An Approach for Reducing the Training Set of the KNN Applied to Remote Sensed Hyperspectral Images Classification

S. L. J. L. Tinoco, D. Menotti
Computing Department
UFOP - Federal University of Ouro Preto
Ouro Preto, MG, Brazil
Email: sjtinoco@yahoo.com.br,menottid@gmail.com

***Abstract*—This work presents a hybrid classification approach combining the use of supervised classification and unsupervised clustering algorithms. The main idea is to reduce the training set by selecting the most appropriated samples for classification using the K-nearest neighbor (KNN) algorithm. Indeed, a clustering algorithm is run in each class and the resulting centers of each cluster are selected to form the new training set of KNN. In this work, we use the K-Means and ISODATA clustering algorithms and present a comparative analysis using these methods. Experiments are carried out on two well-known databases: Indian Pines, acquired by AVIRIS sensor; and Pavia University, acquired by ROSIS sensor. Results show the efficiency of our proposed approach which significantly reduces the time required in the classification step while the effectiveness/accuracy is kept close to the ones of the original KNN.**

***Keywords*-KNN; KMeans; ISODATA; hyperspectral images;**

I. Introduction

The emergence of remote sensed hyperspectral images has brought some challenges to the task of data interpretation. Among them, we may mention the modeling of high dimensional data and their parameter estimation. In multispectral data (dozens of spectrals), Gaussian distribution model has been used for these purposes [1]. However, when dealing with hyperspectral imaging a large number of training samples for each class is required in order to estimate the terms of the large covariance matrices, for example. It should be noted also that a unimodal Gaussian description is not enough to handle multimodal data class [2]. In order to circumvent the above problems, the use of non-parametric algorithms such as the k-nearest neighbors (KNN) can be a good choice, since it has the advantage of not requiring estimated density function for each class [2]. Despite its simplicity, the KNN has been widely used [3], [4], [5], [6], because it has a high degree of accuracy, clear rules and, moreover, it is easy to implement.

The KNN is one of the most simple and intuitive algorithms to supervised classification. It is assumed that nearest samples are in the same class. This notion is used for the classification task and the KNN works as follows. For each unclassified pattern (*testing set*), one seek for the closest known class patterns (*training Set*) in the feature space, *i.e.*, the nearest neighbors. And one uses the class of these classified samples for selecting by majority the class of the unclassified pattern.

The KNN classifier is the one in which learning is based on analogy. The training set is composed by patterns represented by n-dimensional vectors. Each pattern of this group can be seen as a point in a n-dimensional space. In order to determine the class of a pattern which does not belong to the training set, the classifier KNN chooses the patterns of the training set that are closest to this unknown pattern, *i.e.*, having the greatest similarity, usually the smallest Euclidian distance. Computational cost can be high due to the number of comparisons to be made [7], since the similarity/distance from the unclassified pattern to the all training set has to be computed. That is, a large number of spectral distances should be evaluated for each pixel which requires a high computational load, especially when the number of spectral bands and/or the number of training samples is large. This is why the KNN has been primarily limited to the classification of multispectral data. For hyperspectral data, the KNN has been used only after features reduction is achieved [2].

Contributions: Thus, an approach which may take advantage of KNN decreasing its computational cost can be useful and appropriate to classify remote sensed hyperspectral images. With this in mind, the connection between an unsupervised clustering algorithm as Kmeans or ISODATA and the non-parametric KNN classifier is proposed in this work[1]. In order to decrease the computational load, we suggest to reduce the training set size by selecting the most appropriated samples. For each class, these samples are chosen as the resulting center clusters from Kmeans and ISODATA.

The remainder of this paper is organized as follows. Section II formally describes the Kmeans, ISODATA, and KNN algorithms. Section III presents our approach for remote sensed hyperspectral images classification. Sections IV and V describes the experiments performed using two well-know database (Indian Pines, acquired by AVIRIS sensor [8]; and Pavia University, acquired by ROSIS sensor [9].) in order to validate the proposed approach and their respective analyses. Finally, conclusions are pointed out in Section VI.

[1] This work relates to a master's thesis.

II. Theoretical background

A. Kmeans

The Kmeans is a partition clustering algorithm that is characterized by dividing the dataset into disjoint subsets. According to [10], the Kmeans does not require spatial information and has the great advantage of the computational time. It is based on distance, since its similarity function is based in the distance, which seeks to minimize. The most popular clustering algorithm is the Kmeans [11] using Euclidean distance. The idea of the algorithm is to provide a classification according to the data itself, based on analysis and comparisons of their numerical values. Thus, the algorithm provides an automatic classification without the need for human supervision. Because of this feature, the Kmeans algorithm is considered as a data classification unsupervised. According to a pre-defined rule, this method uses values from the data itself as temporary estimates of the average of clusters Km, where Km is the number of clusters specified by the user. Thus, the KMeans form sets having the most similar to each other, consequently separating the least similar.

