Angle-Based Clustering

Anna Beer(✉), Dominik Seeholzer, Nadine-Sarah Schüler, and Thomas Seidl

Ludwig-Maximilians-Universität München, Munich, Germany
{beer,schueler,seidl}@dbs.ifi.lmu.de,
d.seeholzer@campus.lmu.de

Abstract. The amount of data increases steadily, and yet most clustering algorithms perform complex computations for every single data point. Furthermore, Euclidean distance which is used for most of the clustering algorithms is often not the best choice for datasets with arbitrarily shaped clusters or such with high dimensionality. Based on ABOD, we introduce ABC, the first angle-based clustering method. The algorithm first identifies a small part of the data as border points of clusters based on the angle between their neighbors. Those few border points can, with some adjustments, be clustered with well-known clustering algorithms like hierarchical clustering with single linkage or DBSCAN. Residual points can quickly and easily be assigned to the cluster of their nearest border point, so the overall runtime is heavily reduced while the results improve or remain similar.

1 Introduction

If there are clusters in a dataset, most of the points lie rather in the middle of a cluster than at its border, and if the clusters of the border points are known, the assignment of inner points is easy and fast using a simple 1NN classification. To identify border points we suggest an angle based approach inspired by *Angle-Based Outlier Detection* (ABOD) [4], which is robust even for higher dimensionalities.

Our new clustering method ABC (Angle-Based Clustering), consists of three steps: First, by assessing the angles between difference vectors of points to their kNN, we can reliably identify points located at the boundaries of clusters. Secondly, we apply existing clustering techniques on those border points only, which allows us to reduce the number of points to be clustered severely. Finally, inner points are assigned to the same cluster as their nearest border point. As clustering has a higher complexity than the angle-based border point extraction as well as inner point assignment, the total runtime is dramatically reduced by clustering only a small fraction of all data points.

Our main contributions are as follows:

- Based on angles between a point and its kNN we detect the border points bounding clusters
- We apply adapted versions of DBSCAN and Hierarchical Single-Linkage Clustering on the border points

S. Satoh et al. (Eds.): SISAP 2020, LNCS 12440, pp. 312–320, 2020.
https://doi.org/10.1007/978-3-030-60936-8_24

– In experiments we show not only the speedup of algorithms using only the border points, but also the improvement of results regarding quality

2 Related Work

ABOD [4] was the first algorithm to use angles for outlier detection by regarding the variance of angles between the difference vectors of a point to all pairs of other points. Several works extended it regarding, e.g., acceleration [8], streams [13], and or stability [5]. ABSAD [14] uses angles between points and axis-parallel lines for an angle-based subspace anomaly detection method.

We could only find one work which uses angles in the field of clustering: SCUBI [11] combines classical clustering with detecting boundary information using angles to create a highly scalable clustering scheme. In contrast to our approach, they use the angles only for an approximation to an intrinsically density-based boundary extraction. Furthermore, we consider the previously calculated angles also for the clustering step by improving the distance function.

There are diverse approaches to identify border points: density based [11,12], hull based [7], and graph based [6] . Nevertheless, they lead to problems for higher dimensionalities, either regarding meaningfulness, or complexity.

3 Mathematical Background

Angles Between Data Points. Angles in a finite-dimensional real Euclidean vector space $\mathbb{V}^{\mathbb{R}}(\simeq \mathbb{R}^d, d \in \mathbb{N}, d \geq 2)$ are defined between any pair of vectors $A, B \in \mathbb{V}^{\mathbb{R}}$ with:

$$cos\Theta(A,B) = \frac{(A,B)_R}{|A|\,|B|}, \tag{1}$$

where $(A,B)_R = \sum_{k=1}^{d} A_k B_k$ is the scalar product between the two vectors and $|A| = \sqrt{(A,A)_R}$ [9]. For the resulting (real) angle $\Theta(A,B)$ the following holds true: $0 \leq \Theta \leq \pi$.

Directional Angle and Enclosing Angle. Figure 1 (left) shows the minimal angle for a point X between two difference vectors to its neighboring points which "encloses" all other neighboring points (green shape). We call it the *enclosing angle* Θ_{enc} of a point. One way to calculate the *enclosing angle* in two dimensions requires to calculate the directional angle between two vectors. In a 2d vector space with vectors $\overrightarrow{XY} = (u_1, u_2)$, $\overrightarrow{XZ} = (v_1, v_2) \in \mathbb{V}_2$, the counter-clockwise directional angle from $\overrightarrow{XY}$ to $\overrightarrow{XZ}$ is $\Theta_{YZ}(X) = atan2(u_2, u_1) - atan2(v_2, v_1)$. If the resulting Θ_{dir} is negative, we add 2π to receive only positive values between 0 and 2π. Figure 1 shows an example directional angle Θ_{dir}. Note, that if the directional angle is less than π, it will be equal to the cosine angle.

