Journal of Universal Computer Science, vol. 31, no. 9 (2025), 1012-1022
submitted: 9/2/2025, accepted: 8/7/2025, appeared: 14/8/2025 CC BY 4.0

DSGD++: Reducing Uncertainty and Training Time in ...

Aik Tarkhanyan
Mathematics and Mechanics Department at Yerevan State University, 0025 Yerevan, Armenia
https://orcid.org/0009-0000-7015-111X, hayk.tarkhanyan@edu.ysu.am)

Ashot Harutyunyan
(ML Lab at Yerevan State University, 0025 Yerevan, Armenia,
Institute for Informatics and Automation Problems NAS RA, 0014 Yerevan, Armenia
https://orcid.org/0000-0003-2707-1039, harutyunyan.ashot@ysu.am)

Abstract: Several studies have shown that the Dempster–Shafer theory (DST) can be successfully applied to scenarios where model interpretability is essential. Although DST-based algorithms offer significant benefits, they face challenges in terms of efficiency. We present a method for the Dempster-Shafer Gradient Descent (DSGD) algorithm that significantly reduces training time—by a factor of 1.6—and also reduces the uncertainty of each rule (a condition on features leading to a class label) by a factor of 2.1, while preserving accuracy comparable to other statistical classification techniques. Our main contribution is the introduction of a ”confidence” level for each rule. Initially, we define the ”representativeness” of a data point as the distance from its class’s center. Afterward, each rule’s *confidence* is calculated based on *representativeness* of data points it covers. This confidence is incorporated into the initialization of the corresponding Mass Assignment Function (MAF), providing a better starting point for the DSGD’s optimizer and enabling faster, more effective convergence. The code is available at https://github.com/HaykTarkhanyan/DSGD-Enhanced.

Keywords: Dempster-Shafer Theory, Interpretability, KMeans, Mass Assignment Functions, Classification
Categories: I.2.6, I.5, G.3
DOI: 10.3897/jucs.164745

1 Introduction

Dempster-Shafer theory [**?**] has emerged as a powerful framework for developing classification algorithms that prioritize interpretability. This theory provides a mathematical approach for combining evidence from different sources to calculate the probability of an event, utilizing Dempster’s rule of combination. Peñafiel et al. [**?**] have demonstrated for classification tasks that an algorithm combining Dempster-Shafer theory with optimization techniques can offer substantial explainability, even when employing a limited number of rules, without sacrificing accuracy. The algorithm was further extended to the regression case by Baloian et al. [**?**] and to clustering by Valdivia et al. [**?**]. The algorithm is inherently explainable because it operates by combining relatively simple rules for inference. Additionally, recent work by Baloyan et al [**?**] demonstrates the algorithm’s robustness to class imbalance and its ability to capture various numeric interactions among the individual properties (i.e., features) that define each data point.

To make improvements, two issues should be considered. Firstly, the number of subsets in the frame of discernment, growing with a complexity of $\mathcal{O}(2^n)$, makes the

inclusion of non-singleton classes nearly impossible. Secondly, given that each feature in the dataset typically generates three rules, combining even two rules significantly increases training time and further limits the predictive power of the rules. Moreover, one promising direction for improving DST-based methods lies in incorporating rules derived from rule mining algorithms [**?**, **?**, **?**]. However, to make this approach truly feasible, it becomes evident that a strategy to reduce training time is crucial for enhancing DST-based models.

Below, we outline our methodology for addressing these challenges, focusing on improved Mass Assignment Function initialization to reduce training time and uncertainty.

Our approach significantly reduces the optimization time by improving the technique for initializing MAFs associated with the rules. Peñafiel et al. [**?**] relied on random assignment, where the empty set's mass is set to 0, the entire set—which represents total uncertainty—is assigned a value of 0.8, and the remaining 0.2 is randomly distributed among singleton classes (other classes are not considered). Sedláček et al. [**?**] have investigated the use of the statistical distribution of classes in allocating mass values. Here, we propose a clustering-based technique that incorporates additional information about each rule into the MAF.

