only if $\mathcal{S}_i = \mathcal{S}_j$. From (3), we see that the subspace affinity captures a notion of nearness between subspaces that considers all principal angles, rather than only the maximum. Further, it has been shown through the analysis of various algorithms to be a key parameter in measuring the difficulty of the subspace clustering problem. For these reasons, we propose the use of the following pairwise distance between subspaces

$$\text{dist}(\mathcal{S}_i, \mathcal{S}_j) = \sqrt{1 - \text{aff}^2(\mathcal{S}_i, \mathcal{S}_j)}$$

$$= \sqrt{\frac{1}{d_i \wedge d_j} \sum_{l=1}^{d_i \wedge d_j} \sin^2 \theta_l}. \tag{4}$$

The above is closely related to the chordal distance considered in the subspace packing problem [45]. In the case where $d_i = d_j = d$, it is easy to see that (4) is a proper metric by noting that

$$\begin{aligned}
\text{dist}^2(\mathcal{S}_i, \mathcal{S}_j) &= 1 - \frac{1}{d} \left\| U_i^T U_j \right\|_F^2 \\
&= \frac{1}{2d} \left(\|U_i\|_F^2 + \left\| U_j \right\|_F^2 - 2 \left\| U_i^T U_j \right\|_F^2 \right) \\
&= \frac{1}{2} \text{tr}\left(U_i U_i^T + U_j U_j^T - 2 U_i U_i^T U_j U_j^T \right) \\
&= \frac{1}{2} \left\| U_i U_i^T - U_j U_j^T \right\|_F^2.
\end{aligned}$$

3.3. Point-to-point distance

While the point-to-subspace and subspace-to-subspace distances are straightforward to define in terms of familiar quantities, to the best of our knowledge, there does not exist a useful notion of pairwise distances between points lying on a union of

[1] For example, there is not a single subspace clustering algorithm implemented in the widely-used scikit-learn Python package.

subspaces. We now introduce a novel notion of distance between points for this setting that satisfies a number of “common sense” properties. Assume we are given a clustering $\mathcal{C} = \{c_1, \ldots, c_K\}$ with bases $U_1, \ldots, U_K$ corresponding to each cluster. Let P_x denote the orthogonal projection matrix onto the subspace corresponding to the cluster containing the point x, and let $P_x^\perp = I - P_x$. Our proposed point-to-point distance is

$$\mathrm{dist}(x, y) = \frac{1}{2}\Big(x^T P_x^\perp x + x^T P_y^\perp x + y^T P_x^\perp y + y^T P_y^\perp y \\ -2\left|x^T P_x^\perp y\right| - 2\left|x^T P_y^\perp y\right|\Big)^{1/2}. \tag{5}$$

It is easily verified that (5) is a pseudometric taking values between 0 and 1. Further, the distance can be efficiently computed in $O(N^2 + D^2)$ time ($O(N^2)$ if the subspace bases are provided, as with KSS and its variants). We now provide intuition for this distance with a number of observations.

First note that without the projection matrices $P_x^\perp$ and $P_y^\perp$, the proposed distance becomes $\sqrt{1 - |x^T y|}$, indicating that the distance between points is a function of their absolute inner product. The absolute inner product has been utilized widely in subspace clustering methods [16,23,46] and is therefore a useful feature; however, we argue that even orthogonal points should have small distance if they are believed to lie in the same subspace. On the other hand, note that if we drop the absolute value on the last two terms of (5), the distance becomes a Mahalanobis distance with covariance matrix $P_x^\perp + P_y^\perp$. However, in this case, antipodal points do not necessarily have distance zero as desired.

The proposed distance overcomes both of these issues. First, antipodal points always have distance zero due to the final two terms of (5). Second, if $P_x = P_y$ and $x = P_x x$ and $y = P_y y$, then $d(x, y) = 0$. In other words, if x and y are assigned to the same cluster and the subspaces are estimated perfectly, then $d(x, y) = 0$ even when x and y are orthogonal. The maximum value of (5) is 1, which occurs when x and y are orthogonal to each other and each is orthogonal to both the subspaces spanned by P_x and P_y. This instance may occur if the orthogonal points x and y are assigned to the same cluster but neither lies in the subspace corresponding to that cluster, i.e., $P_x = P_y =: \bar{P}$ and $\bar{P}^\perp x = x$ and $\bar{P}^\perp y = y$.

To further motivate the proposed distance, consider the case where the subspaces are perfectly modeled, which yields

$$d(x, y) = \frac{1}{2}\left(\left\|P_y^\perp x\right\|_2^2 + \left\|P_x^\perp y\right\|_2^2\right)^{1/2}.$$

The above is small when each point lies near to the opposing point’s subspace, indicating that points near the intersection of subspaces will have small distance from each other. Further, consider the case where the points are drawn randomly from their respective subspaces, taking $x \sim Ua$ and $y \sim Vb$, where $P_x = UU^T$ and $P_y = VV^T$ and $a, b \sim \mathrm{Unif}(\mathbb{S}^{d-1})$. In this case, we have

$$\mathbb{E}\left[\mathrm{dist}^2(x, y)\right] = \frac{1}{2}\left(1 - \frac{1}{d}\left\|U^T V\right\|_F^2\right) = \frac{1}{2}\mathrm{dist}^2(\mathcal{S}_x, \mathcal{S}_y) \tag{6}$$

indicating that randomly drawn points will have small distance when their corresponding subspaces have small distance from each other. While we do not analyze the case of imperfect subspace modeling here, our empirical results (Section 5.2) indicate that the average pairwise intra-cluster distance remains smaller than the average inter-cluster distance even under significant errors in the subspace modeling.