To obtain the enclosing angle of a point X, we calculate the directional angle between difference vectors to all pairs of neighbors and differentiate two cases:

First, if $\exists Y \in kNN(X) : \forall Z \in kNN(X) : \Theta_{YZ} \geq \pi$ as illustrated in Fig. 1 (middle), the enclosing angle can be calculated as $2\pi - min(\{\Theta_{YZ}|Y, Z \in kNN(X)\})$. Otherwise, the enclosing angle can be calculated as $2\pi - max(\{min(\{\Theta_{YZ}|Z \in kNN(X)\})|Y \in kNN(X)\})$, as shown in Fig. 1 (right). We can use the concept behind *enclosing angle* to characterize the relative position of neighboring points. Points in the center of a cluster tend to have much larger enclosing angles.

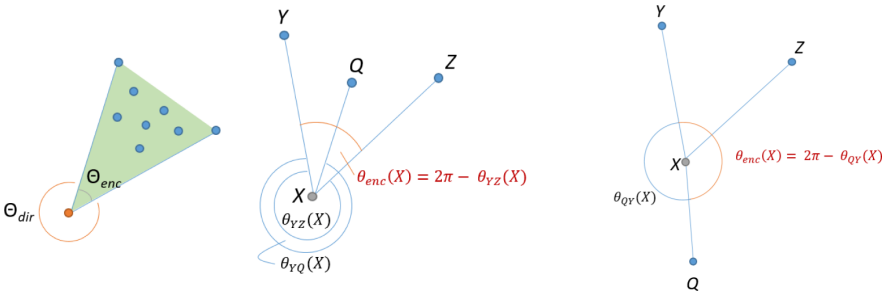

Fig. 1. Left: Enclosing Angle θ_{enc} and counter-clockwise Directional Angle θ_{dir}. Middle and Right: Example calculation of the enclosing angle θ_{enc}. (Color figure online)

4 ABC: Angle-Based Clustering Approach

ABC consists of three steps: First we calculate an angle-based border degree, see Sect. 4.1. The top β points with the highest border degree are the border points. Secondly, we cluster the border points using either an adapted DBSCAN or Hierarchical-Single Linkage Clustering, see Sect. 4.2. Finally, inner non-border points are assigned to cluster of their nearest border point. With a k-d tree this can be done in $O(n \log n)$.

4.1 Border Point Detection Based on Enclosing Angles

Because the nearest neighbors are all located in a similar direction for border points, their enclosing angle (see Sect. 3) tends to be much smaller compared to inner points. As we work with higher dimensionalities we use the following approximation: The *enclosing angle based border degree* is calculated as the maximum of all angles between the vector formed by query point to the kNN-mean and the vector from query point to one of the neighbors. Figure 2 (left) shows a simplified 2d example. The approximated enclosing angle θ_{enc} for border points tends to be much smaller than for inner points. The green shape encompasses the enclosed points.

The complete enclosing angle based border point extraction process proceeds as follows: For each point the kNN, the average distance to them, and the enclosing angles are calculated. For the *direction* of a border point, we use the vector

from the query point to the kNN-mean. Border points are then sorted by border degree and the $\beta \cdot n$ points with the highest border degree are returned as the *Boundary*. Figure 2 (middle and right) shows an example on a two dimensional dataset, where darker points imply a higher border degree.

Parameter Analysis. Small values for k can lead to inner points being falsely identified as border points, high values can lead to inter-cluster border points not being recognized as such, i.e., we only find the global boundary of all clusters. For datasets with many close clusters a small k should be preferred, while far separated clusters yield better results with a larger k.

The parameter β determines the separation threshold between border and inner points. Too high values yield more border points leading to a longer execution time of the subsequent clustering step. Too small values will fail to correctly identify enough cluster boundaries. In general, we have found values for β between 5–20% to yield optimal results.