To advance this method, we introduce the concept of rule "confidence". First of all, operating under the assumption that certain points are more representative of their class than others, we define "representativeness" for each point. This approach is inspired by the post-hoc interpretation technique commonly used in clustering algorithms. In these models, each cluster's "color" is treated as a label, upon which interpretable classifiers are constructed. In our framework, we not only assign a "color" to data points, but also attribute a numeric value representing their "opacity" which we refer to as "representativeness". For spherical data, we employ the $get\_representativeness\_KMeans$ function (see Algorithm **??**), which uses the KMeans clustering algorithm [**?**] to determine the most representative data point. In our previous work ([**?**]) we were simply setting the most representative data point to the arithmetic mean of the data points with the same label instead of using KMeans. In scenarios where data is better suited for density-based clustering, we apply the $get\_representativeness\_DBSCAN$ function (see Algorithm **??**), which is based on DBSCAN [**?**] algorithm. Once we have determined the representativeness of each data point, we then define the confidence of each rule based on the data points it covers, this is achieved by $get\_confidence$ function discussed in Section **??**. The final step involves integrating this confidence into the MAF assignment, which will be discussed further at the end of Section **??**.

In Section **??** we specify the concept of representativeness and its estimation, as well as the rule confidence computation, and the enhanced MAF initialization. Section **??** describes evaluation results for our approaches, while Section **??** concludes the paper with notes on the future work.

2 Principles behind DSGD++

Let's start by introducing the algorithms for **estimation of representativeness** using KMeans clustering (Algorithm **??**) and the DBSCAN-based (Algorithm **??**) approaches. Additionally, we present example figures for synthetic datasets (Figure **??**).

In the **KMeans-based approach**, the data is standardized and then clustered into k groups (equal to the number of classes). Each data point's representativeness is determined by its distance to the cluster centroid: points are first checked for outlier status using a Z-score threshold, and then distances are normalized within each cluster. The

representativeness score is calculated as 1 minus the normalized distance (or 0 if the point is flagged as an outlier).

In the **DBSCAN-based approach**, the algorithm begins with a small initial neighborhood radius (e.g., $\varepsilon = 0.1$) and then increases this radius in small increments until it either reaches *maxEps* or yields the desired number of clusters. After each increment, DBSCAN is applied, and the resulting number of clusters is checked against the target k. If there are too few clusters, ε is incremented further, leading to larger neighborhoods and fewer clusters merging. Once a specific ε value produces the target number of clusters, each data point's representativeness is computed by counting how many points fall within that final radius, and these counts are then scaled via Min-Max normalization.

Note: The following hyperparameters are used:

1. KMeans-based approach: $zScoreThreshold = 2$
2. DBSCAN-based approach: $minPoints = 2 \times$ (number of features); $maxEps = 20$; $step = 0.05$

Algorithm 1 Estimation of Representativeness using KMeans Clustering

Require: X: Dataset, k: Number of clusters (same as dataset's number of classes),
Ensure: $representativenessList$: List of representativeness values for each data point

function GET_REPRESENTATIVENESS_KMEANS(X, k)
    Standardize the dataset X using standard scaling.
    Fit the KMeans clustering algorithm on X with k clusters to obtain centroids $\{C_i\}_{i=0}^{k-1}$.
    $confidenceList \leftarrow$ Empty list
    for each $dataPoint$ in X **do**
        Calculate Euclidean distance from $dataPoint$ to its nearest centroid C_i.
    end for
    Identify outliers among the data points using the Z-score technique, where a data point is considered an outlier if its distance's Z-score exceeds $zScoreThreshold$.
    if $dataPoint$ is not an outlier **then**
        $normRepr \leftarrow dataPoint$'s distance after normalizing using min-max scaling within its cluster.
        $representativeness \leftarrow 1 - normRepr$
    else
        $representativeness \leftarrow 0$
        Append $representativeness$ to $representativenessList$
    end if

    return $representativenessList$
end function

Algorithm 2 Estimation of Representativeness using Density Based Approach

Require: X: Dataset, k: Number of clusters (same as number of unique classes),
Ensure: $confidenceList$: List of confidence scores for each data point

function GET_REPRESENTATIVENESS_DBSCAN(X, k)
  Standardize the dataset X using standard scaling.
  Initialize $eps \leftarrow$ Initial small value (e.g., 0.1)
  while $numCentroids \neq k$ and $eps \leq maxEps$ **do**
    $model \leftarrow$ DBSCAN(eps, $minPoints$)
    Fit DBSCAN on X
    $numCentroids \leftarrow$ Count of unique clusters formed (excluding noise)
    $eps \leftarrow eps + step$
  end while
  $radius \leftarrow eps$
  Initialize $confidenceList \leftarrow$ Empty list
  for each $dataPoint$ in X **do**
    $representativeness \leftarrow \sum_{y \in X} \mathbf{1}_{\|y-x\| \leq radius}$ (number of data points within the specified radius)
    Append $representativeness$ to $representativenessList$
  end for
  $representativenessList \leftarrow$ Min-Max Scaled version of $representativenessList$
  return $representativenessList$
end function