Finally, consider the case where many points are drawn from a subspace but corrupted by noise. Under this setting, the proposed distance indicates the level of noise on a given point, as points that are heavily corrupted will have large distance from those that are nearer to the true subspace. We illustrate this final scenario in Fig. 2, which shows the arrangement of points from the Extended Yale Face Database B. These points are known to lie near a union of 9-dimensional subspaces, each corresponding to images of a different subject. We take $c_1, \ldots, c_K$ to correspond to the true clusters and find the best 9-dimensional basis for each cluster in order to compute the distance between points in clusters 13, 26, and 38. Fig. 2 illustrates the arrangement of after embedding the points into $\mathbb{R}^2$ using multidimensional scaling (MDS) [47] on the proposed pairwise distance. Analyzing the original images shows that the tightly-grouped points correspond to images with low amounts of shadow, while those farther from the cluster centroids correspond to heavily-shadowed images. In fact, we see that images with shadow on the left half of the face form one group of outliers, and likewise for images with shadow on the right half.

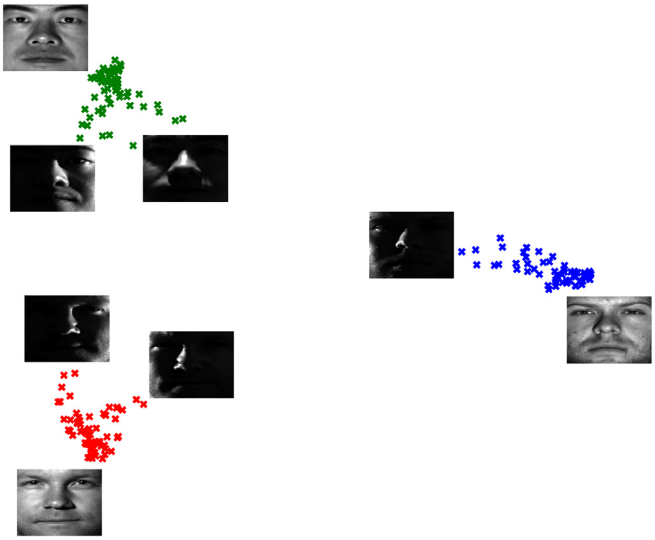

Fig. 2. Two-dimensional embedding of points in Extended Yale Face Database B, subjects 13, 26, and 38 using multidimensional scaling on the proposed point-to-point distance. The proposed distance provides an indication of which points lie near the estimated subspace and groups outliers with similar forms of shadow.

4. Internal validation measures for subspace clustering

Armed with the notions of point-to-subspace, subspace-to-subspace, and point-to-point distances defined in the previous section, we are now ready to define a variety of CQMs for the problem of subspace clustering. We define two quality measures based on the KSS cost function as well as three analogs of existing CQMs adapted to the UoS model. All CQMs require three inputs: the data $\mathcal{X} = \{x_1, \ldots, x_N\}$, the estimated clusters $\mathcal{C} = \{c_1, \ldots, c_K\}$, and a set of subspace dimensions $\mathcal{D} = \{d_1, \ldots, d_K\}$. We define $\mathcal{S}_k$ to be the d_k-dimensional subspace obtained by performing PCA on the points in cluster c_k.

The first CQM we consider is that of the KSS cost, which is defined as

$$m_{KSS}(\mathcal{X}, \mathcal{C}, \mathcal{D}) = \frac{1}{N}\sum_{k=1}^{K}\sum_{x_i \in c_k}\mathrm{dist}^2(x_i, \mathcal{S}_k).$$

The KSS cost is suggested as a method for selecting among the best of several runs of KSS in [9]. However, it is not appropriate for attempting to determine the number of subspaces or the underlying subspace dimensions, since it is a monotonically decreasing function of both of these parameters. In the language of existing CQMs, the KSS cost is a measure of cohesion only, rather than a balance between cohesion and dispersion. Existing approaches such as the gap statistic [48] attempt to quantify an “elbow” in the within-cluster cohesion (e.g., as computed by m_{KSS}) in order to select the number of clusters. However, the gap statistic requires the additional computation of the cohesion for a reference dataset, increasing computational complexity. An alternative method based on examining the singular values of the graph Laplacian was proposed in [13]. However, this method requires selecting yet another tuning parameter and is only applicable to algorithms that rely on an affinity matrix. Further, our empirical results on selecting the number of clusters indicated that both the gap statistic and the Laplacian-based method failed to reliably determine the correct number of clusters across multiple algorithms, even on synthetic data. We therefore propose the following CQM, which we refer to as Normalized KSS Cost (NKSS)

$$m_{NKSS}(\mathcal{X}, \mathcal{C}, \mathcal{D}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{x_i \in c_k} \frac{\mathrm{dist}^2(x_i, \mathcal{S}_k)}{\min_{j \neq k} \mathrm{dist}(\mathcal{S}_j, \mathcal{S}_k)^2}.$$

For both m_{KSS} and m_{NKSS}, smaller values correspond to better clusterings. In the case where all subspaces are orthogonal, we have $\mathrm{dist}(\mathcal{S}_j, \mathcal{S}_k) = 1$ for all j, k, and $m_{NKSS} = m_{KSS}$. However, as the subspace dimension increases, the subspaces “fill up the space,” incurring a penalty. This is made clear by noting that for two d-dimensional subspaces drawn uniformly at random from the Grassmannian, $\mathrm{dist}^2(\mathcal{S}_i, \mathcal{S}_j) \approx \frac{D-d}{D}$ [21, Lemma 3]. Similarly, increasing the number of subspaces decreases the expected minimum pairwise distance between subspaces, increasing the normalization penalty.

We also consider three existing CQMs that rely on the distances defined in the previous section. Since there are numerous existing CQMs based on pairwise distances between points and centroids, we choose three of the best performers in the extensive survey [1].[2] The first is a variant of the Dunn Index (DI) [31], referred to as Generalized Dunn Index 41 (gD41) in [1], which measures cohesion using the maximum cluster diameter and dispersion using the minimum distance between any pair of subspaces.

$$m_{DI}(\mathcal{X}, \mathcal{C}, \mathcal{D}) = \frac{\min_{j \neq k} \mathrm{dist}(\mathcal{S}_j, \mathcal{S}_k)}{\max_{k \in [K]} \max_{x_i, x_j \in c_k} \mathrm{dist}(x_i, x_j)}.$$

Higher values correspond to better clusterings for the Dunn Index.

