Clustering-based Detection of Debye-Scherrer Rings

Rabia Sirhindi
Department of Computer Science
University of the Punjab
Lahore, Pakistan
Email: rabia.sirhindi@pucit.edu.pk

Nazar Khan
Department of Computer Science
University of the Punjab
Lahore, Pakistan
Email: nazarkhan@pucit.edu.pk

Calibration of the X-ray powder diffraction experimental setup is a crucial step before data reduction and analysis, and requires correctly extracting individual Debye-Scherrer rings from the 2D XRPD image. This problem is approached using a clustering-based machine learning framework, thus interpreting each ring as a cluster. This allows automatic identification of Debye-Scherrer rings without human intervention and irrespective of detector type and orientation. Various existing clustering techniques are applied to XRPD images generated from both orthogonal and non-orthogonal detectors and the results are visually presented for images with varying inter-ring distances, diffuse scatter and ring graininess. The accuracy of predicted clusters is quantitatively evaluated using an annotated gold standard and multiple cluster analysis criteria. These results demonstrate the superiority of density-based clustering for the detection of Debye-Scherrer rings. Moreover, the given algorithms impose no prior restrictions on detector parameters such as sample-to-detector distance, alignment of the center of diffraction pattern or detector type and tilt, as opposed to existing automatic detection approaches.

1 INTRODUCTION

Analysis of 2D X-ray powder diffraction (XRPD) data is widely used for determining the crystal structure of unknown materials and compounds. It has applications in geological sciences, pharmaceutical industry, biology, texture analysis, stress and strain analysis of materials and measurement of grain sizes and orientations of unknown solids. X-ray diffraction from solid crystals is used to determine both X-ray wavelengths when the crystal structure is known, and crystal structure of unknown solids when X-ray wavelength is known. A crystal structure consists of regularly repeating three-dimensional patterns of atomic cells. The incidence of a collimated X-ray beam on a poly-crystalline powdered substance produces a number of scattering cones making concentric elliptical patterns on a 2D detector surface mounted co-axially with the beam (as seen in Fig. 1).

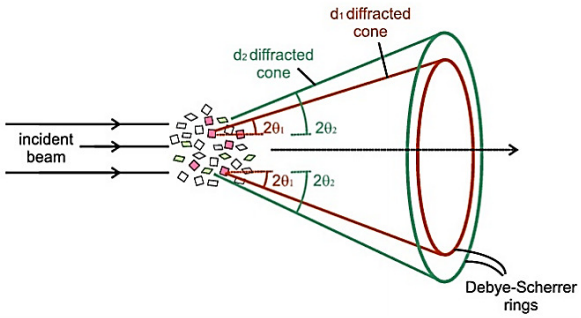


Fig. 1. X-ray powder diffraction. [1]

The diffraction patterns produced are in the form of conic sections, also called Debye-Scherrer rings [3]. Fig. 2 illustrates some 2D XRPD images collected from different area detectors. Detector configurations that are orthogonal to the beamline produce concentric ellipses while hyperbolic patterns are also observed in case of non-orthogonal detectors. Identifying these 2-D patterns in XRPD data is an important pre-requisite of the data reduction step in phase identification [4, 5]. Furthermore, an important pre-requisite of X-ray data analysis is calibration of the XRPD experimental setup. Detection of Debye-Scherrer rings also facilitates the calibration process thereby accurately identifying parameters such as

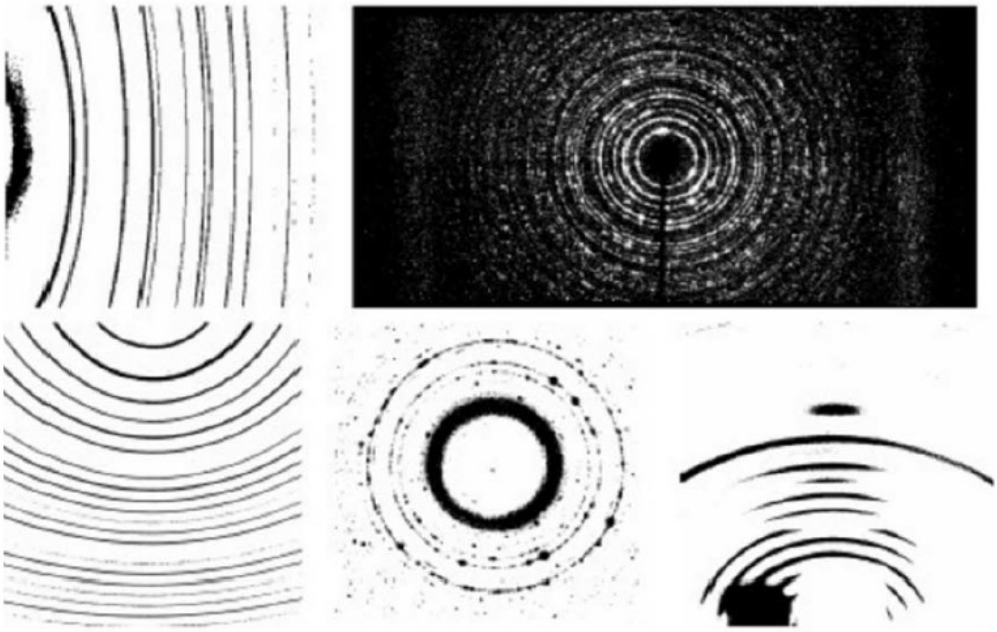


Fig. 2. XRPD images with circular, elliptical and hyperbolic Debye-Scherrer rings [2]

powder-to-detector distance, center of diffraction pattern, and correctness of geometrical errors caused by detector's orientation or tilt.

The problem can be formulated as follows. For any XRPD image I_{pd}, the task of automatically and accurately extracting a set of Debye-Scherrer rings R requires identifying the pixel regions corresponding to an individual ring as a first step. This paper aims at devising methods for automatic identification of these elliptical regions, which is a crucial prerequisite of both the calibration of XRPD setup and data analysis phase. This task becomes challenging since the XRPD images obtained as a result of the diffraction experiments are inherently noisy having small inter-ring distances and spotty rings, sometimes missing several connecting regions. Also, both elliptical and hyperbolic patterns are observed depending on the type of detector used.

Traditional approaches for the detection of Debye-Scherrer rings are based on manual marking of elliptical regions in images. This process has been automated to some extent in numerous open-source software packages designed for calibration and analysis of X-ray data such as d2Dplot [6], Fit2D [7], Power3d [8] and pyFAI [9]. However, these data analysis tools are still dependant on manual marking of points on rings before azimuthally integrating sections of ellipses in the data reduction step. This process is both slow and error-prone. Some automatic ellipse detection techniques have been proposed [10, 11, 12], but are rather restrictive in terms of choice of detector's tilt and parameters. Recently, an incremental ellipse detection algorithm (IED) has been proposed [2] based on computer vision techniques, which automatically detects Debye-Scherrer rings without any manual marking. It uses region growing and ellipse fitting techniques to identify elliptical regions in X-ray images. The IED algorithm performs better than the existing detection techniques and can accurately detect Debye-Scherrer rings in noiseless images. However, the accuracy of ring detection decreases significantly for noisy images where rings are spotty or unconnected and occur very close to each other. For images generated by orthogonal detectors, this problem is addressed by forcing the Debye-Scherrer rings to belong to a family of ellipses with the same major-to-minor axis ratio. Thus, when one ellipse is correctly detected using IED, its major-to-minor axis ratio is used to determine subsequent ellipses if any exist. This approach is based on the fact that all concentric ellipses can be generated using a scaled version of any one detected ellipse. However, the ellipse family constraint based on major-to-minor axis ratio cannot be applied to Debye-Scherrer rings generated by non-orthogonal detectors, where the rings may take parabolic shapes.

This paper presents the application of clustering techniques for the automatic detection of ellipses in XRPD images. Clustering is widely utilized in pattern recognition tasks to improve the performance of industrial manu-

facturing processes[13, 14, 15]. A cluster-based interpretation of XRPD data is presented where points belonging to one elliptical region or ring are identified as one cluster. Once the rings are identified, their parameters can be obtained using any ellipse-fitting technique. It is demonstrated through visual and quantitative results that clustering-based approaches perform well for the identification of Debye-Scherrer rings. It is shown that irrespective of detector type and tilt, clustering methods can detect multiple Debye-Scherrer rings in X-ray diffraction images generated from different area detectors. This includes images containing elliptical as well as hyperbolic rings. It is also highlight that some clustering algorithms are able to successfully detect all or most of the Debye-Scherrer rings in noiseless images. For moderately and very noisy images, these algorithms are able to correctly detect the inner-most ring and a few outer rings. Compared to previous detection approaches, the proposed clustering-based solutions are not restrictive in terms of center of diffraction pattern and sample-to-detector distance and works equally well for orthogonal and non-orthogonal detectors. A thorough comparative analysis of the performance of different clustering algorithms is also presented in this regard. Moreover, it is argued that the regions initially obtained by IED and subsequently grown into ellipses are much higher in number as compared to those identified by clustering, thus incurring more computational overhead.

The main contributions of this research are as follows.

1. Interpretation of Debye-Scherrer rings as clusters which allows their detection using clustering algorithms.
2. Annotation of XRPD data with ground truth labels for quantitative evaluation of various clustering algorithms using standard cluster evaluation criteria.
3. Comparison and analysis of seven clustering methods for the ring detection problem in XRPD images in the presence of varying inter-ring distances, diffuse scattering and graininess.
4. Analysis that density-based and spectral clustering algorithms yield better Debye-Scherrer rings.
5. Automatic estimation of number of rings in a diffraction pattern.
6. Visual comparison of regions identified by region growing and data clustering.

This paper is organized as follows. The main categories of clustering algorithms are introduced in Section 2. This is followed by the results of different clustering algorithms and their limitations for detecting Debye-Scherrer rings in Section 3. The accuracy of clusters depends on how close they are to actual Debye-Scherrer rings in the image. This is evaluated using multiple external criteria and quantitative results are reported in Section 4. In Section 5, the role of various algorithmic constants in determining accurate clusters is discussed. A discussion on how such parameters can be determined automatically from data is also presented in the same section.

2 DATA CLUSTERING

The goal of a clustering algorithm is to divide data points into different partitions or clusters such that the similarity of data points within a cluster is high and that of data points belonging to different clusters is low. The most commonly used similarity measures employ Euclidean distance between pairs of data points, however, other similarity measures can also be used to capture the underlying structure of data and compute similarities accordingly [16, 17, 18]. Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{N \times D}$, the problem of clustering can be formulated as finding c disjoint partitions $\{\mathcal{A}\}_1^{n_1 \times D}, \{\mathcal{A}\}_2^{n_2 \times D}, \ldots, \{\mathcal{A}\}_c^{n_c \times D}$ of S such that for any $\mathbf{x}_j \in \mathcal{A}_k$, $\mathbf{x}_j \notin \overline{\mathcal{A}_k}$ and $\sum_i^c n_i = N$. Various clustering algorithms exist in this regard including centroid-based, hierarchical, spatial density-based and graph theory-based clustering [19]. Fig. 3 illustrates some of these techniques. Centroid-based algorithms such as k-means, are best suited to data having compact groups with well-separated centers. It may not perform well on data with intersecting groups or groups having overlapping means ($\mu_i \approx \mu_j$). When this happens, cluster memberships can be determined using the distance among data points instead of distance from some centroid value. In the following sections some clustering techniques that use this method of grouping points together are discussed.

