IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2021, accepted January 21, 2022, date of publication January 31, 2022, date of current version February 9, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3147951

A Novel Mean-Shift Algorithm for Data Clustering

CLAUDE CARIOU[1], (Member, IEEE), STEVEN LE MOAN[2], (Member, IEEE), AND KACEM CHEHDI[1], (Member, IEEE)

[1]CNRS, IETR, UMR 6164, University of Rennes 1, 22300 Lannion, France
[2]Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway

Corresponding author: Claude Cariou (claude.cariou@univ-rennes1.fr)

ABSTRACT We propose a novel Mean-Shift method for data clustering, called Robust Mean-Shift (RMS). A new update equation for point iterates is proposed, mixing the ones of the standard Mean-Shift (MS) and the Blurring Mean-Shift (BMS). Despite its simplicity, the proposed method has not been studied so far. RMS can be set up in both a kernel-based and a nearest-neighbor (NN)-based fashion. Since the update rule of RMS is closer to BMS, the convergence of point iterates is conjectured based on the Chen's BMS convergence theorem. Experimental results on synthetic and real datasets show that RMS in several cases outperforms MS and BMS in the clustering task. In addition, RMS exhibits larger attraction basins than MS and BMS for identical parametrization; consequently, its kernel variant requires a lower aperture of the kernel function, and its NN variant a lower number of nearest neighbors compared to MS or BMS, to achieve optimal clustering results. In addition, the NN version of RMS does not need to specify a convergence threshold to stop the iterations, contrarily to the NN-BMS algorithm.

INDEX TERMS Data mining, data clustering, mean-shift, kernel-based method, nearest neighbors.

I. INTRODUCTION

Data clustering is a type of unsupervised learning which consists of automatically grouping data points having similar characteristics into identified clusters without training sample points. It is a central task in various application fields such as medicine, genomics, content-based image and video indexing, and Big Data mining to cite a few. Clustering is also increasingly relevant in the context of Artificial Intelligence, in order to unveil the existence of underlying complex structures in datasets [2], and especially in applications where little or no training data is available. Despite several decades of research, clustering remains a challenging task for many applications because of the increasing size (number of data points) and dimensionality (number of features) of modern datasets. This is particularly true for applications that require on-the-fly data partitioning [3].

Clustering methods can be broadly categorized into several families, comprising centroid clustering [4]–[6], hierarchical clustering [7], [8], density-based [9], [10], Mean-Shift and mode seeking [11]–[14], clustering based on mixture resolving [15]–[17], and, more recently, affinity propagation (AP) [18], information theoretic clustering [19], and convex clustering [20].

From a general viewpoint, clustering remains an ill-posed problem [21] because, depending on the partitioning process, several legitimate solutions can be obtained which are all acceptable [22], [23]. In fact, most popular methods claimed as unsupervised require a significant prior knowledge about the data structure, i.e. the number of clusters to be found. This is particularly true for centroid clustering, mixture resolving, and spectral clustering in their baseline implementation. However, while some of their parameters are necessary and can be difficult to tune, several other approaches do not require to specify the number of clusters. For instance, among them are hierarchical methods, as well as DBSCAN [9], AP [18], convex clustering [20], nearest-neighbor density-based (NN-DB) methods [24] and Mean-Shift based methods. In this work, we focus on the latter.

Mean-Shift (MS) was originally proposed by Fukunaga and Hostetler in 1975 [11] essentially as a means to provide the modes of an unknown probability density function (p.d.f.). MS relies on kernel density estimation (KDE), a non-parametric way to estimate a p.d.f. from data samples [25], [26]. In MS, each point of the dataset is moved iteratively by a small amount (the so-called *mean shift*) until convergence to some stationary point, i.e. a local mode of

The associate editor coordinating the review of this manuscript and approving it for publication was Amir Masoud Rahmani.

the estimated p.d.f.. MS has first been used as an unsupervised data clustering method, for which the retained local modes after convergence of the point iterates serve as cluster representatives (or exemplars). A connected component post-processing stage [26] is therefore necessary after the convergence is achieved to assign a cluster label to each of the original data points.

A number of studies have followed the seminal work of Fukunaga and Hostetler [12], [25], [27]–[31], and several proofs pertaining to convergence and p.d.f. estimation have been proposed [1], [12], [25]–[28], [32]–[34]. In [26], Carreira-Perpiñán provides a comprehensive review of MS-based methods and their application to data clustering and data denoising. Mean-Shift has also been successfully applied to image filtering and segmentation in [25].

In the present work, we propose a novel approach to the classical Mean-Shift algorithm focusing data clustering; the KDE problem is not investigated herein. To the best of our knowledge, the proposed approach has not been published. Despite relying on a modification of the original MS algorithm, we demonstrate that our method leads to significantly different features. This method, which we name Robust Mean-Shift (RMS) is a hybridization of the standard Mean-Shift (MS) algorithm [25], and of the so-called Blurring Mean-Shift (BMS) method [27]. It is worth recalling that BMS was actually first proposed in [11], as pointed out in [26]. Our algorithm is based on iteratively moving updates of initial data points, similar to MS and BMS, but the update equation of RMS fundamentally differs from both methods.

The proposed RMS approach proposed in this work has several valuable properties:

- We find experimentally that RMS requires a lower bandwidth parameter (in the kernel-based variant) and a lower number of nearest neighbors (in the NN variant) than MS and BMS to achieve comparable or even better results in the clustering task; this is especially interesting to speed up the computation of point iterates, and in the case of the NN-RMS variant, to reduce the size of the NN graph compared to the ones required by the NN variants of MS and BMS;
- Compared to MS and BMS, RMS generally performs better, as evidenced experimentally through the analysis of various datasets;
- In most experiments, RMS converges faster than BMS, the latter being proved to converge faster than MS [35];
- The (classical) kernel-based RMS can be easily turned to a K-nearest neighbor (KNN) algorithm, similarly to MS and BMS [30], [31];
- The NN variant of RMS does not require a termination threshold, unlike the NN variant of BMS.

The paper is organized as follows. Section II provides a brief overview of related works, including kernel-based and KNN-based Mean-Shift (MS) approaches published so far. In Section III, we introduce the proposed clustering method and explain how it relates to MS and BMS. The convergence of RMS in the kernel-based framework is then discussed in Section IV. Section V describes the NN-based variant of RMS. An experimental study of RMS and its comparison with other similar clustering approaches on various datasets is provided in Section VI. Conclusions and perspectives of this work are given in Section VII.

II. NOTATIONS AND RELATION TO PRIOR WORKS

Let $\mathscr{X} = \{\mathbf{x}_i\}, \mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, N$ the set of data points to classify. Let $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ a kernel function such that $(\mathbf{u}, \mathbf{v}) \mapsto f(\mathbf{u}, \mathbf{v}) \geq 0$, and $f(\mathbf{u}, \mathbf{v})$ decreases monotonically with $\|\mathbf{u} - \mathbf{v}\|$. Kernel density functions are generally tuned with a bandwidth parameter, and they can follow several models; flat, Epanechnikov, biweight, and especially Gaussian kernels are commonly used [27]. However, it is well established that the shape of the kernel has very little effect on the results in comparison to the bandwidth parameter. With that in mind, Gaussian kernels are most commonly used for their convenience.

MS and BMS aim at estimating a p.d.f. and finding its local modes from the observations $\mathscr{X}$. This is done by moving the initial data points $\{\mathbf{x}_i\}_{i=1,\ldots,N}$ iteratively until convergence to stationary points, which are the estimated local modes of the true p.d.f.. Let $\left\{\mathbf{y}_i^{(t)}\right\}_{i=1,\ldots,N}$ be the set of moved points at iteration t, and assume $\mathbf{y}_i^{(0)} = \mathbf{x}_i \ \ \forall i$.

We briefly recall below the original MS and BMS update rules, i.e. the operation applied to the data points at each iteration.

A. MEAN-SHIFT (MS)

MS can be used to partition a dataset by assigning each data point a label corresponding to the unique point it converges to after some (expected) finite number of iterations of an update equation. More precisely, with the above notations, the update equation of MS writes, $\forall i$:

$$\mathbf{y}_i^{(t+1)} = \frac{\sum_{k=1}^{N} f\left(\mathbf{x}_k, \mathbf{y}_i^{(t)}\right) \cdot \mathbf{x}_k}{\sum_{j=1}^{N} f\left(\mathbf{x}_j, \mathbf{y}_i^{(t)}\right)} \ . \tag{1}$$

With the above assumption on the kernel function f, the updated points $\{\mathbf{y}_i^{(t+1)}\}_{i=1,\ldots,N}$ are obtained as a convex linear combination (i.e. with non-negative coefficients summing up to unity) of the *initial* data points $\{\mathbf{x}_i\}_{i=1,\ldots,N}$.

