Nearest neighbor-density-based clustering methods for large hyperspectral images

Claude Cariou, Kacem Chehdi

Nearest neighbor - density-based clustering methods for large hyperspectral images

Claude Cariou and Kacem Chehdi

Institute of Electronics and Telecommunications of Rennes - SHINE/TSI2M team
University of Rennes 1 / Enssat
6, rue de Kerampont, 22300 Lannion, France

ABSTRACT

We address the problem of hyperspectral image (HSI) pixel partitioning using nearest neighbor - density-based (NN-DB) clustering methods. NN-DB methods are able to cluster objects without specifying the number of clusters to be found. Within the NN-DB approach, we focus on deterministic methods, e.g. ModeSeek, knnClust, and GWENN (standing for Graph WatershEd using Nearest Neighbors). These methods only require the availability of a k-nearest neighbor (kNN) graph based on a given distance metric. Recently, a new DB clustering method, called Density Peak Clustering (DPC), has received much attention, and kNN versions of it have quickly followed and showed their efficiency. However, NN-DB methods still suffer from the difficulty of obtaining the kNN graph due to the quadratic complexity with respect to the number of pixels. This is why GWENN was embedded into a multiresolution (MR) scheme to bypass the computation of the full kNN graph over the image pixels. In this communication, we propose to extend these previous works on three aspects. Firstly, similarly to knnClust, the original labeling rule of GWENN is modified to account for local density values, in addition to the labels of previously processed objects. Secondly, we set up a modified NN search procedure within the MR scheme, in order to stabilize of the number of clusters found from the coarsest to the finest spatial resolution. Finally, we show that these extensions can be easily adapted to the three other NN-DB methods (ModeSeek, knnClust, knnDPC) for pixel clustering in large HSIs. Experiments are conducted to compare the four NN-DB methods for pixel clustering in HSIs. We show that NN-DB methods can outperform a classical clustering method such as fuzzy c-means (FCM), in terms of classification accuracy, relevance of found clusters, and clustering speed. Finally, we demonstrate the feasibility and evaluate the performances of NN-DB methods on a very large image acquired by our AISA Eagle hyperspectral imaging sensor.

Keywords: Nearest neighbor, density estimation, clustering, image partitioning, hyperspectral.

1. INTRODUCTION

Clustering is an important tool in data analysis in general, and particularly pixel interpretation in remote sensing images when no or few reference (ground truth) data is available. Semi-supervised or unsupervised classification is also useful when the reference data (objects' labels), if available, is erroneous;[1] it can indeed help reconsidering prior labeling, to leave aside incorrectly labeled objects. However, despite the vast literature in data clustering,[2,3] it is still a difficult task for which no universal solution has been found yet, especially for very large data sets, such as large size hyperspectral images (HSIs).[4]

Nearest-neighbor density-based (NN-DB) methods are able to cluster objects without explicitly specifying the number of clusters to be found. Among the NN-DB approaches, there are deterministic methods, e.g. ModeSeek,[5] knnClust,[6] and GWENN.[7] NN-DB methods require as input the kNN graph, i.e. the set of distances of objects to their NNs, as well as their indices. As such, they only need to specify the number of neighbors k around each object in the representation space. Recently, a new DB clustering method, called Density Peak Clustering (DPC),[8] has been proposed and quickly received attention. kNN versions of DPC have been proposed since and shown their efficiency for a wide range of applications.[9,10] NN-DB clustering methods are specifically well adapted to non-convex clusters as often observed in high dimensional data, and especially for hyperspectral

Further author information:
E-mail: claude.cariou@univ-rennes1.fr, Telephone: +33 296 469 039

images (HSIs). However, they still suffer from the burden of computing the kNN graph due to its quadratic complexity with respect to the number of pixels. A multiresolution (MR) scheme has been proposed recently[7] to bypass the computation of the full kNN graph over the image, by restricting the NN search to descendants of the cluster modes found at each lower level of the Discrete Wavelet Transform approximation coefficients.

The present work extends the work in Ref. 7 on three points. Firstly, the original labeling rule of GWENN is modified to account for local density values, in addition to the labels of previously processed data objects. Secondly, we set up a modified NN search procedure within the MR scheme, in order to stabilize the number of clusters found from the coarsest to the finest spatial resolution. Finally, we show that these extensions are readily adaptable to the three other NN-DB methods (ModeSeek, knnClust, knnDPC), providing a family of NN-DB algorithms for fast pixel clustering in large images.

Experiments are conducted to assess the reliability and compare the four NN-DB methods for HSI pixel clustering. Using the AVIRIS *Salinas* image and its reference map, we show that NN-DB methods can outperform a classical clustering method such as fuzzy c-means (FCM) in terms of overall accuracy for any number of clusters found. Among the NN-DB methods, we show that the modified MR-GWENN method provides a good trade-off in terms of stability and relevance of found clusters, and clustering speed. Another experiment is performed on a very large image acquired by our AISA Eagle hyperspectral imaging sensor, still showing the efficiency of NN-DB methods in terms of clustering quality and speed.

2. RELATION TO PREVIOUS WORKS AND MOTIVATION

In a recent work,[7] a new nearest-neighbor density-based method named GWENN (standing for *Graph WatershEd using Nearest Neighbors*) has been proposed. GWENN can be seen as a generalization of the watershed method on kNN graphs. It aims at finding the higher density regions in the original representation space of the data set, based on a partial knowledge of the similarities between objects of the data set. GWENN requires the availability of a neighborhood graph, uniquely defined by k, the number of nearest neighbors (NNs) to each object of the data set. GWENN is a non iterative clustering method, similarly to DPC[8] and its kNN variants,[9,10] and contrarily to knnClust[6] and ModeSeek.[5] However GWENN differs from these kNN variants since (i) it does not require selecting cluster exemplars from thresholds in a decision graph; (ii) the labeling rule of an object does not rely only on the label of its NN with higher local density, but also accounts for all its kNNs' labels. The pseudo-code of GWENN is given in Algorithm 1. The main idea of GWENN is to progressively assign class labels to objects, starting from the objects for which the local density is the highest, following a simple rule: a given element takes the *mode label* of its previously labeled kNNs. Actually, the `mode` function is feeded with the labels of the previously processed objects, and outputs a single label value based on the most frequent label among them. Unfortunately, unwanted effects are very likely to occur during the processing, typically label assignment ambiguity caused by multiple modes, i.e. different labels values occurring with exactly the same frequency. In this case, a simple disambiguation rule would consist in assigning the label of the mode which is the closest to the current object. Note that this rule has not been implemented in the original work.