2.1 Hierarchical Clustering

Hierarchical clustering builds a tree of data points starting either at the root or the leaf nodes. The former, called *divisive* hierarchical clustering, assigns all data points to a single root cluster and recursively partitions it into most dissimilar smaller sub-groups. An example algorithm is divisive analysis (DiAna) [17]. The latter, called *agglomerative* hierarchical clustering, takes a bottom-up approach considering each point as an individual cluster, merging the similar groups together as the algorithm progresses. Hierarchical algorithms use different methods to measure the similarity (or dissimilarity) of clusters to determine if they can be linked together.

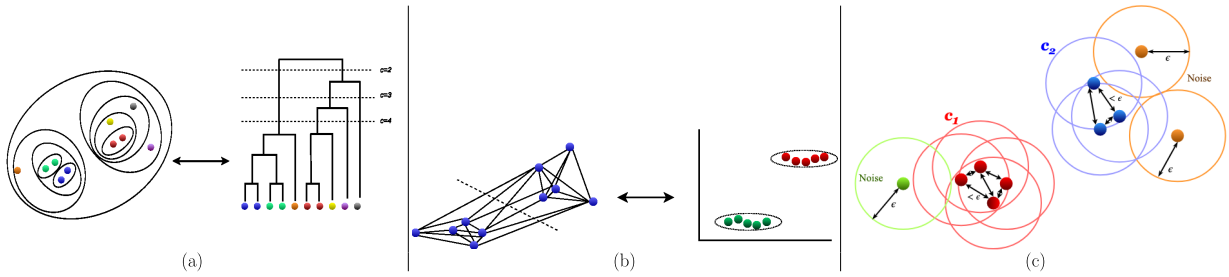


Fig. 3. Types of clustering algorithms illustrated for data having two clusters. (a) Hierarchical clustering for the same data. Each data point is an individual cluster and most similar clusters are iteratively merged to form bigger groups. (b) Spectral clustering using graph min-cut approach. A similarity graph of data is constructed followed by dimensionality reduction where clusters become more pronounced.(c) Density-based clustering where connected components of core points are formed into clusters.

Single linkage measures the distance between two clusters as the distance between their closest pair of elements.

$$\mathcal{D}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} d(x_i, x_j) \tag{1}$$

Complete linkage measures the distance between two clusters as the distance between their farthest pair of elements.

$$\mathcal{D}(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} d(x_i, x_j) \tag{2}$$

Average linkage computes the distance between two clusters as the average distance between all their elements.

$$\mathcal{D}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j) \tag{3}$$

Hierarchical clustering performs better than k-means for data with complex structure. However, the accuracy depends on the linkage method used. Also, in presence of noise, the algorithm fails to correctly capture hierarchical relationships in data.

2.2 Spectral Clustering

Spectral clustering provides an approximation of the graph min-cut problem using the weighted undirected proximity graph $\mathcal{G}$ of data. A variety of methods exist that capture the neighborhood information in this respect, including k-nearest neighbors, mutual kNN, ϵ-neighborhood, and fully connected graphs. A similarity function such as the Gaussian kernel is used to calculate the similarities between data points (edge weights in $\mathcal{G}$) and form the affinity matrix $\mathbf{A}$ of data. Gaussian kernel is especially useful when calculating similarities in a fully connected graph, as it captures the neighborhood information with the help of scaling parameter σ (see Eq. 4). It is important to emphasize that the structure of the affinity matrix depends on the proximity graph and its values depend on the similarity function. Data sets having well-pronounced clusters tend to have a block diagonal affinity matrix which represents higher intra-cluster similarities than inter-cluster similarities [20].

$$a_{ij} = exp^{-\frac{\mathcal{D}(i,j)^2}{2\sigma^2}} \tag{4}$$

where a_{ij} represents the similarity between points $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is the scaling coefficient.

Spectral clustering produces clusters using the eigenvalues and eigenvectors of the graph Laplacian which is obtained from the $\mathbf{A}$. Both un-normalized and normalized variants of graph Laplacian are employed for this purpose. Eq.5 shows how an un-normalized graph Laplacian is computed from the similarity and degree matrix of a graph.

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{5}$$

$\mathbf{D}$ is a diagonal matrix called the degree matrix where $d_{ii} = \sum_j a_{ij}$.

Eq.6 [21] and Eq.7 [22] are used to compute the two versions of the normalized graph Laplacian. Here $\mathbf{L}_{rw}$ represents a random-walk normalized Laplacian where as $\mathbf{L}_{sym}$ is symmetric normalized Laplacian.

$$\mathbf{L}_{rw} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \tag{6}$$

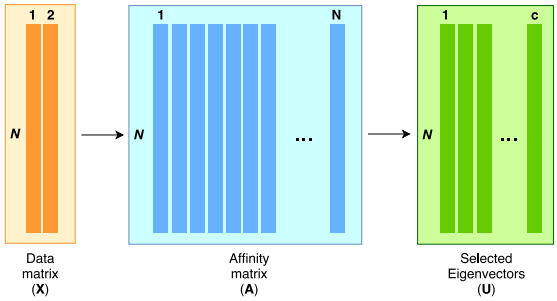


Fig. 4. Mapping D-dimensional data matrix ($\mathbf{X}$) to c-dimensional matrix ($\mathbf{U}$) of eigenvectors.

$$\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{7}$$

The Laplacian matrix obtained as a result is symmetric and positive semi-definite [20]. Moreover, the eigen decomposition of $\mathbf{L}$ yields real and non-negative eigenvalues ($0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$) and the eigenvectors form an orthonormal basis. The eigenspace of Laplacian forms a basis of indicator vectors that correspond to the connected components of the graph $\mathcal{G}$. This set of eigenvectors is denoted by $\mathbf{V} \in \mathbb{R}^{N\times N}$ where N is the number of data points. In the final step of the algorithm, c eigenvectors are selected from $\mathbf{V}$ to form $\mathbf{U} \in \mathbb{R}^{N\times c}$. The rows of this matrix correspond to each data point where a data point from $\mathbb{R}^{N\times D}$ is mapped to $\mathbb{R}^{N\times c}$. K-means is applied on the rows to obtain cluster memberships of each data point (as seen in Fig. 4). Spectral clustering performs considerably well on complex data sets where k-means and hierarchical algorithms fail. However, it is highly dependent on the similarity function and its parameters. Moreover, it is computationally intensive especially for large data sets, and not very robust to noisy data. Many variant algorithms of spectral clustering exist [23, 24] using novel similarity functions to capture the inherent patterns in data.

A spatial density-based approach to clustering robust to noise in data sets is discussed in the following section.

2.3 Density-based Clustering

A density-based spatial clustering algorithm for noisy data sets, abbreviated as DBSCAN, has been proposed in [25]. The main idea is that the density of data points within a cluster is higher than those outside it. Also, noise has a lower density than actual data, therefore points that are far away from majority of points can be labeled as outliers. DBSCAN works by building ϵ-neighborhoods of spatially close data points in the Euclidean space. A data point is called a *core* point if it has more than some minimum number of points ($minPts$) in its ϵ-ball radius. The connected component of each set of core points forms a cluster. A non-core point is assigned to the cluster corresponding to the closest core-point in its ϵ-neighborhood. Points that are not assigned to any of the clusters are marked as noise and removed (see Fig. 3 (c)). DBSCAN does not require prior knowledge of the number of clusters in data as opposed to traditional clustering algorithms. Instead, it grows neighborhoods based on ϵ-distance, very similar to region growing in the IED algorithm [2]. Moreover, it works well for noisy data sets and can detect arbitrary shaped clusters in data. However, similar to other clustering algorithms, it is sensitive to the choice of ϵ and $minPts$ to correctly identify clusters. Also, for very noisy data sets, DBSCAN can inadvertently label and remove some data points as noise. It may not produce correct results for data having varying densities of points. A generic approach to density-based clustering using kernel density estimation is also proposed in [26]. Moreover, a hierarchical density estimation based method HDBSCAN is presented in [27] that builds a cluster tree and extracts the most significant clusters from a complete density-based cluster hierarchy. It only requires the $minpts$ parameter from which it determines the value of ϵ to generate clusters of core points.

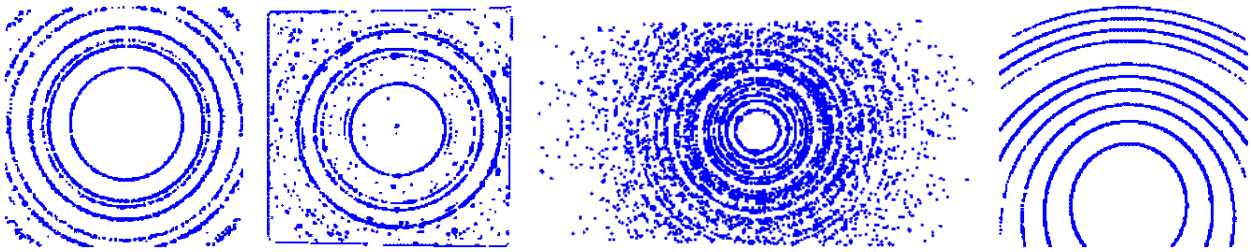
The following section presents visual results of the application of different clustering algorithms on XRPD images.

3 VISUAL RESULTS

An XRPD image is first pre-processed and binarized using the method described in [2] . This involves applying smoothing and morphological operations on the image to reduce the graininess (spottiness) of the rings. It enhances the connectivity of pixels along the elliptical arcs and produces better clustering results. XRPD images generated for calibrants from both orthogonal and non-orthogonal detectors are used in the experiments. Three of these namely Si12, LaB6 and Max1 belong to the family of images generated using orthogonal detectors, while the fourth (Tilted) is generated by a non-orthogonal detector [1]. Orthogonal detectors generate rings that are almost circular in nature where as non-orthogonal detectors produce patterns that may be elliptical but in some cases are modeled best using parabolas.

X-ray data from the binarized image is read in a matrix $\mathbf{X} \in \mathbb{N}^{N\times 2}$, where N is the total number of points of all Debye-Scherrer rings in the image. Each data point in

[1] All images are acquired from the European Synchroton Radiation Facility (ESRF). https://www.esrf.eu/home/education/what-is-the-esrf.html



(a) Max1 (b) Si12 (c) LaB6 (d) Tilted

Fig. 5. 2D XRPD image data from different detectors, and with varying degrees of noise. (a) Noiseless with some rings occurring very close to each other, (b) moderately noisy, (c) highly noisy with a lot of spots around the rings, and (d) noiseless and well-separated hyperbolic rings.

Table 1. Clustering parameters used in experiments given in Sections 3 and 4.

	Number of clusters (c)	Scaling co-efficient (σ)	Minimum points ($mPts$)	Epsilon (ϵ)
Max1	7	0.6	3	5
Si12	6	0.8	3	5
LaB6	14	0.3	2	3
Tilted	10	1.5	3	7

$\mathbf{X}$ is the row-column coordinate location corresponding to a pixel belonging to a ring in the image. Table 1 lists the parameters used for clustering algorithms for different X-ray images. Here c is the number of clusters given as input to the algorithm alongside the data matrix. This parameter is crucial in determining accurate results in all the aforementioned clustering algorithms except DBSCAN. The value of c is chosen according to the number of rings identified from the ground truth of each image. Methods to automatically determine this value are discussed in Section 5. In addition, σ represents the scaling parameter used in Gaussian function for constructing the affinity matrix. Global σ is chosen according to the density of data and how close or far apart the rings are from each other. Two parameters namely $mPts$ and ϵ are specific to DBSCAN and represent the minimum number of points required in an ϵ-ball radius of a point to be considered as a core point.