B. BLURRING MEAN-SHIFT (BMS)

The update rule for BMS is different from MS since it is based on a convex linear combination of the *previously moved* data points (hence the so-called blurring effect, as coined by Cheng in [27]), i.e. $\forall i$:

$$\mathbf{y}_i^{(t+1)} = \frac{\sum_{k=1}^{N} f\left(\mathbf{y}_k^{(t)}, \mathbf{y}_i^{(t)}\right) \cdot \mathbf{y}_k^{(t)}}{\sum_{j=1}^{N} f\left(\mathbf{y}_j^{(t)}, \mathbf{y}_i^{(t)}\right)} \ . \tag{2}$$

It can be noticed that, over the iterations, this update rule progressively 'forgets' the initial data points in $\mathscr{X}$, contrarily to MS.

C. DISCUSSION

1) MAIN ADVANTAGES AND DRAWBACKS

As pointed out in [26] MS and BMS algorithms have several advantages compared to other clustering techniques, among which:

- their parametrization is limited to the choice of an appropriate kernel function, and a single bandwidth (or aperture) parameter for this kernel function;
- they allow to discover non-convex clusters;
- they can automatically determine the number of clusters, depending on the chosen bandwidth parameter;
- the algorithms are totally deterministic.

These advantages make a significant difference with respect to classical clustering methods like k-means or fuzzy c-means for which none of the last three items above is ensured. However, MS-based methods have two main drawbacks which are the lack of scalability to large datasets, and a high sensitivity of clustering performances for high-dimensional datasets with respect to the bandwidth parameter [26]. The latter is a consequence of the so-called *curse of dimensionality* [36] and the fact that distances tend to be less meaningful in high dimensions.

2) CONVERGENCE OF MS-BASED METHODS

Since the early works on MS-based clustering, and until very recently, the convergence of MS and BMS has been studied. A comprehensive study of the convergence of both MS and BMS algorithms is summarized in [26], but specific results were provided for MS in [25], [35], and for BMS in [1], [27].

For MS, the convergence of moved points to local modes is ensured theoretically and practically in the general case. However the situation may vary from one kernel to another as for the number of iterations required to convergence: the latter is proven in a finite number of steps for the Epanechnikov kernel [25], but infinite for the Gaussian kernel, moreover with a linear convergence rate [37].

For BMS, the convergence issue depends on the kernel aperture: for large ones which encompass the whole dataset $\mathscr{X}$, convergence is ensured to a unique mode, whereas for narrower ones with finite support, the BMS update rule allows the iterates to converge quickly to well-separated distinct modes during the first steps. However, pursuing the iterations can eventually lead to the merging of these modes into a single one. Therefore BMS must be stopped before this case occurs [29]. The convergence rate of Gaussian BMS has been proven cubic [35], hence much faster than Gaussian MS.

3) KNN-MS BASED METHODS

Choosing the optimal set of hyperparameters (kernel function and bandwidth) for a specific task can become very challenging. This is why, since the early works on Mean-Shift-based density estimation and data clustering [11], [12], the use of nearest neighbors as an alternative to the standard kernel approach has been proposed by many researchers. Indeed, the NN approach (*i*) does not require to specify an underlying parametric function, so that only one parameter K is required; and (*ii*) the KNN principle makes it possible to maintain the relationship between data points located far from each other, especially on the external border of clusters. In this sense, the NN-based framework is data-adaptive, contrarily to the kernel-based one. In [11], a mean-shift estimate calculated from K nearest neighbors was proposed as a natural way to automatically adapt the density estimation to its local variations. The nearest neighbor paradigm can also be used to estimate the bandwidth parameter of kernel-based MS, for instance as the average distance of each point to its KNNs [26].

Koontz *et al.* [12] adopted a graph-based clustering approach which enables the assignment of any data point (or node) to a parent node, using the number of neighbors found within the constant radius ball centered on it. The clustering result is then produced via a directed tree traversal [38]. In [29], Grillenzoni proposed a Gaussian BMS based on KNNs, which provides a data-driven technique to select the bandwidth and is shown to have low sensitivity to K. Duong *et al.* [30] also provided a data-driven closed-form solution to estimate the optimal number of NNs in NN-MS. Recently, Beck *et al.* [31] proposed an extension of the Nearest Neighbor Gradient Ascent (NNGA) method (which is in fact a KNN version of MS) incorporating Locality Sensitivity Hashing (LSH) to approximate nearest neighbors and ϵ-proximity cluster labeling rule into NNGA, and renamed this method NNGA$^+$. NNGA and NNGA$^+$ have the advantage of being scalable for large datasets.

4) IMPLEMENTATION ISSUES

An important issue of MS-based methods is related to their practical implementation. Actually, there is a major difference between MS and BMS in this regard. Indeed, Eq. (1) shows that each point can be treated independently from others because, as soon as kernel evaluations are computed, the update $y_i^{(t+1)}$ is a linear combination of the *original* data points. In contrast, BMS in Eq. (2) requires the whole set of current iterates to calculate $y_i^{(t+1)}$. Therefore, BMS cannot be parallelized, whereas MS can be parallelized efficiently. For both approaches and for arbitrary data, the complexity is quadratic in the number of points [26].

III. ROBUST MEAN-SHIFT

In this section, we propose another MS-like clustering approach, which we name Robust Mean-Shift (RMS). The rationale behind RMS is to combine MS and BMS, so that the next iterate remains a convex linear combination of the current ones, like BMS in Eq. (2), whereas the kernel weighting is kept identical to the MS update in Eq. (1). Therefore, the proposed update equation writes:

$$\mathbf{y}_i^{(t+1)} = \frac{\sum_{k=1}^{N} f\left(\mathbf{x}_k, \mathbf{y}_i^{(t)}\right) \cdot \mathbf{y}_k^{(t)}}{\sum_{j=1}^{N} f\left(\mathbf{x}_j, \mathbf{y}_i^{(t)}\right)} \quad . \tag{3}$$

The main idea of RMS is based on the following expectations:

- Since the next iterate is a combination of the current ones, a behavior similar to BMS can be anticipated, especially in terms of faster convergence with respect to MS;
- Since the kernel weights remain dependent on the initial data points $\{\mathbf{x}_k\}_{k=1,\dots,N}$, it is expected that iterates remain 'bound' to the initial data points, therefore avoiding convergence to a unique mode for broad kernels, which is a known drawback of BMS [35];
- The kernel-based update rule in Eq. (3) can be easily modified to involve nearest neighbors, similarly to NNGA [31] and the graph-theoretic approach in [12].

Surprisingly, this update equation has not been reported so far in the literature to the best of our knowledge.

Figure 1 illustrates with a simple example the differences between MS, BMS and RMS. For the three methods, the same kernel parametrization is used, i.e.

$$f(\mathbf{u},\mathbf{v}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{\sigma^2}\right) \quad , \tag{4}$$

with bandwidth parameter $\sigma = 0.15$. The dataset is drawn randomly from three 2-D normal distributions with identical diagonal covariance matrices, centered at [0, 0], [0, 1], [1, 1]. One can see that the modes of these distributions are hardly distinguishable on Figure 1-(a). Figures 1-(b-d) display the evolution of each data point to its corresponding mode for the three methods.

In this example, RMS is able to recover the three components of the original distribution, as well as tiny modes or single outliers, whereas MS and BMS identify a higher number of local modes after convergence. Moreover, RMS requires less iterations to converge (9 iterations) than MS (96 iterations) and BMS (33 iterations).

