International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

Pixel Based Approach of Unsupervised Satellite Image Classification

M.Suganthi[1], S.Chidambaram[2]

P.G Scholar, M.E.Communication System, Department of Electronics and Communication Engineering, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[1]

Assistant professor, Department Of Electronics and Communication Engineering, Adhiyamaan College of Engineering Hosur, Tamilnadu, India[2]

Abstract: In this project pixel-based approach for urban land covers classification from high resolution satellite image using K means clustering and ISO data clustering. Pixel based image analysis of image segmentation that is, clustering of pixels into homogenous objects, and subsequent classification or labeling of the pixels, and modeling based on the characteristics of pixels is done using MATLAB GUI model. When applied to a satellite image, the clustering algorithm approach involves two steps. First, each group or cluster is homogeneous; i.e. examples that belong to the same group are similar to each other. Second, each group or cluster should be different from other cluster, i.e. examples that belong to one cluster should be different from the examples of other clusters. The algorithm was implemented in MATLAB GUI model and was tested on remotely sensed images of different sensors, resolutions and complexity levels.

Keywords: pixel based approach, high resolution satellite image.

I. INTRODUCTION

Clustering algorithms for remote sensing images are used to being divided into two categories: pixel-based [1] and object-based [2] approaches. Using pixel based clustering algorithms for high-resolution remote sensing images, one could often find the "pepper and salt" effect in the results because of the lack of spatial information among pixels. In contrast, object-based clustering algorithms are not based on spectral features of individual pixels but on image objects, i.e., segments. Consequently, in terms of semantics, the quality of image objects is heavily dependent on segmentation algorithms. In this letter, a novel clustering algorithm is proposed to detect geo-objects from high-spatial-resolution remote sensing images using both neighborhood spatial information and probabilistic latent semantic analysis model (NSPLSA). The proposed algorithm is not based on either pixels or segments but on densely overlapped sub images, i.e., rectangular regions, with prefixed size. The probabilistic latent semantic analysis model (PLSA), which is also called aspect model, is employed to model all sub images. Every pixel in each sub image has been allocated a topic label. The cluster label of every pixel in the large satellite image is derived from the topic labels of multiple sub images which cover the pixel.

Unsupervised clustering is a fundamental tool in image processing for geosciences and remote sensing applications. For example, unsupervised clustering is often used to obtain vegetation maps of an area of interest. This approach is useful when reliable training data are either scarce or expensive, and when relatively little a priori information about the data is available. The problem of clustering points in multidimensional space can be posed formally as one of a number of well-known optimization problems, such as the Euclidean k-median problem, in which the objective is to minimize the sum of distances to the nearest center, the Euclidean k-center problem, in which the objective is to minimize the maximum distance, and the k-means problem, in which the objective is to minimize the sum of squared distances. Efficient solutions are known to exist only in special cases such as the planar 2-center problem. There are no efficient exact solutions known to any of these problems for general k, and some formulations are known to be NP-hard. Efficient approximation algorithms have been developed in some cases. These include constant factor approximations for the k-center problem, the k-median problem and the k-means problem. There are also approximation algorithms for the k-median and k-means problems, including improvements based on coresets. Work on the k-center algorithm for moving data points, as well as a linear time implementation of a 2-factor approximation of the k-center problem have also been introduced. In spite of progress on theoretical bounds, approximation algorithms for these clustering problems are still not suitable for practical implementation in multidimensional spaces, when k is not a small constant. This is due to very fast growing dependencies in the asymptotic running times on the dimension and/or on k. In practice, it is common to use heuristic approaches, which seek to and a reasonably good clustering, but do not provide guarantees on the quality of the results. This includes randomized approaches, such as clara and clarans, and methods based on neural networks. One of the most popular and widely used clustering heuristics in remote sensing is isodata. A set of n data points in d-dimensional space is given along with an integer k indicating the initial number of clusters and a number of additional parameters. The general goal is to compute a set of cluster centers in d-space. Although there is no specify optimization criterion, the algorithm is similar in spirit to the well-known k-means clustering method, in which the objective is to minimize the average squared distance of each point to its nearest center, called the average distortion. One significant advantage of isodata over k-means is that the user need only provide an initial estimate of the number of clusters, and based on various heuristics the algorithm may alter the number of clusters by either deleting small clusters, merging nearby clusters, or splitting large clusters. The algorithm will be described in the next section. As currently implemented, isodata can run very slowly, particularly on large data sets. Given its wide use in remote sensing, its efficient computation is an important goal. Our objective in this paper is not to provide a new or better clustering algorithm, but rather, to show how computational geometry methods. Can be applied to produce a faster implementation of isodata clustering. There are a number of minor variations of isodata that appear in the literature .These variations involve issues such as termination conditions, but they are equivalent in terms of their overall structure. We focus on a widely used version, called isoclus, which will be presented in the next section. The running times of isodata and isoclustering are dominated by the time needed to compute the nearest among the k cluster centers to each of the n points. This can be reduced to the problem of answering n nearest-neighbor queries over a set of size k, which naively would involve O(kn) time. To improve the running time, an obvious alternative would be to store the k centers in a spatial index such as a kd-tree. However, this is not the best approach, because k is typically much smaller than n, and the center points are constantly changing, requiring the tree to be constantly updated. Kanungo proposed a more efficient and practical approach by storing the points, rather than the cluster centers, in a kd-tree. The tree is then used to solve the reverse nearest neighbor problem, that is, for each center. We compute the set of points for which this center is the closest. This method is called the iltering algorithm. We show how to modify this approach for isoclustering. The modifications are not trivial. First, in order to perform the sort of aggregate processing that the iltering algorithm employs, it was necessary to modify the way in which the isoclustering algorithm computes the degree of dispersion within each cluster.

