Supervised Clustering for Radar Applications: On the Way to Radar Instance Segmentation

Ole Schumann,
Markus Hahn
and Jürgen Dickmann
Daimler AG
{ole.schumann, markus.hahn, juergen.dickmann}@daimler.com
Wilhelm-Runge-Str. 11
89081 Ulm, Germany

Christian Wöhler
Technische Universität Dortmund
Image Analysis Group
Otto-Hahn-Str. 4
44227 Dortmund, Germany

***Abstract*—For many radar applications, a grouping of radar reflections that belong to the same object is needed. Unsupervised clustering algorithms are commonly used for this task. However, the number and density of measured reflections of an object depends on various parameters and therefore unsupervised algorithms often fail to identify all points that should be part of the same cluster. We propose a method to incorporate learned knowledge about the data into the clustering algorithm and show that this new method outperforms unsupervised approaches.**

I. Introduction

Modern high resolution automotive radars can measure multiple reflections on one object and deliver therefore information about the spatial extension of the target. To utilize this information, for example in tracking or classification applications, all targets that belong to the same object have to be grouped together. In addition to the location in space and time of the reflection, also the (ego-motion compensated) Doppler velocity can be used to discern the object's reflections from other nearby measurements. In figure 1 all measurements in a time range of 1 s are displayed and the reflections of one approaching car are enclosed by a blue bounding box.

A number of effects make this a challenging task for every algorithm and even experienced human annotators struggle to find the correct clustering. For example, the number of measured reflections on an object decreases with increasing distance and at the same time the spacing between the reflections increases on average. Micro Doppler effects from moving wheels, arms, legs or any other small moving part of an object cause reflections with apparently zero velocity or twice the average velocity of the object to still belong to the same cluster. Also, the difference in Doppler velocity between nearby reflections decreases on average with distance to the sensor, because micro Doppler induced variations are no longer resolved and the measurements stem from the most reflective parts of the object which are often not the smaller parts. Sensor artifacts, Doppler ambiguity, mirror effects and alignment errors are further challenges to be dealt with. The aim of this paper is to introduce a new method that combines knowledge obtained from a supervised learning task with an unsupervised clustering algorithm. This new method is then compared to other unsupervised clustering algorithms that operate without the domain knowledge. A score function is introduced which enables us to optimize parameters of a clustering algorithm and to compare the results with a ground truth. The paper is structured as follows: In section II, related work is presented and differences to our work are highlighted. This is followed by the introduction of our score function and a review of the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [1] and Hierarchical-DBSCAN (HDBSCAN) [2] clustering algorithms. The optimization and training procedure is explained afterwards, before in section IV the results are presented. In section V conclusions are drawn and and outlook to further work is presented.

II. Related Work

The term "supervised clustering" is used in other works in two different ways. Some authors use this term to describe an algorithm that gets as an input the instances which should be clustered as well as the class label for each instance. The task is then to build clusters which are dense in feature space and contain only members of the same class [3], [4]. The second interpretation of the term is that examples for clusters are provided and the algorithms learns from these examples how a clustering should be performed [5], [6].

For example, in [5] a support vector machine is used to learn a similarity measure between two instances. This approach is closely related to semi-supervised clustering approaches where extra information like "must-link" and "cannot link" are provided for some instances. For example, in [6] pairwise constraints are calculated by a "similarity neural network" during a training phase and are then used for the distance calculation in a constrained k-means clustering.

Kellner et al. [7] propose a DBSCAN variant that is tailored to radar applications. They take the range dependence of possible radar observations into account by modifying the density calculation in DBSCAN. The downside of their approach is that they do not take into account the velocity of the object and rely on manually chosen thresholds.