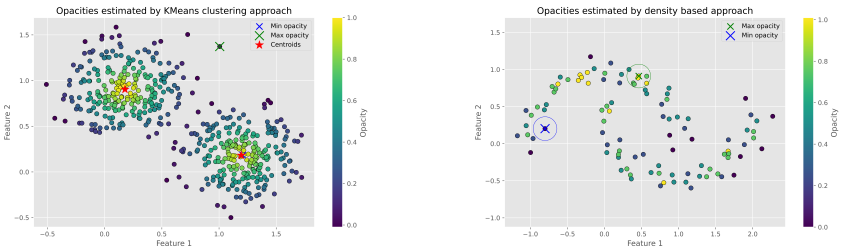

Figure 1: Illustrations of algorithms for estimation of representativeness for datasets with spherical shapes and density-based characteristics. Brighter colors indicate data points that are more representative of their respective classes.

Our next step is aggregating the representativeness values for data points that follow the same rule into a measure of **rule confidence**.

Utilizing the algorithms previously described, we calculate the representativeness for each data point. Afterward, we generalize this data point-specific estimate to an entire rule using the following steps (we encapsulate this in the $get\_confidence$ function):

1. Filter the dataset to retain only the rows that comply with the rule.
2. If the rule does not apply to any rows, set its confidence to 0 (this corresponds to the full uncertainty). Otherwise:
3. Calculate the rule's confidence as the mean representativeness of the rows it covers.
4. If the rows are not homogeneous with respect to their labels, reduce the confidence based on the proportion of the most frequent label among these rows.

Once we have obtained the confidence scores for the rules we can incorporate them into the **MAF initialization**.

In the DSGD classifier, MAFs are key components. Each rule has a MAF associated with it. For a new data point, MAFs of satisfied rules are combined using Dempster's rule. The class with the highest combined mass is typically chosen for prediction. The initial MAF values are crucial as they are the starting point for optimization, affecting convergence speed and model quality.

Our approach uses the following values for initialization. Let $c = get\_confidence(rule)$ represent the confidence derived for a given rule. The label l_{mode}, which is the most frequently occurring label within the subset of data points covered by the rule, receives the confidence value c. The remaining mass, $(1 - c)$, is evenly distributed among all the other labels present in the subset. Formally, for an element l_i in the subset:

$$m(l_i) = \begin{cases} c & \text{if } l_i = l_{\text{mode}}, \\ \frac{1-c}{n-1} & \text{otherwise,} \end{cases}$$

where $m(l_i)$ denotes the mass assigned to label l_i, and n is the total number of elements in the frame of discernment. Note that here by saying label we mean the original label of the data point, not the one assigned to it via clustering algorithm (the color).

In the following section, we compare the results of initializing MAFs with the proposed approach and the traditional random initialization.

3 Evaluation

Here, we demonstrate the effects of the newly defined MAF initialization algorithm (KMeans-based) on the training time, accuracy, and the amount of rule uncertainties. We accomplish this by testing the approach both on controlled scenarios and on some classical datasets. The datasets used for evaluations are summarized in Table **??**.

Table 1: Datasets overview (binary classification)

Dataset	**Rows**	**Cols**	**Description**
Brain Tumor	3762	14	Includes first-order and texture features with target levels.
Breast Cancer Wisconsin	699	9	Clinical reports detailing cell benignity or malignancy.
Gaussian	500	3	Two 2D Gaussian distributions generate this dataset.

Dataset	Rows	Cols	Description
Uniform	500	3	Uniform samples from [-5, 5], with class split by the sign of x.
Rectangle	1263	3	Points in [-1, 1]×[-1, 1], class determined by the y component's sign.

The first two are real-life datasets ([**?**, **?**]), while the last three are controlled scenarios.

Now we will present accuracy and speedup analysis, and Section **??** will cover uncertainty analysis (we also do an evaluation of our newly proposed definition of uncertainty).