Another popular CQM that is shown to perform well in the survey [1] is the Silhouette Index (SI) [33], which measures cohesion using the mean pairwise distance between points in the same cluster and dispersion as the smallest average distance from a point to all points in another cluster.

$$m_{SI}(\mathcal{X}, \mathcal{C}, \mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{x_i \in c_k} \frac{b(i) - a(i)}{\max(a(i), b(i))},$$

where

$$a(i) = \frac{1}{N_k - 1} \sum_{x_j \in c_k x_j \neq x_i} \mathrm{dist}(x_i, x_j),$$

and

$$b(i) = \min_{l \neq k} \frac{1}{N_l} \sum_{x_j \in c_l} \mathrm{dist}(x_i, x_j).$$

Higher values correspond to better clusterings for the Silhouette Index.

Finally, we consider the Calinski-Harabasz (CH) index [34], which measures cohesion using the average distance from points to their respective subspaces and dispersion using the average distance from each subspace to the best subspace of the same dimension for the entire dataset,

$$m_{CH}(\mathcal{X}, \mathcal{C}, \mathcal{D}) = \frac{N - K}{K - 1} \frac{\sum_{k=1}^{K} N_k \mathrm{dist}(\mathcal{S}_k, \mathcal{S}_{\mathcal{X}})}{\sum_{k=1}^{K} \sum_{x_i \in c_k} \mathrm{dist}(x_i, \mathcal{S}_k)},$$

where $\mathcal{S}_{\mathcal{X}}$ denotes the subspace spanned by the entire dataset. Higher values correspond to better clusterings for the Calinski-Harabasz Index.

5. Empirical results

In this section, we compare the proposed CQMs to existing CQMs on a variety of synthetic and real datasets. We first evaluate the proposed point-to-point distance to show its empirical benefits over other existing notions of distance. We then evaluate the ability of each CQM to determine the true number of clusters on synthetic data drawn from a UoS. Finally, we evaluate the performance of the CQMs on three common benchmark datasets in the subspace clustering literature.

Along with the CQMs described in Section 4, we also consider four graph-based CQMs that are shown to perform well in the surveys [36,37,49]: coverage [38], modularity [39], permeance [50], and communitude [51].

When evaluating the various CQMs, we consider six subspace clustering algorithms: SSC [8], SSC-OMP [52], EnSC [53], GSC [44], TSC [16], and EKSS [23]. These algorithms are shown to be scalable, theoretically justified, and perform well on benchmark datasets. Further, they represent a wide range of tuning parameters, including optimization hyperparameters, thresholding parameters, subspace dimension, and number of nearest neighbors. Each algorithm is run for between 50 and 200 hyperparameter configurations (depending on number of tuning parameters and computation time) with parameters chosen linearly from these ranges. See Table 1 for a summary of these parameters and their considered ranges. We evaluate the proposed CQMs on synthetic data as well as three of the most common benchmark datasets in the subspace clustering literature: the Hopkins-155 dataset [54], the cropped Extended Yale Face Database B [6,55], and the COIL-20 [56] object database, with preprocessing steps performed as in [23].

Recall that our proposed CQMs and subspace-based metrics require the underlying subspace dimensions as input. First, we note that allowing each cluster to have a different subspace dimension results in an explosion of the parameter space. Instead, it is common to set all clusters to have dimension equal to some maximum estimated subspace dimension during clustering and then estimate individual subspace dimensions once the clusters have been identified. For GSC and EKSS, which take subspace dimension as an input parameter, we use the same (maximum) subspace dimension during both clustering and evaluation with the CQMs, allowing us to perform model selection on subspace dimension. For algorithms such as SSC and its variants, which do not require subspace dimension as an input, we select the CQM subspace dimension based on accepted values from the literature, taking $d = 9$ for Yale and COIL and $d = 3$ for Hopkins, as in [40]. Although omitted due to lack of space, our initial empirical investigation suggests that our proposed subspace-based CQMs have roughly equal performance over a wide range of subspace dimensions, and the automatic selection of this parameter (e.g., via explained variance or Bayesian methods [57]) is an interesting topic for future work.

5.1. Evaluation metric

A variety of metrics for evaluating and comparing CQMs are proposed throughout the literature. Often, CQMs are used to select a parameter with a true value, such as the number of clusters, in which case it is common to evaluate the ability to select this value correctly. Alternatively, Spearman’s rank correlation coefficient [58] may be used to measure how well the ranking of clusterings according to a given CQM compares to an external validation measure, such as the Jaccard coefficient [29] or Adjusted Rand index [30]. We use a variation on this approach.

[2] We experimented with thirteen total existing CQMs studied in [1] and chose the top three performers to report here.

Table 1
Subspace clustering algorithms considered and their corresponding parameters with ranges considered.

Algorithm	Parameter 1	Description	Parameter 2	Description
SSC-ADMM	$\rho \in [0.1, 10]$	thresholding parameter	$\alpha \in [5, 2000]$	hyperparameter
SSC-OMP	$k_{\max} \in \{1, 50\}$	maximum coefficients	–	–
EnSC	$\lambda \in [0.01, 0.99]$	hyperparameter	$\alpha \in [3, 100]$	hyperparameter
GSC	$k_{\max} \in \{1, 20\}$	# neighbors	$d \in \{1, 20\}$	subspace dimension
TSC	$q \in \{2, 100\} \cup \{N\}$	thresholding parameter	–	–
EKSS	$q \in \{2, 100\} \cup \{N\}$	thresholding parameter	$d \in \{1, 20\}$	subspace dimension

First, the most widely used *external* validation measure for subspace clustering is the clustering error, which is computed by matching the true labels and the labels output by a given clustering algorithm,

$$\varepsilon = 100\left(1 - \max_{\pi} \frac{1}{N} \sum_{i,j} Q^{\text{out}}_{\pi(i)j} Q^{\text{true}}_{ij}\right),$$

where π is any permutation of the cluster labels, and Q^{out} and Q^{true} are the output and ground-truth labelings of the data, respectively, where the (i, j)th entry is one if point j belongs to cluster i and is zero otherwise. We define the *oracle error* as the lowest clustering error among all parameter configurations considered (see Table 1) and emphasize that this error can only be determined in light of the ground-truth labels. Hence, the goal of any CQM is to discover the parameter configuration(s) that result in the oracle error *without* knowledge of the true labels.