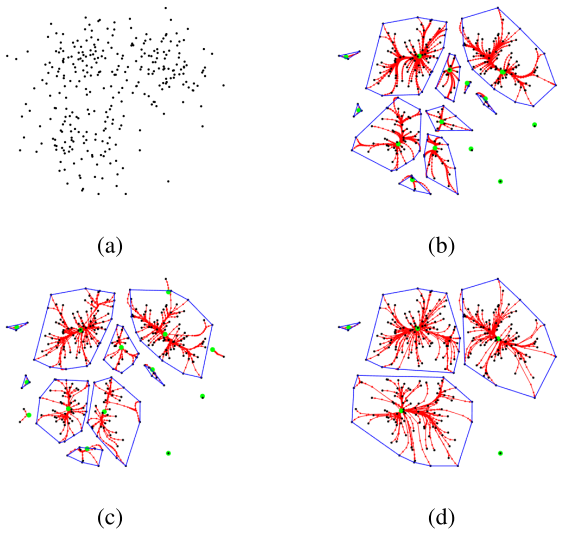


FIGURE 1. Illustration of the differences between MS, BMS and RMS mode seeking algorithms, for a Gaussian kernel with bandwidth parameter $\sigma = 0.15$. (a) 300 2-D data points drawn randomly from three normal distributions $\mathcal{N}([0, 0], 0.09\mathbf{I})$, $\mathcal{N}([0, 1], 0.09\mathbf{I})$, $\mathcal{N}([1, 1], 0.09\mathbf{I})$. Successive moves of iterates towards their corresponding mode with (b) the MS algorithm (14 modes); (c) the BMS algorithm (14 modes); (d) the RMS algorithm (5 modes). RMS recovers the main three modes, and two extra modes due to outliers. Final modes are displayed as green dots, and the convex hull of the original points belonging to a same final cluster is drawn in blue.

IV. CONVERGENCE OF RMS

In this section, we discuss the convergence property of RMS. Though it is not fully theoretically established here, the convergence of RMS is conjectured, based on the adaptation of the BMS convergence theorem of Chen [1]. This theorem states that there exists $\{\mathbf{y}_1^*, \dots, \mathbf{y}_N^*\}$ s.t. $\lim_{t\to\infty} \mathbf{y}_i^{(t)} = \mathbf{y}_i^*$. This theorem is based on three lemmas:

- The convex hull of all updated data points along the iterations are nested and converge to a limiting convex hull (Lemma 1);
- For each vertex of the converged convex hull, at least one sequence of the data points converge to this vertex (Lemma 2);
- The influence of the vertices of one converging convex hull to other data points outside this convex hull vanishes along the iterations (Lemma 3).

Based on these partial results, we discuss below the convergence of RMS under the same assumptions on f.

First, it is easy to show that Lemma 1 in [1] still holds for RMS. Let $C_1^{(t)}$ be the convex hull of the set of data points $\{\mathbf{y}_1^{(t)}, \dots, \mathbf{y}_N^{(t)}\}$ at iteration t. Since $\mathbf{y}_i^{(t+1)}$ in the updating equation (3) is still a convex linear combination of $\{\mathbf{y}_k^{(t+1)}\}_{k\in\{1,\dots,N\}}$ (though different from the one in BMS), then $\mathbf{y}_i^{(t+1)} \in C_1^{(t)}$. The same is true $\forall i$, which implies:

$$C_1^{(t)} \supseteq C_1^{(t+1)} \quad \forall t.$$

The proof of Lemma 2 in [1] requires showing that, for a large enough t, no exchange can happen at iteration $(t+1)$ between a point $\mathbf{y}_{j'}^{(t)}$ in the interior of the convex hull $C_1^{(t)}$ and one of its vertices $\mathbf{y}_j^{(t)}$, i.e. $\mathbf{y}_{j'}^{(t+1)}$ cannot 'pass' $\mathbf{y}_j^{(t+1)}$ to become a new vertex of $C_1^{(t+1)}$. The transposition of this lemma to the RMS update equation remains to be formally proven, but our computer simulations indicate that the same lemma can be conjectured.

The Lemma 3 of Chen can also be transposed to RMS. More precisely, the adaptation of Eq. (6) in [1] writes:

$$\sum_{k\neq i} f\left(\mathbf{x}_k, \mathbf{y}_i^{(t)}\right) \cdot \left(\mathbf{y}_k^{(t)} - \mathbf{y}_i^{(t+1)}\right) = f\left(\mathbf{x}_i, \mathbf{y}_i^{(t)}\right) \cdot \left(\mathbf{y}_i^{(t+1)} - \mathbf{y}_i^{(t)}\right) \tag{5}$$

where without loss of generality $\mathbf{y}_i^{(t)}$ is assumed being the only point converging to a vertex of $C_1^{(t)}$. For large enough t, since $\mathbf{y}_i^{(t+1)} - \mathbf{y}_i^{(t)}$ vanishes to zero from the above adapted Lemma 1, then depending on k, either $f\left(\mathbf{x}_k, \mathbf{y}_i^{(t)}\right)$ or $\mathbf{y}_k^{(t)} - \mathbf{y}_i^{(t+1)}$ go down to zero. In the first case, $\mathbf{y}_i^{(t)}$ is no longer influenced by $\mathbf{x}_k$, and $\mathbf{y}_k^{(t)}$ will converge to another limit point, whereas in the second case, $\mathbf{y}_k^{(t)}$ will converge to the same limit point as $\mathbf{y}_i^{(t)}$. Note that this only sketches the proof of the third Lemma.

In summary, convergence of RMS can be conjectured at this point, but further investigations would be necessary to thoroughly prove it. In addition, why RMS provides larger attraction basins than MS and BMS remains an open question.

V. NEAREST-NEIGHBOR RMS

Similarly to MS and BMS, the kernel-based RMS algorithm can be cast to a nearest-neighbor-based algorithm, by setting f as a variable-radius flat kernel based on the KNNs:

$$f_{KNN}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \text{KNN}(\mathbf{y}) \\ 0 \text{ else} \end{cases} . \tag{6}$$

This yields the following nearest-neighbor robust Mean-Shift (NN-RMS) update rule:

$$\begin{aligned} \mathbf{y}_i^{(t+1)} &= \frac{\sum_{k=1}^{N} f_{KNN}\left(\mathbf{x}_k, \mathbf{y}_i^{(t)}\right) \cdot \mathbf{y}_k^{(t)}}{\sum_{j=1}^{N} f_{KNN}\left(\mathbf{x}_j, \mathbf{y}_i^{(t)}\right)} \\ &= \frac{1}{K} \sum_{k:\mathbf{x}_k \in \text{KNN}(\mathbf{y}_i^{(t)})} \mathbf{y}_k^{(t)} \end{aligned} \tag{7}$$

The corresponding algorithm is detailed in Algorithm 1.

Notice that, contrarily to the kernel-based RMS approach, the new one is point-wise (local), i.e. its radius around one data point $\mathbf{y}_j$ is equal to the distance to its Kth NN in $\mathscr{X}$. By doing so, one can expect that NN-RMS better captures the local complexity of the data distribution, compared to the kernel-based version.

Algorithm 1 NN-RMS

Input:
$\mathscr{X} = \{\mathbf{x}_m\}, \mathbf{x}_m \in \mathbb{R}^n, m = 1, \ldots, N$; the dataset
K, the number of NNs;
Output: The vector of points' labels $\mathbf{c} = [c_1, \ldots, c_N]^t$;
the set of cluster exemplars $\mathscr{E}$;
1) Initialize $\mathscr{Y}^{(0)} = \left\{\mathbf{y}_m^{(0)}\right\} = \{\mathbf{x}_m\}, m = 1, \ldots, N$;
$t = 1$; $\left\{\mathbf{y}_m^{(1)}\right\} = \left\{\mathbf{y}_m^{(0)}\right\}$;
2) Core loop
while $t = 1 \vee \left\{\mathbf{y}_m^{(t)}\right\} \neq \left\{\mathbf{y}_m^{(t-1)}\right\}$ **do**
  for $m = 1 : N$ **do**
    $\mathbf{j} = [j_1, j_2, \ldots j_K] = \text{KNN}(\mathscr{X}, \mathbf{y}_m^{(t)})$;
    $\mathbf{y}_m^{(t+1)} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{y}_{j_k}^{(t)}$;
    $c_m^{(t+1)} = j_1$;
    $t \leftarrow t + 1$;
  end for
end while
3) Set $\mathscr{E}$ as the set of unique indices in $\mathbf{c}^{(t)}$;
4) Remap $\mathscr{E}$ to cluster labels in $[1 \ldots C]$, $C = |\mathscr{E}|$;

Contrarily to NN-BMS, NN-RMS does not update anymore after a small number of iterations. This can be explained easily: the KNN search in NN-RMS operates so as to find the NNs of the current updates within the set of original data points, which remain static. On the one hand, this is different from NN-BMS in which the KNN search is performed within the set of *current* updates, which do continuously move along the iterations. On the other hand, the KNN search is similar to NN-MS (or NNGA [31]), but since NN-RMS is essentially a BMS algorithm (because next updates are weighted sums of the current moving points), its convergence is faster than MS. Indeed, NN-RMS stops if all the moving points have reached the condition that they share the *same* NNs within the *original* dataset, even if the last current iterates are close to each other, but distinct.