We focus on the KMeans approach, which we will refer to as the "clustering" MAF method. Additionally, we will use "MED." to denote the median and "AVG." to denote the average.

Table **??** presents a comparison of various metrics across different MAF initialization methods (Confidence and Random) and datasets. For evaluating the classifier's predictive power, we have calculated the accuracy (ratio of correctly predicted instances to the total instances) and F1 score. For evaluating the optimizer, we have reported the training time in seconds, the number of epochs, the minimum loss, and the initial loss.

MAF method	dataset	accuracy	f1	training_time	epochs	min_loss	initial_loss
clustering	Brain Tumor	0.981	0.98	138.469	117	0.018	0.181
random	Brain Tumor	0.983	0.981	157.075	132	0.026	0.245
clustering	Breast Cancer	0.976	0.966	17.243	73	0.023	0.228
random	Breast Cancer	0.976	0.966	19.63	117	0.031	0.308
clustering	Gaussian	0.987	0.988	15.792	100	0.017	0.083
random	Gaussian	0.987	0.988	39.059	264	0.024	0.265
clustering	Rectangle	1	1	61.032	167	0.006	0.252
random	Rectangle	1	1	98.64	275	0.008	0.235
clustering	Uniform	0.973	0.97	20.683	120	0.035	0.197
random	Uniform	0.973	0.97	39.208	273	0.037	0.255

Table 2: Comparison of various metrics across different MAF initializations and datasets

We can see that although F1 scores and accuracies are nearly identical for both methods, our proposed methods consistently requires fewer epochs (on average by 42%) to converge. Additionally, the results also support our hypothesis that MAF initialization provides a better starting point, for every dataset the optimizer starts from a better point and achieves a lower final loss.

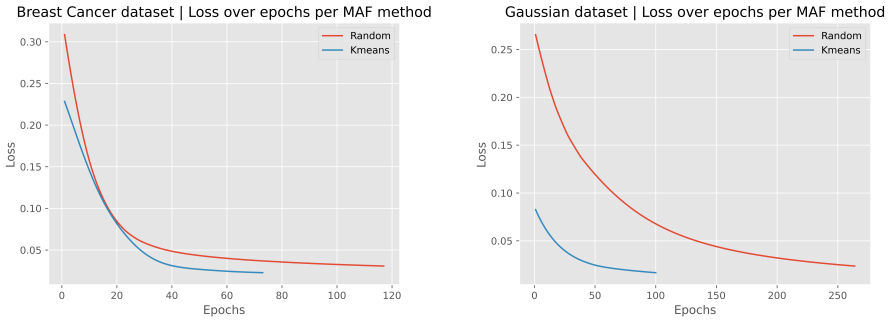

Figure 2: The figure demonstrates the benefits offered by MAF initialization, mainly decreased number of epochs and better starting point.

Table **??** deals with the speedup and predictive performance tradeoff.

Dataset	Accuracy Ratio	F1 Ratio	Training Time Speedup, x
Brain Tumor	1	1	1.13
Breast Cancer	1	1	1.14
Gaussian	1	1	2.47
Rectangle	1	1	1.62
Uniform	1	1	1.90

Table 3: Ratios of accuracy, F1 score, and training time for Random and Confidence MAF initializations across datasets

The results in Table **??** were rounded up to 2 digits. We can infer that accuracy and the f1 score remain unchanged while we experience a speedup of 1.65x.

3.1 Uncertainty Analysis and a New Approach to Rule Importance Estimation

After the model has been trained, the question arises of how reliable the rules used by the model are. To determine this, we must examine the MAFs associated with these rules. In this subsection, we will describe the traditional definition of uncertainty, demonstrate its weaknesses, suggest an alternative definition, and analyze both traditional and newly suggested uncertainties for different MAF initialization methods.

Traditionally, uncertainty has been defined by the mass of the complete set, with lower uncertainty indicating a more reliable rule. However, this is not always the best approach. Table **??** demonstrates an example.

Rule	Mass First Class	Mass Second Class	Uncertainty	Ratio
1	0.49	0.51	0	1.04
2	0.01	0.09	0.9	9

Table 4: Illustration of the Pitfalls of the Traditional Approach

Rule 1 shows zero uncertainty, which might seem ideal. But the nearly equal masses for the two classes make it hard for the algorithm to distinguish between them. On the flip side, Rule 2, despite its high uncertainty, has a significant difference in class masses (a ratio of 9), which enhances its ability to clearly separate the classes.