The initial groups are formed around centers provided. Then, for each cluster formed is calculated an average value among the values that form the cluster. This average will be a new center, around which would form a new cluster with the elements closest to this center, considering all elements of the data set. This process continues iteratively until the elements stop changing clusters or until a maximum number of iterations predetermined is reached.

B. ISODATA

The ISODATA algorithm [12] is an enhancement of KMeans. The clusters resulting from KMeans depend strongly on the value of k, the number of clusters chosen *a priori*, that does not always represent the real amount (natural) of possible groupings. The ISODATA allows the deletion, fusion and division of clusters, so that the resulting number of clusters can be different from the k, informed by the user. Thus, the resulting number of cluster are closer to the natural amount of clusters from the data set. Like the KMeans, ISODATA based on minimization of a cost measure, in this case, the internal distance between the patterns of one group. Considering:

K: Number of clusters initial;

I: Maximum number of iterations;

θ_M: Threshold for minimum number of samples in each cluster can contain, for discarding clusters;

θ_S: Threshold for standard deviation, for split operation;

θ_C: Threshold for pairwise distances, for merge operation.

The ISODATA algorithm, extracted from [13], as follow:

Step 1: Arbitrarily choose k (not necessarily equal to K) initial cluster centers: $m_1, m_2, ..., m_k$ from the data set $x_i; i = 1, 2, ..., N$.

Step 2: Assign each of the N samples to the closest cluster center:

$x \in w_j$, if $D_L(x, m_j) = max\{D_L(x, m_i), i = 1, 2, ..., k\}$.

Step 3: Discard clusters containing too few members, *i.e.*, if $K_j < \theta_M$, then discard w_j and reassign its members to other clusters. $k = k - 1$.

Step 4: Update each cluster center:

$m_j = \frac{1}{K_j} \sum_{x \in w_j} x; (j = 1, 2, ..., k)$.

Step 5: Compute the average distance D_j of samples in cluster w_j from their corresponding cluster center:

$D_j = \frac{1}{K_j} \sum_{x \in w_j} D_L(x, m_j); (j = 1, 2, ..., k)$.

Step 6: Compute the overall average distance of the samples from their respective cluster centers:

$D = \frac{1}{\sum_{j=1}^{k} K_j} \sum_{j=1}^{k} K_j D_j = \sum_{j=1}^{k} P_j D_j, P_j = \frac{K_j}{\sum_{j=1}^{k}}$.

Step 7: If $k <= K/2$ (too few clusters), go to Step 8 (split); else if $k > 2K$ (too many clusters), go to Step 11 (merge); else go to Step 13 (continue).

(Steps 8 through 10 are for split operation, Steps 11 through 13 are for merge operation.)

Step 8: Find the standard deviation vector:

$\sigma^{(j)} = [\sigma_1^{(j)}, ..., \sigma_N^{(j)}]^T$

for each cluster:

$\sigma_n^{(j)} = \sqrt{\frac{1}{K_j} \sum_{x \in w_j} (x_n - m_n^{(j)})^2}, (n = 1, ..., N; j = 1, ..., k)$

where $m_n^{(j)}$ and $\sigma_n^{(j)}$ are the nth component of the mean vector m_j e o desvio padro das amostras w_j. K_j is the number of samples in w_j.

Step 9: Find the maximum component of each σ_j and denote it by $\sigma_{max}^{(j)}$. Do this for all $j = 1, ..., k$.

Step 10: If for any $\sigma_{max}^{(j)}, (j = 1, ..., k)$, all of the following are true:

$\sigma_{max}^{(j)} > \theta_S$

$D_j > D$

$K_j > 2\theta_M$

then split m_j into two new cluster centers m_j^+ e m_j^-, by adding ±δ to the component of m_j corresponding to $\sigma_{max}^{(j)}$ where δ can be $\alpha\sigma_{max}^{(j)}$, for some $\alpha > 0$. Then delete m_j and let $k = k + 1$. Goto Step 2, else Go to Step 13.