Figs. 6 presents the clustering results for the XRPD images corresponding to hierarchical clustering. Here HC refers to hierarchical agglomerative clustering with different linkage methods. Sng stands for single linkage, Comp for complete linkage, and Avg for average linkage as defined in Eqs. 1, 2 and 3. Diana refers to the divisive hierarchical clustering algorithm that works in a top down manner. Fig. 7 presents the results of different variants of spectral clustering. SC refers to un-normalized spectral clustering using $\mathbf{L}$ where as SC-Shi using $\mathbf{L}_{rw}$ and SC-NJW using $\mathbf{L}_{sym}$ are its normalized variants, as defined in Eqs. 5, 6 and 7 respectively. Moreover, SC-ST refers to self-tuning spectral clustering [23] that proposes using k-nearest neighbor distance to locally determine the values of scaling co-efficient σ to compute the affinity matrix as given in Eq. 4. SC-LD refers to local density adaptive spectral clustering [24]. This technique scales the global value of sigma by the common nearest neighbors of each pair of data points and subsequently computes the affinity between them using the Gaussian kernel. Finally, Fig. 8 presents the results of two variants of density-based clustering namely DBSCAN and HDBSCAN [27].

The results of hierarchical agglomerative clustering are demonstrated with various linkage methods. It can be seen that complete and average linkage (HC-Comp and HC-Avg) produce very similar clusters to k-means. They fail to identify any Debye-Scherrer rings even for noiseless X-ray images. Since average linkage takes the mean of distances between all points of two clusters, it converges as k-means. Complete linkage fails because the nature of Debye-Scherrer rings is such that based on maximum distance, regions of the same ring are considered further apart than the regions across two different rings. This results in points of adjacent rings being grouped into a single cluster, as opposed to the points lying on the circumference of the same ring. In contrast, the single linkage method (HC-Sng) is able to detect some individual rings in noiseless images (Max1 and Tilted). Single linkage takes the distance between two clusters as the min-

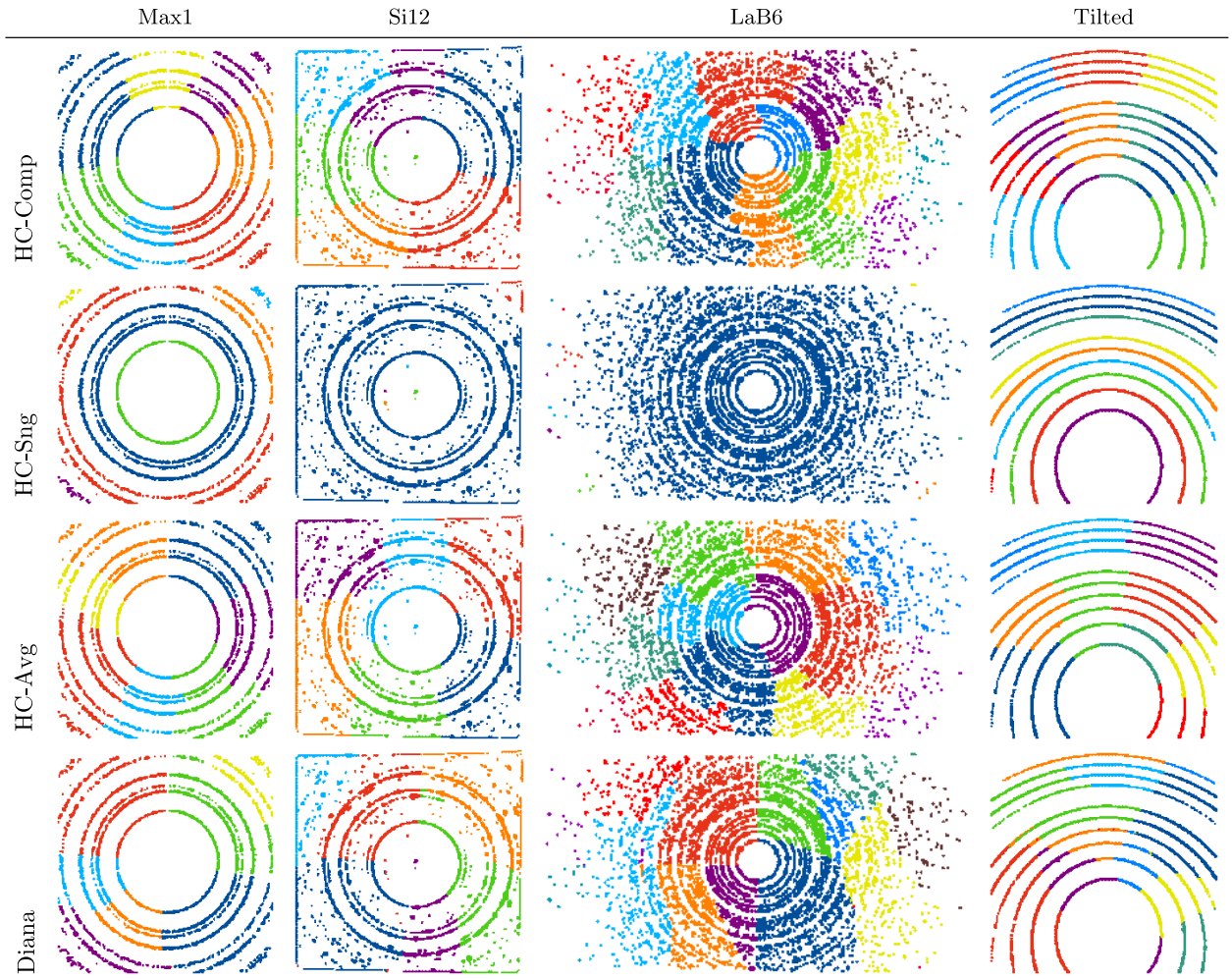


Fig. 6. Results of hierarchical clustering algorithms for XRPD images from both orthogonal (cols 1-3) and non-orthogonal (col 4) detectors. Each color in the XRPD image represents one cluster. The results show that among all agglomerative hierarchical algorithms only HC-Sng is able to detect some rings in noiseless images (cols 1 and 4). Also, divisive hierarchical clustering is not able to detect any ring accurately.

imum of distances between all points, therefore, regions of the same ring appear closer to each other and are combined in a cluster. The algorithm keeps merging neighboring regions on a ring until there are no more points left. However, single linkage combines some rings together that occur very close to each other even in noiseless image (Max1). Linkage methods such as single and complete that make use of a single pair of points to make a merging decision are always susceptible to noise. Therefore, single linkage performs poorly when images have moderate to high noise (Si12 and LaB6), grouping all the rings together. It can be observed that divisive hierarchical clustering Diana also produces results similar to HC-Comp since it uses highest average dissimilarity between points in a group to determine how to split a cluster into multiple sub-clusters.

The clusters identified by spectral clustering are considerably better than the hierarchical approaches. Unnormalized spectral clustering (SC) generates similar clusters as the single linkage hierarchical method. However, the algorithm performs better when normalized Laplacian $\mathbf{L}_{rw}$ is used (SC-Shi). It identifies the inner most ring correctly in Max1, but clusters narrowly spaced rings together in the same image. For noisy images Si12 and LaB6, the algorithm is unable to give accurate results, clustering all rings together in a single cluster. Nor-

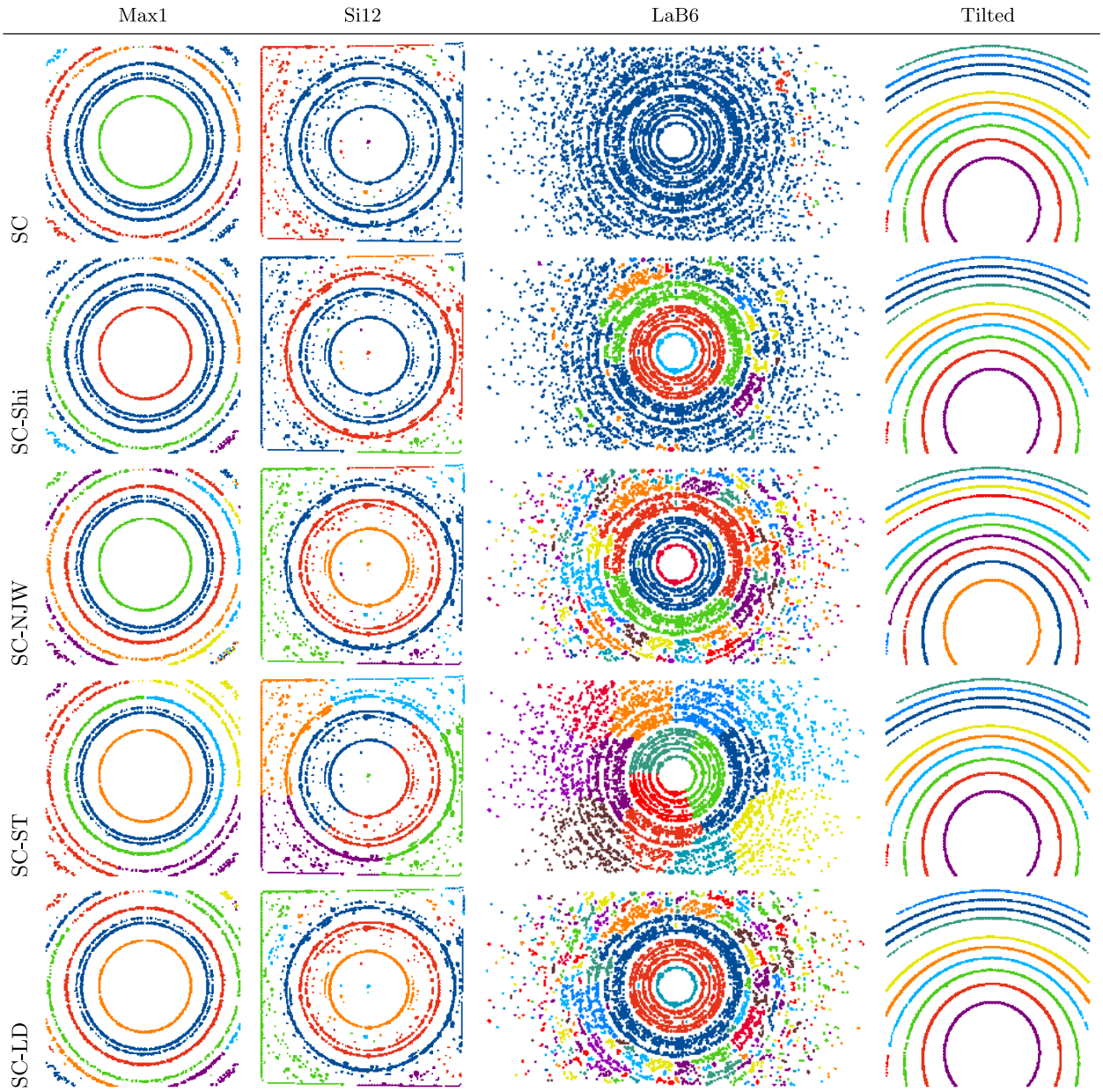


Fig. 7. Results of various spectral clustering algorithms for XRPD images from both orthogonal (cols 1-3) and non-orthogonal (col 4) detectors. Each color in the XRPD image represents one cluster. Most spectral algorithms correctly identify some or all rings in noiseless images(cols 1 and 4), however their performance decreases as the spottiness of the rings increases (cols 2 and 3). For the most noisy image (col 3), spectral clustering is only able to detect the inner most ring at best.