In our work we also use the second interpretation of the term "supervised clustering". That is, we use a manually

978-1-5386-1725-0/18/$31.00 ©2018 IEEE

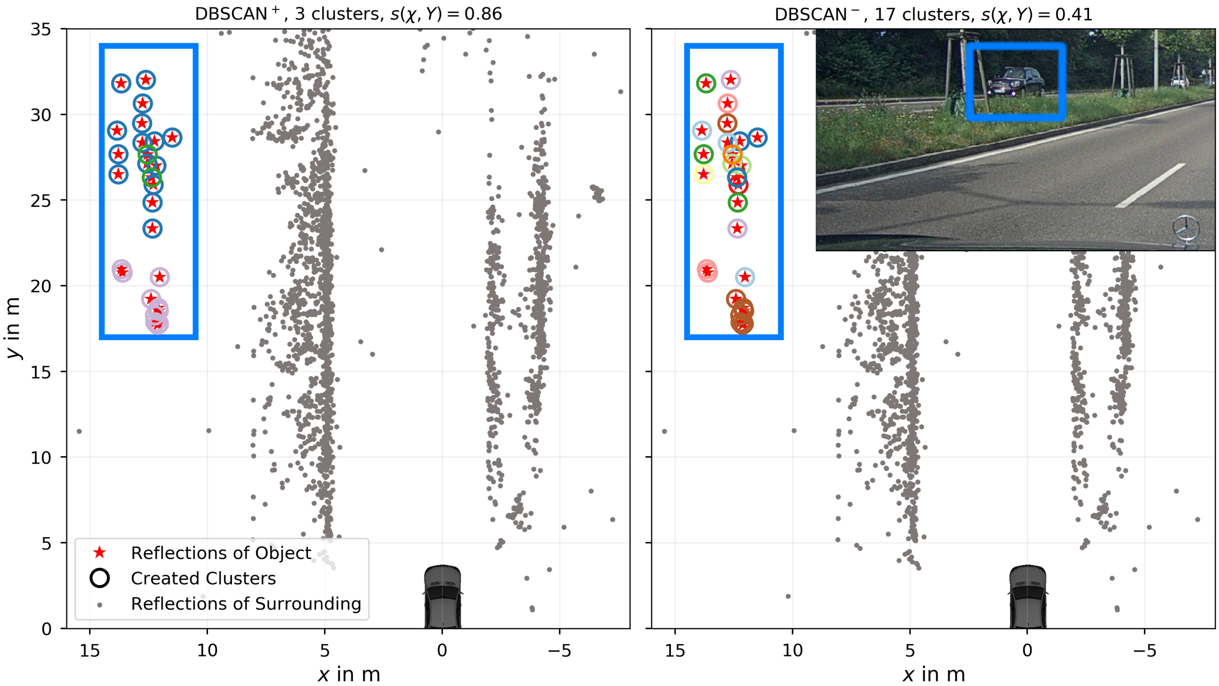

Fig. 1. Comparison of DBSCAN$^+$ (left) and DBSCAN$^-$ (right) for one object. In addition to the radar measurements, the camera image of the scene is displayed. Only the clusters created for the car (blue frame) are shown. The plot shows all measurements that were collected during 1 s.

clustered training set to learn optimal parameters for the hyper-parameters in the DBSCAN algorithm. Even though we seek clusters which contain only radar reflections from one object and therefore our clusters should be class-uniform, we cannot exploit this class information: our classification algorithms use clusters as an input and therefore at test time no class information is available for the clustering algorithm itself.

III. Methods

In our setting, the task is to assign all radar reflections which originate from the same object to one cluster. A cluster $X = \{p_1, \dots, p_n\} \subset \mathcal{P}$ is therefore a subset of the whole set of radar reflections $\mathcal{P} = \{p_1, \dots, p_N\}$ that were collected within some time range. For each measured radar reflection p, the positions p_x and p_y, the measurement time p_t and the ego-motion compensated Doppler velocity p_v were collected.

For each object i, a ground truth cluster $Y_i = \{p_1, \dots, p_m\}$ is obtained by manually grouping the corresponding radar targets. We manually grouped over 2.4 million radar reflections of more than 5000 different objects from the classes car, truck, pedestrian, pedestrian group and bike into clusters.