Figure 2 exemplifies the differences between three NN-based MS algorithms for mode seeking, namely Nearest-Neighbor Mean-Shift (NN-MS), Nearest-Neighbor Blurring Mean-Shift (NN-BMS), and NN-RMS. The dataset is randomly drawn from three 2-D normal distributions with identical diagonal covariance matrices, and centered at [0, 0], [0, 1], [1, 1]. This dataset is challenging for the clustering task, the modes of the mixture distribution being hard to distinguish on Figure 2-(a). Figures 2-(b-d) display the evolution of each data point to its corresponding mode for the three methods. Notice that NN-RMS is run until strict fixed-point convergence, whereas NN-MS and NN-BMS must be stopped as soon as the mean squared difference between successive iterates $\mathscr{Y}^{(t)}$ and $\mathscr{Y}^{(t-1)}$ is below some threshold ϵ. In this experiment, we set $\epsilon = 10^{-8}$. It can be seen that NN-RMS again creates larger attraction basins than its MS and BMS counterparts. NN-RMS is able to recover the exact number of components in the actual distribution, whereas NN-MS and NN-BMS still identify a much higher number of local modes after convergence. Moreover, NN-RMS requires less iterations to converge (14 iterations) than NN-MS (16 iterations) and NN-BMS (55 iterations), as shown in Figure 3.

VI. EXPERIMENTS

In this section, we provide experimental results obtained with several datasets, both synthetic and real, in order to assess the performance of the proposed RMS approach for clustering, and to compare it with the state-of-the-art MS and BMS, in both configurations, i.e. kernel-based and KNN-based.

A. DATASETS

1) SYNTHETIC DATASETS

To perform the experiments and compare our approach with other clustering algorithms, we have selected a number of publicly available synthetic and real datasets. The synthetic datasets are displayed in Figure 4 with their actual (ground truth) label shown in specific colors. These datasets show diverse configurations, from well separated to highly overlapped clusters, from convex to non-convex and highly intricate clusters, and from balanced to unbalanced clusters.

2) REAL DATASETS

The real datasets used in the experiments are summarized in Table 1. They show different configurations, from low to moderate dimensionality, and various number of instances

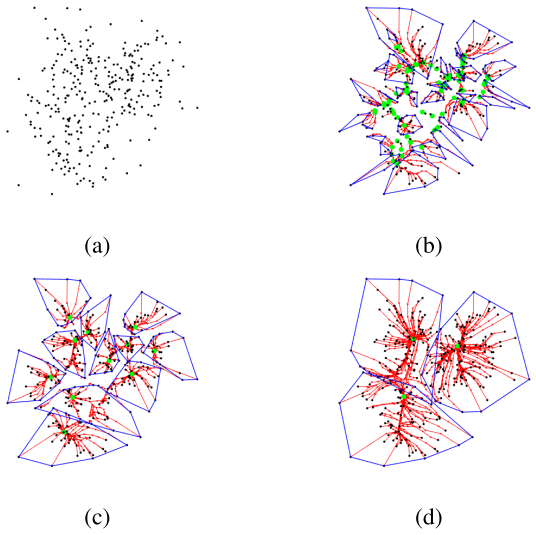


FIGURE 2. Illustration of the differences between NN-MS, NN-BMS and NN-RMS mode seeking algorithms, for $K = 17$. (a) 300 2-D data points drawn randomly from 3 normal distributions $\mathcal{N}([0, 0], 0.16\mathbf{I})$, $\mathcal{N}([0, 1], 0.16\mathbf{I})$, $\mathcal{N}([1, 1], 0.16\mathbf{I})$. Successive moves of iterates towards their corresponding mode with (b) the NN-MS algorithm (56 distinct modes); (c) the NN-BMS algorithm (11 distinct modes); (d) the NN-RMS algorithm. NN-RMS recovers the exact number of 3 modes. The final modes are displayed as green dots, and the convex hull of the original points belonging to a same final cluster is drawn in blue.

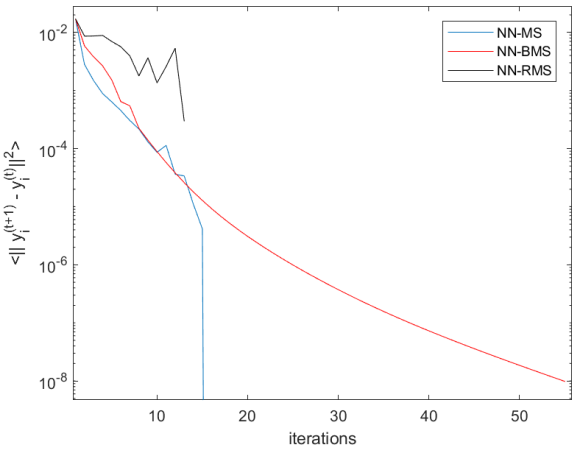


FIGURE 3. Evolution of the mean squared error (in log scale) between successive iterates for NN-MS, NN-BMS and NN-RMS for the dataset shown in Figure 2. Notice that NN-RMS is stopped at iteration 14 with zero mean squared error.

and number of clusters. All these datasets have been used without any pre-processing, except the *AttFace* dataset for which the original dimension $n = 4096$ has been reduced to $n = 20$ by means of principal component analysis (PCA).

B. SELECTED METHODS

With regard to similar methods based on the MS principle and their NN variants, we have compared both the NN-based and the kernel-based RMS proposed method to their equivalent MS and BMS counterparts. For comparison, we selected the

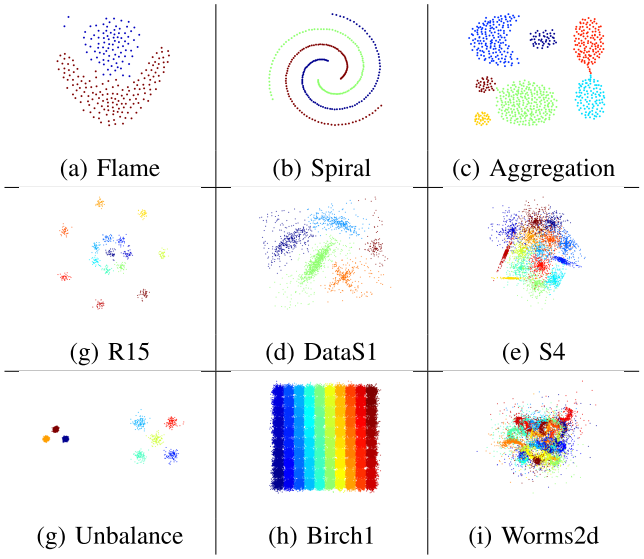


FIGURE 4. Various 2D synthetic datasets used in experiments. All but the DataS1 dataset (available at https://www.science.org/doi/10.1126/science.1242072) can be downloaded from http://cs.joensuu.fi/sipu/datasets/.

TABLE 1. Real datasets.

Name	N	n	C_{true}	Location
Iris	150	4	3	https://archive.ics.uci.edu/
Ecoli	336	7	8	https://archive.ics.uci.edu/
Seeds	210	7	3	https://archive.ics.uci.edu/
Banknote	1372	4	2	https://archive.ics.uci.edu/
Segment	2310	19	7	https://archive.ics.uci.edu/
AttFace	400	20	40	https://scikit-learn.org/

MedoidShift method proposed in [38], which was adapted to the KNN case, as suggested by the authors. We also compared our approach with two nearest-neighbor density-based clustering methods, namely kNN-DPC, and GWENN. These methods were recently improved and compared in [24] for their applicability to pixel clustering in hyperspectral images. They were chosen because they require the same input parameter K as the NN-based MS methods. Also, due to the specific convergence of NN-BMS as illustrated in Figure 3, an additional parameter ϵ was used to stop the algorithm. In all our experiments, we set $\epsilon = 10^{-6}$. For kernel-based MS methods, we have chosen the Gaussian kernel detailed in Eq. (4).

C. SELECTED VALIDATION CRITERIA

To allow the comparison of different results, we used the following cluster validation criteria:

- Since all the datasets include a ground truth labeling as an external data for cluster assessment, the overall accuracy (OA), average accuracy (AA), kappa index, can be obtained after optimal pairing of cluster labels with

the actual ground truth classes owing to the Munkres assignment algorithm [39];
- The purity and normalized mutual information (NMI) indices [40] are also based on the relationship between the ground truth and the predicted labels, but do not require label reassignment;
- The consistency violation ration (CVR) [41] is a clustering index based on an information-theoretic concept, which is also fitted to non-convex clusters; a lower CVR indicates a better clustering result. Contrarily to the previous criteria, the CVR index does not require the ground truth labels, which makes it useful to assess the results of unsupervised classification.

