Spectral Spatial Classification Of Hyper Spectral Image Using Mean Shift Clustering

1 D.Geethamani, 2 Dr.S.Thavamani
1 Department Of Computer Science, Dr. N.G.P College Of Arts And Science(Autonomous)
2 Department Of Computer Applications, Snr Sons College (Autonomous)
Tamilnadu, India.

Abstract:-We propose a novel approach for solving the perceptual grouping problem in vision. Rather than focusing on local features and their consistencies in the image data, our approach aims at extracting the global impression of an image. In existing method image segmentation using the normalized cut. The normalized cut criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. We show that an efficient computational technique based on a generalized Eigen value problem can be used to optimize this criterion. In our proposed method mean shift algorithm is used for segmenting the region. It requires low computational complexity and is therefore very feasible for real-time image segmentation processing. It preprocesses an image by using the MS algorithm to form segmented regions that preserve the desirable discontinuity characteristics of the image.

Keywords: Hyper spectral, segmentation, classification, mean shift clustering

INTRODUCTION:
Segmentation is an image processing technique which is nearly an unavoidable preprocessing step. There are many techniques used for segmentation in the field of IP, which can be adapted to remote sensing applications. Adapting the algorithm to remote sensing is same as using the techniques for real world datasets. For a remote sensing system the desirable output of segmentation are well defined regions or features of object which can be distinguished from another. These features are affected by the homogeneity of regions. Homogeneity can be recognized as same color or same texture for a region. The remote sensing images of a natural region such as forest have regions of greenery more. This helps in the segmentation of those regions which may be a good part of the image. If it is an urban image the number of regions will be more with lots of small homogenous regions.

In real world remote sensing imagery, the homogeneity may not be evident as expected. There will often be small gradient and textural variation. The images also may contain large number of regions, which may not be known prior to execution. These interfere in the segmentation steps and further processing using those regions. Some of the earlier version of segmentation included edge based, contour based, model based, template based and also region based segmentation. It has stated the usage of edge based segmentation as having the problem of giving large number of edges, due to trees in natural images and blocks in urban areas. The edges were needed then to group and form meaningful geometry. The algorithms are not viable, due to the reason that edge grouping to segment a region with specific geometry is of non-polynomial complexity. Using region based segmentation techniques such as K-Means segmentation techniques has the inherent disadvantage of knowing the number of regions prior to segmentation, which is not known in real world cases, Contour based techniques such as Active contour models [3], sometimes used in remote sensing also has the problem of knowing the approximate location of the region and also the noises will hinder its performance. A template based approach [4] is not scale invariant and using multiple scales is not suited in real time detection. Color based segmentation is also not suited for the problem due to the fact that the object may have gradient variation and color ranges. It has also given an overview of the use of segmentation technique in remote sensing. He has stated some of the method with example and also given some of the applications in which segmentation can be used. This paper is organized such that the next section describes about the challenges in segmentation, followed by the solution to the challenges which is identified as the technique known as mean shift segmentation. A case study is also presented as proof.

Exploiting manifold geometry in hyperspectral imagery

A new algorithm for exploiting the nonlinear structure of hyperspectral imagery is developed and compared against the de facto standard of linear mixing. This new approach seeks a manifold coordinate system that preserves geodesic distances in the high-dimensional hyperspectral data space. Algorithms for deriving manifold coordinates, such as isometric mapping (ISOMAP), have been developed for other applications. ISOMAP guarantees a globally optimal solution, but is computationally practical only for small datasets because of computational and memory requirements. Here, we develop a hybrid technique to circumvent ISOMAP's computational cost. We divide the scene into a set of smaller tiles. The manifolds derived from the individual tiles are then aligned and stitched together to recompose the scene. Several alignment methods are...
discussed. This hybrid approach exploits the fact that ISOMAP guarantees a globally optimal solution for each tile and the presumed similarity of the manifold structures derived from different tiles. Using land-cover classification of hyperspectral imagery in the Virginia Coast Reserve as a test case, we show that the new manifold representation provides better separation of spectrally similar classes than one of the standard linear mixing models. Additionally, we demonstrate that this technique provides a natural data compression scheme, which dramatically reduces the number of components needed to model hyperspectral data when compared with traditional methods such as the minimum noise fraction transform.