A. DBSCAN and HDBSCAN

The DBSCAN algorithm [1] is one of the most prominent density based clustering algorithms. A distance threshold ϵ and a requirement on a minimum number of neighboring points m_{pts} determine the algorithm. Core points are defined as points which have at least m_{pts} points in an ϵ region around themselves. An ϵ-region of a point $p \in \mathcal{P}$ is defined as the set of points with distance smaller than ϵ: $N_\epsilon(p) = \{q \in \mathcal{P} | d(p,q) \leq \epsilon\}$. The distance function d can be any metric defined on the feature space of the points. Core points and density-reachable points define the clusters [1].

Hierarchical DBSCAN only uses the m_{pts} parameter and creates a clustering tree with all possible clusterings of DBSCAN, where m_{pts} is fixed and ϵ is variable. From this tree-like structure, only the most stable clusters are extracted. Details on this can be found in [2]. This algorithm allows to identify clusters with varying density without the explicit specification of distance thresholds.

B. Score function

To evaluate the clustering, a score function is needed which compares the clustering result with the ground truth. The clustering result is a set of n_{clst} clusters $\chi_i = \{X_1, \dots, X_{n_{\text{clst}}}\}$ that were created for object i with ground truth cluster Y_i. The score $s(\chi_i, Y_i) \in [0, 1]$ is maximal for a perfect clustering and the score is set to zero if no cluster was created for the object under consideration. Let $\mathcal{X} = \bigcup_{j=1}^{n_{\text{clst}}} X_j$ be the union of the created clusters, TP the true positives $TP = |\{p \mid p \in Y_i \;\wedge\; p \in \mathcal{X}\}|$, FP the false positives $FP = |\{p \mid p \notin Y_i \;\wedge\; p \in \mathcal{X}\}|$ and FN the false negatives $FP = |\{p \mid p \notin Y_i \;\wedge\; p \in \mathcal{X}\}|$. The F_1 score is the harmonic mean of precision $Pr = TP/(TP+FP)$ and recall $Re = TP/(TP+FN)$ and is a suitable measure for how well the algorithm managed to assign points to clusters or to leave them out. A clustering algorithm that creates many clusters for one object should be considered worse than an algorithm that creates only one cluster. Therefore, the *variety* measure V is introduced, which penalizes the creation of many small clusters:

$$V = 1 - \eta \cdot \tanh(\alpha[n_{\text{clst}} - 1]), \quad \eta = 1 - \max_j \frac{|X_j \cap Y_i|}{|Y_i|}.$$

The factor η takes into account the size of the created clusters. For example, a clustering result with one large cluster and a few very small clusters should be scored better than a result with just as many equally sized clusters. The parameter α determines how fast the score decreases with the number of clusters. In this paper, $\alpha = 0.3$ was selected.

The total score is now given by the harmonic mean of the F_1 score and the variety V:

$$s(\chi_i, Y_i) = 2\frac{F_1(\chi_i, Y_i) \cdot V(\chi_i, Y_i)}{F_1(\chi_i, Y_i) + V(\chi_i, Y_i)}. \quad (1)$$

An example for an output of a clustering algorithm along with ground truth and score is shown in figure 2. For this object, two clusters were created which contain two and three noise points, respectively. Three reflections were erroneously excluded from the clusters. Only the cartesian position of the reflections are plotted here and the time/velocity information of the reflections are left out for brevity.

Fig. 2. Toy data with ground truth clustering (left) and possible output of a clustering algorithm (right). The resulting score is $s(\chi, Y) = 0.81$.