In Figure 5, we illustrate the evaluation of the results obtained by applying NN-MS, NN-BMS, NN-RMS, and NN-MedoidShift to the *Flame* dataset, as a function of the number of NNs K in the range $[6, 50]$. The results clearly show that NN-RMS provides an estimation of the correct number of clusters close to the actual one for a much lower number of NNs K. In comparison, NN-MS gives a very high number of clusters, as well as NN-BMS and NN-MedoidShift. The latter reaches the correct number of clusters for a higher number of NNs than NN-RMS, and for NN-BMS even higher. The OA index shown on Figure 5-(b) reveals the potential of NN-RMS to quickly approach the ideal partition for very low K. On Figure 5-(c), the low CVR indices within a large range of K confirms the robustness of NN-RMS with respect to the selection of the parameter K.

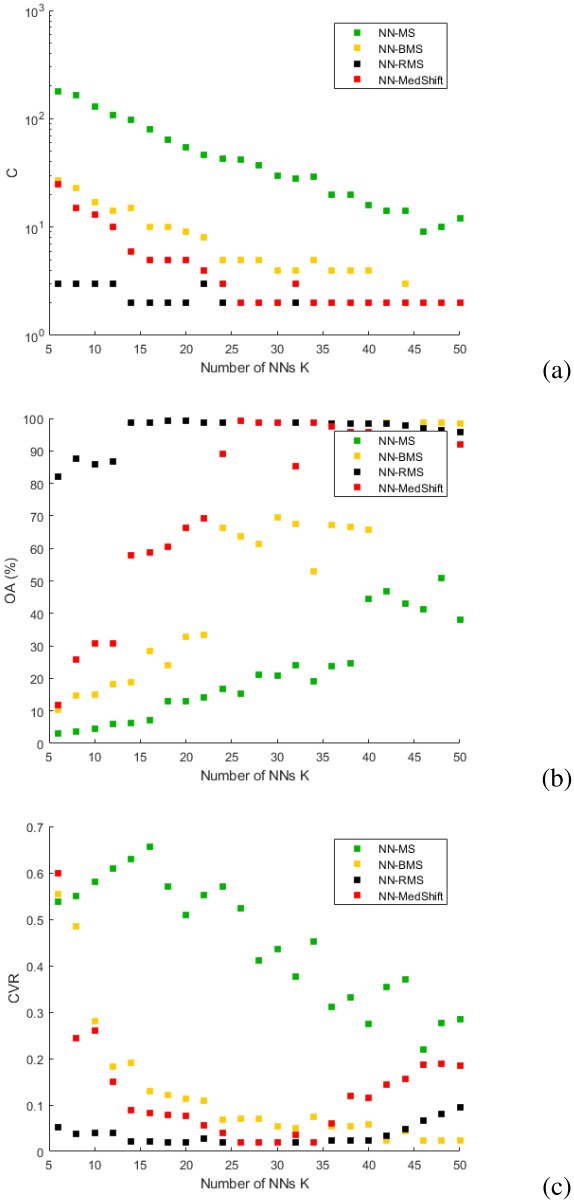


FIGURE 5. Clustering results for the *Flame* dataset for various methods, as a function of the number of neighbors K. (a) Number of clusters; (b) Overall Accuracy; (c) Consistency Violation Ratio (CVR).

D. RESULTS

1) SYNTHETIC DATA

The results obtained on synthetic datasets are given in Table 2. In addition to the cluster indices mentioned above, we also added the number of output clusters, the computation time and the number of iterations for each method used for comparison. For each dataset, the clustering task was performed with two groups of methods, namely NN-based and kernel-based methods. The values displayed in Table 2 correspond to the specific values of K (for NN-based methods) or σ (for the kernel-based methods) providing the best kappa index.

Concerning the NN-based methods, it can be seen that NN-RMS provided the best kappa indices on five over the nine datasets (*Aggregation*, *DataS1*, *S4*, *Unbalance* and *Worms2d*). Furthermore, NN-RMS provided the second best kappa on three other datasets (*Flame*, *Spiral* and *Birch1*). One important issue of this comparison is that these best kappa results for NN-RMS were obtained for the lowest number of NNs among all the compared methods on eight over the nine datasets. This result can be considered as significant since most of the optimal values of K for the other methods are often more than twice the optimal values for NN-RMS. It is also noticeable that NN-RMS found the correct number of clusters in seven over nine datasets, and that this number is very close to the actual one in the remaining cases.

With regard to kernel-based Mean-Shift methods, over the nine datasets, G-RMS again outperforms G-MS and G-BMS on six datasets (*Flame*, *Spiral*, *Aggregation*, *DataS1*, *S4* and *Worms2d*). Also, in all cases, G-RMS never requires a higher Gaussian kernel aperture than G-MS and G-BMS to achieve the results with the best kappa indices, and for most datasets the optimal aperture is well below the ones provided by G-MS and G-BMS. This finding is in accordance with the results of the NN-based MS methods.

Note that for the *R15* dataset, all methods (NN- or kernel-based) perform equally well, except NN-MS and NN-MedShift with lower kappa, NMI and purity, and higher CVR.

TABLE 2. Comparison of clustering methods for various synthetic datasets.