Step 11: Compute the pairwise distances D_{ij} between every two cluster centers:

$D_{ij} = D_L(m_i, m_j), (1 \leq i, j \leq k, i > j)$

sort them in ascending order:

$D_{i1j1} \leq D_{i2j2} \leq ... \leq D_{ikjk}$

Step 12: Find the P smallest distances all satisfying $D_{ij} < \theta_C$, and perform pairwise merge of the corresponding clusters. Specifically, for $l = 1, ..., P$, if neither of m_{ij} nor m_{ji} has been used in this iteration, merge them to form a new center:

$m = \frac{1}{K_{ij}+K_{ji}} [K_{ij} m_{ij} + K_{ji} m_{ji}]$

Delete $m_i j$ and $m_j i$, and let $k = k - 1$

Go to Step 2.

Step 13: Terminate if maximum number of iterations I is reached. Otherwise go to Step 2.

C. KNN

The K-Nearest Neighbor (KNN) [7] is one of the most simple and intuitive to supervised classification, it is assumed that within the close any instances of attributes are the same class. For each pattern with unknown class, look for the patterns of known class (Training Set) closest in feature space, the nearest neighbors, and uses the class closer to these standards for classification, choosing the class corresponding to majority. The KNN classifier is one in which learning is based on analogy. The training set consists of n-dimensional vector and each element of this set represents a point in n-dimensional space. To determine the class of an element that does not belong to the training set, the KNN classifier seeks to K elements of the training set that are closest to this unknown element, that is, having the shortest distance.

Let x be an unknown pixel vector, w_i be a label for k_i neighbors, so that
$\sum_{i=1}^{M} k_i = k$
Let a general KNN rule be

$$x \in \omega_i, \text{ if } m_i > m_j \text{ for all } j \neq i \qquad (1)$$

where $m_i(x)$ is the membership that pixel vector x belongs to class i. For the basic KNN

$$m_i(x) = k_i \qquad (2)$$

In this work, Euclidean distance is used as the spectral distance measure.

III. The proposed approach

The proposed approach aims to obtain a reduced training set such that the KNN classification algorithm run faster than in its original way. Moreover, we expect that the instances chosen for each class, which are cluster centers of Kmeans, could keep the classification effectiveness similar to the one when all training set is used. A flowchart of our proposed approach is shown in Fig. 1.

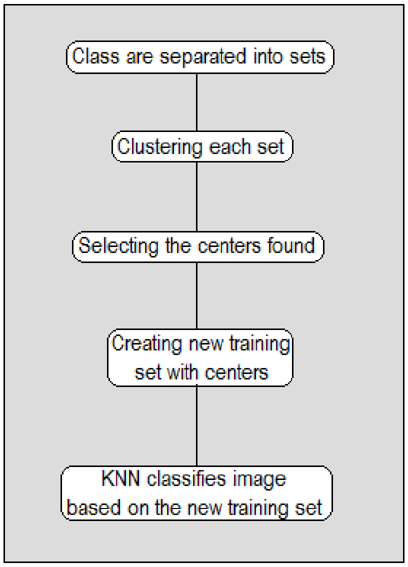

Class are separated into sets
Clustering each set
Selecting the centers found
Creating new training set with centers
KNN classifies image based on the new training set

Fig. 1. Scheme of the proposed approach

The proposed strategy is the implementation of the clustering algorithm on each set of instances of the same class in the original image. Then from each cluster obtained, their centers will be selected and form the new training set for the KNN. This new training set is then formed by data representing best determined class, thus diminishing the effect of intra-class, and of noise in the set clustered. In the proposed strategy, as was done with KMeans, the ISODATA was used in each class and the resulting cluster centers were selected as the new training set of KNN.

IV. Experiments

In order to determine the reliability of the constructed model with the data available, the N-fold cross validation scheme is employed, in which the dataset is divided into N subsets. Among these subsets, one is retained to be used as testing and the remaining $N-1$ subsets are used for training. The validation procedure is repeated N times until each subset is used exactly once as testing data, as illustrated in Fig. 3. In this way, the N average effectiveness of the classifier in testing is obtained.