As a validation metric for the various CQMs, one could compute the Spearman correlation between the clusterings sorted according to true error (smallest to largest) and best clustering according to each CQM, as is done in [49]. However, it is less important that the order of clusterings returned by a CQM match the oracle order exactly than that the top few clusterings (according to each CQM) be ones of low error. For this reason, we propose the following *ratio of area under the curves (R-AUC)* metric. Consider a set of p clusterings—i.e., outputs of a clustering algorithm on p different configurations of hyperparameters—to be evaluated by a CQM m, and let $\varepsilon_m \in \mathbb{R}^p$ be the vector of clustering errors resulting from the p clusterings sorted from best to worst according to m. Further, let ε_* be the vector of errors sorted according to the true (oracle) clustering error. Let A_m denote the area under the normalized cumulative sum of ε_m, i.e.,

$$A_m = \frac{1}{\sum_{i=1}^{p} \varepsilon_m(i)} \sum_{j=1}^{p} \sum_{i=1}^{j} \varepsilon_m(i),$$

where $\varepsilon_m(j)$ denotes the jth element of the vector ε_m. Let A_* be similarly defined, and note that smaller values of A_m correspond to better orderings of the clusterings. The R-AUC is then

$$\text{R-AUC} = \frac{A_m}{A_*}. \tag{7}$$

The R-AUC ≥ 1, with a lower ratio implying better performance.

5.2. Empirical evaluation of proposed point-to-point distance

We begin by demonstrating that the proposed point-to-point distance (5) outperforms existing distances in terms of providing an embedding by which points lying on a UoS are well separated. Fig. 3 compares the resulting two-dimensional embedding of points from a union of five 7-dimensional subspaces of $\mathbb{R}^{100}$ using t-SNE [59] on the pairwise distance matrix formed using the Euclidean distance, the inner product-based distance $\text{dist}(x, y) = \left(1 - \left|x^T y\right|\right)^{1/2}$, and the proposed distance (5), where we assume the subspaces are modeled perfectly. The figure demonstrates that the proposed metric is the only one that results in points that are clearly separated according to subspace membership.

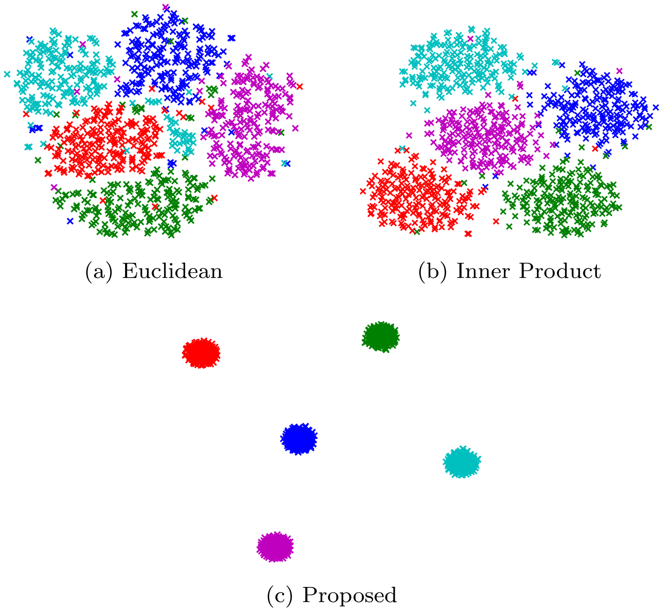
(a) Euclidean
(b) Inner Product
(c) Proposed

Fig. 3. t-SNE embedding of points drawn from a union of five 7-dimensional subspaces of $\mathbb{R}^{100}$ using (a) Euclidean, (b) inner product, (c) proposed pairwise distance.

As stated above, the separation shown in Fig. 3 is obtained assuming the subspaces are modeled perfectly, which is unlikely to be the case in practice. We now study the impact of mismodeling the subspaces by considering the *distance gap* between inter-cluster and intra-cluster distances. In general, a CQM will want the inter-cluster distances to be large and intra-cluster relatively smaller. Let the average difference between these inter-cluster and intra-cluster distances be called the “distance gap.” In Fig. 4, we display the distance gap as a function of both the estimation error in the subspaces (i.e., in P_x and P_y) and the distance between the true subspaces the points are drawn from. We consider the case of two ten-dimensional subspaces of $\mathbb{R}^{100}$, drawing 1000 points uniformly at random from the unit sphere intersected with each subspace. The subspace estimates $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$ are each generated to have the same distance from their respective true subspaces, displayed on the vertical axis. The horizontal axis indicates the distance between the true subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. For a fixed distance between the true subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, the figure demonstrates a significant distance gap even when there is nontrivial estimation error in the subspaces. For example, in the case where $\text{dist}(\mathcal{S}_1, \mathcal{S}_2) = 0.5$, the distance gap is still greater than 0.13, even when $\text{dist}(\mathcal{S}_i, \hat{\mathcal{S}}_i) = 0.5$. By comparison, if we were to use the inner product-based distance (as in Fig. 3(b)), the resulting distance gap would be 0.05. Hence, the integration of both the arrangement of points and the subspace estimates results in a metric that robustly differentiates points lying on a UoS.

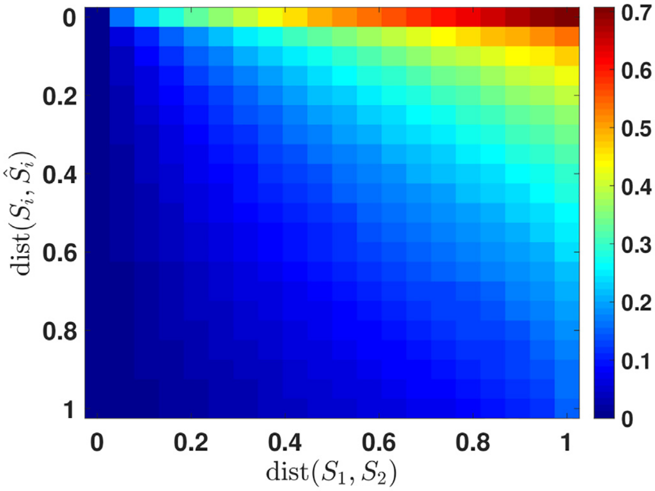

Fig. 4. Distance gap as a function of subspace estimation error (vertical axis) and distance between the true subspaces (horizontal axis) computed using the proposed metric (5). The larger the distance gap, the better clustered is the dataset. Points are drawn from two ten-dimensional subspaces of $\mathbb{R}^{100}$. Using the proposed metric, points in different subspaces are well separated even under significant subspace estimation error.