Dataset	Method	K / σ	C	AA	OA	kappa	CVR	NMI	purity	time (s)	nitr
Flame $n=2$ $N=240$ $C_{true}=2$	**kNNDPC**	28	**2**	**100**	**100**	**1**	0.026	**1**	**1**	**0.00186**	5
	GWENN	26	**2**	98.5	98.8	0.973	0.0197	0.899	0.988	0.00943	2
	NN-MS	48	10	53.5	50.8	0.332	0.277	0.479	0.508	0.00662	35
	NN-BMS	42	**2**	98.3	98.8	0.973	0.0238	0.911	0.988	4.03	843
	NN-RMS	**18**	**2**	99.1	99.2	0.982	**0.0186**	0.927	0.992	0.0665	12
	NN-MedShift	26	**2**	99.1	99.2	0.982	**0.0186**	0.927	0.992	0.0265	5
	G-MS	1.7	**2**	98.5	98.8	0.973	0.0197	0.899	0.988	0.824	123
	G-BMS	1.7	5	75.8	70.4	0.538	0.0605	0.622	0.704	0.0826	10
	G-RMS	**1.08**	**2**	**99.1**	**99.2**	**0.982**	**0.0186**	**0.927**	**0.992**	0.201	22
Spiral $n=2$ $N=321$ $C_{true}=3$	**kNNDPC**	14	**3**	**100**	**100**	**1**	0.0686	**1**	**1**	**0.0026**	7
	GWENN	12	**3**	85.9	85.9	0.789	**0.0432**	0.634	0.859	0.0127	2
	NN-MS	22	108	27.5	27.6	0.129	0.231	0.314	0.279	0.00957	21
	NN-BMS	16	14	26.5	26.6	0.194	0.119	0.318	0.266	0.411	71
	NN-RMS	**8**	**3**	90.7	90.7	0.861	0.064	0.721	0.907	0.0883	12
	NN-MedShift	12	12	34.3	34.3	0.258	0.118	0.357	0.343	0.033	6
	G-MS	1.76	21	50.4	50.3	0.403	0.241	0.611	0.503	7.11	1e+03
	G-BMS	**1.39**	26	26.3	26.3	0.16	0.344	0.43	0.263	0.00778	208
	G-RMS	**1.39**	**3**	**95.5**	**95.5**	**0.933**	**0.139**	**0.841**	**0.955**	0.179	16
Aggregation $n=2$ $N=788$ $C_{true}=7$	**kNNDPC**	100	4	57.1	85.7	0.808	**0**	0.86	**1**	**0.0098**	6
	GWENN	120	4	57.1	85.7	0.808	0.0135	0.838	0.977	0.0258	2
	NN-MS	110	40	71	63.1	0.578	0.352	0.659	0.666	0.0456	19
	NN-BMS	100	5	64.3	72.6	0.669	0.0338	0.794	0.805	0.865	36
	NN-RMS	**20**	**7**	**100**	**100**	**1**	**0**	**1**	**1**	0.456	17
	NN-MedShift	100	5	65.8	76.9	0.716	0.0586	0.782	0.84	0.0847	3
	G-MS	1.25	27	65.3	46.6	0.418	0.128	0.718	0.466	8.13	344
	G-BMS	1.19	15	70.6	52	0.472	0.0601	0.784	0.52	1.43	81
	G-RMS	**0.824**	**7**	**100**	**100**	**1**	**0**	**1**	**1**	2.92	118
R15 $n=2$ $N=600$ $C_{true}=15$	**kNNDPC**	16	**15**	**99.7**	**99.7**	**0.996**	**0.00181**	**0.994**	**0.997**	**0.00252**	4
	GWENN	16	**15**	**99.7**	**99.7**	**0.996**	**0.00181**	**0.994**	**0.997**	0.0157	2
	NN-MS	38	16	99.5	99.5	0.995	0.0025	0.993	0.995	0.0239	8
	NN-BMS	22	**15**	**99.7**	**99.7**	**0.996**	**0.00181**	**0.994**	**0.997**	0.286	18
	NN-RMS	**10**	**15**	**99.7**	**99.7**	**0.996**	**0.00181**	**0.994**	**0.997**	0.207	12
	NN-MedShift	16	**15**	99.3	99.3	0.993	0.0024	0.989	0.993	0.0873	4
	G-MS	0.424	**15**	**99.7**	**99.7**	**0.996**	**0**	**0.994**	**0.997**	0.247	13
	G-BMS	0.424	**15**	**99.7**	**99.7**	**0.996**	**0**	**0.994**	**0.997**	0.0186	7
	G-RMS	**0.267**	**15**	**99.7**	**99.7**	**0.996**	**0**	**0.994**	**0.997**	0.238	13
DataS1 $n=2$ $N=2000$ $C_{true}=5$	**kNNDPC**	200	4	78.5	92.8	0.899	0.0173	0.897	**0.986**	**0.0121**	7
	GWENN	180	4	78.5	92.8	0.898	0.0198	0.892	0.985	0.0804	2
	NN-MS	180	29	83	70.3	0.643	0.0886	0.721	0.703	0.391	16
	NN-BMS	140	10	74.4	59	0.528	0.0337	0.714	0.59	4.81	21
	NN-RMS	**60**	**5**	**97.5**	**98.4**	**0.977**	**0.0167**	**0.938**	0.984	3.17	15
	NN-MedShift	190	**5**	66.7	71.2	0.63	0.0322	0.73	0.749	1.19	5
	G-MS	0.0815	14	95.6	95.9	0.943	0.0442	0.9	0.959	2.5	24
	G-BMS	0.0815	7	95.7	95.9	0.943	0.0212	0.896	0.959	1.25	15
	G-RMS	**0.0409**	6	**98.7**	**99.4**	**0.991**	**0.012**	**0.976**	**0.994**	0.977	9
S4 $n=2$ $N=5000$ $C_{true}=15$	**kNNDPC**	190	**15**	79.8	79.9	0.785	0.0215	0.728	79.9	**0.0387**	6
	GWENN	140	**15**	79.9	80	0.785	0.0227	0.726	80	0.263	2
	NN-MS	200	36	73.2	73.3	0.716	0.0416	0.704	73.3	1.83	26
	NN-BMS	200	**15**	79.8	79.9	0.784	0.0234	0.722	79.9	1.04	14
	NN-RMS	**40**	**15**	**80.2**	**80.3**	**0.789**	**0.0209**	**0.732**	**80.3**	0.504	14
	NN-MedShift	90	**15**	79.8	79.9	0.784	0.0235	0.722	79.9	0.167	4
	G-MS	4.4e4	17	79.8	79.9	0.785	0.0232	0.728	0.799	36.3	63
	G-BMS	3.5e4	26	79.1	79.1	0.776	0.0236	0.724	0.791	69.4	61
	G-RMS	**2.2e4**	19	**79.9**	**79.9**	**0.785**	**0.019**	**0.729**	**0.799**	36.3	23
Unbalance $n=2$ $N=6500$ $C_{true}=8$	**kNNDPC**	170	4	50	93.8	0.914	**0**	0.954	**1**	**0.0321**	7
	GWENN	150	4	50	93.8	0.914	**0**	0.954	**1**	0.186	2
	NN-MS	200	73	58.4	26.2	0.213	0.0836	0.564	0.266	4.37	48
	NN-BMS	190	24	32.6	21.8	0.167	0.0297	0.584	0.257	70.8	28
	NN-RMS	**50**	**8**	**100**	**100**	**1**	7e-5	**1**	**1**	50.2	22
	NN-MedShift	200	12	30.7	46.4	0.374	0.0348	0.681	0.525	16	6
	G-MS	**1.05e4**	**8**	**100**	**100**	**1**	**7e-5**	**1**	**1**	34.8	39
	G-BMS	**1.05e4**	**8**	**100**	**100**	**1**	**7e-5**	**1**	**1**	2.95	1e+03
	G-RMS	**1.05e4**	**8**	**100**	**100**	**1**	**7e-5**	**1**	**1**	788	1e+03
Birch1 $n=2$ $N=100000$ $C_{true}100$	**kNNDPC**	**300**	**100**	97.8	97.8	0.977	**0.00615**	0.973	97.8	**0.808**	7
	GWENN	500	**100**	**99.3**	**99.3**	**0.993**	0.00664	**0.99**	**99.3**	7.05	2
	NN-MS	500	183	83.2	83.2	0.83	0.014	0.957	83.2	85.3	31
	NN-BMS	500	122	88.7	88.6	0.885	0.00865	0.974	88.6	172	59
	NN-RMS	500	101	99.2	99.2	0.992	0.00673	0.989	99.2	50.9	18
	NN-MedShift	**300**	**100**	99	99	0.99	0.00688	0.986	99	9.61	5
	G-MS	3.06e4	102	**99.4**	**99.4**	**0.994**	0.00668	**0.991**	**99.4**	2.36e3	36
	G-BMS	2.43e4	**100**	99.4	99.4	0.993	0.00657	0.991	99.4	205	10
	G-RMS	**1.53e4**	**100**	99	99	0.99	**0.0064**	0.986	99	186	11
Worms2d $n=2$ $N=105600$ $C_{true}=35$	**kNNDPC**	800	40	56.2	56.5	0.551	**0.00738**	0.639	59.2	**1.62**	9
	GWENN	750	40	55.3	55.6	0.543	0.00805	0.636	58.5	10.6	2
	NN-MS	1e+03	331	42.5	39.4	0.382	0.0341	0.583	40.9	1e+03	175
	NN-BMS	950	75	39.6	32.8	0.318	0.0115	0.598	32.9	455	78
	NN-RMS	**350**	36	**60.5**	**59.6**	**0.582**	0.00985	**0.646**	**63.2**	47.6	21
	NN-MedShift	950	46	45	44.1	0.426	0.00994	0.604	47.2	32.9	6
	G-MS	89.9	182	49.4	53.2	0.517	**0.0105**	0.626	59.5	4.08e4	469
	G-BMS	89.9	229	49.2	43.6	0.423	0.0114	0.603	46.9	2.00e3	23
	G-RMS	**28.4**	613	**55.8**	**54.5**	**0.531**	0.0143	**0.627**	**61.4**	3.78e3	54

TABLE 3. Comparison of clustering methods for various real datasets.