C. Training and Testing

To tailor DBSCAN to our application, we modified the distance calculation slightly: Instead of using one ϵ threshold in DBSCAN and a metric that combines all four features p_x, p_y, p_t and p_v, we used three thresholds ϵ_r, ϵ_t and ϵ_v and defined the ϵ-region around a point p as $N_\epsilon(p) = \{q \in \mathcal{P} \,|\, |p_x - q_x| \le \epsilon_r \wedge |p_y - q_y| \le \epsilon_r \wedge |p_t - q_t| \le \epsilon_t \wedge |p_v - q_v| \le \epsilon_v\}$. The two spatial dimensions were treated equally because the clustering should be rotationally invariant. The goal is to find optimal values for $\epsilon_r, \epsilon_t, \epsilon_v$ and m_{pts} that yield the best clustering result, i.e. the highest score when compared to the ground truth, eq. (1).

As already stated in section I, the distance between nearest neighbors in the spatial dimensions and in the velocity space depends on the velocity of the object and its distance to the radar sensor. It can therefore not be expected that a single set of ϵ thresholds is optimal for all distances and velocities. Given the changes in nearest neighbor distances and the distribution of our training data in different range and velocity regions, we split the data into six spatial regions and sorted the objects further by their average radial Doppler velocity into five different categories. By this, we obtained 30 individual distance-velocity regions with varying number of sectors within each region. A sector is a collection of the object's reflections and surrounding measurements up to $6\,\text{m}$ away from the object borders. If an object left a distance-velocity region during its observation (for example an approaching car), the measured reflections were split into multiple regions. To find only clusters of moving objects, DBSCAN was only allowed to consider those reflections as core points for a cluster, which have an ego-motion compensated Doppler velocity of at least $0.4\,\text{m/s}$. This allows us to still include stationary points in our clusters if they are density-reachable.

During training, the parameters $\epsilon_r, \epsilon_t, \epsilon_v$ and m_{pts} were optimized for each distance-velocity region by using adaptive simulated annealing [8], [9]. The score function, eq. (1), was used as an objective that had to be maximized. At the end of the training, 30 sets of parameters were obtained and tested on previously unseen data. Ten-fold cross validation was used for our analysis, i.e. each distance-velocity region was split into ten folds from which nine were used for training and one for testing. At test time, the scores were computed using the ground truth clustering and eq. (1).

In addition to this parametrized and trained DBSCAN algorithm (abbreviated from now on as DBSCAN$^+$), clustering was also done with HDBSCAN [10] and DBSCAN with a fixed parameter set of $\epsilon_r = 1\,\text{m}, \epsilon_t = 0.2\,\text{s}, \epsilon_v = 5\,\text{m/s}$ and $m_{\text{pts}} = 1$ (abbreviated as DBSCAN$^-$). These parameters are an educated guess by an expert and are reasonable choices for our application. For HDBSCAN a metric has to be defined. For example, if an euclidean metric is used, the distance between two points is calculated as $d(p,q)^2 = \sum_i (p_i - q_i)^2$, $i \in \{x, y, v, t\}$. This expression is only meaningful and useful, if each dimension is scaled by a characteristic length ℓ_i, so that the dimensions have the same impact on the distance. We tried two approaches to find suitable values for the ℓ_i: i) We used the ground truth clusters to calculate nearest neighbor differences in space, time and velocity for different range-velocity regions and thereby obtained a set of ℓ_i for each region. ii) We used the learned parameters $\epsilon_r, \epsilon_t, \epsilon_v$ from DBSCAN$^+$ for the ℓ_i. We will refer to the two approaches as HDBSCAN$^{(i)}$ and HDBSCAN$^{(ii)}$, respectively.