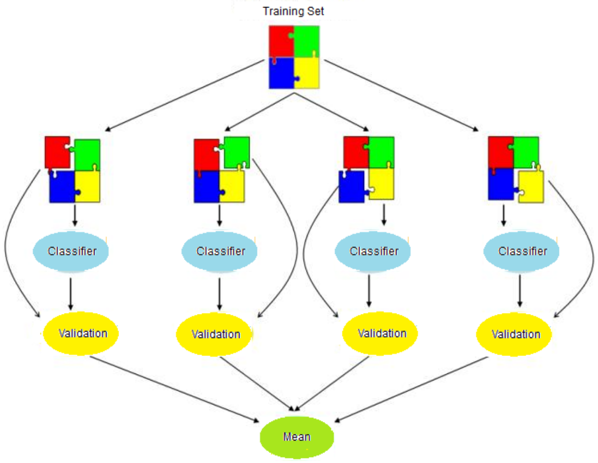


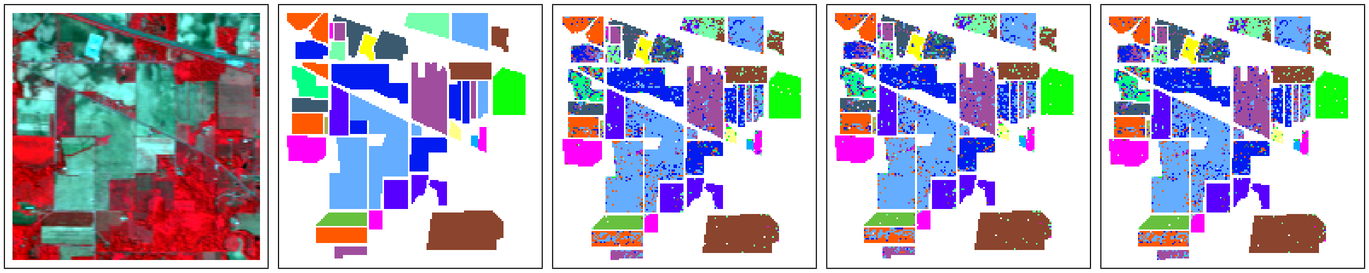
Fig. 3. N-folds cross validation scheme

The dataset division is performed as follows. The labeled pixels are divided into sets, in which each set represents a class. Then each class set is equally divided into five subsets ($N = 5$). The resulting subsets are grouped so that each contain $1/5$ of the labeled pixels of each class.

In order to verify the effectiveness and efficiency of our approach, tests are performed with two training sets.

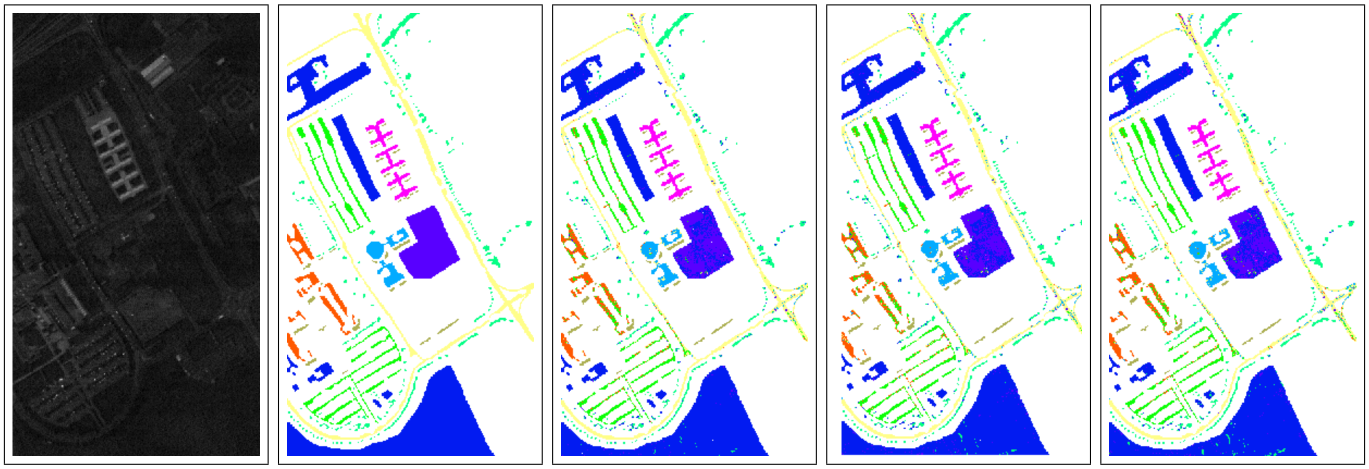
First experiment: Experiments are performed using the *Indian Pines* datasets, acquired by AVIRIS airborne sensor data [8], which cover an area of agriculture and forest in northeastern Indiana, USA, $145 \times 145 \times 220$ pixels. Noise bands are removed, that is, the indexed from 104 to 108, from 150 to 163 and 220, remaining a total of 200 bands. A representation of this hyperspectral image can be seen in Fig. 2a, in which there are sixteen classes or categories as highlighted in Fig. 2b.