Dataset	Method	K	C	AA	OA	kappa	CVR	NMI	purity	time (s)	nitr
Iris $n=4$ $N=150$ $C_{true}=3$	**kNNDPC**	20	**3**	90.7	90.7	0.86	0.0172	0.806	0.907	**0.00255**	4
	GWENN	20	**3**	89.3	89.3	0.84	0.0223	0.791	0.893	0.00741	2
	NN-MS	46	5	84	84	0.776	0.182	0.735	0.84	0.00403	11
	NN-BMS	48	**3**	**94**	**94**	**0.91**	0.13	**0.817**	**0.94**	0.0362	6
	NN-RMS	**12**	**3**	90	90	0.85	**0.0151**	0.798	0.9	0.0516	12
	NN-MedShift	26	**3**	90	90	0.85	0.0812	0.778	0.9	0.0107	4
Ecoli $n=7$ $N=336$ $C_{true}=8$	**kNNDPC**	40	3	34.7	76.5	0.662	0.0612	0.662	**0.946**	0.0371	4
	GWENN	26	3	34.7	76.2	0.659	**0.0437**	**0.663**	0.943	**0.00897**	2
	NN-MS	50	6	36.4	75.6	0.658	0.209	0.636	0.836	0.0161	15
	NN-BMS	46	4	**41.7**	**78**	**0.694**	0.108	0.648	0.86	0.204	11
	NN-RMS	**8**	3	34.7	76.2	0.659	0.0445	**0.663**	0.943	0.18	13
	NN-MedShift	40	3	32.1	72.9	0.604	0.114	0.585	0.899	0.0631	4
Seeds $n=7$ $N=210$ $C_{true}=3$	**kNNDPC**	18	**3**	89	89	0.836	0.136	0.719	0.89	**0.00452**	4
	GWENN	20	**3**	90	90	0.85	0.147	0.723	0.9	0.00619	2
	NN-MS	46	**6**	84.8	84.8	0.778	0.402	0.663	0.848	0.00595	21
	NN-BMS	46	**3**	90	90	0.85	0.203	0.704	0.9	0.111	12
	NN-RMS	**8**	**3**	90.5	90.5	0.857	**0.129**	0.731	0.905	0.14	18
	NN-MedShift	16	**3**	**91.9**	**91.9**	**0.879**	0.184	**0.743**	**0.919**	0.0677	5
Banknote $n=4$ $N=1372$ $C_{true}=2$	**kNNDPC**	180	**2**	76.4	73.8	0.498	**0.00974**	0.342	0.738	**0.00847**	6
	GWENN	180	**2**	74.7	72.3	0.469	0.0636	0.257	0.723	0.0783	2
	NN-MS	440	**2**	70.6	68.4	0.392	0.125	0.162	0.684	0.36	35
	NN-BMS	320	**2**	70.3	68.4	0.388	0.0951	0.146	0.684	9	46
	NN-RMS	**120**	**2**	**91.9**	**91.6**	**0.832**	0.0587	**0.594**	**0.916**	3.7	24
	NN-MedShift	160	3	77.3	75	0.583	0.155	0.531	0.75	0.732	4
Segment $n=19$ $N=2310$ $C_{true}=7$	**kNNDPC**	90	13	65	65	0.6	0.215	0.627	0.67	**0.0127**	6
	GWENN	90	8	66.4	66.4	0.609	0.218	0.598	0.689	0.0791	2
	NN-MS	180	19	45.1	45.1	0.371	0.744	0.435	0.494	0.925	36
	NN-BMS	200	**7**	59.5	59.5	0.527	0.409	0.525	0.595	38.9	71
	NN-RMS	**50**	**7**	**71.2**	**71.2**	**0.664**	**0.141**	**0.643**	**0.74**	8.31	18
	NN-MedShift	120	**7**	59	59	0.521	0.559	0.488	0.597	2.25	4
AttFace $n=20$ $N=400$ $C_{true}=40$	**kNNDPC**	6	42	72	72	0.713	0.47	**0.866**	0.762	**0.00229**	3
	GWENN	5	45	70.3	70.3	0.695	0.438	0.856	0.748	0.0143	2
	NN-MS	10	103	53.5	53.5	0.526	**0.404**	0.778	0.555	0.0263	11
	NN-BMS	7	37	**75.8**	**75.8**	**0.751**	0.464	0.846	**0.797**	0.44	30
	NN-RMS	**4**	**40**	70	70	0.692	0.456	0.847	0.76	0.188	12
	NN-MedShift	5	43	67	67	0.662	0.438	0.8	0.693	0.0586	4

2) REAL DATA

Table 3 provides the clustering results obtained on the real datasets described above. Here, only NN-based MS methods were considered, as well as kNNDPC, GWENN and NN-MedShift. Again, these results correspond to the optimal parameter K in terms of output kappa index. The variety in data size (from $N = 150$ to 2310), dimensionality (up to $n = 20$), and cluster shape makes it difficult to draw clear conclusions from these observations. Here, over the six datasets, in terms of kappa index, NN-RMS performs better than the other methods in two cases (*Banknote* and *Segment*), whereas NN-BMS is better for three others (*Iris*, *Ecoli* and *Attface*). However, NN-RMS was able to provide the correct number of clusters in all cases, except for the *Ecoli* dataset for which all the methods failed. This is particularly true for the *Attface* dataset, despite the relative high dimensionality ($n = 20$) and low populated classes (only 10 data points per class) which makes it a challenging clustering problem.

Finally, similarly as above for the synthetic datasets, the best results in terms of kappa index reported for NN-RMS in all cases correspond to lower values of K than that of the other methods.

3) APPLICATION TO PIXEL CLUSTERING IN HYPERSPECTRAL IMAGES

We provide here early experimental results of the application of NN-RMS to hyperspectral image pixel clustering. Hyperspectral images are composed of hundreds of spectral bands covering a specific spectral range, generally including the visible range, the near-infrared range and sometimes the short-wave infrared range. Each pixel can be viewed as a high-dimensional vector of spectral radiances (or reflectances) from which a high amount of valuable information can be extracted to remotely identify objects or land cover types when the hyperspectral camera is operated on board an aerial platform (aircraft of UAV). We have selected a publicly available hyperspectral image [42]. It comprises 86×83 pixels, has 204 spectral bands ($n = 204$) and includes six classes of vegetation cover. Figure 6-(a) and (b) show respectively a color composite of the hyperspectral image and the corresponding ground truth map used for clustering assessment. In this experiment, we applied the same protocol as above, i.e. we applied several nearest-neighbor-based clustering method with K varying in the range from 100 to 400 by steps of 20. Four methods were compared: kNNDPC and GWENN as density-based clustering methods, and NN-MedShift and NN-RMS as Mean-Shift-type methods. For each method, we retained the best result as the one maximizing the kappa index, because this index mixes both the OA and the AA issued from the confusion matrix (after label reassignment) and generally better represents the clustering quality. Figure 6-(c)-(f) displays the corresponding clustering maps with an effort to keep the same color scale than the ground truth. It is interesting to notice that all four methods can discover two additional clusters with respect to the available ground truth. These clusters are visually

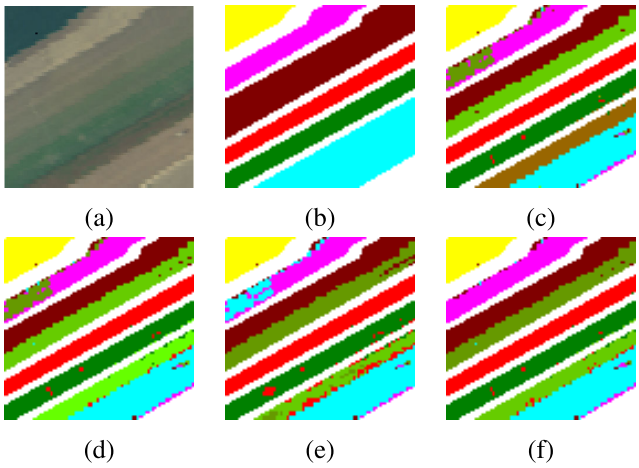


FIGURE 6. Comparison of the best results in hyperspectral image pixel clustering with NN-density based methods and NN-MS methods. (a) *Salinas-A* hyperspectral image, color composite of spectral bands (R: band 30; G: band 20; B: band 10) (b) Ground truth map (white pixels are unlabeled); (c) kNNDPC ($K = 240$, $C = 12$, kappa = 0.666); (d) GWENN ($K = 280$, $C = 9$, kappa = 0.663); (e) NN-MedShift ($K = 380$, $C = 8$, kappa = 0.658); (f) NN-RMS ($K = 200$, $C = 8$, kappa = 0.701).

coherent both from the spectral viewpoint as can be seen from the composite image, and from the spatial viewpoint since the additional segments in the maroon and light blue regions of the ground truth map follow the same spatial structure as the labeled ones. Despite the high dimensionality of the dataset, here again, NN-RMS provides the best overall clustering result among the four methods, still with the lowest number of nearest neighbors K.

VII. CONCLUSION AND PERSPECTIVES

In this paper, we have proposed a novel Mean-Shift-like method to data clustering, called Robust Mean-Shift (RMS). This approach differs from the standard Mean-Shift (MS) and Blurring Mean-Shift (BMS) ones by its update equation. More precisely, RMS uses a linear combination of the current point iterates (similarly to BMS) with weights depending on the similarity (or distance) of these iterates to the original data points (similarly to MS). Surprisingly, the proposed method does not seem to have been studied so far despite its simplicity.

The RMS update equation has been set up in both a kernel-based and a nearest-neighbor-based version. In the kernel-based case, the convergence of point iterates has been conjectured based on the BMS convergence theorem of Chen [1].