IV. Results

For each object i in the test sets, the total score $s(\chi_i, Y_i)$ and its components $V(\chi_i, Y_i), Pr(\chi_i, Y_i)$ and $Re(\chi_i, Y_i)$ were calculated and stored. From these values, the median $\hat{s}$, the arithmetic means $\overline{s}, \overline{V}, \overline{Pr}, \overline{Re}$ as well as the standard deviations $\sigma_s, \sigma_V, \sigma_{Pr}$ and σ_{Re} were computed. In Table I, these values are presented for the four different clustering approaches. Our learned DBSCAN outperforms the other methods with regard to the mean and median values of the total scores $s(\chi_i, Y_i)$. The higher average total score of our new method is mainly a result of a higher average variety score $\overline{V}$. This means that DBSCAN$^+$ creates on average fewer small clusters and tends to create single large clusters which capture most of the points. This however, seems to be done on the cost of losing some precision in the clustering. In our method, the precision drops to 0.79 whereas the completely unsupervised DBSCAN$^-$ has an average precision of 0.86. The recall values on the other hand are similarly high, indicating that almost all object reflections were assigned to one of the created clusters. In

Figure 1, the clustering results of DBSCAN$^+$ and DBSCAN$^-$ are displayed for one approaching car. Because of a Doppler ambiguity, DBSCAN$^+$ did not include two central points into the main cluster (blue circles) and one further cluster was created for points which were measured later in time and closer to the ego-vehicle. DBSCAN$^-$ completely failed to create a reasonable clustering because of high variations in the Doppler velocity and a too small ϵ_v threshold for this situation.

TABLE I
OVERVIEW OF THE CLUSTERING SCORES.

Algorithm	$\overline{s} \pm \sigma_s$	$\hat{s}$	$\overline{V} \pm \sigma_V$	$\overline{Pr} \pm \sigma_{Pr}$	$\overline{Re} \pm \sigma_{Re}$
HDBSCAN$^{(i)}$	0.69 ± 0.33	0.84	0.93 ± 0.14	0.65 ± 0.34	0.95 ± 0.12
HDBSCAN$^{(ii)}$	0.66 ± 0.34	0.80	0.93 ± 0.13	0.62 ± 0.36	0.94 ± 0.13
DBSCAN$^-$	0.81 ± 0.21	0.88	0.80 ± 0.24	$\mathbf{0.86 \pm 0.22}$	0.99 ± 0.07
DBSCAN$^+$	$\mathbf{0.87 \pm 0.18}$	**0.93**	$\mathbf{0.94 \pm 0.12}$	0.79 ± 0.25	$\mathbf{0.99 \pm 0.05}$

The scores of the two HDBSCAN approaches are even below the scores of DBSCAN$^-$. This might be explained as follows: For DBSCAN we could restrict the core points to those with some minimal Doppler velocity. By design, this is not possible for HDBSCAN. To stop HDBSCAN from adding too many static points from the surrounding to a cluster, one would have to manually increase the distance between moving and static points. This, however, would also hinder HDBSCAN to add static points to a cluster which in fact should be included, for example the bottom part of a turning wheel. HDBSCAN therefore does not seem to be a good choice for our task.

The high standard deviations in all the computed scores demand some extra consideration. In Fig. 3, the distribution of the score values obtained from DBSCAN$^-$ and DBSCAN$^+$ are plotted and the count differences of each bin are visualized. This representation displays that the score distributions are highly skewed towards higher scores. The high standard deviation in the scores stems from the fact that in both methods the full range of scores is covered and the score counts decrease only slowly in the direction of lower scores. The visualization of the count differences shows that the greatest difference exists in the region of scores greater than 0.85: Significantly more sectors obtained a high score when clustered with DBSCAN$^+$. On the other end of the score range, there is almost no difference in the counts because both algorithms have problems with a few rare situations in which the measured reflections have low velocities or only a few reflections with high spatial separation were captured.

Despite the high standard deviations in the average score, it is justified to claim that our proposed method works better than the other approaches: the 95 % bias corrected and accelerated bootstrap confidence interval [11] for the mean difference of the scores $s(\chi_i, Y_i)^+ - s(\chi_i, Y_i)^-$ is given by $[0.058, 0.063]$. The symbols $s(\chi_i, Y_i)^+$ and $s(\chi_i, Y_i)^-$ indicate the score of the i-th sample obtained by using DBSCAN$^+$ and DBSCAN$^-$, respectively. For bootstrapping, 30 000 iterations were used.