Table 2
Comparison of CQMs using Euclidean distance and proposed subspace-based distance from Section 3. Values indicate the average R-AUC across all six algorithms (lower better).

	Euclidean			Proposed		
Dataset	DI	SI	CH	DI	SI	CH
Synthetic	1.09	1.95	1.49	1.12	1.08	1.08
Hopkins-155	1.45	4.88	3.13	1.48	1.13	1.23
Yale B	1.08	1.19	1.06	1.10	1.12	1.05
COIL-20	1.08	1.13	1.16	1.06	1.14	1.04
Iris	1.13	1.15	1.13	1.25	1.17	1.27
Balance	1.07	1.08	1.09	1.09	1.05	1.07
Sonar	1.03	1.04	1.03	1.04	1.05	1.04

5.3. CQM comparison: proposed vs. euclidean distance

Next, we evaluate the impact of utilizing the proposed metrics from Section 3 on the DI, SI, and CH CQMs. For each dataset, we run each of the six algorithms in Table 1 for a grid of parameters in the range specified. Aside from the benchmark datasets described above, we also consider synthetic data drawn from a union of $K = 5$ subspaces, each having dimension $d = 5$, drawn uniformly at random in ambient space $D = 100$. We draw $N_k = 100$ points from each subspace and corrupt them with Gaussian noise with variance $\sigma^2 = 0.05$. We generate ten instances of data according to this arrangement and report the average value. Table 2 shows the average R-AUC (taken over all algorithms) for each of these three CQMs using both Euclidean distance (left columns) and our proposed subspace-based distances (right columns). For the Hopkins dataset, where the UoS structure is known to be strong, the use of subspace-based metrics has an especially large impact on the performance of both the SI and CH, yielding a R-AUC of roughly 1/4 and 1/3, respectively, of the Euclidean-based variants. On other datasets, the improvement in R-AUC is more mild, though the resulting error selected by the subspace-based CQMs is typically significantly lower than that of the Euclidean-based CQMs. We confirm this finding in Table 3, where we display the average difference between the error selected by the various CQMs and the oracle error. In this case, a value of zero would indicate that the CQM selected the best possible clustering among all configurations for each algorithm. The table displays the dramatic benefit of utilizing the proposed metric for datasets known to have strong UoS structure. For example, on Hopkins-155, the Euclidean-based SI selects errors that are an average of 24.56% greater than the minimum error, while the proposed subspace-based SI results in errors that are only 3.94% above the oracle. For the DI, the choice of metric appears to have less impact; however, we will show in Section 5.5 that the DI performs poorly overall when compared to the proposed KSS and NKSS CQMs. For completeness, we also consider the Balance, Iris, and Sonar datasets from the UCI Machine Learning Repository [60], none of which is expected to exhibit UoS structure. Our results show that the subspace-based CQMs perform on par with their Euclidean counterparts, even showing a mild improvement in some cases (e.g., for the Balance dataset). Hence, our proposed subspace-based metrics result in significant benefits in the case where the underlying UoS structure is strong and do not appear to be harmful even in the case where there is no such structure.

Table 3
Average difference between best and oracle clustering error (%) according to CQMs using Euclidean distance and proposed subspace-based distance from Section 3. Average is taken across the six algorithms listed in Table 1.

	Euclidean			Proposed		
Dataset	DI	SI	CH	DI	SI	CH
Synthetic	0.85	27.24	19.95	0.56	7.57	0.37
Hopkins-155	9.81	24.56	19.56	6.73	3.94	3.93
Yale B	12.37	54.69	26.11	24.66	35.47	8.90
COIL-20	9.46	27.97	42.15	12.26	24.20	6.55
Iris	9.89	17.22	4.44	18.11	17.11	12.00
Balance	16.32	15.09	11.68	9.97	10.24	9.31
Sonar	11.06	8.01	8.09	10.58	12.58	12.58

5.4. Selecting the number of clusters

We next consider the problem of selecting the number of clusters K on synthetic data as well as the Hopkins, Yale, and COIL datasets. For the synthetic data, we generate data from $K = 7$ subspaces of dimension $d = 5$ drawn uniformly at random from the Grassmannian in ambient dimension $D = 100$. For $k = 1, \ldots, K$, we draw $N_k = 100$ points from the subspace spanned by U_k as $x_i \sim \mathcal{N}(0, U_k U_k^T)$, corrupt them with independent and identically distributed Gaussian noise with variance $\sigma^2 = 0.05$, and then normalize the points to have unit norm. We generate ten instances of data according to this arrangement and report the average values below.

For each dataset, we run each algorithm listed in Table 1 over a range of 10 values of K and select the best clustering according to each CQM. The resulting average absolute deviation from the true value of K is given in Table 4, and the average R-AUC (across algorithms) is given in Table 5. For the synthetic data, the KSS cost uniformly chooses the wrong K, choosing $K = 8$ for all algorithms and trials, while the normalization in NKSS selects the correct value in all instances. However, examination of the R-AUC shows that the KSS cost does a better job of selecting low-error clusterings, even though the number of clusters may be wrong. The SI also selects the correct value for all algorithms and trials, while modularity and communitude selected K correctly for all but SSC-OMP and EKSS. In the case of EKSS, inspection of the affinity matrices reveals that both CQMs favored sparse affinity matrices, a phenomenon noted in [36]. For the benchmark data, both the KSS cost and NKSS are among the top performers in terms of absolute deviation and R-AUC, while the CH achieves the best R-AUC scores for several datasets despite having larger deviations from the true number of clusters. This highlights the fact that selecting the “correct” number of clusters does not always result in the lowest clustering error.