Given these insights, we propose a fresh approach that factors in both the uncertainty and the ratio between the most probable and second most probable masses.

Our method is inspired by the F1 score, which is the harmonic mean of precision and recall. To optimize our evaluation, we focus on maximizing the value of $1 - \textit{uncertainty}$ and the class mass *ratio*. This approach allows us to prioritize both a high ratio and low uncertainty simultaneously. To unify these metrics into a single measure, we calculate their harmonic mean. However, since 1 - *uncertainty* ranges from 0 to 1 and the *ratio* can vary widely, we first normalize the *ratio* using min-max scaling across all rules before applying the harmonic mean. This scaling ensures that both metrics contribute equally to the final score. To put this formally

– **Uncertainty Adjustment:**

$$U' = 1 - U$$

where U is the original uncertainty.

– **Normalization of the Ratio:**

$$R' = \frac{R - \min(R)}{\max(R) - \min(R)}$$

where R is the original ratio (when dividing the values of two masses we add $\varepsilon = 0.01$ to denominator to avoid zero division error), and $\min(R)$ and $\max(R)$ are the minimum and maximum values of the ratio across all rules, respectively.

– **Harmonic Mean Calculation:**

$$H = \frac{2 \cdot U' \cdot R'}{U' + R'}$$

where U' is the adjusted uncertainty value and R' is the normalized ratio.

With these definitions in place, we now present the experimental evaluation of the effects of our new MAF initialization both based on traditional approach for importance calculation, and on newly proposed one. Below, we will define improvement as the ratio of $MED.Random$ and $MED.Clustering$.

Table **??** shows that on average the clustering approach yields in uncertainty reduction by a factor of 2.12.

Dataset	AVG. Clust.	AVG. Rand.	Med. Clust.	Med. Rand.	Improvement
Brain Tumor	0.258	0.713	0.246	0.732	2.979
Breast Cancer	0.314	0.710	0.262	0.703	2.685
Gaussian	0.225	0.330	0.229	0.302	1.318
Rectangle	0.201	0.448	0.203	0.434	2.135
Uniform	0.150	0.262	0.096	0.144	1.493

Table 5: Average and median uncertainties for Random and Clustering MAF initializations

Dataset	AVG. Clust.	AVG. Rand.	Med. Clust.	Med. Rand.	Improvement
Brain Tumor	0.247	0.313	0.075	0.336	4.471
Breast Cancer	0.156	0.359	0.075	0.411	5.486
Gaussian	0.633	0.625	0.807	0.757	0.937
Rectangle	0.473	0.536	0.457	0.571	1.249
Uniform	0.732	0.700	0.935	0.876	0.937

Table 6: Average and median newly defined uncertainties for Random and Clustering MAF initializations

Table **??** shows that on average the clustering approach yields in uncertainty reduction (harmonic mean approach) by a factor of 2.61.

See below Figure **??** for an example of rule uncertainties for the Breast Cancer dataset [**?**].

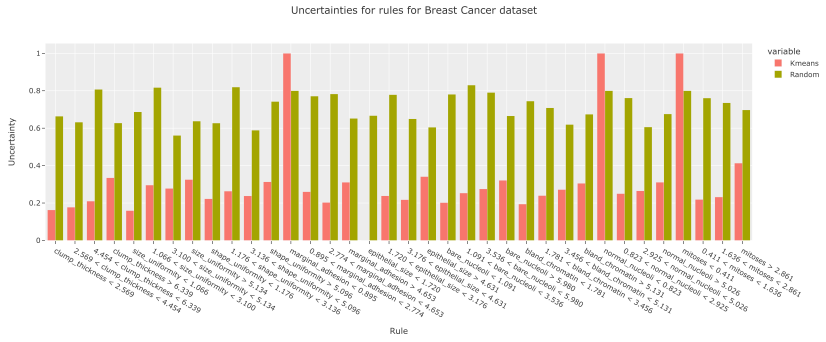

Figure 3: Uncertainties per rule for different MAF initialization methods

4 Conclusions and Future Work

We saw that by introducing the concept of *representativeness* for individual data points—and using it to guide clustering-based MAF initialization—we achieved a 1.6× reduction in training time for DSGD classifier. The proposed initialization method not only offers a more favorable starting point for optimization, but also reduces the number of required epochs, all while preserving predictive performance equivalent to baseline method.