This image is classified by KNN using the full training set and using the training set procuced by the proposed approach, and the respective obtained thematic maps are shown in Fig.'s 2c, 2d and 2e. The obtained results are shown in Table I. As we can observe, the accuracy of KNN is higher,



(a) Original (b) Ground Truth (c) Thematic Map for 1-NN (d) Thematic Map for 1-NN and 60 clusters, KMeans (e) Thematic Map for 1-NN and 100 clusters, ISODATA

Fig. 2. *Indian Pines* dataset, 200 bands, AVIRIS sensor



(a) Original (b) Ground Truth (c) Thematic Map for 3-NN (d) Thematic Map for 3-NN and 60 clusters, KMeans (e) Thematic Map for 3-NN and 140 clusters, ISODATA

Fig. 4. *Pavia University*, 103 bands, ROSIS sensor

but the time required for classification is greater than that of the proposed approach. In these experiment, we used all 200 image bands, which may influenced the low accuracy. Note that the time required for clustering is taken into account.

Second experiment: An image of the *University of Pavia*, Italy, acquired by the sensor ROSIS, 610 × 340 × 103 pixels is used [9]. This image presentes nine classes as highlighted in Fig. 4b, where Fig.'s 4a shows a representation of that region. The experiments are performed using all 103 bands of this image. The results obtained for this image by the KNN and proposed approach can be seen in Table II. This image is classified by KNN using the full training set and using the training set procuced by the proposed approach, and the respective obtained thematic maps are shown in Fig.'s 4c, 4d and 4e. The proposed approach obtained an accuracy slightly lower than the KNN using the full training data, however its running time is quite smaller.

V. Results and Discussion

It is important to note that the classes in both images/datasets is quite unbalanced, *i.e.*, some few classes contain the majority of pixels while others have many few, as can be seen in column *Samples* in Tables III and IV. Therefore, when applying the clustering algorithm and selecting the cluster centers found, it may happen that the classes with more elements are not well represented, since the number of centers is equal for all classes. This procedure can reduce the accuracy of classification. In order to find a better representation for the new training set extracted from the cluster centers, we adopted the following strategy. Firstly, we account the number of elements Q from each class and calculated the median M between the values found. M is divided by the number of clusters K, previously reported, resulting in R. Then the new number of clusters NK_n to be used in each class is now the value for the quantity Q of the respective cluster divided by R, as follow:

$M = median(Q_1, Q_2, ..., Q_n)$

$R = M/K$

$NK_n = Q_n/R$

where C = number of classes, $n = 1, 2, ..., C$ and K = number of clusters.

A. Performances

Analyzing the results shown in Table I (*Indian Pines*-AVIRIS), we can observe that the accuracies reached by the proposed approach is very close to the ones reached by the

TABLE I

EXPERIMENTS ON ACCURACY AND RUN TIME (IN SECONDS) FOR *Indian Pines* - TESTING FOR A ZERO MEAN, 95% CONFIDENCE.