The NN-DB methods investigated in the present work, despite their differences, share several similarities, beyond the fact that they uniquely require a NN graph as input. First, they all implement a labeling decision rule involving local densities. While the calculation of the local density around each object is in the real core of ModeSeek, DPC and kNN variants, and knnClust, it has not yet been extended to GWENN in other way than for ordering objects before propagating labels. Actually, the main differences between these methods rely on the specific labeling decision strategies using information derived from the kNN graph. These similarities and differences have not yet been investigated in the literature to the best of our knowledge. One important issue at this point is the choice of the density function assigned to each object. In the present work, in order to align the four NN-DB methods considered, we chose a unique density function, as follows:[7,11]

$$\rho(\mathbf{x}_m) = \frac{k}{\sum_{\mathbf{x}_j \in k\mathrm{NN}(\mathbf{x}_m)} d(\mathbf{x}_m, \mathbf{x}_j)} = \frac{k}{\sum_{j=1}^{k} \mathbf{D}(m,j)} \quad , \quad 1 \leq m \leq M \quad , \tag{1}$$

where N is the number of objects, k is the number of NNs, $k\mathrm{NN}(\mathbf{x}_m)$ is the set of nearest neighbors to $\mathbf{x}_m$ according to the distance $d(.,.)$, chosen here as the Euclidean metric. Note that other choices of the metric (e.g.

ℓ_1) or the combination of kNN distances to estimate the local density (e.g. maximum distance, median distance) may apply. Though this question would deserve attention, we have not investigated it in the present work.

The motivation behind the present work is threefold. The first objective is to propose modifications of the original NN-DB methods (especially GWENN and knnDPC) in order to align them on a same baseline, i.e. that all the methods only require the availability of a kNN graph, from which pointwise local densities can be easily obtained and used in the labeling decision rules. This effort should then help understanding and explaining the similarities and differences observed in their behavior when clustering large data sets. We should notice here that the optimization of the input parameter k will not be considered an issue in the present work. The second objective is to improve a recent framework in order to tackle the problem of pixel clustering for large images.[7] The choice of a MR scheme is motivated by the difficulty to compute an exact kNN graph for large data sets (with several millions of objects). In this study we propose a modified scheme in order to improve the stability of the number of found clusters from one resolution level to the upper level. The third objective is to observe the distinct behaviors of the four NN-DB methods when used into the MR framework. Due to their proper label assignment rules which lead to different clustering results for the same data set (i.e. MR level), one may expect an amplification of these differences along the MR scheme up to the final cluster map.

Algorithm 1 GWENN

Require:
$\mathbf{X} = \{\mathbf{x}_m\}, \mathbf{x}_m \in \mathbb{R}^n, m = 1, \ldots, M$; % The set of data objects to classify
k, the number of NNs;
Ensure: The vector of objects' labels $\mathbf{c} = [c_1, \ldots, c_M]^t$; the set of exemplars $\mathcal{E}$;
1) Compute $\mathbf{D}$, the $M \times k$ array of distances (in ascending order) between each object and its kNNs.
2) Compute $\mathbf{J} = \{\mathbf{j}_m\}_{m=1,\ldots,M}, \mathbf{j}_m = \left[j_m^1, j_m^2, \ldots, j_m^k\right]$, the $M \times k$ array of indices of each object's kNNs.
3) Compute the pointwise densities vector $\boldsymbol{\rho}$ following Eq. (1).
4) Compute $\boldsymbol{\rho}' = \texttt{DescendSort}(\boldsymbol{\rho})$, keep kNNs indices $\mathbf{i} = [i_1, i_2, \ldots, i_M]^t : \boldsymbol{\rho}' = \boldsymbol{\rho}(\mathbf{i})$.
5) $NC = 1$; % NC is the current number of clusters
$c_{i_1} = NC$; % The "denser" object takes the first label
$\mathcal{E} = \{i_1\}$; % The set of exemplars is initialized with the index of the denser object
$P = \emptyset$;
for $m = 2 : M$ **do**
  $P \leftarrow P \cup i_{m-1}$;
  $Q = P \cap \mathbf{j}_{i_m}$;
  if $Q \neq \emptyset$ **then**
    $c_{i_m} = \texttt{mode}(\mathbf{c}_Q)$;
  else
    $NC = NC + 1$;
    $c_{i_m} = NC$;
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{i_m\}$; % Add i_m to the set of exemplars
  end if
end for

3. NN-DB METHODS: SIMILARITIES, DIFFERENCES AND IMPROVEMENTS

In this section we briefly recall the principles of each NN-DB method, while highlighting their similarities and differences. The objective is to propose a set of methods requiring the same data as input (the kNN graph) and from which are derived the local density values used in the labeling decision rules.

3.1 ModeSeek[5]

ModeSeek is essentially a two-stage algorithm. The first stage is aimed at assigning one and only one neighbor among its kNNs to each object of the data set. This neighbor is chosen among the set of each object's NNs as the one having the highest local density. The second stage is iterative with respect to the data objects; at

each iteration, an object is assigned the label of its selected neighbor. The implementation of ModeSeek is straightforward and leads to a very fast and effective algorithm. Moreover, the convergence to a labeling result is insured and requires only a few number of iterations (less than 10) in most cases. In the present work, only the ModeSeek algorithm is kept unchanged with respect to the original algorithm*.