Table 4
Ability of various CQMs to select the correct number of clusters. Values indicate the average absolute deviation between the true and estimated number of clusters. Lowest values in each row are bolded.

Dataset	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
Synthetic	1.00	**0.00**	0.57	**0.00**	4.75	4.22	0.13	2.02	0.10
Hopkins-155	**0.02**	0.53	0.63	0.64	0.38	0.70	0.08	0.33	0.15
Yale B	**1.00**	**1.00**	1.83	**1.00**	**1.00**	1.67	**1.00**	3.00	**1.00**
COIL-20	1.00	1.00	1.50	1.83	1.00	3.67	**0.67**	**0.67**	1.67

Table 5
Ability of various CQMs to select the correct number of clusters. Values indicate the average R-AUC across all six algorithms (lower better).

Dataset	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
Synthetic	**1.03**	1.13	1.91	1.21	1.54	1.90	1.17	1.52	1.25
Hopkins-155	1.23	1.17	1.17	1.12	**1.11**	1.14	1.27	1.17	1.20
Yale B	1.10	1.14	1.49	1.08	**1.04**	1.36	1.25	1.61	1.15
COIL-20	**1.03**	**1.03**	1.52	1.49	**1.03**	1.97	1.16	1.13	1.31

Table 6
Ability of various CQMs to select tuning parameters on synthetic UoS data. Algorithm values (rows 2–7) indicate the average best clustering error (%) according to various CQMs. Oracle denotes the best overall clustering error. Final row shows the average R-AUC (lower better).

Algorithm	Oracle	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
SSC-ADMM	1.90	**1.90**	1.98	2.20	20.16	1.96	54.66	40.64	46.98	49.50
SSC-OMP	4.54	**4.54**	4.86	4.82	**4.54**	**4.54**	56.12	56.12	6.04	56.12
EnSC	2.28	**2.28**	2.44	2.70	2.40	2.34	2.70	2.70	6.30	2.70
GSC	0.50	**2.02**	**2.02**	2.08	3.82	**2.02**	26.70	2.14	2.12	27.52
TSC	0.56	**0.66**	0.78	0.94	0.76	0.76	1.68	1.68	0.80	1.68
EKSS	0.38	0.82	0.94	0.76	23.90	0.74	0.72	0.70	**0.62**	41.22
Average R-AUC	1	1.08	**1.07**	1.12	1.08	1.08	1.25	1.19	1.30	1.25

Table 7
Ability of various CQMs to select tuning parameters on Hopkins-155 dataset. Algorithm values (rows 2–7) indicate the average best clustering error (%) according to various CQMs. Oracle denotes the best overall clustering error. Final row shows the average R-AUC (lower better).

Algorithm	Oracle	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
SSC-ADMM	1.07	3.31	4.25	4.89	2.81	**2.59**	18.15	16.01	13.52	18.61
SSC-OMP	25.25	33.10	33.55	33.43	31.91	**31.77**	40.87	43.06	36.30	42.24
EnSC	9.75	13.88	15.25	16.46	12.97	**12.52**	21.85	25.80	22.91	23.66
GSC	2.07	4.83	5.95	6.77	6.24	**4.74**	19.98	26.53	10.23	22.67
TSC	11.82	**16.25**	17.79	19.94	16.21	16.50	22.15	27.45	25.04	25.36
EKSS	0.26	6.15	8.21	9.13	**3.70**	5.67	17.96	30.08	18.24	28.65
Average R-AUC	1	1.38	1.41	1.48	**1.13**	1.23	3.18	3.33	2.68	3.10

Table 8
Ability of various CQMs to select tuning parameters on Yale B dataset. Algorithm values (rows 2–7) indicate the best clustering error (%) according to various CQMs. Oracle denotes the best overall clustering error. Final row shows the R-AUC (lower better).

Algorithm	Oracle	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
SSC-ADMM	9.83	**9.83**	23.68	31.37	84.09	32.69	80.84	76.27	76.27	84.09
SSC-OMP	13.28	**13.28**	27.59	30.35	38.16	27.59	79.15	79.15	79.15	79.15
EnSC	18.87	31.58	28.99	58.92	42.85	31.58	**21.30**	**21.30**	63.36	**21.30**
GSC	20.27	31.87	31.87	31.87	**21.71**	22.78	69.98	69.98	69.98	69.98
TSC	22.20	**22.20**	**22.20**	41.24	39.27	**22.20**	49.34	49.34	49.34	49.34
EKSS	16.00	**17.02**	22.94	54.65	87.21	**17.02**	81.62	50.37	35.49	84.95
R-AUC	1	1.04	**1.03**	1.10	1.12	1.05	1.10	1.08	1.11	1.10

5.5. General parameter selection

Finally, we consider the problem of selecting arbitrary algorithm parameters, including optimization hyperparameters, thresholding parameters, and number of neighbors, on the three benchmark datasets described above. For each dataset, we run each of the six algorithms listed in Table 1 for a grid of parameters in the range specified. We provide the oracle parameters, i.e., those resulting in the lowest clustering error, in Table 10 at the end of this section.

Tables 6–9 show the resulting errors obtained and R-AUC for each CQM on the Synthetic, Hopkins, Yale, and COIL datasets, respectively, where Synthetic refers to the dataset described in Section 5.3. While no single CQM stands out as the best performer across all datasets, several useful observations can be made. First, the proposed CQMs that explicitly account for existing UoS structure consistently outperform those that are based solely on the

Table 9
Ability of various CQMs to select tuning parameters on COIL-20 dataset. Algorithm values (rows 2–7) indicate the best clustering error (%) according to various CQMs. Oracle denotes the best overall clustering error. Final row shows the R-AUC (lower better).