RMS has several advantages over MS and BMS:

- For a same kernel bandwidth (in the kernel-based implementation) or number of NNs (in nearest-neighbor-based implementation), RMS shows larger basins of attraction than MS and BMS. One consequence is that the size of the NN graph required to achieve the same number of clusters is smaller for NN-RMS than for NN-MS and NN-BMS.
- Experimental results on synthetic and real datasets show that RMS in most cases outperforms MS and BMS in the clustering task.
- Though the RMS update equation is closer in spirit to BMS, their NN-based versions have different behaviors when the iterates get close to their fixed-point limit: whereas NN-BMS iterates continue to evolve until the mean squared error between the current and the previous iterates reach a specified small value, NN-RMS stops as soon as all iterates share the same set of NNs within the original dataset. This property is also valid for NN-MS (or NNGA).

Perspectives of this work are two-fold, and will concern (*i*) the optimization of RMS parametrization (kernel definition, kernel aperture, and number of NNs), and (*ii*) theoretical proofs of RMS convergence properties, especially with regard to convergence rate and larger attraction basins with respect to the standard MS and BMS clustering methods.

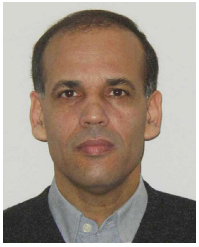
REFERENCES

[1] T.-L. Chen, "On the convergence and consistency of the blurring mean-shift process," *Ann. Inst. Stat. Math.*, vol. 67, pp. 157–176, Oct. 2015.
[2] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.
[3] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221006706
[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, Jan. 1967, pp. 281–297.
[5] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York, NY, USA: Plenum Press, 1981.
[6] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[7] J. H. Ward, Jr., "Hierarchical grouping to optimize an objective function," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 236–244, 1963.
[8] P. Sneath and R. Sokal, *Numerical Taxonomy. The Principles and Practice of Numerical Classification*. London, U.K.: Freeman, 1973.
[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Intern. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[10] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Philadelphia, PA, USA, 1999, pp. 49–60.
[11] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 1, pp. 32–40, Jan. 1975.
[12] Koontz, Narendra, and Fukunaga, "A graph-theoretic approach to nonparametric cluster analysis," *IEEE Trans. Comput.*, vol. C–25, no. 9, pp. 936–944, Sep. 1976.
[13] R. P. W. Duin, A. L. N. Fred, M. Loog, and E. Pekalska, "Mode seeking clustering by knn and mean shift evaluated," in *SSPR/SPR* (Lecture Notes in Computer Science), vol. 7626. Berlin, Germany: Springer, 2012, pp. 51–59.
[14] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., B Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.
[16] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Comput. Statist. Data Anal.*, vol. 14, no. 3, pp. 315–332, 1992.
[17] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems (NIPS)*, S. A. Solla, T. K. Leen, and K. R. Müller, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 554–560.

[18] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[19] M. Sugiyama, G. Niu, M. Yamada, M. Kimura, and H. Hachiya, "Information-maximization clustering based on squared-loss mutual information," *Neural Comput.*, vol. 26, no. 1, pp. 84–131, Jan. 2014.
[20] T. Hocking, J.-P. Vert, F. R. Bach, and A. Joulin, "ClusterPath: An algorithm for clustering using convex fusion penalties," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 745–752.
[21] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba, "Persistence-based clustering in Riemannian manifolds," *J. ACM*, vol. 60, no. 6, p. 38, 2013.
[22] F. Masulli and S. Rovetta, "Clustering high-dimensional data," in *Proc. 1st Intern. Workshop Clustering High–Dimensional Data*, vol. 7627. New York, NY, USA: Springer-Verlag, 2015, pp. 1–13.
[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[24] C. Cariou, S. Le Moan, and K. Chehdi, "Improving K-nearest neighbor approaches for density-based pixel clustering in hyperspectral remote sensing images," *Remote Sens.*, vol. 12, no. 22, p. 3745, Nov. 2020.
[25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[26] M. A. Carreira-Perpiñán, "A review of mean-shift algorithms for clustering," *CoRR*, vol. abs/1503.00687, pp. 1–4, Oct. 2015.
[27] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.
[28] M. A. Carreira-Perpinan, "Generalised blurring mean-shift algorithms for nonparametric clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[29] C. Grillenzoni, "Design of blurring mean-shift algorithms for data classification," *J. Classification*, vol. 33, no. 2, pp. 262–281, Jul. 2016.
[30] T. Duong, G. Beck, H. Azzag, and M. Lebbah, "Nearest neighbour estimators of density derivatives, with application to mean shift clustering," *Pattern Recognit. Lett.*, vol. 80, pp. 224–230, Sep. 2016.
[31] G. Beck, T. Duong, M. Lebbah, H. Azzag, and C. Cérin, "A distributed approximate nearest neighbors algorithm for efficient large scale mean shift clustering," *J. Parallel Distrib. Comput.*, vol. 134, pp. 128–139, Dec. 2019.
[32] X. Li, Z. Hu, and F. Wu, "A note on the convergence of the mean shift," *Pattern Recognit.*, vol. 40, no. 6, pp. 1756–1762, Jun. 2007.
[33] Y. Aliyari Ghassabeh, "A sufficient condition for the convergence of the mean shift algorithm with Gaussian kernel," *J. Multivariate Anal.*, vol. 135, pp. 1–10, Mar. 2015.
[34] K. Huang, X. Fu, and N. D. Sidiropoulos, "On convergence of epanechnikov mean shift," in *Proc. AAAI*, S. A. McIlraith and K. Q. Weinberger, Eds., 2018, pp. 3263–3270.
[35] M. A. Carreira-Perpiñán, "Fast nonparametric clustering with Gaussian blurring mean-shift," in *Proc. ICML*, W. W. Cohen and A. W. Moore, Eds., vol. 148, 2006, pp. 153–160.
[36] R. Bellman, *Adaptive Control Processes*. Princeton, NJ, USA: Princeton Univ. Press, 1961.
[37] M. A. Carreira-Perpinan, "Gaussian mean-shift is an EM algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 767–776, May 2007.
[38] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoidshifts," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[39] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1995.
[40] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[41] G. Ver Steeg, A. Galstyan, F. Sha, and S. DeDeo, "Demystifying information-theoretic clustering," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 1–8.
[42] Grupo de Inteligencia Computacional (UPV/EHU). (2021). *Hyperspectral Remote Sensing Scenes*. [Online]. Available: http://www.ehu.es/ccwintco/index.php/Hyperspectral Remote Sensing Scenes

CLAUDE CARIOU (Member, IEEE) received the Ph.D. degree in electronics from the University of Brest, France, in 1991. Since 1992, he has been with the Engineering School of Applied Sciences and Technology (ENSSAT), University of Rennes 1, where he is currently with the Image Department, Institut d'Électronique et des Technologies du numéRique - UMR CNRS 6164. His research interests include image analysis, pattern recognition, unsupervised classification, texture modeling and segmentation, image registration and feature extraction/selection, mostly dedicated to multispectral and hyperspectral imagery.

STEVEN LE MOAN (Member, IEEE) received the Ph.D. degree in image processing from the University of Burgundy, France, in 2012. He then worked as a Postdoctoral Researcher at the Technical University of Darmstadt, Germany, and the Gjøvik University College, Norway. Until 2021, he was a Senior Lecturer in electronics, information and communication systems with Massey University, New Zealand. He is currently an Associate Professor of color imaging at the Norwegian University of Science and Technology, Gjøvik, Norway. His research interests include multi/hyperspectral image analysis, color science, and visual perception.



KACEM CHEHDI (Member, IEEE) received the Ph.D. and "Habilitation à Diriger des Recherches" degrees in signal processing and telecommunications from the University of Rennes 1, France, in 1986 and 1992, respectively. From 1986 to 1992, he was an Assistant Professor at the University of Rennes 1. Since 1993, he has been a Professor of signal and image processing at the University of Rennes 1. From 1998 to 2003, he has been the Head of the LASTI Laboratory, and of the TSI2M Laboratory (Signal and Multicomponent/Multimodal Image Processing), since 2004. His research activities concern adaptive processing at every level of the pattern recognition chain. In the framework of blind restoration and blind filtering, his main interests are the identification of the physical nature of image degradations and the development of adaptive algorithms. In the segmentation and registration topics, his research concerns the development of unsupervised, cooperative, and adaptive systems. The main applications under current investigation concern multispectral and hyperspectral image processing and analysis.