Nonlinear dimensionality reduction by locally linear embedding

Many areas of science depend on exploratory data analysis and visualization. The need to analyze large amounts of multivariate data raises the fundamental problem of dimensionality reduction: how to discover compact representations of high-dimensional data. Here, we introduce locally linear embedding (LLE), an unsupervised learning algorithm that computes low-dimensional, neighbor-hood-preserving embeddings of high-dimensional inputs. Unlike clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by images of faces or documents of text.

A global geometric framework for nonlinear dimensionality reduction

Scientists working with large volumes of high-dimensional data, such as global climate patterns, stellar spectra, or human gene distributions, regularly con-front the problem of dimensionality reduction: finding meaningful low-dimensional structures hidden in their high-dimensional observations. The human brain confronts the same problem in everyday perception, extracting from its high-dimensional sensory inputs—30,000 auditory nerve fibers or 106optic nerve fibers—a manageably small number of perceptually relevant features. Here we describe an approach to solving dimensionality reduction problems that uses easily measured local metric information to learn the underlying global geometry of a data set. Unlike classical techniques such as principal component analysis (PCA) and multidimensional scaling (MDS), our approach is capable of discovering the nonlinear degrees of freedom that underlie complex natural observations, such as human handwriting or images of a face under different viewing conditions. In contrast to previous algorithms for nonlinear dimensionality reduction, ours efficiently computes a globally optimal solution, and, for an important class of data manifolds, is guaranteed to converge asymptotically to the true structure.

Semi-Supervised Graph-Based Hyperspectral Image Classification

This paper presents a semi-supervised graph-based method for the classification of hyperspectral images. The method is designed to handle the special characteristics of hyperspectral images, namely, high-input dimension of pixels, low number of labeled samples, and spatial variability of the spectral signature. To alleviate these problems, the method incorporates three ingredients, respectively. First, being a kernel-based method, it combats the curse of dimensionality efficiently. Second, following a semi-supervised approach, it exploits the wealth of unlabeled samples in the image, and naturally gives relative importance to the labeled ones through a graph-based methodology. Finally, it incorporates contextual information through a full family of composite kernels. Noting that the graph method relies on inverting a huge kernel matrix formed by both labeled and unlabeled samples, we originally introduce the Nystro umlm method in the formulation to speed up the classification process. The presented semi-supervised-graph-based method is compared to state-of-the-art support vector machines in the classification of hyperspectral data. The proposed method produces better classification maps, which capture the intrinsic structure collectively revealed by labeled and unlabeled points. Good and stable accuracy is produced in ill-posed classification problems (high dimensional spaces and low number of labeled samples). In addition, the introduction of the composite-kernel framework drastically improves results, and the new fast formulation ranks almost linearly in the computational cost, rather than cubic as in the original method, thus allowing the use of this method in remote-sensing applications.

Hyperspectral Imager for the Coastal Ocean: Instrument description and first images

The Hyperspectral Imager for the Coastal Ocean (HICO) is the first spaceborne hyperspectral sensor designed specifically for the coastal ocean and estuarial, riverine, or other shallow-water areas. The HICO generates hyperspectral images, primarily over the 400–900 nm spectral range, with a ground sample distance of ≈90 m (at nadir) and a high signal-to-noise ratio. The HICO is now operating on the International Space Station (ISS). Its cross-track and along-track fields of view are 42 km (at nadir) and 192 km, respectively, for a total scene area of 8000 km². The HICO is an innovative prototype sensor that builds on extensive experience with airborne sensors and makes extensive use of commercial off-the-shelf components to build a space sensor at a small fraction of the usual cost and time. Here we describe the

Geethamani, Thavamani…. (IJ0SER) September–2016
instrument’s design and characterization and present early images from the ISS.

Existing method:

Ncut
The set of points in an arbitrary feature space can be described as a weighted undirected graph \( G = (V, E) \), where the nodes of the graph are the points in the feature space, and an edge is formed between every pair of nodes. The weight on each edge, \( W(i, j) \), is a function of the similarity between nodes \( i \) and \( j \). A graph \( G = (V, E) \) can be partitioned into two disjoint sets, \( A, B, A \cup B = V \), \( A \cap B = \emptyset \), we expected that the intra similarity is high and the intergroup similarity is low. In graph theoretic language, a mathematical formulation of a cut is

\[
W(u, v)
\]

Cut type:

1. Minimum cut
2. Ratio cut
3. Normalized cut

Formula:

\[
Ncut(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A)} + \frac{\text{cut}(A, B)}{\text{assoc}(B)}
\]