Algorithm	Oracle	KSS Cost	NKSS	DI	SI	CH	Cov	Mod	Per	Comm
SSC-ADMM	13.19	15.28	17.50	33.68	47.22	16.32	63.68	**13.19**	57.15	63.68
SSC-OMP	27.29	**27.29**	**27.29**	36.67	87.92	**27.29**	64.72	**27.29**	36.67	64.72
EnSC	8.26	8.47	17.36	21.60	46.67	17.36	**8.26**	**8.26**	21.67	**8.26**
GSC	2.99	12.50	12.50	12.50	10.28	12.50	61.32	**3.40**	10.62	68.96
TSC	15.62	17.22	20.62	16.60	16.88	**15.83**	21.46	16.25	23.33	21.46
EKSS	14.03	31.39	31.39	33.89	**17.64**	31.39	47.71	40.35	27.99	48.96
R-AUC	1	1.06	1.05	1.06	1.14	**1.04**	1.08	1.05	1.11	1.08

Table 10
Parameter configuration resulting in the lowest clustering error for each algorithm on each dataset. Description of parameters for each algorithm is given in Table 1. For Hopkins-155 dataset, the mode of each parameter is displayed.

Algorithm	Synthetic	Hopkins-155	Yale B	COIL-20
SSC-ADMM	(1.0, 5.0)	(0.1, 226.67)	(0.10, 670)	(0.8, 5)
SSC-OMP	2	2	2	2
EnSC	(3, 0.01)	(98, 0.01)	(3, 0.88)	(3,0.99)
GSC	(14, 1)	(2, 1)	(12, 5)	(11, 9)
TSC	6	3	3	8
EKSS	(87, 5)	(2, 3)	(18, 7)	(7, 12)

affinity matrix, reinforcing the notion that geometric structure in the data should be leveraged when it is known to exist. Second, the KSS Cost is a strong performer across all three datasets, though it should be noted that this is in light of a fixed number of clusters K, and in the case of algorithms that do not have subspace dimension as input, a fixed subspace dimension d. In the case of GSC and EKSS, where the subspace dimension is selected, KSS Cost selects $d = 20$ for both the Yale and COIL datasets. However, these still correspond to clusterings of low error, as indicated in the table. Third, the CH obtains strong performance across all datasets, while the SI is the best CQM on the Hopkins dataset but performs poorly on Yale and COIL. Upon closer inspection, we found that the SI favored clusterings in which one or two clusters contain the overwhelming majority of the points.

Based on the above observations, the results indicate that when the number of clusters is unknown, the NKSS and SI provide the most reliable performance, though practitioners should take care to verify that the SI does not select clusterings with unwarranted class imbalance. In the case where the number of clusters is known in advance, the KSS Cost and CH provide the most reliable indications of clustering quality.

6. Conclusions & future work

In this work, we present the first comprehensive study of internal clustering validation for the problem of subspace clustering. We propose a novel point-to-point distance designed to capture the salient features of points lying on a union of subspaces, and we demonstrate empirically that this pseudometric has favorable properties. We then propose a variety of measures of clustering quality that can be used to select the “best” configuration of parameters for any subspace clustering algorithm. Our results show that while no single CQM is clearly dominant, measures such as the proposed normalized KSS cost and Silhouette Index can be used to select the number of clusters, while the KSS cost and Calinski-Harabasz index provide strong results on selecting the algorithm parameters.

As this is a first approach to the clustering validation problem for subspace clustering, we believe that it will enable researchers and practitioners to develop new CQMs based on the proposed distances. Finally, the proposed point-to-point metric resembles a Mahalanobis distance, as noted in Section 3.3. In light of this fact, it would be interesting to incorporate our distance into the problem of metric learning with pairwise constraints, as in [61], which may open a new avenue for the development of subspace clustering algorithms.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the anonymous reviewers for their comments on this manuscript. L. Balzano was supported by DARPA grant 16-43-D3M-FP-037, NSF CAREER award CCF-1845076, AFOSR YIP award FA9550-19-1-0026, and ARO YIP award W911NF1910027. J. Lipor was supported by National Science Foundation DMS 1624776 and by the U.S. Army Basic Research Program under PE 61102, Project T25, Task 02 “Network Science Initiative,” managed at the U.S. Army ERDC with Portland State University under Cooperative Agreement Number W912HZ-17-2-0005.

References

[1] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (1) (2013) 243–256.

[2] T. Chakraborty, A. Dalmia, A. Mukherjee, N. Ganguly, Metrics for community analysis: a survey, ACM Comput. Surv. (CSUR) 50 (4) (2017) 54.

[3] B. Desgraupes, Clustering indices, Univ. Paris Ouest Lab Modal X 1 (2013) 34.

[4] T.A. Bailey Jr., R. Dubes, Cluster validity profiles, Pattern Recognit. 15 (2) (1982) 61–83.

[5] R. Vidal, S.S. Sastry, Y. Ma, Generalized Principal Component Analysis, Springer-Verlag, 2016.

[6] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.

[7] Z. Lin, M. Chen, L. Wu, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, Coordinated Science Laboratory Report no. UILU-ENG-09-2215, DC-247, 2009.

[8] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2765–2781.

[9] R. Vidal, Subspace clustering, IEEE Signal Process. Mag. 28 (2011) 52–68.

[10] S. Kumar, Y. Dai, H. Li, Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion, Pattern Recognit. 71 (2017) 428–443.

[11] T. Wu, Graph regularized low-rank representation for submodule clustering, Pattern Recognit. (2019) 107145.

[12] Q. Li, Z. Sun, Z. Lin, R. He, T. Tan, Transformation invariant subspace clustering, Pattern Recognit. 59 (2016) 142–155.

[13] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 663–670.

[14] L. Fei, Y. Xu, X. Fang, J. Yang, Low rank representation with adaptive distance penalty for semi-supervised subspace classification, Pattern Recognit. 67 (2017) 252–262.

[15] Y. Wang, Y.-X. Wang, A. Singh, Graph connectivity in noisy sparse subspace clustering, in: Artificial Intelligence and Statistics, 2016, pp. 538–546.

[16] R. Heckel, H. Bölcskei, Robust subspace clustering via thresholding, IEEE Trans. Inf. Theory 24 (11) (2015) 6320–6342.

[17] R. Heckel, E. Agustsson, H. Bolcskei, Neighborhood selection for thresholding-based subspace clustering, in: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 2014, pp. 6761–6765.