In order to further improve execution times, we have also introduced an approximate version of the ltering algorithm. A user-supplied approximation error bound i > 0 is provided to the algorithm, and each point is associated with a center whose distance from the point is not farther than (1 + i) times the distance to its true nearest neighbor. This result may be of independent interest because it can be applied to k-means clustering as well. The running time of the iltering algorithm is a subtle function of the structure of the clusters and centers, and so rather than presenting a worst-case asymptotic analysis, we present an empirical analysis of its efficiency based on both synthetically generated data sets, and actual data sets from a common application in remote sensing and geostatistics. As the experiments show, depending on the various input parameters (that is, dimension, data size, number of centers, etc.), the algorithm presented runs faster than a straightforward implementation of isoclustering by factors ranging from 1.3 to over 50. In particular, the improvements are very good for typical applications in geostatistics, where the data size n and the number of centers k are large, and the dimension d is relatively small. Thus, we feel that this algorithm can play an important role in the analysis of geostatistical data analysis and other applications of data clustering. The remainder of the paper is organized as follows. We describe in the next section a variant of isodata, called isoclustering, whose modification is the focus of this paper. In Section 3 we provide background, concerning basic tools such as the kd-tree data structure and the iltering algorithm that will be needed in our efficient implementation of isoclustering.

II. THE ISOCLUS ALGORITHM

We begin by presenting the particular variant of isodata, called isoclustering, whose modification will be presented later. Although our description is not exhaustive, it contains enough information to understand our various modifications. The algorithm tries to and the best cluster centers through an iterative approach. It also uses a number of different heuristics to determine whether to merge or split clusters. At a high level, the following tasks are performed in each iteration of the algorithm: Points are assigned to their closest cluster centers, cluster centers are updated to be the centroids of their associated points, clusters with very few points are deleted, large clusters satisfying some conditions are split, and small clusters satisfying other conditions are merged. The algorithm continues until the number of iterations exceeds a user-supplied value. Let us present the algorithm in more detail. There are a number of user-supplied parameters. These include the following. (In parentheses we give the variable name of the parameter used in Ref. [13].

kinit: initial number of clusters (numclus)

nmin: minimum number of points that can form a cluster (samprm)