3.2 knnDPC

The kNN version of the Density Peaks Clustering (DPC) algorithm[8] which is set up in this work is a simplified version of the original algorithm†. More precisely, the connections between data objects are restricted to kNNs, similarly to Ref.(9) and Ref.(10). In our implementation, knnDPC is essentially similar to ModeSeek in its overall structure. The first stage is dedicated to the selection of the unique neighbor among the kNNs of each object, and the second stage is strictly the same as in ModeSeek (label propagation). Actually, the first stage differs between the two algorithms in one subtle point: in ModeSeek this neighbor is selected as the one having the highest local density, but in knnDPC it is chosen as the closest object having higher density than the current object (see Fig. (1)). This has two consequences. First, exemplars provided by the two methods do coincide: this comes from the fact that in both methods an exemplar is an object which has itself as NN, i.e. none of its kNNs has a higher local density than itself. Therefore, the number of clusters found by the two methods is identical for a given kNN graph. The second consequence is that the structure of the 1NN graph created by knnDPC is different from the one given by ModeSeek. Indeed, the connections between objects are shorter in average due to the specific NN selection criterion, therefore yielding a higher density of paths: yet a higher number of local paths must be followed for any object to reach its corresponding exemplar.

With the proposed method, the use of a decision graph is no longer required, contrarily to the original DPC algorithm and to many ones derived from it. This implies a great simplification with respect to the original algorithm, with no further need to set up thresholds in the decision graph: obtaining the cluster representatives (exemplars) is fast and straightforward. However, the outliers' detection capability (halo[8]) is lost, since each object is directly assigned one specific label among those of the retained exemplars. Besides, this allows to keep the method similar to the other NN-DB methods.

3.3 GWENN and knnClust: Mode Weighting

One important improvement in the present work is based on a refinement of the rule used in GWENN for assigning the label of the current object with respect to previously labeled objects. This improvement is justified by the fact that, as said above, only the labels of the higher local density objects, and those of the current object's NNs are required to assign it a label. The idea is therefore to rely additionally on the local densities of these NNs, by weighting the count of each label found among the set of nearest neighbors considered by the local densities of the latter. Recall that the local densities of the objects are first computed and sorted in GWENN and therefore already available for this function. The weighted mode of one object $\mathbf{x}_m$ is therefore the label which, among those of its extended neighbors Q (as defined in Algorithm 1), has the maximum sum of local densities:

$$c(\mathbf{x}_m) = c_m = \arg \max_{c \in \cup \mathbf{c}_Q} \sum_{\mathbf{x} \in Q} \mathbf{1}_{(c_\mathbf{x} = c)} \cdot \rho_\mathbf{x} \tag{2}$$

The modified algorithm, called GWENN-WM (for GWENN with Weighted Mode) replaces the original `mode` function by a `wmode` function integrating this new labeling strategy. Figure 2 illustrates with a simple case the difference in labeling decisions between `mode` and `wmode` functions. Optionally, a second labeling pass is performed to suppress isolated exemplars, i.e. objects which do not represent other objects than themselves.
Concerning knnClust, the modification brought to the original algorithm,[6] which already integrates density estimation, was to strictly apply the same label assignment rule as in GWENN-WM for each object. Note that the sequential ordering of visited objects during the iterations in knnClust is lexicographic, which is not guaranteed to be optimal, despite the authors claim that the results are independent of it. The modified knnClust method will accordingly be referred to as knnClust-WM in the sequel.

*http://svn-mede.ewi.tudelft.nl/trac/MM-ICT/PRTOOLS/browser/modeseek.m?rev=641

†http://people.sissa.it/∼laio/Research/Res_clustering.php

4. APPLICATION TO HSIS: MR-NN-DB FRAMEWORK

Clustering image pixels with the NN-DB methods described above can raise computational problems due to the number of objects to classify. These methods therefore need to be adapted to avoid an exhaustive NN search. Hence, based on our previous works,[7] we propose a multi-resolution framework, which can produce clustering maps at each approximation level, from the coarsest (lowest level) to the finest (uppermost, original level). The approach can be summarized as follows. Firstly, a S-level discrete wavelet transform (DWT) of the original image is performed, with $S \in \mathbb{N}$. For multicomponent images, each image component is processed similarly. Secondly, at the coarsest scale S, an exhaustive search of the k NNs is performed using the pixels of the coarsest approximation image, followed by the application of the chosen NN-DB method, hence on a limited number of pixels. Cluster exemplars are then obtained as pixels of the S-level approximation image. Then we store in lexicographical order the indices of each exemplar pixel as well as its descendants at the upper scale. Thirdly, at each upper scale $s < S$ of the DWT, the kNN search is operated differently: for each exemplar pixel, the four corresponding pixels after image upsampling are added to a pool of candidate exemplars to represent all the pixels of the approximation image at the current scale. More precisely, if the number of found clusters at a given scale is NC, then the number of candidate exemplars at the next upper scale is $NC' = 4NC$ (see Figure 3). This is justified by the consideration that clustering pixels at a given scale should benefit from the analysis at the lower scales, and therefore these pixels must in priority be compared to the descendants of the previously found exemplars. This scheme allows to highly reduce the complexity of the kNN search since for each pixel at the current scale resolution, its kNNs are sought only among a limited set of candidate exemplars. The distances between each query pixel of the corresponding scale approximation and the NC' exemplars are then computed. In the present work, the kNN search is performed using a fixed value of $k = 4$ for any value of $s < S$. This limitation is justified to preserve the consistency of the local density estimation used in the NN-DB methods: limiting the search to 4 NNs is likely to avoid mixing candidate exemplars from close, but different clusters identified at the lower scale, for density estimation. One expected advantage of the 4NN search is the stabilization of the number of clusters found along the inverse discrete wavelet transform (IDWT) reconstruction steps, though this is not true for all the NN-DB methods studied here as will be shown below.