To create a partition, one aims to choose a partitioning \( A, B \) that minimizes. After some algebraic manipulation, this can be reformulated as

\[
\min_v \frac{\nu f_L v}{\nu f_D v}
\]

Spatial spectral graph:

In this section, we present our main algorithm, which uses the Laplacian eigenmap / normalized cuts algorithm to segment and/or perform dimensionality reduction on a hyperspectral image. In particular, we present a graph coupling and weighting structure that naturally encapsulates both the spatial and spectral information present in HSI data. Once the graph has been defined, we solve the generalized eigenvalue problem for the smallest nonzero eigenvalues and eigenvectors. Once the eigenvalue problem has been defined, the analysis can proceed in one of two ways: in the first case, the first \( k \) eigenvectors can be used to define a (nonlinear) dimensionality reduction, exactly as in Laplacian eigenmaps. In particular, if we let \( X \) by the \( n \)-by-\( k \) matrix whose columns are the eigenvectors, then the rows of \( X \) define our projected data (here, \( n \) is the total number of pixels in the image, and \( k \) is the reduced dimension).

Alternatively, one can use the single smallest eigenvector to partition the scene; ideally, the eigenvector will be bimodal, with some subset of the entries clustered close to a single (positive) value, with the remaining entries clustered around a second (negative) value. By choosing an appropriate threshold, the eigenvector entries are partitioned into two groups; the corresponding pixels form the clusters in image space.

Once an initial split has been made, we simply recurse on each of the partitions. Note that the original graph contains all the information we need, and only needs to be calculated once. Subsequent iterations simply require us to select the rows and columns from the original adjacency matrix to create a new (smaller) reduced matrix. From there, the new degree and Laplacian matrices are easily formed, and the next Eigen decomposition calculated.

Some care must be taken as the recursion progresses, since the graph may cease to be connected (when clusters are no longer spatially contiguous), and/or no ‘good’ partition is possible. The former case can be handled by simply checking to see if the graph is connected before running the eigen analysis; if not, the graph is split into its individual connected components, and the recursion is continued on each component separately.

The latter case is not as easy to handle, especially in automated fashion. In essence, the problem reduces to deciding if given distributions (in this case, the values of the eigenvector) are bimodal or not. Currently, we use a statistical test known as guide the user; the final determination of whether or not to further partition the given cluster is made by the user. is to histogram the eigenvector and calculate the ratio between the maximal and minimal bin counts.

To define the spatial-spectral graph, we follow the normalized cuts algorithm: image pixels define the nodes, and each node is coupled to all other pixels that are spatially contiguous – that is, in some \( r \)-by-\( r \) window centered at the given pixel. The weights are given by heat kernel using a spectral distance measure, scaled by a ‘spatial dampening’ that is also a heat kernel (in pixel coordinates).

Problem identification:

- It’s hard to judge performance of segmentation.
- Texton-based and Luminance-based have different advantages and disadvantage.

Proposed method

Block diagram:

Mean shift segmentation:

Mean shift algorithm is a kind of density estimation algorithm based on nonparametric kernel function and the kernel function
is increasing probability density along the direction of density gradient, the final convergence to the local probability density near the maximum points. While applying mean shift algorithm on the image segmentation assignment, we denote the kernel function to be. Mean (Gx) is

\[ M_h(x) = m_h(x) - x \]

\[ m_h(x) = \frac{\sum_{i=1}^{n} G \left( \frac{x_i - x}{h} \right) w(x_i) x_i}{\sum_{i=1}^{n} G \left( \frac{x_i - x}{h} \right) w(x_i)} \]

We choose \( x \) pixels as seed points as the allowable error, mean shift algorithm according to the following steps to realize image pixel clustering:

Algorithm steps:

Step1: Start from the initial node \( x \) to calculate the \( h m x \), assign the \( h m x \) value to \( x \).

Step2: Determine the step length and re-calculate the next \( h m x \).

Step3: If \( h m x \), we mark the node and all through the nodes from its starting point the holiday point the same marking, end of cycle; Otherwise, continue with Step 1 and Step 2, until all nodes are marked.

Step4: Merge homogenous area, realize the image segmentation. The number of original image nodes (pixels) is 25x33. We will regard the mean shift every small area.