KNN	**KMeans**	**ISODATA**	**Accuracy**		**Time**	**Confidence Interval**			
			mean	std		accuracy	relate accuracy	relate run time	
1	-	-	83.52%	5.26	2min 58s	83.41%	82.45% ⊢⊣ 84.38%		
3	-	-	83.32%	5.22	2min 57s	83.47%	82.68% ⊢⊣ 84.26%		
5	-	-	82.88%	5.22	2min 58s	83.04%	82.19% ⊢⊣ 83.88%		
1	5	-	72.19%	5.26	19s	71.60%	70.58% ⊢⊣ 72.62%	-30.93 ⊢⊣ 53.59	116.73 ⊢⊣ 201.26
1	20	-	80.62%	5.27	1min 11s	80.54%	80.17% ⊢⊣ 80.90%	-39.40 ⊢⊣ 45.20	64.69 ⊢⊣ 149.30
1	60	-	83.23%	5.27	3min 35s	83.35%	82.25% ⊢⊣ 84.44%	-42.01 ⊢⊣ 42.59	-79.30 ⊢⊣ 5.30
3	5	-	69.32%	5.17	19s	69.11%	68.65% ⊢⊣ 69.56%	-27.74 ⊢⊣ 55.74	116.25 ⊢⊣ 199.74
3	20	-	79.41%	5.21	1min 11s	79.32%	78.59% ⊢⊣ 80.04%	-37.99 ⊢⊣ 45.81	64.09 ⊢⊣ 147.90
3	60	-	82.96%	5.21	3min 34s	82.97%	81.87% ⊢⊣ 84.07%	-41.54 ⊢⊣ 42.26	-78.90 ⊢⊣ 4.90
5	5	-	67.29%	5.19	20s	68.12%	67.68% ⊢⊣ 68.56%	-26.23 ⊢⊣ 57.41	116.17 ⊢⊣ 199.82
5	20	-	78.55%	5.22	1min 17s	78.53%	77.92% ⊢⊣ 79.15%	-37.61 ⊢⊣ 46.27	59.05 ⊢⊣ 142.94
5	60	-	82.58%	5.21	3min 43s	82.69%	81.74% ⊢⊣ 83.63%	-41.60 ⊢⊣ 42.20	-86.90 ⊢⊣ -3.09
1	-	60	71.87%	4.96	2min 23s	72.03%	70.76% ⊢⊣ 73.31%	-29.43 ⊢⊣ 52.73	-6.08 ⊢⊣ 76.08
1	-	100	83.54%	5.26	2min 57s	83.58%	82.38% ⊢⊣ 84.77%	-42.28 ⊢⊣ 42.24	-41.26 ⊢⊣ 43.26
1	-	140	83.45%	5.26	2min 54s	83.37%	82.04% ⊢⊣ 84.69%	-42.19 ⊢⊣ 42.33	-38.71 ⊢⊣ 45.81
3	-	60	68.77%	4.82	2min 22s	68.53%	67.92% ⊢⊣ 69.13%	-25.82 ⊢⊣ 54.92	-5.37 ⊢⊣ 75.37
3	-	100	83.35%	5.22	2min 57s	82.96%	81.25% ⊢⊣ 84.67%	-41.97 ⊢⊣ 41.91	-41.94 ⊢⊣ 41.94
3	-	140	83.41%	5.22	2min 50s	83.42%	82.58% ⊢⊣ 84.27%	-42.03 ⊢⊣ 41.85	-34.94 ⊢⊣ 48.94
5	-	60	66.95%	4.77	2min 21s	67.81%	67.26% ⊢⊣ 68.36%	-24.25 ⊢⊣ 56.11	-3.18 ⊢⊣ 77.18
5	-	100	82.99%	5.22	2min 56s	82.79%	82.13% ⊢⊣ 83.45%	-42.05 ⊢⊣ 41.83	-39.94 ⊢⊣ 43.94
5	-	140	83.19%	5.22	3min 03s	83.50%	82.28% ⊢⊣ 84.72%	-42.25 ⊢⊣ 41.63	-46.94 ⊢⊣ 36.94

TABLE II

EXPERIMENTS ON ACCURACY AND RUN TIME (IN SECONDS) FOR *Pavia University* - TESTING FOR A ZERO MEAN, 95% CONFIDENCE.

KNN	**KMeans**	**ISODATA**	**Accuracy**		**Time**	**Confidence Interval**			
			mean	std		accuracy	relate accuracy	relate run time	
1	-	-	89.96%	1.74	27min 47s	89.86%	89.64% ⊢⊣ 90.08%		
3	-	-	90.83%	1.72	27min 16s	90.71%	90.49% ⊢⊣ 90.93%		
5	-	-	90.82%	1.72	26min 58s	90.93%	90.35% ⊢⊣ 91.51%		
1	5	-	81.21%	1.72	2min 17s	81.39%	80.74% ⊢⊣ 82.04%	5.93 ⊢⊣ 11.56	1527 ⊢⊣ 1532
1	20	-	86.64%	1.73	2min 48s	86.48%	85.98% ⊢⊣ 86.98%	0.49 ⊢⊣ 6.14	1496 ⊢⊣ 1501
1	60	-	88.18%	1.76	8min 31s	88.20%	87.89% ⊢⊣ 88.51%	-1.06 ⊢⊣ 4.62	1153 ⊢⊣ 1158
3	5	-	80.05%	1.73	2min 17s	79.57%	78.93% ⊢⊣ 80.21%	7.97 ⊢⊣ 13.58	1496 ⊢⊣ 1501
3	20	-	86.26%	1.71	2min 40s	86.47%	86.23% ⊢⊣ 86.71%	1.78 ⊢⊣ 7.35	1473 ⊢⊣ 1478
3	60	-	88.46%	1.74	8min 19s	88.47%	88.31% ⊢⊣ 88.64%	-0.44 ⊢⊣ 5.18	1134 ⊢⊣ 1139
5	5	-	78.94%	1.71	2min 19s	79.07%	78.01% ⊢⊣ 80.13%	9.09 ⊢⊣ 14.66	1476 ⊢⊣ 1481
5	20	-	85.33%	1.70	2min 42s	85.27%	85.00% ⊢⊣ 85.55%	2.70 ⊢⊣ 8.27	1453 ⊢⊣ 1458
5	60	-	87.86%	1.73	8min 27s	87.73%	87.21% ⊢⊣ 88.25%	0.15 ⊢⊣ 5.76	1108 ⊢⊣ 1113
1	-	60	87.75%	1.76	6min 28s	87.88%	87.43% ⊢⊣ 88.33%	-0.63 ⊢⊣ 5.05	1276 ⊢⊣ 1281
1	-	100	88.10%	1.73	7min 45s	88.09%	87.76% ⊢⊣ 88.43%	-0.96 ⊢⊣ 4.68	1199 ⊢⊣ 1204
1	-	140	88.36%	1.77	11min 30s	88.22%	88.17% ⊢⊣ 88.27%	3.42 ⊢⊣ 9.59	973 ⊢⊣ 980
3	-	60	87.83%	1.74	6min 16s	87.72%	87.49% ⊢⊣ 87.96%	0.18 ⊢⊣ 5.81	1257 ⊢⊣ 1262
3	-	100	88.12%	1.70	7min 30s	88.27%	87.91% ⊢⊣ 88.63%	-0.07 ⊢⊣ 5.49	1183 ⊢⊣ 1188
3	-	140	88.79%	1.75	11min 06s	88.86%	88.65% ⊢⊣ 89.07%	-0.78 ⊢⊣ 4.86	967 ⊢⊣ 972
5	-	60	87.17%	1.74	6min 16s	87.04%	86.78% ⊢⊣ 87.29%	0.83 ⊢⊣ 6.46	1239 ⊢⊣ 1244
5	-	100	87.51%	1.68	7min 41s	87.68%	87.35% ⊢⊣ 88.01%	0.54 ⊢⊣ 6.07	1154 ⊢⊣ 1159
5	-	140	88.31%	1.75	11min 22s	88.31%	88.01% ⊢⊣ 88.62%	4.80 ⊢⊣ 10.45	933 ⊢⊣ 933