In our experiments on images, the MR scheme was implemented using the Haar DWT. Of course, using a multiresolution scheme adds an extra parameter to the method, i.e. the number of DWT levels S. In the following, we have not investigated the issue of clustering performances regarding S. Also, one should notice that no model of spatial dependence between pixels is assumed when applying a specific NN-DB method at a given MR scale. The only spatial dependence in this framework comes from the upsampling step between MR levels.

5. EXPERIMENTAL STUDY

In this section, we investigate the efficiency of the proposed method to pixel clustering in hyperspectral images, including large images acquired by our AISA Eagle sensor.

5.1 Clustering quality vs. number of NNs

We first studied the behavior of the respective NN-DB methods in the proposed MR framework. To this end, we selected the *Salinas* HSI test image[‡]. This image was acquired by the AVIRIS sensor in 1998 over agricultural fields. The last column of the HSI was removed to allow exact reconstruction with the Haar DWT, and therefore the HSI has a spatial size of 512×216 pixels, and 204 spectral bands. The reference map is composed of 54,129 pixels distributed in 16 vegetation classes. Figure 4 shows a color composite image, as well as the associated reference map. A two level DWT ($S = 2$) was set up in this experiment, and therefore the number of data objects at the first clustering stage (i.e. coarsest scale) is 6,912. In this experiment, the features (spectral bands of the HSI) were normalized in the interval $[0, 1]$ because the average classification performances were found better than by using the original data.

[‡]http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

The study was intended to analyze the clustering performances with respect to the size of the kNN graph. More precisely, we compared the four NN-DB methods in terms of external validation criteria (average correct classification rate (ACCR), overall correct classification rate (OCCR), kappa index) using the reference data.

Figure 5 first shows the evolution the number of clusters found versus the number of NNs k (by steps of 2) for the four NN-DB methods considered. As expected, increasing the size of the graph induces a decrease in the number of clusters for most methods because the high connectivity between objects in the NN graph leads to a limited number of retained exemplars. Contrarily, reducing k would lead to the situation where each object is its own exemplar. From Figure 5, one can fist observe a differentiated behavior of knnDPC w.r.t. the three other methods with a high number of clusters found, especially above $k = 104$. This can be explained by the densely connected 1NN structure created between objects as mentioned above, which leads to create a higher number of exemplars once the interpolation stages of the MR scheme is entered. Besides, the three other methods provide rather coherent number of clusters, and among them ModeSeek and GWENN-WM show an almost monotonic decrease of the latter, with two distinguishable plateaus around 15 clusters, and then 9 clusters for higher k. This stability of cluster number could be used to optimize the kNN graph.

Figure 6 shows the evolution of the computation time between methods as a function of NC. It can be seen that ModeSeek and knnDPC (with $k \leq 104$) perform nearly equally, and better than GWENN-WM and knnClust-WM, the latter requiring an extensive iterative labeling procedure at each stage of the MR scheme. The present experiments were done using Matlab on a 12-core HP Z800 computer.

Figure 7 displays the Average Correct Classification Rate (ACCR), the Overall Correct Classification Rate (OCCR), and the kappa index of agreement. Among the four methods, knnDPC performs the best around $k = 80$ in terms of ACCR, and a similar behavior is observed for ModeSeek, knnClust-WM and GWENN-WM for the same value of k. Overall, knnClust-WM seems to provide the best results in OCCR and kappa for k outside this range, but performances look quite unstable which would not be of practical use to predict an optimal value of k. Contrarily, ModeSeek and GWENN-WM provide more stable results with increasing k, though lower. knnDPC results are not reported here for $k > 104$ for the reason mentioned above. It is interesting to notice that contrarily to ACCR, optimum OCCR and kappa are achieved by the NN-DB methods for different values of k. Figure 8 reports another way to analyze the compared performances by plotting the same indices versus the number of found clusters. This allows to compare the results with the semi-supervised FCM method, for which the number of clusters is explicitly specified. Here, FCM is used in the same MR scheme as detailed above, the difference being that the cluster maps obtained at each resolution level are reported to the upper scale by upsampling, thus providing a deterministic initialization of the algorithm. Therefore, only the initialization of the membership matrix at the coarsest resolution is kept random. With this setting, FCM results still remain conditioned to this random initialization, and this is why 20 runs of this method were performed to compute average indices and standard deviations, for $16 \leq NC \leq 30$. These results indicate that FCM is not better than NN-DB approaches in terms of ACCR, whereas it is worse than the latter in OCCR and kappa indices as can be seen in Figure 8-(b) and (c). Indeed, differences beyond the standard deviation of FCM, especially for $20 \leq NC \leq 25$ are frequently observed. Among the NN-DB methods, the highest indices are obviously obtained with knnDPC in the same range of NC, with a rather well marked peak at $NC = 23$. The best indices for the three other NN-DB methods were obtained for lower cluster numbers, typically below $NC = 21$ for ACCR, and below $NC = 15$ for OCCR and kappa.

5.2 Clustering pixels in large hyperspectral images

In this section, we aim to demonstrate the feasibility of using the NN-DB methods detailed above to large scale HSIs, i.e. composed of several millions of pixels acquired in tenths of spectral bands. The objective is to demonstrate the operational capability to produce reliable classification maps in reasonable time, using only two parameters, i.e. k and S.