**Bandwidth**

In the classical MS algorithm, the kernel function selection is important. But the kernel function selection is not too sensitive to the image processing result [3] in classical mean shift algorithm. So the only need to be sure is the bandwidth of kernel function. It not only determines the number of sampling points in iterative process, but also influences convergence speed and accuracy of mean shift.

At present there are two major kinds of bandwidth calculating methods: automatic calculation method and self-adaptive calculation method. Automatic bandwidth calculating method is also called fixed bandwidth method, in which the bandwidth always remains unchanged in the iterative process. So the key of this method is to calculate the global optimal bandwidth according to the whole situation of sampling points. The optimal bandwidth associated with the kernel density estimator is defined as the bandwidth that achieves the best compromise between the bias and variance of the estimator, i.e., minimizes AMISE. There is no consensus view of the error measure formula by which the bandwidth selection is optimal. When sampling points included a variety of mode, it is difficult to calculate the global optimal bandwidth. Therefore, self-adaptive bandwidth calculating method must be used according to the local structure of sample points. That is, small bandwidth is used in large density area, and large bandwidth is used in small density area. From literature [6] we can know that: the normal kernel function is used for mean shift when sampling points meet normal distribution; the mode of the mean drift vector is took the maximum. when bandwidth = \( \Sigma \). From literature [6] we can know that: multi-scale bandwidth calculating method is used for many mode sampling points and multivariable sampling points in mean shift algorithm, but the value scope of scale must be known by use of this method.

**Mean shift clustering**

The mean shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. Given \( n \) data points \( x_i, i = 1, ..., n \) on a \( d \)-dimensional space \( Rd \), the multivariate kernel density estimate obtained with kernel \( K(x) \) and window radius \( h \) is

\[ f(x) = \frac{1}{n h^d} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) \]

For radially symmetric kernels, it suffices to define the profile of the kernel \( k(x) \) satisfying

\[ K(x) = c_{k,d} k(\|x\|^2) \]

where \( c_{k,d} \) is a normalization constant which assures \( K(x) \) integrates to 1. The modes of the density function are located at the zeros of the gradient function \( \nabla f(x) = 0 \).

The gradient of the density estimator (1) is

\[ \nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} g \left( \frac{\|x - x_i\|^2}{h} \right) - x \]

where \( g(s) = -k0(s) \). The first term is proportional to the density estimate at \( x \) computed with
The mean shift vector always points toward the direction of the maximum increase in the density. The mean shift procedure, obtained by successive
• computation of the mean shift vector $m_h(x_t)$,
• translation of the window $x_{t+1} = x_t + m_h(x_t)$
is guaranteed to converge to a point where the gradient of density function is zero.

Filtering
The filtering step of the mean shift segmentation algorithm consists of analyzing the probability density function underlying the image data in feature space. Consider the feature space consisting of the original image data represented as the (x, y) location of each pixel, plus its colour in $L^*u^*v^*$ space ($L_-, u_-, v_-$). The modes of the pdf underlying the data in this space will correspond to the locations with highest data density. In terms of a segmentation, it is intuitive that the data points close to these high density points (modes) should be clustered together. Note that these modes are also far less sensitive to outliers than the means of, say, a mixture of Gaussians would be.

The mean shift filtering step consists of finding the modes of the underlying pdf and associating with them any points in their basin of attraction. Unlike earlier techniques, the mean shift is a non-parametric technique and hence we will need to estimate the gradient of the pdf, $f(x)$, in an iterative manner using kernel density estimation to find the modes. For a data point x in feature space, the density gradient is estimated as being proportional to the mean shift vector:

$$\nabla f(x) \propto \frac{\sum_{i=1}^{n} x_i g \left( \frac{\|x - x_i\|}{h} \right)}{\sum_{i=1}^{n} g \left( \frac{\|x - x_i\|}{h} \right)} - x$$

where $x_i$ are the data points, $x$ is a point in the feature space, $n$ is the number of datapoints (pixels in the image), and $g$ is the profile of the symmetric kernel $G$. We use the simple case where $G$ is the uniform kernel with radius vector $h$. Thus the above equation simplifies to:

$$\nabla f(x) \propto \frac{1}{|S_{x,h_s,h_r}|} \sum_{x_t \in S_{x,h_s,h_r}} x_t - x$$