4.2 ABC-DBSCAN/ABC-Hierarchical-SL

To cluster the boundary points we can use an adaption of DBSCAN [1] in which we regard also the *direction* of each border point to its neighbors. As border points that lie close to each other but have opposing directions are unlikely to belong to the same cluster, we use the following new the distance function instead of the Euclidean:

Definition 1. *Direction-Angle modified Distance Function*
Given two border points $A, B \in \mathcal{D}$ and their respective direction vectors $\boldsymbol{a}, \boldsymbol{b}$ as well as the Euclidean distance $d(A,B)_{eucl}$ between the points and the angle $\Theta(A,B)$ between their direction vectors. Then, given a direction-angle modifier σ_{mod} , the ***direction-angle modified distance*** *$d(A,B)_{mod}$ is calculated as:*

$$d(A,B)_{mod} = d(A,B)_{eucl} * (1 + (\frac{\sigma_{mod} - 1}{\pi}) * cos\Theta(A,B)) \tag{2}$$

A larger angle between the direction vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ results in a larger modified distance, where σ_{mod} controls the maximum. A higher σ_{mod} leads to more influence of direction-angle similarity compared to the Euclidean distance. When $\sigma_{mod} = 1$, then $d(A,B)_{mod} = d(A,B)_{eucl}$. A value of $\sigma_{mod} < 1$ increases the distance between points with different angle. Note, that this distance function does not represent a metric, since the triangle inequality does not always hold.

Another well suited approach to cluster border points is hierarchical agglomerative clustering using single linkage (Hierarchical-SL) [3]. Again with a complexity of $O(n^2)$, potential time savings using Angle-Based border point clustering are high. Also here we use the modified distance as described in Definition 1.

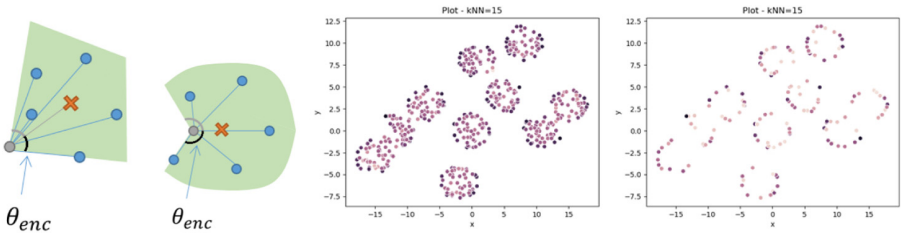

Fig. 2. Left: Approximated Enclosing Angles for border point and inner point. The red cross marks the mean of the blue kNN of the regarded gray point. Right: Border degree and selected border points ($k = 15$, $\beta = 0.2$). (Color figure online)

Complexity Analysis. Calculating the border degree requires an kNN query with complexity $O(n \log n)$ using a k-d tree [10]. The border degree calculation itself has complexity $O(n*k)$, as an angle between each nearest neighbor of each point and the mean of all its kNN is calculated. The sorting and selection of border points is $O(n \log n)$. In total, we get $O(n \log n + nk)$. As k is typically very small ($k \leq \log n$) the overall complexity is then $O(n \log n)$.

5 Experiments and Results

The following Sect. 5.1 covers results of experiments analyzing the runtime of algorithms. The quality on different kinds of datasets, both synthetic and real, are compared in Sect. 5.2 based on the Adjusted Rand Index (ARI).

5.1 Runtime

As ABC only requires to cluster a small fraction of all data points it is highly scalable and well suited for big datasets. Figure 3 (left) summarizes the experiments on how long each of the main three steps (border degree calculation, border point clustering and inner point assignment) take for an increasing number of points. As clustering is the most time consuming task with growing number of observations, reducing the amount of points having to be clustered significantly saves time.

As seen in Fig. 3 (right), ABC-DBSCAN outperforms the naive implementation of DBSCAN with time complexity $O(n^2)$. Even with the use of optimized index structures, the complexity of DBSCAN cannot be reduced below $O(n^{4/3})$ for higher dimensional data [2]. Thus, for large enough datasets, the ABC version with $O(n \log n)$ outperforms even optimized variants of DBSCAN.