For this, we used a HSI which was acquired thanks to our AISA Eagle sensor in 2010 over the region of Murcia, Spain, for purposes of invasive cane detection and discrimination. The area surveyed is located in the city of Guardamar, and the HSI used is part of a single transect acquisition line (without correction). Its size is 8192 lines and 960 samples per line (approx. 8 Mpixels), and it is composed of 62 spectral bands covering the [400, 960] nm range at 10 nm spectral width per band. Each band is coded with two bytes per pixel, and

the HSI amounts a total size of 930 Mbytes. Figure 9 shows a color composition of the HSI as well as the clustering results. A variety of themes is clearly visible, such as vegetation, agricultural fields, water, buildings, roads, etc. Note that the NN-DB methods were implemented in C++ language for integration in our software platform[§] dedicated to decision helping for environmental purposes using hyperspectral imaging. In order to keep a sufficient amount of pixels to process at the coarsest scale of the MR scheme, we fixed $S = 5$, i.e. the clustering procedure using the full NN search is performed on 7680 pixels, which is close to the situation detailed in the above experiment. The analysis was performed with the four NN-DB methods described above, and only the size of the neighborhood k was varied. In the present experiment, the original spectral bands of the HSI were kept unchanged (no normalization).

The clustering results of the four NN-DB methods also displayed in Figure 9 were selected among the results obtained for k ranging from 10 to 40 by steps of 5, after a posterior analysis of the classification performances with respect to an available ground reference map. More precisely, Figure 10 displays a zoomed portion of 300×300 pixels (square area visible in Figure 9) of the whole image, and the corresponding clustering results. Over this sub-image, a ground truth map is available (Figure 10-(b)), showing two regions containing the cane species to discriminate, i.e. *Arundo donax* and *Phragmites australis*. In this experiment, the clustering results of the NN-DB methods were chosen so as to maximize the ACCR performance index with respect to k. The choice of ACCR is motivated by the high imbalance between the two classes to distinguish in the reference data (10771 pixels for *Phragmites australis* vs. 365 pixels for *Arundo donax*). Table 1 provides a summary of the results. It can be observed that optimal values of k differ among the NN-DB methods, ranging from $k = 20$ for GWENN-WM, to $k = 30$ for knnDPC and knnClust-WM. For these neighborhood sizes, Modeseek, knnClust-WM and GWENN-WM give rather similar numbers of clusters (between 16 and 21), whereas knnDPC provides a higher one with $NC = 62$. The best ACCR index is provided by GWENN-WM, closely followed by ModeSeek and knnClust-WM, whereas knnDPC gives the worst index. These results are in accordance with the corresponding clustering maps shown in Figure 10, in which the two vegetation species are more clearly distinguishable with GWENN-WM, ModeSeek and knnClust-WM than with knnDPC, especially for the *Arundo donax* class. Table 1 also reports the computation time spent for each method. These show that NN-DB methods can be used for pixel partitioning in very large HSIs within acceptable time. In comparison, FCM achieves the best clustering results for $NC = 15$ clusters with an ACCR of 56.54%, therefore above knnDPC, but still below GWENN-WM, ModeSeek and knnClust-WM. It is also important to notice that the computation times for the NN-DB methods is approximately 20 times lower than the one given by a single run of FCM using the same MR scheme, which makes these methods very attractive for the processing of very large HSIs.

Table 1. Comparison of NN-DB clustering methods for the *Guardamar* HSI, using a 5-level DWT. The results obtained for a single FCM run using the same MR scheme are also given.

Results	ModeSeek	knnDPC	knnClust-WM	GWENN-WM	*FCM*
optimal k	25	30	30	20	
NC	20	62	16	21	*15*
ACCR (%)	60.29	52.55	58.04	60.75	*56.54*
Computation time (min)	10.18	15.49	11.55	12.59	*254*

6. CONCLUSION

In this communication, we have addressed the use of nearest neighbor - density-based (NN-DB) clustering methods for hyperspectral image (HSI) pixel partitioning. NN-DB methods can partition data objects using a kNN graph, instead of specifying the number of clusters to be found like KMeans or FCM; they are also deterministic since they do not require any random initialization stage. In particular, we have investigated the similarities and differences between ModeSeek, knnDPC, knnClust-WM, and GWENN-WM, once aligned on the same baseline regarding pointwise density estimation. Due to the difficulty to obtain the full kNN graph over a large data set, the multiresolution scheme recently proposed for GWENN was extended to the three other methods, after a slight modification to ensure the stability of the number of classes. Experiments were

[§] http://tsi2m.enssat.fr/

conducted to compare the four NN-DB methods for pixel clustering in HSIs, using a multiresolution setting. We have first evaluated the clustering performances of the proposed methods in terms of classification accuracy owing to a benchmark HSI and its reference map, and experimentally observed their proper clustering stability with respect to the number of neighbors k. Though knnDPC provides the best clustering accuracy, ModeSeek and GWENN-WM exhibit a high clustering stability with respect to k, which could be useful to estimate its optimal value. We have also shown that NN-DB methods can outperform fuzzy c-means, in terms of classification accuracy, whatever the number of clusters. Finally, we have demonstrated the applicability and relevance of NN-DB methods when embedded in a multiresolution scheme for pixel clustering in very large HSIs, with several millions of pixels.

REFERENCES

[1] Chehdi, K. and Cariou, C., "True-false ground truths: what interest?," in [*Proc. SPIE Image and Signal Processing for Remote Sensing XXII*], Bruzzone, L. and Bovolo, F., eds., **10004** (Sept. 2016).

[2] Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: a review," *ACM Computing Surveys* **31**(3), 264–323 (1999).

[3] Jain, A. K., "Data clustering: 50 years beyond k-means.," *Pattern Recognition Letters* **31**(8), 651–666 (2010).

[4] Chehdi, K., Soltani, M., and Cariou, C., "Pixel classification of large size hyperspectral images by affinity propagation," *Journal of Applied Remote Sensing* **8** (August 2014).