malized spectral clustering produces the most accurate clusters when $\mathbf{L}_{sym}$ is used (SC-NJW). It detects more rings in Max1 than SC-Shi, where as the result in Tilted is also better producing distinct clusters for each ring. It can be observed that this method identifies more and considerably better rings for noisy images as compared to the other two spectral clustering algorithms (SC and SC-Shi). The results in Si12 and LaB6 demonstrate that at least one ring is correctly identified. Particularly in LaB6, the clusters produced are mostly elliptical in nature as compared to hierarchical methods which either fail to identify any elliptical structure at all, or cluster all points into

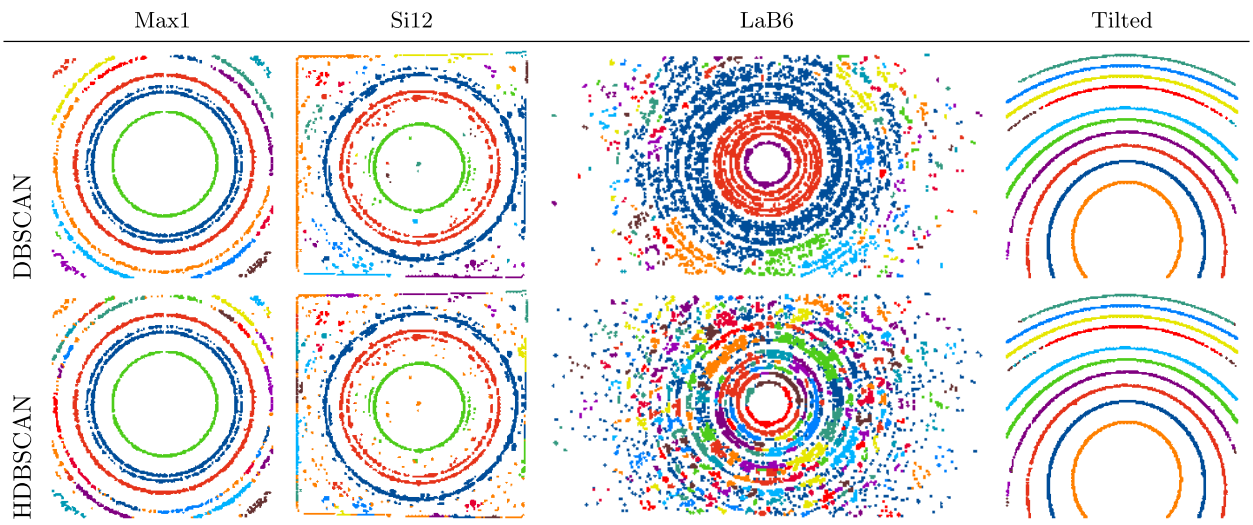


Fig. 8. Results of density-based clustering algorithms for XRPD images from both orthogonal (cols 1-3) and non-orthogonal (col 4) detectors. Each color in the XRPD image represents one cluster. Both algorithms produce nearly similar results, correctly identifying some or all rings in noiseless images. Both algorithms perform well for moderately noisy images (col 2), but are only able to detect one ring in the high noise XRPD image (col 3).

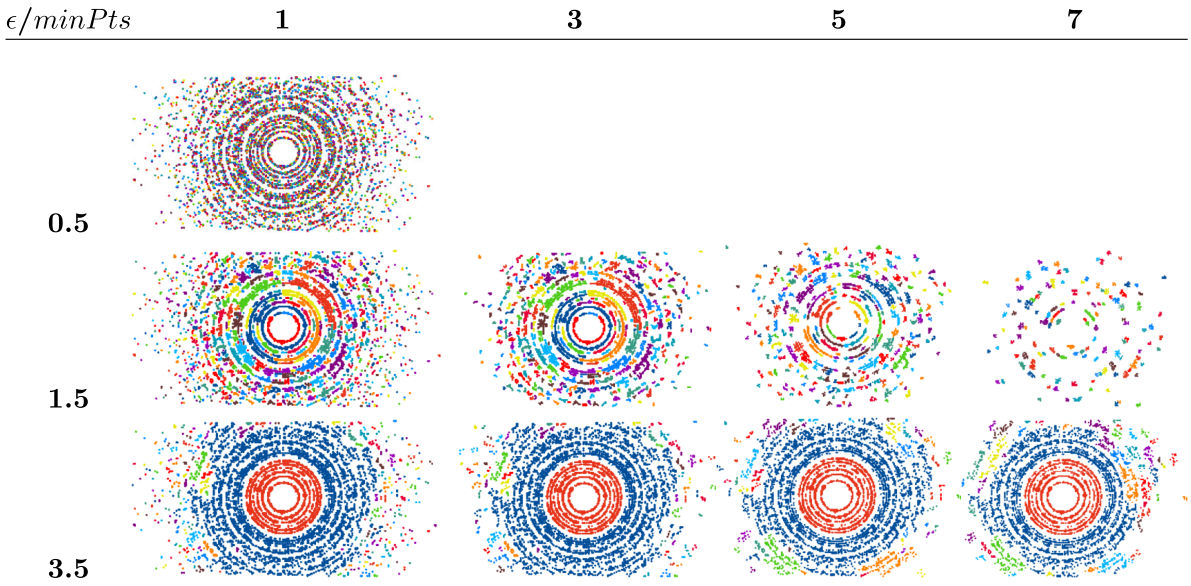


Fig. 9. Role of parameters $minpts$ and ϵ in DBSCAN for high noise XRPD image LaB6. The points from the outer rings are gradually removed as outliers by the algorithm when $minPts$ is increased from 1 to 7 for $\epsilon = 1.5$. Notice how the rings tend to get clustered together when ϵ is increased. Also, for very small ϵ (0.5) the algorithm results in empty clusters as $minPts$ is increased.

one group. Spectral clustering methods SC-ST and SC-LD which define a modified similarity function and subsequently compute L_{sym}, produce similar results as SC-NJW only for Tilted. For all the other images, SC-ST fails to identify any elliptical structure in data and produces results similar to the hierarchical algorithms. SC-LD, however, obtains clusters comparable to SC-NJW and identifies a few inner-most rings correctly. This shows that using a modified similarity function does not necessarily capture the underlying patterns in data more effectively than the baseline normalized algorithm SC-NJW.

Finally, the density-based method DBSCAN pro-

duces clusters comparable in accuracy to the best performing normalized spectral clustering algorithm (SC-NJW). It does not require any prior information on the number of clusters, but the choice of $minPts$ and ϵ affect the clustering results. DBSCAN works well both for noiseless and moderately noisy images. For example in Max1, Si12 and Tilted at least two rings per image were detected correctly. Since DBSCAN is robust to noise, it is able to detect multiple rings in Si12. However, it is not able to correctly identify rings in the high noise XRPD image (LaB6). This happens because rings occurs very close to each other in noisy images which makes it difficult to choose an ϵ radius that yields clusters which are neither too small nor too large. Increasing the value of ϵ tends to include points of adjacent rings in one cluster, where as decreasing it can split a ring in multiple parts thus forming more than one clusters of it. The results produced by HDBSCAN are comparable to DBSCAN for noiseless to moderately noisy images. However, it fails to identify any rings accurately for very noisy images.

It can be observed in Fig. 9 that noisy images have varying densities of points in rings, where the inner most rings have a higher spatial density and the outer rings have more scattered points with a lower density. Since DBSCAN is designed to remove outliers in data, it starts labeling and removing data points as noise in high noise XRPD images if appropriate values of ϵ and $minPts$ are not chosen. As the value of ϵ is increased all the inner rings get clustered together. Similarly, increasing the value of $minpts$ results in less data points being qualified as core points and therefore being removed as noise. This behavior is observed for all values of ϵ. Table 2 presents the quantitative results corresponding to Fig. 9 for different parameters values of $minpts$ and ϵ.

The visual results of clustering demonstrate that normalized spectral clustering and DBSCAN detect better Debye-Scherrer rings as compared to agglomerative clustering. Both normalized spectral clustering and DBSCAN perform equally well for noiseless and moderately noisy images. For highly noisy images, both algorithms identify the inner most ring accurately. However, DBSCAN does not require the knowledge of c in advance, which is a pre-requisite for spectral algorithms. Also, it is computationally much faster than spectral clustering which takes $O(N^3)$ time during the eigen decomposition step which may become a performance bottleneck for very large data sets, such as the XRPD images. The worst case running time of DBSCAN is $O(N^2)$ and may be reduced to $O(N \log N)$ with optimized implementations.This makes DBSCAN the algorithm of choice for detecting Debye-Scherrer rings from XRPD images. Empirical comparison of running times of the two algorithms for different

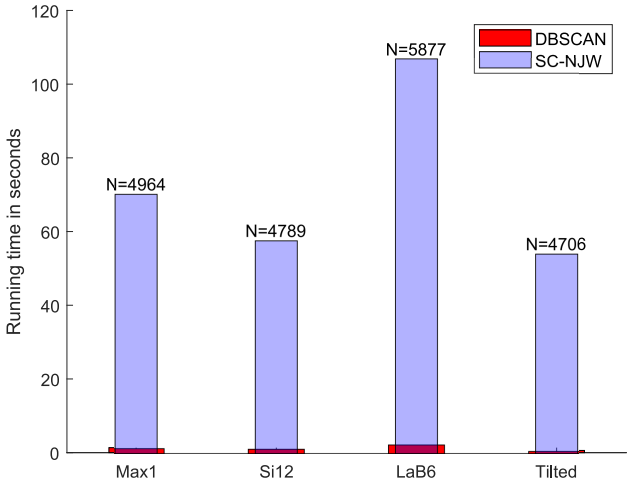


Fig. 10. Plot of average running times in seconds of DBSCAN and SC-NJW for different XRPD images. It can be seen that DBSCAN is computationally much more efficient than SC-NJW for all images having varying number of data points.

images with varying number of data points is presented in Fig. 10. The two different bars represent the average time in seconds of the two algorithms for 50 runs. DBSCAN is approximately 59 times more efficient as compared to spectral clustering on average.

3.1 Effects of removing noise

This section explores how clustering behaves when XRPD images are subject to noise removal techniques to get cleaner rings. The noisy spots around rings can be removed by finding and filtering connected components in the image based on their sizes, where the threshold of size is given by percentiles. So, all connected components below a given size percentile are removed from the image. Fig. 11 shows how the results of HC-Sng and SC improve after filtering noise from Si12, where the algorithm starts to successfully detect the rings in the image as noise is progressively removed.