TABLE III

NUMBER OF PIXELS IN EACH SAMPLE AND WITH BALANCING CLASSES FOR ISODATA, *Indian Pines*.

Class Number	**Class**	**Samples**	**Balancing = 60**		**Balancing = 100**		**Balancing = 140**	
			Expected	**Observed**	**Expected**	**Observed**	**Expected**	**Observed**
1	Alfafa	54	7	10	11	10	15	10
2	Corn-notill	1434	175	286	291	286	407	286
3	Corn-mintill	834	102	166	169	166	237	166
4	Corn	234	28	46	47	46	66	46
5	Grass-pasture	497	61	99	101	99	141	99
6	Grass-trees	747	91	149	152	149	212	149
7	Grass-pasture-mowed	26	3	5	5	5	7	5
8	Hay-windrowed	489	59	97	99	97	139	97
9	Oats	20	2	4	4	4	6	4
10	Soybean-notill	968	118	193	196	193	275	193
11	Soybean-mintill	2468	302	493	501	493	701	493
12	Soybean-clean	614	75	122	125	122	174	122
13	Whea	212	26	42	43	42	60	42
14	Woods	1294	158	258	262	258	367	258
15	Buildings-Grass-Trees-Drives	380	47	76	77	76	108	76
16	Stone-Steel-Towers	95	12	19	19	19	27	19

TABLE IV

NUMBER OF PIXELS IN EACH SAMPLE AND WITH BALANCING CLASSES FOR ISODATA, *Pavia University*.

Class Number	**Class**	**Samples**	**Balancing = 60**	**Final = 60**	**Balancing = 100**	**Final = 100**	**Balancing = 140**	**Final = 140**
1	Asphalt	6631	130	100	216	119	303	118
2	Meadows	18649	365	288	609	365	852	323
3	Gravel	2099	41	30	69	39	96	35
4	Trees	3064	60	45	100	60	140	50
5	Painted metal sheets	1345	26	21	44	26	61	20
6	Bare Soil	5029	98	80	164	93	230	93
7	Bitumen	1330	26	20	43	23	61	21
8	Self-Blocking Bricks	3682	72	55	120	77	168	59
9	Shadows	947	19	14	31	18	43	16

KNN using the full training dataset, for most k values of the KMeans and ISODATA Algorithms. However, for k = 20, the proposed approach obtained run times much lower than the KNN when using the full training dataset. In Table II (*Pavia University*-ROSIS) we can observe that the accuracies reached by the proposed approach is very close to the ones reached by the KNN and, thus, obtained run times much lower than the KNN when using the full training dataset.