[5] Duin, R. P. W., Fred, A. L. N., Loog, M., and Pekalska, E., "Mode seeking clustering by knn and mean shift evaluated.," in [*SSPR/SPR*], *Lecture Notes in Computer Science* **7626**, 51–59, Springer (2012).

[6] Tran, T. N., Wehrens, R., and Buydens, L. M. C., "Knn-kernel density-based clustering for high-dimensional multivariate data," *Computational Statistics & Data Analysis* **51**, 513–525 (Nov. 2006).

[7] Cariou, C. and Chehdi, K., "A new k-nearest neighbor density-based clustering method and its application to hyperspectral images," in [*IEEE Intern. Geoscience and Remote Sensing Symposium*], 6161–6164 (2016).

[8] Rodriguez, A. and Laio, A., "Clustering by fast search and find of density peaks," *Science* **344**(6191), 1492–1496 (2014).

[9] Du, M., Ding, S., and Jia, H., "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems* **99**, 135–145 (2016).

[10] Xie, J., Gao, H., Xie, W., Liu, X., and Grant, P. W., "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors.," *Information Sciences* **354**, 19–40 (2016).

[11] Goodenough, D. G., Chen, H., Richardson, A., Cloude, S., Hong, W., and Li, Y., "Mapping fire scars using radarsat-2 polarimetric SAR data," *Can. J. Remote Sensing* **37**(5), 500–509 (2011).

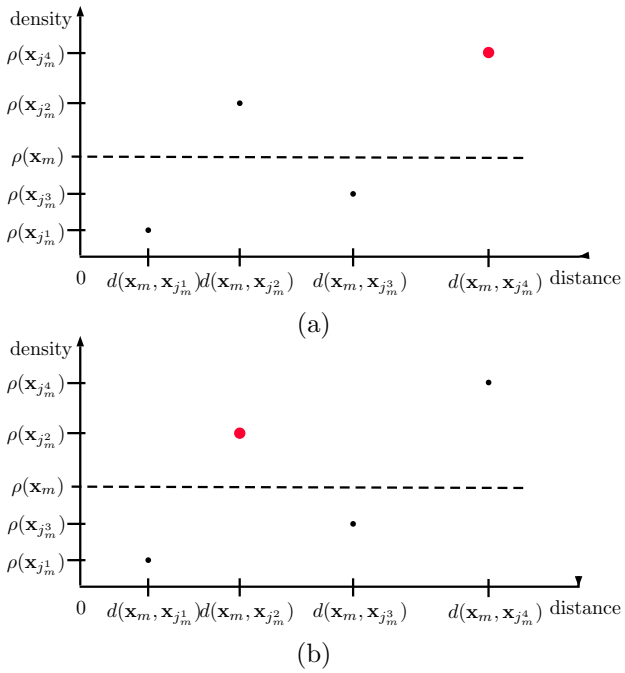


Figure 1. Illustration of the difference in nearest neighbor seeking between ModeSeek (a) and knnDPC (b). ModeSeek retains, among the current object $\mathbf{x}_m$ and its k nearest neighbors, the one with the highest density (here the fourth NN), whereas knnDPC retains the closest with higher density than the current object (here the second NN).

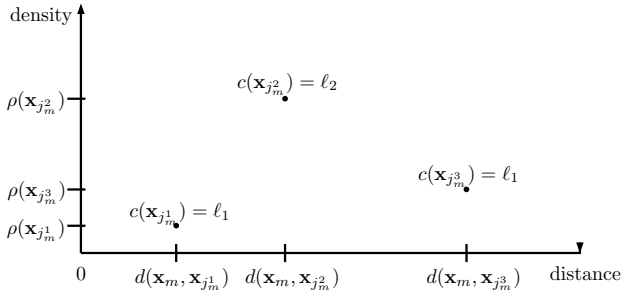


Figure 2. Illustration of the `wmode` function vs. the `mode` function. The `mode` function in GWENN would assign label ℓ_1 to the current object $\mathbf{x}_m$, since two of the three NNs have this label. The `wmode` function in GWENN-WM and knnClust-WM rather assigns label ℓ_2 because the sum of local densities of objects labeled as ℓ_1 does not exceed the density of the object labeled as ℓ_2.

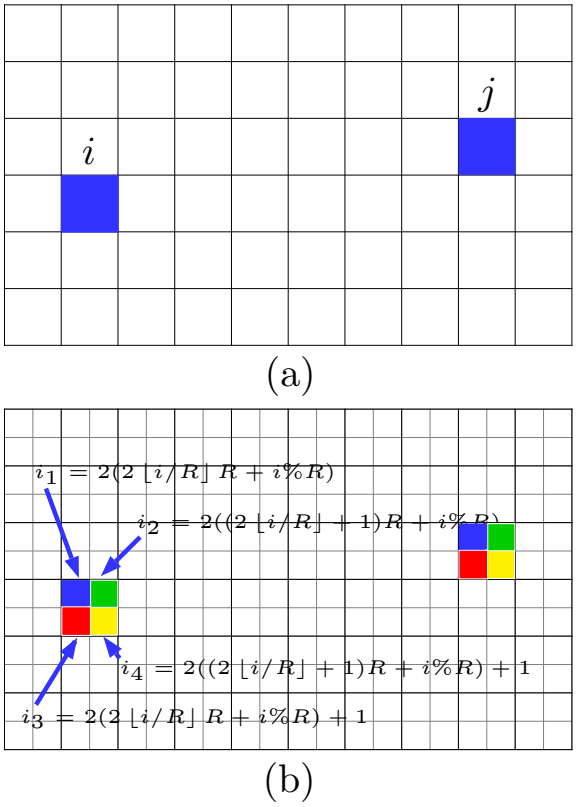


Figure 3. Interpolation of cluster exemplar pixels after IDWT. (a): the pixels i and j are the exemplars of two clusters after applying a given NN-DB method at the coarser scale; (b): the pixels i_1, i_2, i_3, i_4 are the candidate exemplars for the cluster indexed by i after IDWT, which are used in the kNN search before applying the NN-DB method at the upper scale. R is the number of rows in the first image.