Fig. 11 also illustrates the behavior of better-performing clustering algorithms such as SC-NJW and DBSCAN after removing noisy spots. It can be seen that the denoising method employed also removes the signal along with noise. This means that the components that act as links between different regions of a ring are filtered when the size threshold is increased. This leads to producing disconnected regions in outer rings that relied on such connected components to be clustered together. Hence, it can be observed that the rings identified by both SC-NJW and DBSCAN deteriorate in quality above

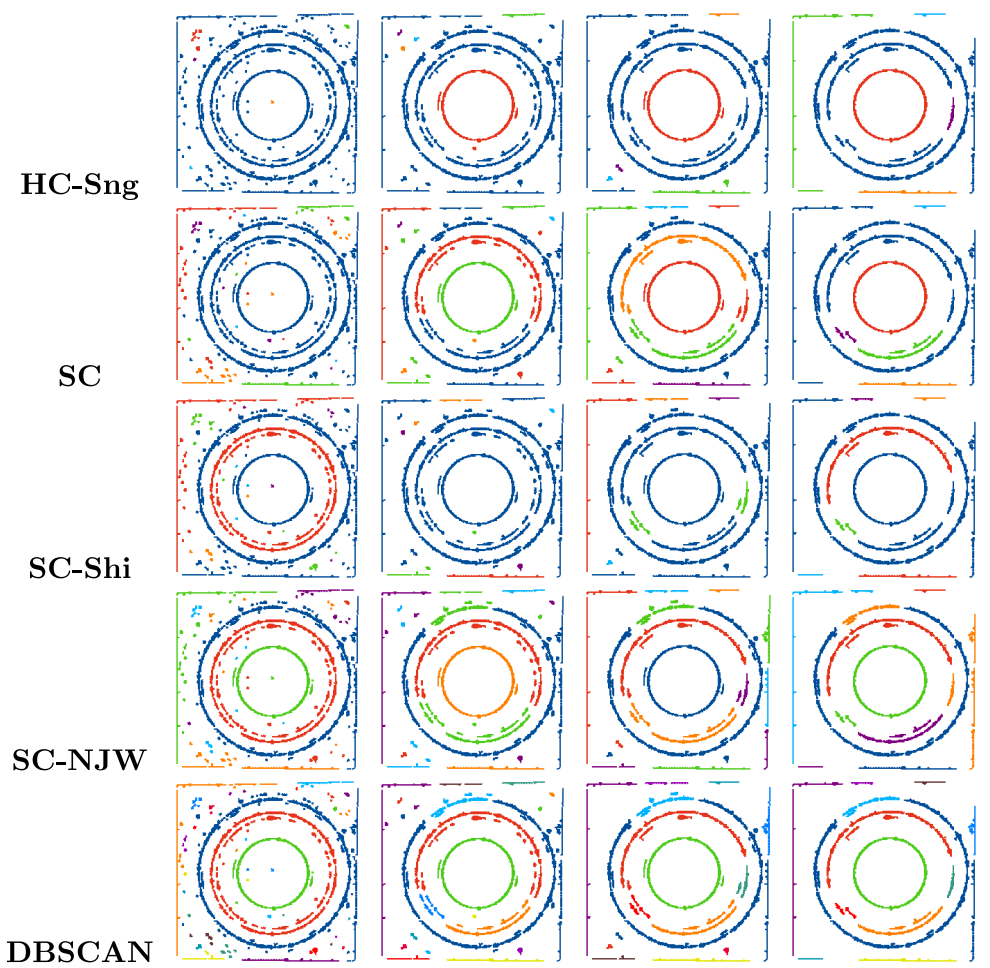


Fig. 11. Performance of hierarchical clustering with single linkage on Si12 after reducing noisy spots in the image. The noisy spots around rings are removed by finding and filtering connected components in the image based on their sizes, where the threshold of size is given by percentiles. The results show how the algorithm progresses from identifying no rings in the first column when the noise threshold is set at 50^{th} percentile, to identifying the inner most ring and some segments of outer rings for higher percentiles.

the 50^{th} percentile value. Spectral clustering depends on strong intra-cluster similarities between points to achieve an optimal cut on the graph. These similarities decrease as a result of removing intermediate points. Therefore spectral clustering fails to capture the neighborhood information correctly and breaks a ring into multiple clusters. Similarly, DBSCAN identifies groups of core points and builds clusters around these. The algorithm creates clusters of isolated components when linking neighborhood points are removed. Therefore, it results in identifying multiple disconnected clusters in each ring. This phenomenon is not observed for the inner most ring for both the algorithms. This is mainly because no components have been removed as noise from this ring.

4 QUANTITATIVE RESULTS

The accuracy of clusters identified by a clustering algorithm need to be evaluated using multiple criteria [28]. *External* evaluation requires creating ground truth cluster labels or a gold standard, to which predicted clusters labels are compared. This external benchmark is usually created by human experts by labeling each data point with a true cluster label. In *internal* evaluation the predicted clustering is checked against a given criteria, for example how close or separated are the data points in different

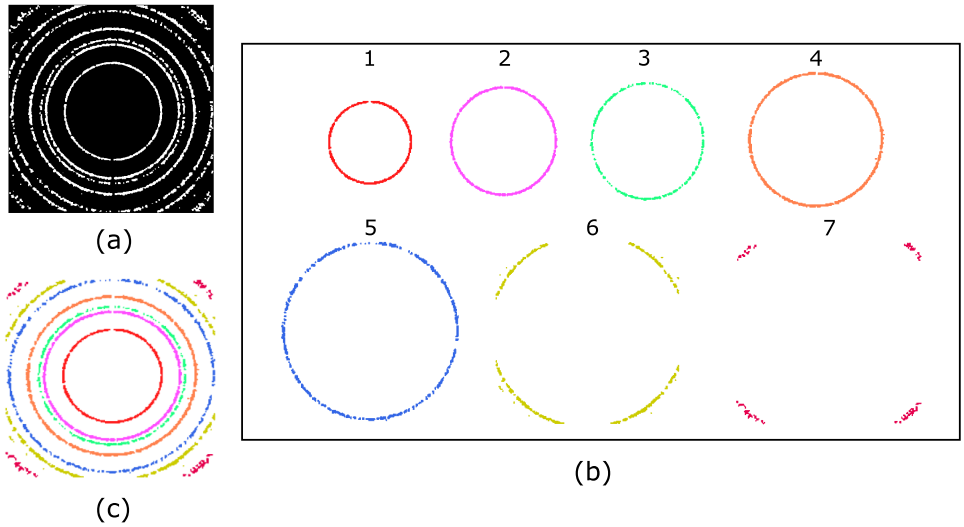


Fig. 12. Manually marked ground truth Debye-Scherrer rings. (a) The original binarized XRPD image having $r = 7$ rings. (b) Each ring is saved as a separate image and its data points are assigned a unique ground truth label. (c) Marked rings combined in an image which is input to the clustering algorithms.

Table 2. Quantitative scores for DBSCAN with different parameter values corresponding to Fig. 9

$minpts$ / ϵ	1			3			5			7		
	NMI	RI	ARI	NMI	RI	ARI	NMI	RI	ARI	NMI	RI	ARI
0.5	0.53	0.91	-5.2e-6	0.007	0.09	3.5e-5	0	0.08	0	0	0.08	0
1.5	0.51	0.8	0.13	0.5	0.8	0.14	0.48	0.78	0.17	0.49	0.72	0.18
3.5	0.25	0.34	0.02	0.26	0.34	0.02	0.31	0.39	0.03	0.35	0.5	0.03

clusters.

4.1 External Evaluation Criteria

For quantitative evaluation using external criteria, the data points belonging to individual rings in a binarized XRPD image are manually marked with ground truth labels. Fig. 12 shows this marking process. Each ring is manually isolated from the rest and stored as a separate binary image. Data points from these separate binary images are read in a vector and points belonging to each ring are assigned a unique label (1, 2, 3, ..., r), where r is the total number of rings in the image. All images are of size 200×200 and the results are reported as mean values of 50 runs of each algorithm on each image. The clustering accuracy of the algorithms is evaluated using multiple external criteria.

Normalized mutual information is used to estimate the quality of clustering in an information theoretic manner [29]. NMI takes values in $[0, 1]$, where higher values of NMI show good clustering results, with a value of 1 indicating that the two clusterings (ground truth and predicted) are identical. Since it is normalized, the values of NMI for different clustering results (with varying number of clusters) can be compared. Table 3 gives the NMI values of different clustering algorithms when applied to XRPD images.

In addition to NMI, *Rand index* (RI) [30] can be used to measure cluster quality. Theoretically, RI lies between 0 and 1, with 1 indicating a perfect match between ground truth C and predicted clustering $\hat{C}$. However, in practice RI of even a pair of random partitions is centered around values close to 1, especially as the numbers of clusters (c) becomes large. This is because RI does not take into account chance cluster assignments. Instead, it rewards mistakes made in the clustering, thereby increasing the numerator value and therefore the overall RI score. Table 4 provides the RI values for different clustering algorithms.

Table 3. NMI score for clustering algorithms for XRPD images corresponding to Figs. 6, 7 and 8

	Max1	Si12	LaB6	Tilted
HC-Comp	0.03	0.06	0.20	0.33
HC-Single	0.67	0.05	0.03	0.97
HC-Avg	0.03	0.05	0.26	0.40
Diana	0.14	0.03	0.23	0.35
SC	0.67±3.6e-16	0.27±0.2120	0.06±0.0139	0.97±0.0034
SC-Shi	0.67±3.5e-16	0.50±0.0862	0.46±0.0420	0.97±0.0069
SC-NJW	0.70±0.0167	**0.68±0.0226**	**0.44± 0.0083**	**0.99±0.110**
SC-ST	0.68±0.022	0.34±0.016	0.37± 0.078	0.97±0.011
SC-LD	0.71±0.022	0.67±0.023	0.43±0.011	0.97±0.011
DBSCAN	**0.83**	0.65	0.43	**0.99**
HDBSCAN	0.75	0.65	0.43	0.98

Table 4. RI score for clustering algorithms for XRPD images corresponding to Figs. 6, 7 and 8

	Max1	Si12	LaB6	Tilted
HC-Comp	0.73	0.69	0.81	0.84
HC-Single	0.77	0.23	0.12	0.98
HC-Avg	0.74	0.70	0.82	0.84
Diana	0.72	0.69	0.79	0.82
SC	0.77±1.1e-15	0.41±0.1956	0.15 ±0.0098	0.9853±0.0023
SC-Shi	0.77±1.1e-15	0.62±0.1207	0.68±0.0846	0.98±0.0082
SC-NJW	0.88±0.0107	**0.83±0.0201**	**0.84±0.0023**	**0.99±0.0059**
SC-ST	0.85±0.022	0.74±0.0182	0.82± 0.003	0.98±0.002
SC-LD	0.88±0.018	0.83±0.008	0.83±0.003	0.98±0.0081
DBSCAN	**0.92**	**0.83**	0.69	**0.99**
HDBSCAN	0.91	0.85	0.67	0.99

Unlike RI, the *adjusted Rand index* (ARI) [31] takes into account chance cluster assignments. The ARI is typically lower than RI and has an upper bound of 1 for a perfect match and a value of 0 represents random agreement of the two clusterings. Occasionally, ARI can also assume negative values close to zero, indicating that the RI of two clusterings is less than the expected RI. This means that the two clusterings are completely different and their similarity is less than the expected similarity as calculated using random clustering. Table 5 provides the ARI values for different clustering algorithms.