Imax: maximum number of iterations (maxiter)

Imax: maximum standard deviation of points from their cluster center along each axis (stdv)

Lmin: minimum required distance between two cluster centers (lump)

Pmax: maximum number of cluster pairs that can be merged per iteration(maxpair)

Here is an overview of the algorithm. (See Ref. [11] for details.) Let S = fx1; : : : ; xng denote the set of points to be clustered. Each point xj = (xj1; : : : ; xjd) is treated as a vector in real d-dimensional space, Rd. Let n denote the number of points. If the original set is too large, all of the iterations of the algorithm, except the last, can be performed on a random subset of S of an appropriate size. Throughout, let kxk denote the Euclidean length of the vector x.

(1) Letting k = kinit, randomly sample k cluster initial centers Z = fz1; z2; : : : ; zkg from S.
(2) Assign each point to its closest cluster center.

That is, for any x 2 S,

$$\mathbf{x} \in S_j \quad \text{if} \quad \|\mathbf{x} - \mathbf{z}_j\| < \|\mathbf{x} - \mathbf{z}_i\|, \quad \forall i \neq j.$$

(Ties for the closest center are broken arbitrarily.) Let nj denote the numberof points of Sj .

(3) Remove cluster centers with fewer than nmin points. (The associated points of S are not deleted, but are ignored for the remainder of the iteration.) Adjust the value of k and relabel the remaining clusters S1 : : : ; Sk accordingly.

(4) Move each cluster center to the centroid of the associated set of points. That is,

$$\mathbf{z}_j \leftarrow \frac{1}{n_j} \sum_{\mathbf{x} \in S_j} \mathbf{x}, \qquad \text{for } 1 \leq j \leq k.$$

If any clusters were deleted in Step 3, then the algorithm goes back to Step 2.

(5) Let nj be the average distance of points of Sj to the associated cluster center zj , and let be the overall average of these distances.

$$\Delta_j \leftarrow \frac{1}{n_j} \sum_{x \in S_i} \|x - z_j\|, \quad \text{for } 1 \leq j \leq k. \qquad \Delta \leftarrow \frac{1}{n} \sum_{j=1}^{k} n_j \Delta_j.$$

(6) If this is the last iteration, then set Lmin = 0 and go to Step 9. Also, if 2k > kinit and it is either an even numbered iteration or k _ 2kinit, then go to Step 9.

(7) For each cluster Sj , compute a vector vj = (v1; : : : ; vd) whose ith coordinate is the standard deviation of the ith coordinates of the vectors directed from zj to every point of Sj . That is,

$$v_{ji} \leftarrow \left(\frac{1}{n_j} \sum_{\mathbf{x} \in S_j} (x_i - z_{ji})^2 \right)^{1/2} \quad \text{for } 1 \leq j \leq k \text{ and } 1 \leq i \leq d.$$

Let vj;max denote the largest coordinate of vj .

(8) For each cluster Sj , if vj;max > _max and either

$$((\Delta_j > \Delta) \text{ and } (n_j > 2(n_{\min} + 1))) \text{ or } k \leq \frac{k_{\text{init}}}{2},$$

then increment k and split Sj into two clusters by replacing its center with two cluster centers centered around zj and separated by an amount and direction that depends on vj;max.37 If any clusters are split in this step, then go to Step 2.

(9) Compute the pairwise intercluster distances between all distinct pairs of cluster Centers.

$$d_{ij} \leftarrow \|\mathbf{z}_i - \mathbf{z}_j\|, \qquad \text{for } 1 \leq i < j \leq k.$$

(10) Sort the intercluster distances of Step 9 in increasing order, and select a subset of at most Pmax of the closest such pairs of clusters, such that each pair has an intercluster distance of at most Lmin. For each such pair (i; j), if neither Si nor Sj has been involved in a merger in this iteration, replace the two clusters Si and Sj with a merged cluster Si [Sj , whose associated cluster center is their weighted average

$$\mathbf{z}_{ij} \leftarrow \frac{1}{n_i + n_j}(n_i \mathbf{z}_i + n_j \mathbf{z}_j).$$

Relabel the remaining clusters and decrease k accordingly.
(11) If the number of iterations is less than Imax, then go to Step 2.If the algorithm is implemented in the most straightforward manner, and if it is assumed that the number of clusters, k, is much smaller than the total number of points, n, then the most time-consuming stage of the algorithm is Step 2. Computing naively the distances from each of the n points of S to each of the k centers for a total of O(kn).