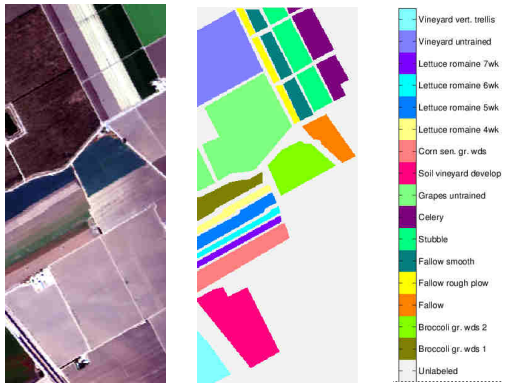


Figure 4. *Salinas* hyperspectral image. (a): Color composite (bands 30, 20, 10); (b): Reference map.

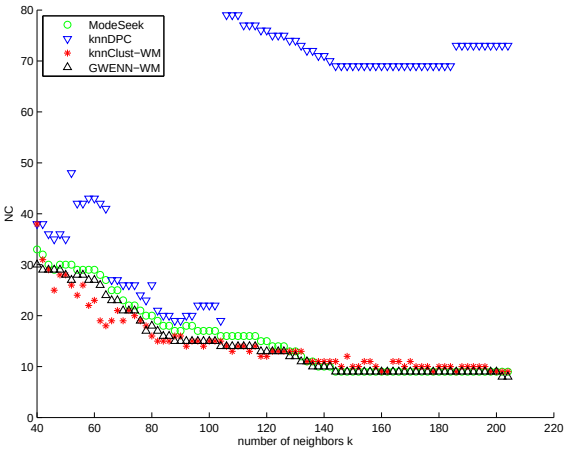


Figure 5. Evolution of the number of clusters NC versus the number of NNs k for *Salinas* hyperspectral image, in a 2-level MR scheme for all NN-DB methods.

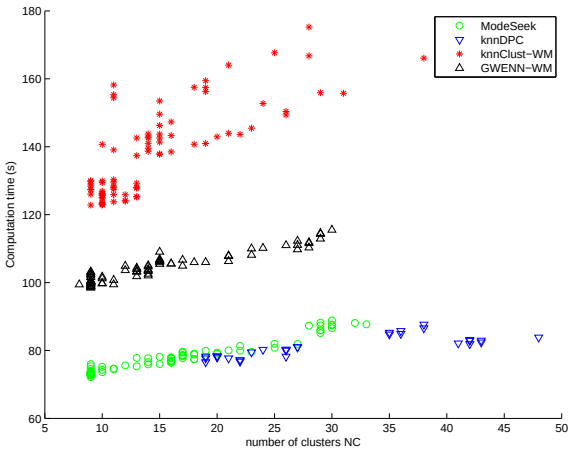
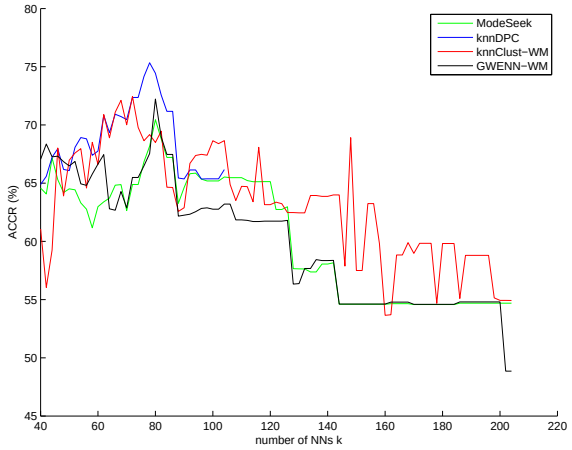
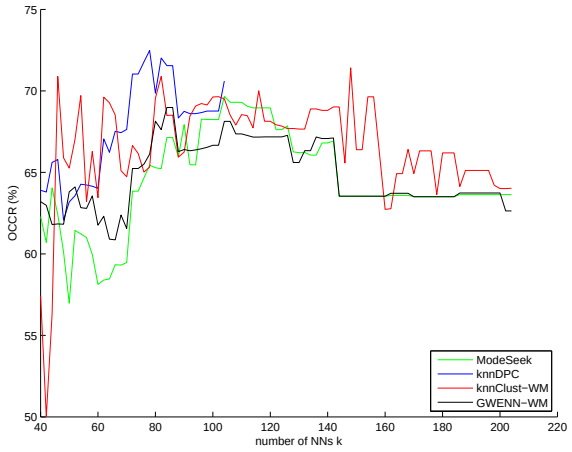


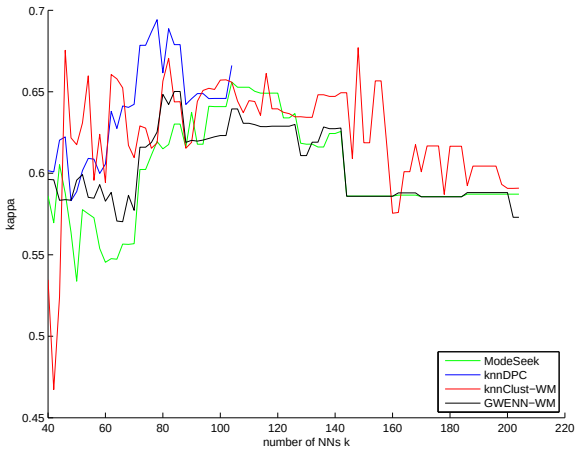
Figure 6. Computation time versus the number of clusters NC for *Salinas* hyperspectral image, in a 2-level MR scheme.