[18] P.S. Bradley, O.L. Mangasarian, *k*-plane clustering, J. Global Optim. 16 (2000) 23–32.

[19] P. Tseng, Nearest q-flat to m points, J. Optim. Theory Appl. 105 (1) (2000) 249–252.

[20] P.K. Agarwal, N.H. Mustafa, K-means projective clustering, in: Proceedings of the twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database systems, ACM, 2004, pp. 155–165.

[21] A. Gitlin, B. Tao, L. Balzano, J. Lipor, Improving *k*-subspaces via coherence pursuit, IEEE J. Sel. Top. Signal Process. 12 (6) (2018) 1575–1588.

[22] M. Rahmani, G.K. Atia, Coherence pursuit: fast, simple, and robust principal component analysis, IEEE Trans. Signal Process. 65 (23) (2017) 6260–6275.

[23] J. Lipor, D. Hong, Y.S. Tan, L. Balzano (2017). Subspace Clustering using Ensembles of k-Subspaces. arXiv preprint arXiv:1709.04744.

[24] A.L. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.

[25] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of internal clustering validation measures, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE, 2010, pp. 911–916.

[26] J. Hou, W. Liu, E. Xu, H. Cui, Towards parameter-independent data clustering and image segmentation, Pattern Recognit. 60 (2016) 25–36.

[27] M. Sause, A. Gribov, A.R. Unwin, S. Horn, Pattern recognition approach to identify natural clusters of acoustic emission signals, Pattern Recognit. Lett. 33 (1) (2012) 17–23.

[28] I. Lapidot, H. Guterman, A. Cohen, Unsupervised speaker recognition based on competition between self-organizing maps, IEEE Trans. Neural Netw. 13 (4) (2002) 877–887.

[29] P. Jaccard, The distribution of the flora in the alpine zone., New Phytol. 11 (2) (1912) 37–50.

[30] L. Hubert, P. Arabie, Comparing partitions, J. Class. 2 (1) (1985) 193–218.

[31] J.C. Dunn, Well-separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1) (1974) 95–104.

[32] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. (2) (1979) 224–227.

[33] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

[34] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. Theory Methods 3 (1) (1974) 1–27.

[35] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, 1, Springer Series in Statistics New York, 2001.

[36] H. Almeida, D. Guedes, W. Meira, M.J. Zaki, Is there a best quality metric for graph clusters? in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 44–59.

[37] J. Creusefond, T. Largillier, S. Peyronnet, On the evaluation potential of quality functions in community detection for different contexts, in: International Conference and School on Network Science, Springer, 2016, pp. 111–125.

[38] U. Brandes, M. Gaertler, D. Wagner, Experiments on Graph Clustering Algorithms, Springer, 2003.

[39] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.

[40] J. Lipor, L. Balzano, Leveraging union of subspace structure to improve constrained clustering, in: International Conference on Machine Learning, 2017, pp. 2130–2139.

[41] I. Guyon, U. Von Luxburg, R.C. Williamson, Clustering: science or art, in: NIPS 2009 Workshop on Clustering Theory, 2009, pp. 1–11.

[42] G. Golub, C.V. Loan, Matrix Computations, Johns Hopkins University Press, 2012.

[43] P.-A. Absil, R. Mahony, R. Sepulchre, Riemannian geometry of Grassmann manifolds with a view on algorithmic computation, Acta Appl. Math. 80 (2) (2004) 199–220.

[44] D. Park, C. Caramanis, S. Sanghavi, Greedy subspace clustering, in: Advances in Neural Information Processing Systems, 2014, pp. 2753–2761.

[45] J.H. Conway, R.H. Hardin, N.J. Sloane, Packing lines, planes, etc.: packings in grassmannian spaces, Exp. Math. 5 (2) (1996) 139–159.

[46] A. Jalali, R. Willett, Subspace clustering via tangent cones, in: Advances in Neural Information Processing Systems, 2017, pp. 6744–6753.

[47] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1) (1964) 1–27.

[48] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. R. Stat. Soc. 63 (2) (2001) 411–423.

[49] T. Chakraborty, A. Dalmia, A. Mukherjee, N. Ganguly, Metrics for community analysis: a survey, arXiv preprint arXiv:1604.03512 (2016).

[50] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, S. Bhowmick, On the permanence of vertices in network communities, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 1396–1405.

[51] A. Miyauchi, Y. Kawase, What is a network community? A novel quality function and detection algorithms, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1471–1480.

[52] C. You, D. Robinson, R. Vidal, Scalable sparse subspace clustering by orthogonal matching pursuit, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3918–3927.

[53] C. You, C.-G. Li, D.P. Robinson, R. Vidal, Oracle based active set algorithm for scalable elastic net subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3928–3937.

[54] R. Tron, R. Vidal, A benchmark for the comparison of 3-D motion segmentation algorithms, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[55] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.

[56] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Technical Report, Columbia University, 1996.

[57] T.P. Minka, Automatic choice of dimensionality for PCA, in: Advances in Neural Information Processing Systems, 2001, pp. 598–604.

[58] Y. Dodge, The Concise Encyclopedia of Statistics, Springer Science & Business Media, 2008.

[59] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.

[60] D. Dua, C. Graff, UCI Machine Learning Repository [Http://Archive.ics.uci.edu/Ml], University of California, School of Information and Computer Science, Irvine, CA, 2019.

[61] E.P. Xing, M.I. Jordan, S.J. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information, in: Advances in Neural Information Processing Systems, 2003, pp. 521–528.

John Lipor is an assistant professor in Electrical and Computer Engineering at Portland State University. He received his Ph.D. in Electrical Engineering from the University of Michigan and was an awardee of the NSF Graduate Research Fellowship. His research interests include active learning and subspace methods.

Laura Balzano is an associate professor of Electrical Engineering and Computer Science at the University of Michigan. She received an NSF CAREER award, an ARO Young Investigator award, and an AFOSR Young Investigator award. She is also a Fulbright scholar, an Intel Early Career Faculty Honor Fellow and a 3M Non-tenured Faculty Awardee. Her main research focus is optimization for low-rank and matrix factorization models with high-dimensional data that are highly incomplete or corrupted.