From the numerical scores it can be seen that normalized spectral clustering with $\mathbf{L}_{sym}$ [22] yields the highest NMI, RI and ARI values in all results for noisy images. For noiseless XRPD images also, it gives higher values as compared to un-normalized and normalized spectral clustering using $\mathbf{L}$ and $\mathbf{L}_{rw}$ respectively. For noiseless images Max1 and Tilted, the NMI, RI and ARI scores of DBSCAN are higher than SC-NJW. DBSCAN gives scores as good as SC-NJW for moderately noisy Si12. How-

Table 5. ARI score for clustering algorithms for XRPD images corresponding to Figs. 6, 7 and 8

	Max1	Si12	LaB6	Tilted
HC-Comp	0.01	0.02	0.04	0.13
HC-Single	0.42	-0.01	-0.0014	0.92
HC-Avg	0.01	0.03	0.08	0.17
Diana	0.04	0.04	0.06	0.15
SC	0.42±4.4e-16	0.12±.1599	-0.004±0.001	0.93±0.011
SC-Shi	0.42±4.5e-16	0.28±0.121	0.19±0.0642	0.91±0.0357
SC-NJW	0.61±0.0254	**0.54±0.0389**	**0.17±0.0083**	0.98±0.0289
SC-ST	0.65±0.024	0.17±0.018	0.13± 0.078	0.94±0.002
SC-LD	0.60±0.019	0.54±0.028	0.17±0.0021	0.95±0.003
DBSCAN	**0.73**	0.52	0.13	**0.9955**
HDBSCAN	0.66	0.56	0.06	0.99

ever, the scores for LaB6 tend to be slightly lower than that of normalized spectral clustering algorithm SC-NJW. This supports the visual results given in Figs. 7 and 8 where DBSCAN detects only one ring for LaB6. Since XRPD images are inherently noisy, therefore, clustering techniques that are robust to noise may effectively identify Debye-Scherrer rings thereby preventing errors in the data reduction phase. It is also observed that the RI scores are relatively higher than the NMI and ARI values for the same algorithms. Consequently, the RI for agglomerative clustering with complete and average linkages are considerably higher as compared to their NMI and ARI scores. The NMI and ARI values for these algorithms are consistent with the visual results as these methods fail to detect any elliptical patterns correctly. This reiterates the previous discussion that higher RI values are observed due to chance cluster assignments even when the two clusterings (predicted and ground truth labels in this case) have very less overlap. This becomes most prominent in case of noisy image LaB6 which has ARI values centered around 0 for agglomerative methods but has RI values as high as 0.8. It can be concluded from the quantitative results that spectral and density-based clustering perform well on complex and noisy XRPD data, detecting either all or at least a few inner most Debye-Scherrer rings. However, the eigen decomposition step in spectral algorithms can become computationally intensive for large data sets,thus taking more time than the density-based clustering methods, as previously discussed in Section 3.

5 CHOICE OF CLUSTERING PARAMETERS

Clustering results are sensitive to the choice of parameters. Different sets of parameters are required by different clustering algorithms to produce clusters that are representative of the intrinsic structure of data. The most important of these is the number of clusters c needed by most algorithms and mandatory for producing correct clustering results. It is especially true in the case of noisy XRPD images where sections of separate rings may get assigned to the same cluster for a certain value of c. Changing the value of c can produce different results for the same data which can lead to the detection of different rings in the same image. This phenomenon can further result in inaccuracies during the calibration and data reduction steps. It can be seen in Fig. 13 that for the given image, $c = 12$ and $\sigma = 1.8$ also produce the most number of Debye-Scherrer rings, despite the number of clusters being different from the actual rings identified in the ground truth for this image (i.e., $c = 7$). Finding c in clustering algorithms is a non-trivial problem and highly data-dependant. However, some heuristics can be employed to estimate c from data and are discussed in Section 5.1.

In addition to c, the scaling co-efficient σ used in the Gaussian similarity function plays a crucial role in how accurately spectral clustering identifies good clusters. It controls how fast the affinity between two points decreases as their Euclidean distance increases [22]. Clustering results are highly sensitive to its value and it is observed that the same data may produce different clusters for slightly different σ. Each value of σ generates a dif-

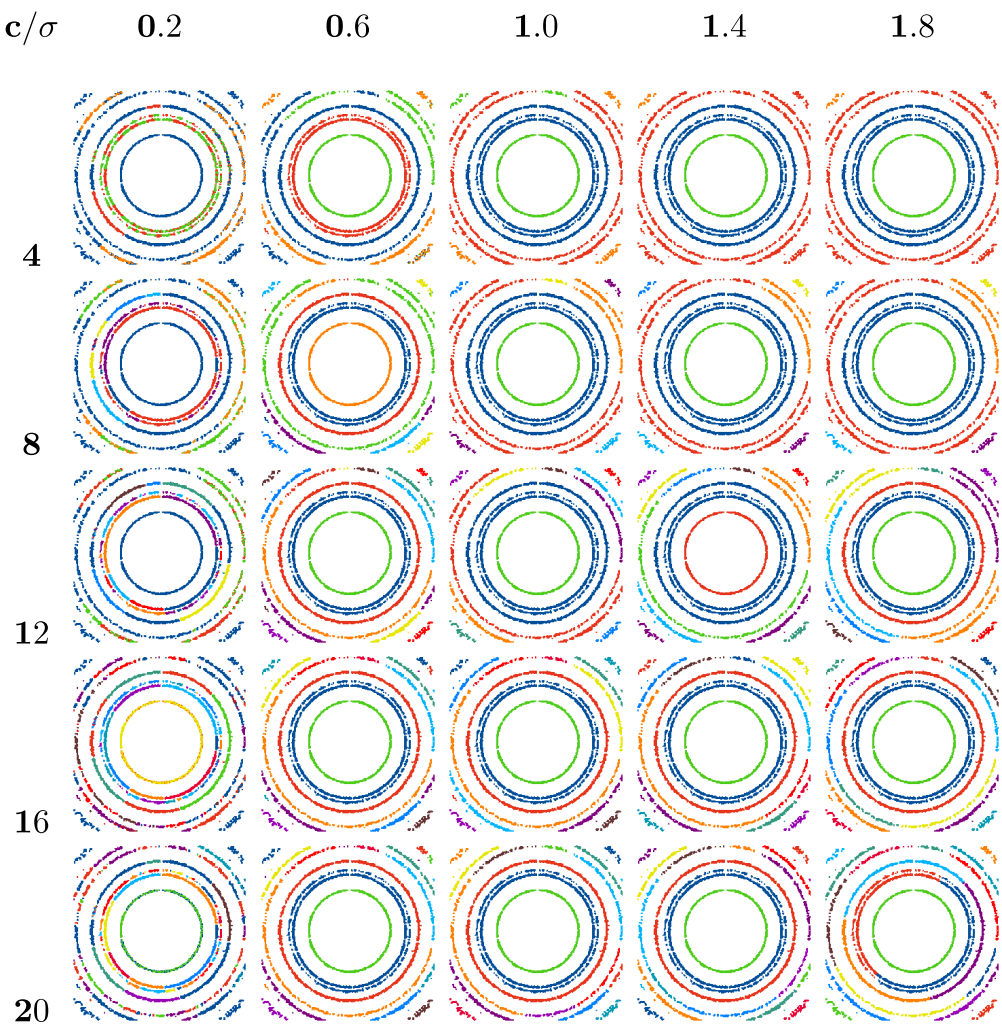


Fig. 13. Normalized spectral clustering results for varying values of parameters c and σ. (Row-wise) Each value of σ generates a different affinity matrix, thus producing different clusters for the same value of c. (Column-wise) The value of c controls how many clusters are identified in the image.

ferent affinity matrix, thus producing different clusters for the same value of c. Fig. 13 shows that some values of σ (0.6) yield good clustering results for different variations of c. However, for other values (1.0 - 1.8), clustering results vary when c is changed, with over-estimated number of clusters giving better results as compared to ground truth value. Thus, even if a good estimate for the number of clusters is found, the results are heavily dependant on choosing a good value of σ (see row 2 in Fig. 13). Similar to c, there are no theoretical results on how to find the optimal sigma and is dependent on the structure and noise in data. Ng *et al.* suggest choosing the best value of sigma that produces c tight clusters of data through parameter tuning [22].

5.1 Estimating Number of Clusters

The accuracy of most clustering techniques (with the exception of DBSCAN and agglomerative clustering) depends on prior knowledge of the number of clusters in data. However, real data sets are not supplemented with this information beforehand. The automatic estimation of the number of clusters during clustering is a non-trivial task and greatly depends on the type of data. This section discusses some heuristics that can be used to approximate the number of clusters c from data, specifically for spectral clustering techniques.

Eigengap One method to estimate c in spectral clustering algorithms is to use the *spectral* or *eigen* gap heuristic as proposed in [20]. The idea is to find the first

largest *jump* in eigenvalues of the graph Laplacian of data, starting with the *largest* eigenvalue λ_n corresponding the largest eigenvectors (for $\mathbf{L}_{sym}$). The eigengap can be found as $|\lambda_{i+1} - \lambda_i|$. The values of c_e as determined using the eigengap heuristic are given in Table 6. Normalized graph Laplacian $\mathbf{L}_{sym}$ is used to determine number of clusters for different XRPD images. It does not produce the exact number of clusters as identified in the ground truth. However, for images with well pronounced clusters and minimal noise, the estimated number of clusters yield a few rings correctly. In contrast, for XRPD data containing noise and non-uniformly spaced rings, the number of clusters c_e estimated by the eigengap heuristic has overestimated values.

Multiplicity of eigenvalues The graph Laplacian of data exhibits certain properties as discussed in Section 2.2. The eigen decomposition of $\mathbf{L}$ yields one eigenvector corresponding to eigenvalue 0 if there is a path between all vertices of the graph, thus forming one connected component. This will lead to generating only one cluster encompassing all data points.

Let us assume that the data has $c > 1$ clusters. Let us also assume that the data points belonging to each cluster are ordered one after the other such that $C_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}\}, C_2 = \{\mathbf{x}_{n_1+1}, \ldots, \mathbf{x}_{n_1+n_2}\}, \ldots, C_c = \{\mathbf{x}_{\sum_{j=1}^{c-1} n_j+1}, \ldots \mathbf{x}_{\sum_{j=1}^{c-1} n_j+n_c}\}$, where n_i is the cardinality of cluster C_i . The graph of this data has multiple connected components and the graph Laplacian takes a block diagonal form,

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_1 & 0 & \ldots & 0 \\ 0 & \mathbf{L}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \mathbf{L}_c \end{pmatrix} \tag{13}$$

where $\mathbf{L}_j$ is the graph Laplacian corresponding to the j^{th} connected component of the graph. The eigenvalues of $\mathbf{L}$ are a union of the eigenvalues of all $\mathbf{L}_1, \mathbf{L}_2, \ldots, \mathbf{L}_c$. This also implies that eigenvalue 0 has c occurrences, one for each $\mathbf{L}_j$. Thus, the multiplicity of eigenvalue 0 is equal to the number of connected components of the graph and consequently the number of clusters present in data. This is true even when the data is not ordered according to clusters, which is generally the case. For normalized graph Laplacian $\mathbf{L}_{sym}$, the multiplicity of eigenvalue 1 serves the same purpose, because the largest eigenvectors of the Laplacian are selected. This multiplicity value can be used to estimate the number of clusters for spectral clustering algorithms.