B. Quality

Statistical tests, Confidence Interval and Testing for a Zero Mean, both with 95% Confidence, are done. Table I shows the results for the degree of accuracy achieved with the use of KNN and the use of the proposed approach on the AVIRIS image. It was found that improvements proposed approach failed. Table I also shows the results for the runtime. For the KNN K equal to 1 and 3 with KMeans k equal to 5 and 20, the intervals obtained do not include zero. This shows that the times were reduced significantly.

Table II shows the results of tests performed on the image Rosis. For the degree of accuracy, the Testing for a Zero Mean shows that the proposed approach achieved slightly worse results than KNN. In Table II are the results for the runtime, the proposed approach achieved better results than KNN. One can therefore say that in all experiments, for the image Rosis, the runtimes of the proposed approach were significantly better than those of KNN.

In the tables I and II, the balancing (expected) field are the values for the initial number of centers and in the final(observed) field are the values found by ISODATA.

VI. Conclusion

In this paper, we presented a hybrid approach for remote sensed hyperspectral images classification, linking clustering (Kmeans and ISODATA) and a supervised non-parametric classification (KNN) algorithms. From the experiments using two well-know databases (Indian Pines, acquired by AVIRIS sensor [8]; and Pavia University, acquired by ROSIS sensor [9]), we can observe that the obtained accuracy by the proposed approach is close to the ones obtained by the KNN when using the full training data. Regarding the runtime, the proposed approach achieved promising results being up to ten times faster than KNN.

As future work, we plan to study other clustering algorithms such as: DBSCAN [14], DenClust [15], Xmeans [16], Optimum-path forest [17], [18], [19], etc. We also plan to study algorithms developed for sub-spaces clustering on high dimensional [20] such that the KNN can process the full training data. In this way, we expect to decrease even more the KNN run time keeping the obtained accuracy close to the original values.

References

[1] J. P. Hoffbeck and D. A. Landgrebe, “Classification of remote sensing images having high spectral resolution,” *Remote Sensing of Environment*, vol. 57, no. 3, pp. 119–126, 2009.

[2] X. Jia and J. A. Richards, “Fast k-nn classification using the cluster-space approach,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 225–228, 2005.

[3] E. Blanzieri and F. Melganin, “Nearest neighbor classification of remote sensing images with the maximal margin principle,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1804–1811, 2008.

[4] D. O. Mcinerney and M. Nieuwenhuis, “A comparative analysis of knn and decision tree methods for the irish national forest inventory,” *International Journal of Remote Sensing*, vol. 30, no. 19, pp. 4937–4955, 2009.

[5] Y. Li and B. Cheng, “An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification,” *In IEEE International Conference on GeoInformatics, 2009*, pp. 1–4, 2009.

[6] L. Samaniego, A. Brdossy, and K. Schulz, “Supervised classification of remotely sensed imagery using a modified k-nn technique,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2112–2125, 2008.

[7] P. Cover, T. e Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory, 1967*, vol. 13, no. 1, pp. 21–27, 1967.

[8] MultiSpec, “https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html,” last visit on June, 25th, 2012.

[9] U. del Pais Vasco, “http://www.ehu.es/ccwintco/index.php/Hyperspectral-Remote-Sensing-Scenes,” last visit on August, 25th, 2012.

[10] T. N. Tran, R. Wehrens, and L. M. C. Buydens, “Clustering multispectral images: a tutorial,” *Chemometrics and Intelligent Laboratory Systems, 2005*, vol. 77, no. 1, pp. 3–17, 2005.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: A review,1999,” *Computational Statistics and Data Analysis, 1999*, vol. 31, no. 3, pp. 264–323, 1999.

[12] D. J. Hall and G. B. Ball, “Isodata : A novel method of data analysis and pattern classification,” *Journal of Machine Learning Research*, 1965.

[13] R. Wang, “http://fourier.eng.hmc.edu/e161/lectures/classification/node11.html,” last visit on June, 25th, 2012.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Knowledge Discovery and Data Mining*, 1998, pp. 226–231.

[15] A. Hinneburg and D. A. Keim, “An efficient approach to clustering in large multimedia databased with noise,” in *Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

[16] D. Pelleg and A. W. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *International Conference on Machine Learning (ICML)*, 2000, pp. 727–734.

[17] J. P. Papa, A. X. Falco, and C. T. N. Suzuki, “Supervised pattern classification based on optimum-path forest,” *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[18] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falco, “Data clustering as an optimum-path forest problem with applications in image analysis,” *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[19] J. P. Papa, A. X. Falco, V. H. C. de Albuquerque, and J. M. R. Tavares, “Efficient supervised optimum-path forest classification for large datasets,” *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[20] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, 2007.