III. K – MEANS ALGORITHM

This algorithm has as an input, a predefined number of clusters (i.e., the K from its name. Means stands for an average, an average location of all the members of a particular cluster). When dealing with clustering techniques, one has to adopt a notion of a higher dimensional space, or space in which orthogonal dimensions are all attributes. The value of each attribute of an example represents a distance of the example from the origin along the attribute axes. Of course, in order to use this geometry efficiently, the values in the data set must all be numeric (categorical data must be transformed into numeric ones) and should be normalized in order to allow fair computation of the overall distances in a multi-attribute space.

The K-means algorithm is a simple, iterative procedure, in which a crucial concept is the one of " *centroid* ". Centroid is an artificial point in the space of records which represents an average location of the particular cluster. The coordinates of this point are averages of attribute values of all examples that belong to the cluster. The steps of the k-means algorithm are given below.

- Select randomly *k* points (it can also be examples) to be the seeds for the centroids of *k* clusters.
- Assign each example to the centroid closest to the example, forming in this way *k* exclusive clusters of examples.
- Calculate new centroids of the clusters. For that purpose, average all attribute values of the examples belonging to the same cluster (centroid).
- Check if the cluster centroids have changed their "coordinates". If yes, start again from the step (ii). If not, cluster detection is finished and all examples have their cluster memberships defined.

Usually this iterative procedure of redefining centroids and reassigning the examples to clusters needs only a few iterations to converge. For a discussion on cluster detection see.

To summarise, clustering techniques are used when there are natural grouping in a data set. Clusters should then represent groups of items that have a lot in common. Creating clusters prior to application of some other data mining technique might reduce the complexity of the problem by dividing the space of data set. This space

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

partitions can be mined separately and such two steps procedure might give improved results as compared to data mining without using clustering.

IV. DATA SELECTION

The image chosen in this project is a high resolution satellite image. The image contains three layers (RGB). It has covered a large area.

V. RESULT AND DISCUSSION

The various experiment carried out in MATLAB v7.5.0 (R2007b).

TABLE I EXPERIMENTAL RESULTS OF ENTROPY AND PURIFY

ITERATION	ALGORITHM	ENTROPY	PURITY
Iteration 1	LDA-MRF	12.5244	0.00460685
	Iso data	10.8645	1.00461
	Fuzzy logic	8.05146	2.0046
Iteration 2	LDA-MRF	16.4335	0.00332835
	Iso data	10.7756	1.00334
	Fuzzy logic	10.307	2.00332
Iteration 3	LDA-MRF	12.9367	0.00542203
	Iso data	8.98036	1.00542
	Fuzzy logic	7.78543	2.00544
Iteration 4	LDA-MRF	13.1847	0.00636605
	Iso data	10.7172	1.00631
	Fuzzy logic	8.06243	2.00637
Iteration 5	LDA-MRF	10.4323	0.00513297
	Iso data	8.43231	1.00513
	Fuzzy logic	8.1213	2.00513
Iteration 6	LDA-MRF	15.5887	0.00519272
	Iso data	13.567	1.00519
	Fuzzy logic	10.7366	2.0052
Iteration 7	LDA-MRF	11.6921	0.0013227
	Iso data	11.3893	1.00133
	Fuzzy logic	9.14164	2.00132
Iteration 8	LDA-MRF	10.0793	0.00411609
	Iso data	12.646	1.00411
	Fuzzy logic	8.23281	2.00409
Iteration 9	LDA-MRF	14.4179	0.00196894
	Iso data	12.2419	1.00197
	Fuzzy logic	11.4227	2.00197
Iteration 10	LDA-MRF	13.506	0.00795486
	Iso data	11.3858	1.00799
	Fuzzy logic	9.21555	2.00799
Average	LDA-MRF	13.07956	0.004541156
	Iso data	11.099997	1.00454
	Fuzzy logic	9.107692	2.004543