(a)

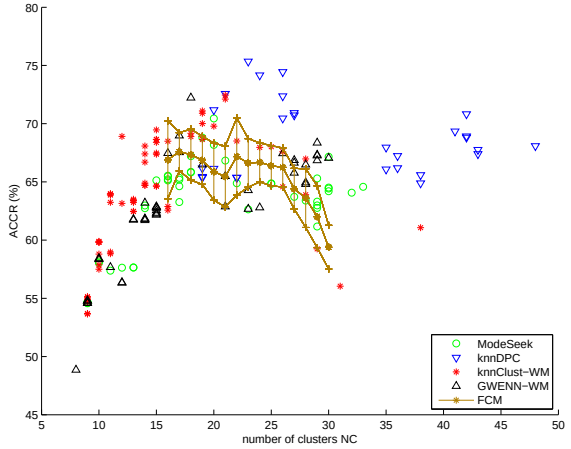


(b)

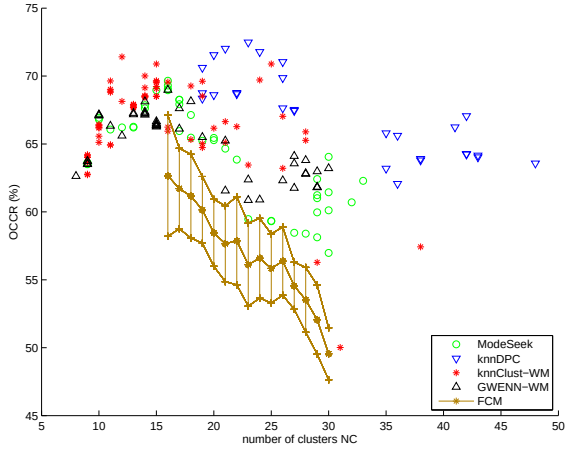


(c)

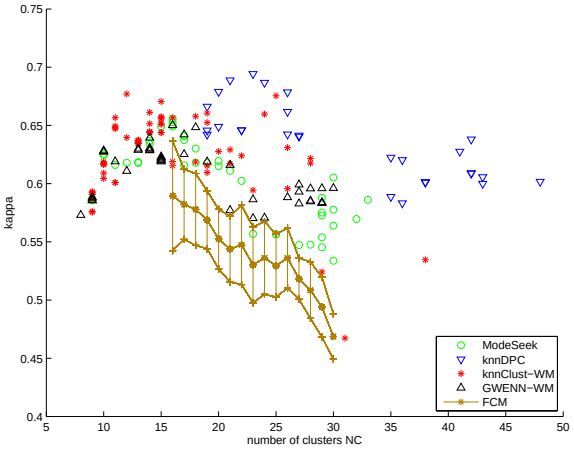
Figure 7. External performance indices for *Salinas* hyperspectral image, as function of the number of NNs k, in a 2-level MR scheme. (a): ACCR; (b): OCCR; (c): kappa index.



(a)



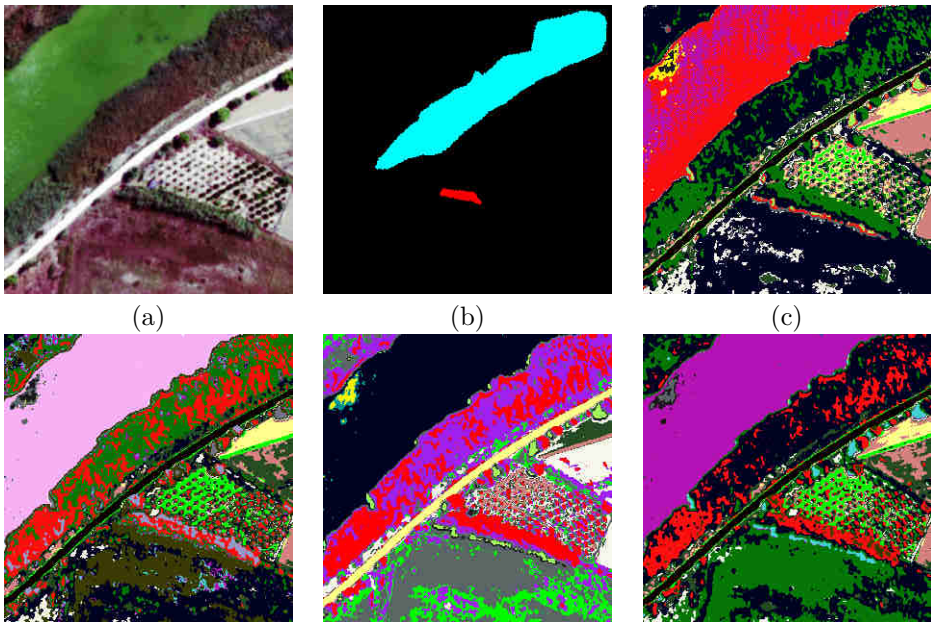
(b)



(c)

Figure 8. External performance indices for *Salinas* hyperspectral image, as function of the number of clusters NC, in a 2-level MR scheme. (a): ACCR; (b): OCCR; (c): kappa index. FCM results are shown as average index $\pm$ standard deviation.

Figure 9. Clustering results using NN-DB methods on a very large HSI (8192 × 960 pixels, 62 bands). From top to bottom: *Guardamar* hyperspectral image (Color composite, bands 30, 20, 10); ModeSeek; knnDPC ; knnClust-WM; GWENN-WM.



(a) (b) (c)

(d) (e) (f)

Figure 10. Detailed views of the clustering results. (a): *Guardamar* hyperspectral image (Color composite, bands 30, 20, 10); (b): Ground truth map (red: *Arundo donax*; cyan: *Phragmites australis*); (c) ModeSeek; (d) knnDPC; (e) knnClust-WM; (f) GWENN-WM.