where $S_{x,h_s,h_r}$ represents the sphere in feature space centered at $x$ and having spatial radius $h_s$ and colour (range) radius $h_r$, and the $x_t$ represent the data points within that sphere. For every data point (pixel in the original image) $x$ we can iteratively compute the gradient estimate in Eqn. 2 and move $x$ in that direction, until the gradient is below a threshold. Thus we have found the points where $crf(x_0) = 0$, the modes of the density estimate. We can then replace the point $x$ with $x_0$, the mode with which it is associated. Finding the mode associated with each data point helps to smooth the image while preserving discontinuities. Intuitively, if two points $x_i$ and $x_j$ are far from each other in feature space, then $x_i \notin S_{x_j,h_s,h_r}$ and hence $x_j$ doesn’t contribute to the mean shift vector gradient estimate and the trajectory of $x_i$ will move it away from $x_j$. Hence, pixels on either side of a strong discontinuity will not attract each other. However, filtering alone does not provide a segmentation as the modes found are noisy. This “noise” stems from two sources. First, the mode estimation is an iterative process, hence it only converges to within the threshold provided (and with some numerical error). Second, consider an area in feature space larger than $S_{x,j,h_s,h_r}$ and where the colour features are uniform or have a gradient of 1. Since the pixel coordinates are uniform by design, the mean shift vector will be 0 in this region, and the data points will not move and hence not converge to a single mode. Intuitively, however, we would like all of these datapoints to belong to the same cluster in the final segmentation. For these reasons, mean shift filtering is only a preprocessing step, and a second step is required in the segmentation process: clustering of the filtered data points $\{x_0\}$.

Clustering
After mean shift filtering, each data point in the feature space has been replaced by its corresponding mode. As described above, some points may have collapsed to the same mode, but many have not despite the fact that they may be less than one kernel radius apart. In the original mean shift segmentation paper [1], clustering is described as a simple post-processing step in which any modes that are less than one kernel radius apart are grouped together and their basins of attraction are merged. This suggests using single linkage clustering, which effectively converts the filtered points into segmentation.

The only other paper using mean shift segmentation that speaks directly to the clustering is [2]. In this approach, a region adjacency graph (RAG) is created to hierarchically cluster the modes. Also, edge information from an edge detector is
combined with the colour information to better guide the clustering. This is the method used in the publicly available EDISON system, also described in [2]. The EDISON system is the implementation we use here as our mean shift segmentation system.

**Discussion**

Mean shift filtering using either single linkage clustering or edge-directed clustering produces segmentations that correspond well to human perception. However, as we discuss in the experiments section, this algorithm is quite sensitive to its parameters. The mean shift filtering stage has two parameters corresponding to the bandwidths

(radii of the kernel) for the spatial (hs) and color (hr) features. Slight variations in hr can cause large changes in the granularity of the segmentation. By adjusting the color bandwidth we can produce over-segmentations, which show every minute detail, to reasonably intuitive segmentations, which delineate objects or large patches, to under-segmentations, which obscure the important elements completely. This issue is a major stumbling block with respect to using mean shift segmentation as a reliable preprocessing step for other algorithms, such as object recognition. For an object recognition system to actually use a segmentation algorithm, it requires that the segmentations produced be fairly stable under parameter changes and that the same parameters produce stable results for different images, thus easing the burden of parameter tuning. In an attempt to improve stability, we consider a second algorithm.

**CONCLUSION**

In this paper a novel image segmentation algorithm has been proposed and designed based on the mean shift algorithm. The effectiveness and robustness of the proposed algorithm have verified by some experimental results to express an improved performance compared to the Ncut algorithm. Also the significant reduction to the computational cost of the proposed algorithm in the experiments is favorable for practical applications.
where $F(i) = \{ R(i), G(i), B(i) \}$ is the color vector of region $i$, if the image is the gray-level image, then $F(i)$ is the gray value. and $X(i)$ is the spatial location of region node $i$. In this paper, we defined the $F(i)$ is the average gray of the region $i$, and the $X(i)$ is the center of the region $i$. In this case, the select of $F(i)$ and $X(i)$ are more representative.

Mean shift segmentation:

The mean-shift segmentation algorithm also considers the probability density of feature vectors $F \sim (-x)$ obtained from a given image. However, a non-parametric density model is used instead of an MoG.

The scale of the mean-shift kernel (controlled by $\Sigma$) roughly determines the size and shape of the extracted regions. There is a trade-off between maintaining the salient boundaries but suffering over-segmentation, versus missing some of the important boundaries and under-segmenting the image.

REFERENCES