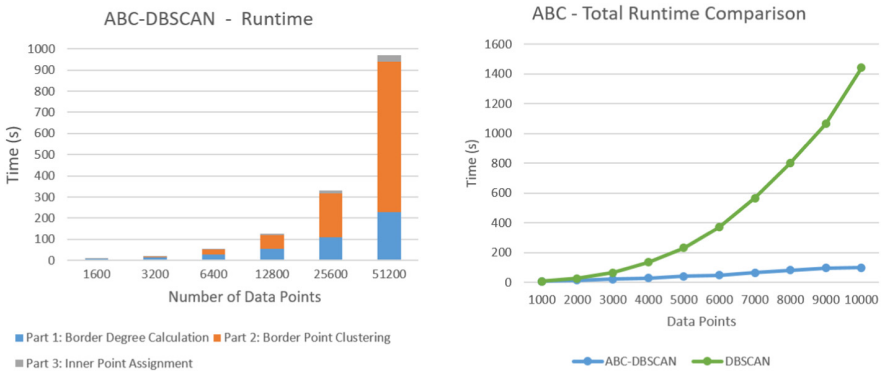

Fig. 3. Left: ABC-DBSCAN components runtime with $d = 5$, $\beta = 0.2$, $k = 10$. Right: Total runtime of DBSCAN and ABC-DBSCAN.

5.2 Quality

Datasets. First, we compare the quality of results on synthetic Gaussian data while modifying either cluster count, dimension count or standard deviation (the last one was left out due to the lack of space, even though ABC constantly outperformed the competitors slightly). The default dataset consists of $n = 1000$ data points, $c = 5$ clusters, $d = 5$ dimensions and a standard deviation $\sigma = 0.1$. Then, we test the algorithms on synthetic complex shaped data sets with and without noise. Finally, we investigate how they perform on real data sets.

Algorithms. We compare **ABC-DBSCAN** to the classic DBSCAN. Additionally, we compare it to **ABC-SCUBI-DBSCAN**, for which we adapt the idea of [11] and exclude a point from the DBSCAN ε-range if its angle is greater than $\pi/2$ (instead of our combined distance measure), but still use our border-degree measurement. Then, we compare the **ABC-Hierarchical-SL** approach to the classic Hierarchical-SL algorithm.

For ABC-DBSCAN and DBSCAN the same range of parameters is tested and the best result is kept. ABC-Hierarchical-SL and Hierarchical-SL get the correct amount of clusters given as the maximum cluster parameter. For the border point calculation, we used parameters $\beta = 0.3$ and $k = 15$. For the direction-angle modifier for ABC-DBSCAN and ABC-Hierarchical, we tested values $\sigma_{mod} \in \{0.1, 0.2, 0.3, 0.5, 1, 2, 5\}$ for different weightings of the angle compared to distance and kept the best result.

5.3 Synthetic Gaussian Distributed Data

Based on the dataset described above we varied the number of clusters c from 1 to 500, as shown in Fig. 4 (top). ABC-Hierarchical-SL outperforms the classical Hierarchical-SL, especially for higher c, where the latter only performs poorly. For the DBSCAN versions, the overall performance decreases with increasing c, but the ABC versions yield constantly better results than the original DBSCAN.

For varying dimensionalities $d \in [2, 1000]$. ABC-Hierarchical-SL as well as Hierarchical-SL converge towards an ARI of 1. The ABC version works slightly better even for small d. All DBSCAN based algorithms suffer from the *"curse of dimensionality"*, dropping to an ARI of 0 for high $d \geq 70$. ABC-DBSCAN still performs well for a much higher d than the classic DBSCAN.

5.4 Benchmark Datasets

To evaluate more complex cluster shapes, we also tested our algorithms with the *Complex9* dataset and its noisy version *Cluto-t7*. Both contain nine different types of clusters including blobs, moons and anisotropically distributed shapes. As depicted in Fig. 4 (bottom), ABC-Hierarchical-SL achieves near perfect results and outperforms the original, since the single link effect connecting two different Hierarchical-SL clusters is prevented by using our adapted distance measure. ABC-DBSCAN and ABC-SCUBI-DBSCAN are slightly outperformed by the original DBSCAN. In such cases, ABC could still be chosen with a trade-off between a huge improvement of the runtime and a rather small decrease of the quality. Results for the noisy dataset *Cluto-t7* show similar behavior, except for a significant improvement from ABC-Hierarchical-SL over the original.

Finally, we applied all algorithms on the real datasets *Iris*, *Seed*, and *Ecoli* from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml). In summary, the ABC versions performed at least comparatively well, in many cases even better than the original, as shown exemplarily in Fig. 4.