V. Conclusion and Outlook

In this paper, a new method for clustering radar data was presented. Optimal parameters $\epsilon_r, \epsilon_v, \epsilon_t$ and m_{pts} for the DBSCAN algorithm were learned by using simulated annealing as an optimizer and a new score function as an objective. A comparison with HDBSCAN and DBSCAN with fixed parameters showed that our new method performs better on our dataset. However, our study revealed that there are still situations in which our parametrized DBSCAN performs poorly. On the one hand, this could be caused by an overfitting to our training data and hence a bad parameter choice for DBSCAN. On the other hand, DBSCAN itself could be the limiting factor, because sometimes no parameter set exists which clusters an object perfectly. More training data could heal the first problem, but the second point needs to be addressed more rigorously. The situations in which DBSCAN$^+$ fails to find a proper clustering displays the difficulty of the task. A more sophisticated algorithm is needed that can use information of the local and global surroundings and alter thresholds for each object. Combined learning of clustering and classification of the resulting clusters could further leverage the performance.

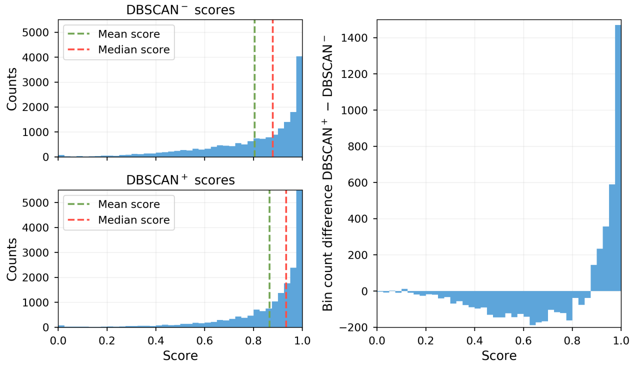

Fig. 3. Distribution of the scores $s(\chi_i, Y_i)$ for DBSCAN$^-$ and DBSCAN$^+$ (left) and difference between the bin counts (right).

References

[1] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[2] R. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," in *Advances in Knowledge Discovery and Data Mining*. Springer, Berlin, Heidelberg, 2013.

[3] C. Eick, N. Zeidat, and Z. Zhao, "Supervised clustering - algorithms and benefits," in *16th IEEE International Conference on Tools with Artificial Intelligence*. IEEE Comput. Soc, 2004.

[4] S. H. Al-Harbi and V. J. Rayward-Smith, "Adapting k-means for supervised clustering," *Applied Intelligence*, vol. 24, no. 3, jun 2006.

[5] T. Finley and T. Joachims, "Supervised clustering with support vector machines," in *Proceedings of the 22nd international conference on Machine learning - ICML '05*. ACM Press, 2005.

[6] S. Melacci, M. Maggini, and L. Sarti, "Semi-supervised clustering using similarity neural networks," in *International Joint Conference on Neural Networks*. IEEE, jun 2009.

[7] D. Kellner, J. Klappstein, and K. Dietmayer, "Grid-based DBSCAN for clustering extended objects in radar data," in *Intelligent Vehicle Symposium*, Alcala de Henares, 2012.

[8] L. Ingber, "Adaptive simulated annealing (ASA): Lessons learned," *Control and Cybernetics*, vol. 25, no. 1, 1996.

[9] ——, "Adaptive Simulated Annealing (ASA)," Pasadena, 1993. [Online]. Available: https://www.ingber.com/#ASA-CODE

[10] L. McInnes, J. Healy, and S. Astels, "HDBSCAN: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, mar 2017.

[11] B. Efron, "Better bootstrap confidence intervals," *Journal of the American Statistical Association*, vol. 82, no. 397, mar 1987.