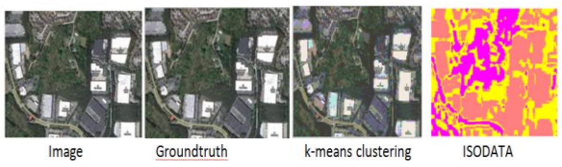

Fig .1. Classification results of pixel based image. The first image shows the input images, second image shows the k-means clustering of iteration 1 and the third image shows the k-means clustering of iteration 2, and finally classification results of ISODATA clustering is displayed.

k-means clustering, iterative self-organizing data analysis technique (ISODATA) clustering and pixel-Oriented Classification (Fuzzy Based) are examined .On the one hand, the qualitative evaluation is examined in terms of the semantics of the segmentation results. On the other hand, the results are also quantitatively evaluated in terms of the purity and entropy. By comparing the above methods better classification results are examined.

REFERENCES

[1] KanXu,Wen Yang, Gang Liu, and Hong Sun, (2013) “Unsupervised Satellite Image Classification Using Markov Field Topic Model”.

[2] Delong. A, Osokin.A, Isack H.N, and Boykov.Y,(2010) “Fast approximate energy minimization with label costs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 2173–2180.

[3] Liénou.M, Maître.H, and Datcu.M, (2010), “Semantic annotation of satellite images using latent Dirichlet allocation,” IEEE Geosci. Remote Sens. Lett, vol. 7, no. 1, pp. 28–32.

[4] Larlus.D and Jurie.F, Apr.(2009) “Latent mixture vocabularies for object categorization and segmentation,” Image Vis. Comput., vol. 27, no. 5, pp. 523–534

[5] MacKay.D.J.C,(2003)”Information Theory, Inference, and Learning Algorithms”, vol. 8. Cambridge, U.K.: Cambridge Univ.,p. 12.

[6] Memarsadeghi.N,Mount.D,Netanyahu.N.S, Le Moigne.J, and de Berg.M,(2007) “A fast implementation of the ISODATA clustering algorithm,” Int. J. Comput. Geom. Appl., vol. 17, no. 1, pp. 71–103.

[7] Rosenberg and J. Hirschberg, “V-measure: A conditional entropy based external cluster evaluation measure,” in Proc. Joint Conf. EMNLPCoNLL, 2007, pp. 410–420.

[8] Tang. W. Yi, H, and Chen.Y, “An object-oriented semantic clustering algorithm for high-resolution remote sensing images using the aspect model “May (2011) IEEE Geosci. Remote Sens. Lett., vol. 8, no. 3, pp. 522–526.

[9] Verbeek.J and Triggs.B, “Region classification with Markov field aspect models,” (2007) in Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 1–8.

[10] Yang.W, Dai.D, Wu.J, and He.C, “Weakly supervised polarimetric SAR image classification with multi-modal Markov aspect model,” (2010) in Proc. ISPRS, TC VII Symp. (Part B), 100 Years ISPRS—Advancing Remote Sensing Science, Vienna, Austria, Jul. 5–7, pp. 669–673.

[11] Zhu.S.C and Yuille.A.L, Sep(1996) “Region competition: Unifying, snakes, region growing, and Bayes/MDL for multiband image, segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 18, no. 9, pp. 884–900.

[12] Civanlar,R.,Trussel,H.,(1986)Constructing membership functions using statistical data. IEEE Fuzzy Sets and Systems 18, 1 –14.. Curlander, J., Kober, W., 1992.

[13] Suganthi.M,Chidambaram.S”object based analysis of high resolution satellite image classification using fuzzy logic” IJECT vol.5 Issue 1,jan to march (2014)