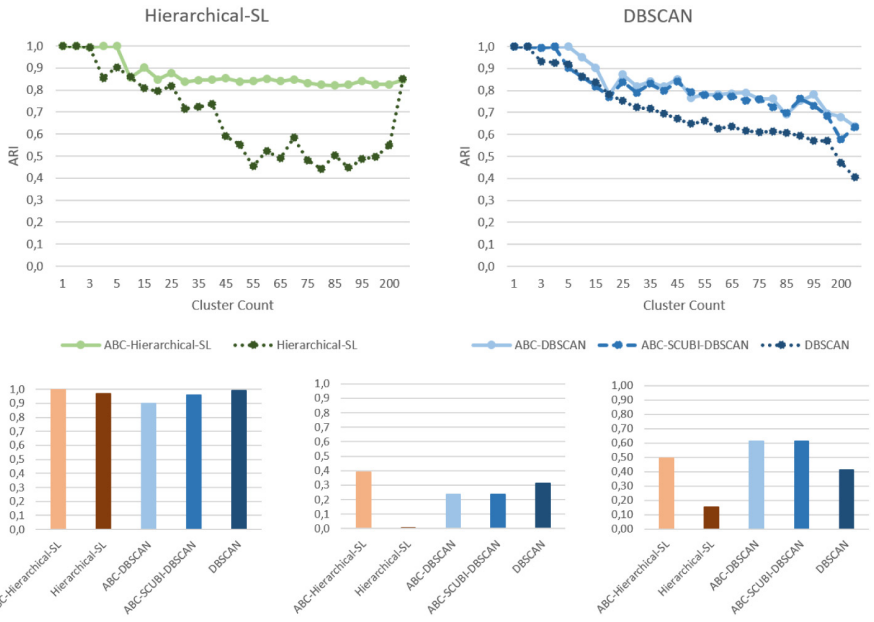

Fig. 4. Top: ARI of synthetic Gaussian distributed data for increasing number of clusters Bottom: ARI of Complex9 (left), Noisy Cluto-t7 (middle) and Ecoli (right)

6 Conclusion

We developed ABC, an angle-based clustering method, which is based on common clustering algorithms like DBSCAN and hierarchical Single-Link clustering, but many times faster as only the few cluster border points, have to be clustered by the respective algorithm. The points lying in the middle of a cluster can easily be assigned to the cluster of their nearest border point. We developed a method to detect those border points based on the angle enclosing their nearest neighbors, which is significantly smaller for points bordering a cluster than for those lying in the inner part. Experiments show that the results are similar or slightly better than those of the original algorithms on synthetic as well as on real world data.

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

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
2. Gan, J., Tao, Y.: DBSCAN revisited: mis-claim, un-fixability, and approximation. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 519–530. ACM (2015)
3. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**, 241–254 (1967). https://doi.org/10.1007/BF02289588
4. Kriegel, H.P., Zimek, A., et al.: Angle-based outlier detection in high-dimensional data. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 444–452. ACM (2008)
5. Li, X., Lv, J.C., Cheng, D.: Angle-based outlier detection algorithm with more stable relationships. In: Handa, H., Ishibuchi, H., Ong, Y.-S., Tan, K.C. (eds.) Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1. PALO, vol. 1, pp. 433–446. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-13359-1_34
6. Liu, D., Nosovskiy, G.V., Sourina, O.: Effective clustering and boundary detection algorithm based on Delaunay triangulation. Pattern Recogn. Lett. **29**(9), 1261–1273 (2008)
7. Moreira, A., Santos, M.Y.: Concave hull: a k-nearest neighbours approach for the computation of the region occupied by a set of points (2007)
8. Pham, N., Pagh, R.: A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 877–885. ACM (2012)
9. Scharnhorst, K.: Angles in complex vector spaces. Acta Applicandae Mathematica **69**(1), 95–103 (2001). https://doi.org/10.1023/A:1012692601098
10. Sproull, R.F.: Refinements to nearest-neighbor searching ink-dimensional trees. Algorithmica **6**(1–6), 579–589 (1991). https://doi.org/10.1007/BF01759061

11. Tong, Q., Li, X., Yuan, B.: A highly scalable clustering scheme using boundary information. Pattern Recogn. Lett. **89**, 1–7 (2017)
12. Xia, C., Hsu, W., Lee, M.L., Ooi, B.C.: Border: efficient computation of boundary points. IEEE Trans. Knowl. Data Eng. **18**(3), 289–303 (2006)
13. Ye, H., Kitagawa, H., Xiao, J.: Continuous angle-based outlier detection on high-dimensional data streams. In: Proceedings of the 19th International Database Engineering & Applications Symposium, pp. 162–167. ACM (2015)
14. Zhang, L., Lin, J., Karim, R.: An angle-based subspace anomaly detection approach to high-dimensional data: with an application to industrial fault detection. Reliab. Eng. Syst. Safety **142**, 482–497 (2015)