Table 6. Estimated number of clusters using eigengap and multiplicity heuristics.

	Max1	Si12	LaB6	Tilted
Ground truth (c_g)	7	6	14	10
Eigengap (c_e)	38	44	4	11
Multiplicity (c_m)	31	43	4	11

The values of c_m determined by multiplicity of eigenvalue 1 using symmetric normalized graph Laplacian $\mathbf{L}_{sym}$ for different XRPD images are given in Table 6. It can be seen that for noiseless images having well-separated rings (Tilted), this value matches the number of Debye-Scherrer rings as identified manually in ground truth. However, for images where either rings are spotty or very close to each other (Max1, Si12 and LaB6), the estimated number of clusters c_m using multiplicity heuristic is greater than the ground truth estimation c_g.

Fig. 14 shows eigenvalue plots of different X-ray images. It can be seen that for noisy images both multiplicity and spectral gap heuristics yield more clusters than the ground truth estimation.

The results produced by spectral clustering using the above heuristics are given in Fig. 15. It shows three different sets of results for number of clusters estimated via ground truth, multiplicity and eigengap denoted as c_g, c_m and c_e respectively. The values of these parameters are taken from Table 6. The clusters are ranked in descending order with respect to their cardinality and colors are assigned based on the rank, as can be seen in the histogram corresponding to each result. It can be seen from Table 6 and Fig. 15, both eigengap and multiplicity heuristics over estimate the number of clusters. However, a pattern similar to the histograms generated from clustering results using c_g can be observed in the histograms for results using c_e and c_m. The inner most rings are correctly identified in all three sets of images. Thus it can be concluded that even with overestimated number of clusters, the main clusters present in results using c_g also exist in the results for values estimated via eigengap and multiplicity, where some points are assigned to smaller clusters. Tables 7 and 8 present the quantitative NMI, RI and ARI values corresponding to number of clusters estimated using the eigengap and multiplicity heuristics respectively. It can be seen that the despite the overestimation of c_e and c_m, the numeric scores corresponding to these heuristics coincide with those obtained when using the ground truth values of c as given in Tables 3, 4 and 5.

It can also be observed from Fig. 15 that over es-

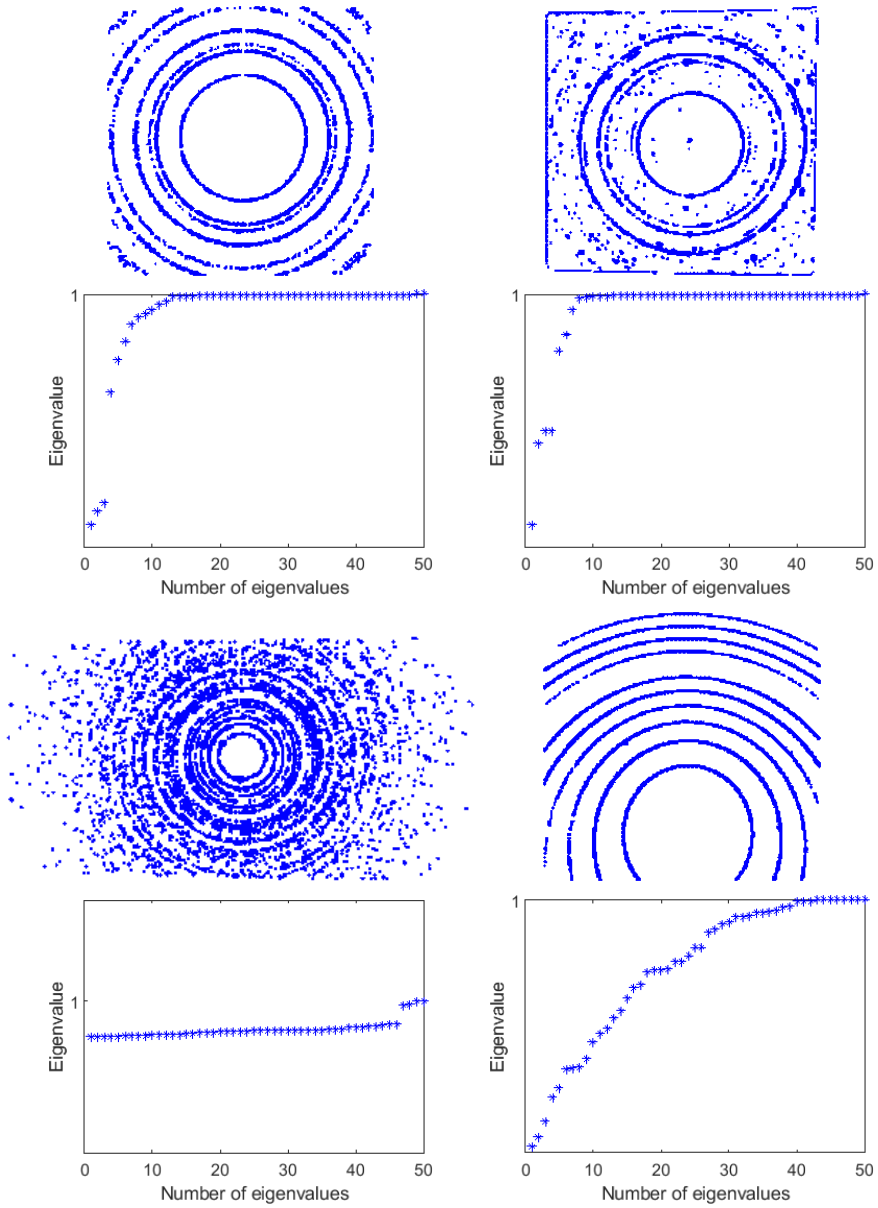


Fig. 14. Eigenvalue plot of different XRPD images. (Top) X-ray image. (Bottom) Plot of the largest forty five eigenvalues $\lambda_{n-49}, \ldots, \lambda_n$ of $\mathbf{L}_{sym}$ corresponding to each image. It can be seen that for noisy images both multiplicity and eigengap heuristics do not yield correct number of clusters.

timating number of clusters can actually benefit the accurate detection of Debye-Scherrer rings. All clustering methods try assigning each point to some cluster, thereby assigning noise to at least one cluster. This leads to clustering noise with rings, thus corrupting at least one (or more) Debye-Scherrer rings. The phenomenon can be seen in Fig. 16 for the moderately noisy image Si12. Fig 16 shows that while overestimation of c does not seem to be a good option, but in the presence of noise, it might actually be beneficial since noise gets assigned to a separate cluster instead of being grouped with some legitimate clusters.

6 CONCLUSIONS

This paper presents the application and analysis of clustering algorithms on XRPD images for the automatic

Table 7. NMI, RI and ARI scores for SC-NJW using the estimated number of clusters c_e via eigengap

	Max1	Si12	LaB6	Tilted
NMI	0.76±0.001	0.64±0.0003	0.39±0.03	0.99±1.2e-16
RI	0.89±0.0003	0.83±7.2e-5	0.71±0.02	0.99±1.2e-16
ARI	0.61±0.002	0.52±0.0002	0.15±0.03	0.99

Table 8. NMI, RI and ARI scores for SC-NJW using the estimated number of clusters c_m via multiplicity

	Max1	Si12	LaB6	Tilted
NMI	0.80±0.0016	0.64±0.0003	0.39±0.03	0.99±1.2e-16
RI	0.92±0.0004	0.83±1.5e-5	0.71±0.02	0.99±1.2e-16
ARI	0.70±0.0017	0.52±5.3e-5	0.15±0.03	0.99

detection of Debye-Scherrer rings. The state-of-the-art solutions are mostly semi-automatic, whereas the few automatic approaches are highly dependent on the area detector's type and tilt. It provides a cluster-based interpretation of Debye-Scherrer rings and demonstrates visual and quantitative results of agglomerative, spectral, and density-based clustering algorithms on low to high noise XRPD images. The given approaches offer two benefits over traditional ring detection techniques: (i) clustering algorithms are not restrictive in terms of (or dependant on sample-to-detector distance and alignment of ellipses within the image, and (ii) they perform well for data generated from both orthogonal and non-orthogonal detectors. Experimental results prove that both spectral and density-based clustering are most effective in detecting Debye-Scherrer rings in noisy as well as noiseless XRPD images. Among the two, the density-based algorithms are computationally much more efficient as compared to spectral clustering and yield 59 times better running times. These also do not require a priori information on the number of clusters which make density-based algorithms the best choice for detecting Debye-Scherrer rings. It can be concluded from this research that spectral and density-based clustering generate encouraging results for the detection of Debye-Scherrer rings. These can be combined with traditional detection methods such as IED, in place of region growing to improve both the rings detection accuracy and time.

REFERENCES

[1] Barnes, P., Jacques, S., and Vickers, M., 2006, "Powder diffracion,".

[2] Shahzad, S., Khan, N., Nawaz, Z., and Ferrero, C., 2018, "Automatic Debye–Scherrer elliptical ring extraction *via* a computer vision approach," *Journal of Synchrotron Radiation,* **25**(2), Mar, pp. 439–450.

[3] Cullity, B., 1957, "Elements of x-ray diffraction," *American Journal of Physics,* **25**(6), pp. 394–395.

[4] Bramble, M., Flemming, R., and McCausland, P., 2014, "Grain size,'spotty'xrd rings, and chemin: Two-dimensional x-ray diffraction as a proxy for grain size measurement in planetary materials," In Lunar and Planetary Science Conference, Vol. 45, p. 1658.

[5] Bramble, M. S., Flemming, R. L., and McCausland, P. J., 2015, "Grain size measurement from two-dimensional micro-x-ray diffraction: Laboratory application of a radial integration technique," *American Mineralogist,* **100**(8-9), pp. 1899–1911.

[6] Vallcorba, O., and Rius, J., 2019, "d2dplot: 2d x-ray diffraction data processing and analysis for through-the-substrate microdiffraction," *Journal of Applied Crystallography,* **52**(2).

[7] Hammersley, A., 2016, "Fit2d: a multi-purpose data reduction, analysis and visualization program," *Journal of Applied Crystallography,* **49**(2), pp. 646–652.

[8] Hinrichsen, B., and Dinnebier, R., 2006, "Powder3d: An easy to use program for data reduction and graphical presentation of large numbers of powder diffraction patterns," *Zeitschrift fur Kristallographie Supplements,* **2006**, 06.

[9] Ashiotis, G., Deschildre, A., Nawaz, Z., Wright, J. P., Karkoulis, D., Picca, F. E., and Kieffer, J., 2015, "The fast azimuthal integration python library: pyfai," *Journal of applied crystallography,* **48**(2), pp. 510–519.