We further addressed the need for a more flexible definition of rule uncertainty by proposing a measure that takes into account the ratio of masses of singleton classes. Our MAF initialization approach reduced uncertainty by a factor of 2.6 using the newly proposed measure and by 2.1 using the traditional measure.

Future work could extend this approach to multi-label datasets, explore additional clustering techniques, and evaluate performance on larger datasets. Additionally, the reduced training time achieved by our method opens opportunities to experiment with incorporating non-singleton classes or generating a larger set of rules. Given that Dempster-Shafer theory effectively integrates evidence from multiple sources, comparing our approach with expert knowledge-based initialization might yield valuable insights. Finally, employing rule-mining algorithms such as RIPPER, C5.0, or SkopeRules [**?**, **?**, **?**] in combination with our MAF initialization strategy could potentially enhance model performance and further reduce uncertainty.

Acknowledgements

The research was supported by ADVANCE Research Grants from the Foundation for Armenian Science and Technology.

References

[Baloian, 24] Baloian, N., Davtyan, E., Petrosyan, K., Poghosyan, A., Harutyunyan, A., Penafiel, S.: ”Embedded Interpretable Regression using Dempster-Shafer Theory”; In: Proceedings of the 4th Codassca Workshop on Data Science and Reliable Machine Learning (2024).

[Baloyan, 24] Baloyan, A., Aramyan, A., Baloian, N., Poghosyan, A., Harutyunyan, A., Penafiel, S.: ”An Empirical Analysis of Feature Engineering for Dempster-Shafer Classifier as a Rule Validator”; In: Proceedings of the 4th Codassca Workshop on Data Science and Reliable Machine Learning (2024).

[Bohaju, 20] Bohaju, J.: ”Brain Tumor”; In: Kaggle 2020, DOI: 10.34740/KAGGLE/DSV/1370629, https://www.kaggle.com/dsv/1370629.

[Cohen, 95] Cohen, W.W.: ”Fast Effective Rule Induction”; In: Prieditis, A., Russell, S. (eds.) Machine Learning Proceedings 1995, Morgan Kaufmann (1995), 115-123.

[Ester, 96] Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: ”A density-based algorithm for discovering clusters in large spatial databases with noise”; Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD’96), AAAI Press (1996), 226-231.

[Goix, 20] Goix, N.: ”skope-rules: Interpretable rules in Python”; scikit-learn-contrib/skope-rules v1.0.1 (2020).

[MacQueen, 67] MacQueen, J.: ”Some Methods for Classification and Analysis of Multivariate Observations”; Proc. 5th Berkeley Symp. on Math. Statist. and Prob., Vol. 1 (1967), 281-297.

[Peñafiel, 20] Peñafiel, S., Baloian, N., Sanson, H., Pino, J.A.: ”Applying Dempster-Shafer Theory for Developing a Flexible, Accurate and Interpretable Classifier”; Expert Systems with Applications, 148 (2020), 113262.

[Salzberg, 94] Salzberg, S.L.: ”C4.5: Programs for Machine Learning by J. Ross Quinlan”; Machine Learning, 16(3) (1994), 235-240.

[Sedláček, 24] Sedláček, O., Bartoš, V.: ”Fusing Heterogeneous Data for Network Asset Classification – A Two-layer Approach”; In: 2024 IEEE Network Operations and Management Symposium (NOMS) (2024).

[Shafer, 76] Shafer, G.: ”A Mathematical Theory of Evidence”; Princeton University Press, Princeton (1976).

[Tarkhanyan, 24] Tarkhanyan, A., Harutyunyan, A.: ”Improving the DSGD Classifier with an Initialization Technique for Mass Assignment Functions”; Codassca 2024, Logos (2024), 137–142.

[Valdivia, 24] Valdivia, R., Baloian, N., Chahverdian, M., Adamyan, A., Harutyunyan, A.: ”An Explainable Clustering Algorithm using Dempster-Shafer Theory”; In: Proceedings of the 4th Codassca Workshop on Data Science and Reliable Machine Learning (2024).

[Wolberg, 90] Wolberg, W.H., Mangasarian, O.L.: ”Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology”; Proc. Nat. Acad. Sci., 87(23) (1990), 9193-9196.