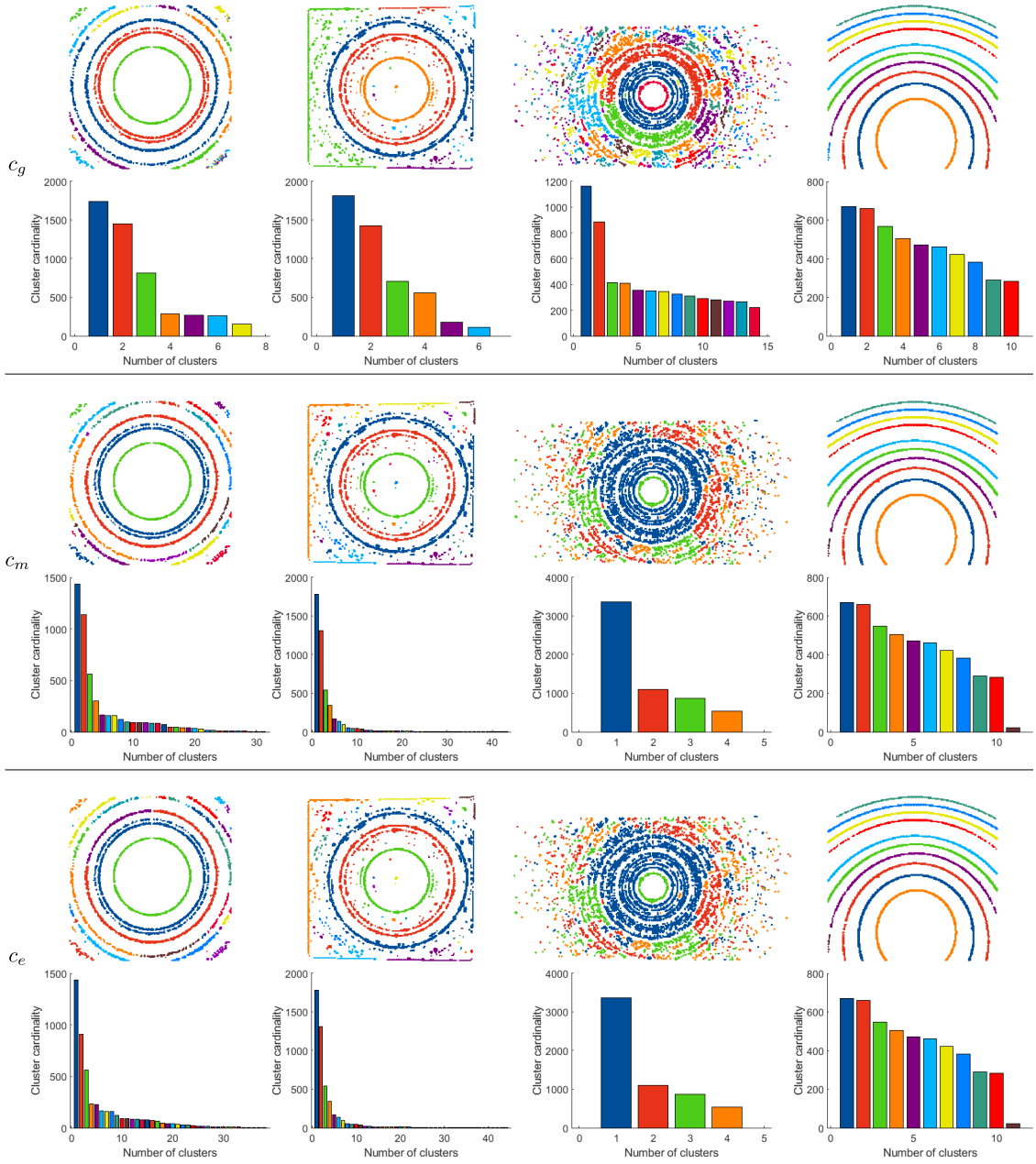


Fig. 15. Result of normalized spectral clustering using $\mathbf{L}_{sym}$ with automatic estimation of number of clusters. (Top row) Clustering results using ground truth estimation c_g along with histogram of cluster cardinalities. (Middle row) Results with estimation of c_m using multiplicity of eigenvalue 1 along with histograms. (Bottom row) Results with estimation of c_e using the eigengap along with histograms.

[10] Hart, M. L., and Drakopoulos, M., 2013, "Weighted least squares fit of an ellipse to describe complete or spotty diffraction rings on a planar 2d detector," *arXiv preprint arXiv:1311.5430*.

[11] Cervellino, A., Giannini, C., Guagliardi, A., and Ladisa, M., 2006, "Folding a two-dimensional powder diffraction image into a one-dimensional scan: a new procedure," *Journal of Applied Crystallography*, **39**(5), pp. 745–748.

[12] Rajiv, P., Hinrichsen, B., Dinnebier, R., Jansen, M., and Joswig, M., 2007, "Automatic calibration of powder diffraction experiments using two-

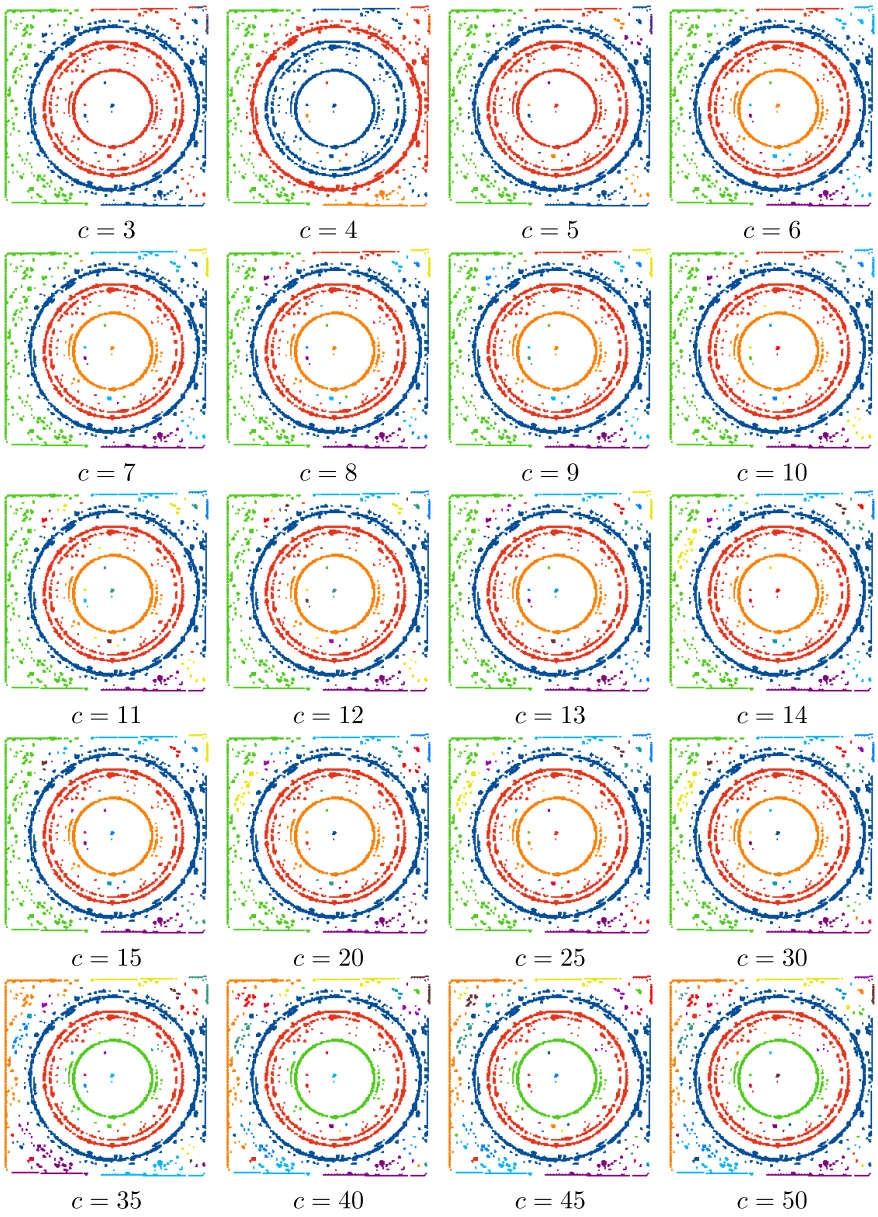


Fig. 16. Clustering results with increasing values of number of clusters for moderately noisy Si12. It can be observed that large values of c tend to produce separate clusters for noise for e.g. in $c = 50$, thus identifying better Debye-Scherrer rings. For smaller values of c, the noise gets clustered as part of at least one ring for e.g. $c = 3$.

dimensional detectors," *Powder diffraction,* **22**(1), pp. 3–19.

[13] Nagargoje, A., Kankar, P. K., Jain, P. K., and Tandon, P., 2021, "Performance Evaluation of the Data Clustering Techniques and Cluster Validity Indices for Efficient Toolpath Development for Incremental Sheet Forming," *Journal of Computing and Information Science in Engineering,* **21**(3), 02 031001.

[14] Xie, J., Mao, S., Zhang, Z., and Liu, C., 2022, "Data-Driven Approaches for Characterization of Aerodynamics on Super High-Speed Elevators," *Journal of Computing and Information Science in Engineering*, 06, pp. 1–17.

[15] Sabbagh, R., and Ameri, F., 2019, "A Framework Based on K-Means Clustering and Topic Modeling for Analyzing Unstructured Manufacturing Capabil-

ity Data," *Journal of Computing and Information Science in Engineering,* **20**(1), 09 011005.

[16] Choi, S.-S., Cha, S.-H., and Tappert, C. C., 2010, "A survey of binary similarity and distance measures," *Journal of Systemics, Cybernetics and Informatics,* **8**(1), pp. 43–48.

[17] Kaufman, L., and Rousseeuw, P. J., 2009, *Finding groups in data: an introduction to cluster analysis*, Vol. 344 John Wiley & Sons.

[18] Aryal, S., Ting, K., Washio, T., and Haffari, G., 2017, "Data-dependent dissimilarity measure: an alternative to geometric distance measure," *Knowledge and Information Systems,* **53**(2), pp. 479–506.

[19] Xu, R., and Wunsch, D., 2005, "Survey of clustering algorithms," *IEEE Transactions on neural networks,* **16**(3), pp. 645–678.

[20] Von Luxburg, U., 2007, "A tutorial on spectral clustering," *Statistics and computing,* **17**(4), pp. 395–416.

[21] Shi, J., and Malik, J., 2000, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107.

[22] Ng, A. Y., and Weiss, Y., 2002, "On spectral clustering: Analysis and an algorithm," In Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Conference, Vol. 2, MIT Press, p. 849.

[23] Zelnik-Manor, L., and Perona, P., 2004, "Self-tuning spectral clustering," *Advances in neural information processing systems,* **17**.

[24] Zhang, X., Li, J., and Yu, H., 2011, "Local density adaptive similarity measurement for spectral clustering," *Pattern Recognition Letters,* **32**(2), pp. 352–358.

[25] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., 1996, "A density-based algorithm for discovering clusters in large spatial databases with noise," In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, AAAI Press, p. 226–231.

[26] Hinneburg, A., and Keim, D. A., 2003, "A general approach to clustering in large databases with noise," *Knowledge and Information Systems,* **5**(4), pp. 387–415.

[27] Campello, R. J., Moulavi, D., and Sander, J., 2013, "Density-based clustering based on hierarchical density estimates," In Pacific-Asia conference on knowledge discovery and data mining, Springer, pp. 160–172.

[28] Pfitzner, D., Leibbrandt, R., and Powers, D., 2009, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowledge and Information Systems,* **19**(3), pp. 361–394.

[29] Strehl, A., and Ghosh, J., 2002, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research,* **3**(Dec), pp. 583–617.

[30] Rand, W. M., 1971, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association,* **66**(336), pp. 846–850.

[31] Hubert, L., and Arabie, P., 1985, "Comparing partitions," *Journal of classification,* **2**(1), pp. 193–218.