Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2011) 1–10
Procedia Computer Science

International Conference on Computational Science, ICCS 2011

GX-Means: A model-based divide and merge algorithm for geospatial image clustering

Ranga R. Vatsavai[a], Christopher T. Symons[a], Varun Chandola[a], Goo Jun[b]

[a] *Computational Sciences and Engineering Division, Oak Ridge National Laboratory*
[b] *Center for Statistical Genetics, University of Michigan*

Abstract

One of the practical issues in clustering is the specification of the appropriate number of clusters, which is not obvious when analyzing geospatial datasets, partly because they are huge (both in size and spatial extent) and high dimensional. In this paper we present a computationally efficient model-based split and merge clustering algorithm that incrementally finds model parameters and the number of clusters. Additionally, we attempt to provide insights into this problem and other data mining challenges that are encountered when clustering geospatial data. The basic algorithm we present is similar to the G-means and X-means algorithms; however, our proposed approach avoids certain limitations of these well-known clustering algorithms that are pertinent when dealing with geospatial data. We compare the performance of our approach with the G-means and X-means algorithms. Experimental evaluation on simulated data and on multispectral and hyperspectral remotely sensed image data demonstrates the effectiveness of our algorithm.

Keywords: Clustering, EM, GMM, K-means, G-means, X-means

1. Introduction

Clustering algorithms play a key role in earth science data mining, as they do in many areas of computational science. They are often used to analyze large volumes of complex, multivariate geospatial data, such as remotely sensed images, sensor measurements, field observations, etc., as a first step in gaining insights into the structure or natural groupings within the data. Clustering is also used in compression, exploratory analysis, and summarization of the data. Unfortunately, existing clustering algorithms often perform poorly on geospatial data for reasons described below, and alternative approaches are required.

When dealing with geospatial data, we are presented with several key challenges to current data mining techniques. The mere size of geospatial data is daunting. For example, a standard single image like Landsat ETM+ with a

*Corresponding author
Email address: vatsavairr@ornl.gov (Ranga R. Vatsavai)

[1] This research is supported through the LDRD program at the Oak Ridge National Laboratory. Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

relatively small spatial scope (185KM swath width) may easily consist of more than 60 million pixels. While much of the data in an image is considered redundant for most analyses, sampling in a manner that ensures adequate model representation can be problematic. Thus, any model dealing with geospatial data is likely to encounter problems in terms of scalability and generalization. Moreover, the feature space is not sparse, and this rules out of the use of many approaches that rely on such sparsity to ensure tractability.

Furthermore, there are constant challenges similar to those that typically occur when applying models across domains. For example, gradual changes in image reflectance values due to weather or seasonal variations make the construction of static classifiers nearly meaningless. In addition, ground truth for remote sensing data is particularly costly to obtain, and class definitions for image categorization are often fuzzy at best. Thus, it is commonly necessary to cluster the data in order to get a sense of the natural groupings before attempting to apply any sort of supervised method for categorization, change detection, etc.

Advancements in our ability to gather data, e.g. in terms of frequency, resolution, etc., have led to a variety of challenges in leveraging this data in the analysis of geospatial phenomena. For example, the increase in difficulty in clustering geospatial data at higher resolutions goes beyond issues of scalability. Increases in spatial resolution result in more variance in spectral analyses [1], and this often leads to more difficult clustering challenges that can be problematic for many current approaches. For example, the G-means algorithm works well only as long as there is good separability of the clusters [2]. However, as we see later, most of the cluster distributions are highly overlapping in multivariate geospatial image data.

Determining the number of clusters is one of the key challenges in clustering. And this determination is critical to the usefulness of analytical tools based on clustering. Manual specification of an optimal number of clusters is not feasible in geospatial clustering given the complexity and volume of the data sets involved, and current methods for automatically discovering the optimal number of clusters from the data have not performed well on geospatial data in our experience.

1.1. Related Work and Our Contributions

Clustering is a fertile research area with applications cutting across many domains. A large number of clustering algorithms can be found in the literature [3, 4]. These algorithms can be broadly categorized into: hierarchical, partitional, density-based, and grid-based methods. Sometimes the distinctions among these categories diminish, and some algorithms can be classified into more than one group. More details on various clustering methods can be found in a recent survey paper [5]. Partitional clustering algorithms, especially the k-means algorithm, are very popular in several application domains, including earth sciences. One of the primary inputs to the k-means algorithm is the specification of k, the number of clusters. However, as mentioned above, determining an optimal number of clusters manually is not feasible given the size and complexity of geospatial data sets.

Considerable research has gone into finding the optimum number of clusters directly from the data itself [6, 7, 8, 9, 10, 11]. In [8], the authors have presented a Bayesian information criterion (BIC) based optimization technique for automatically finding the number of clusters. This algorithm is known as X-means. In [10], the authors presented a model based clustering algorithm to derive the number of clusters directly from the data by optimizing the global Bayesian information criterion (BIC). In our experiments we found that the individual components of the model produced by the X-means algorithm are not necessarily Gaussian. In [2, 11], the authors proposed the G-means algorithm, which attempts to automatically discover the number of clusters. The basic idea behind G-means is simple. The initial number of clusters (k) determined using k-means is incremented by splitting each cluster that doesn't pass a statistical test. The clustering process is repeated until all the clusters have passed this statistical test. In many practical situations there is a danger of overestimating the number of clusters, especially if the model is assumed to be a Gaussian Mixture Model (GMM), which is an appropriate and commonly utilized model for representing geospatial image data. We addressed this problem by introducing a merge step in our earlier work [12]. As our experimental results demonstrate, the GMM is clearly an accurate representation of the geospatial data we analyzed. Adding a merge step avoided finding lot of small non-gaussian clusters.

In this paper we provide a new algorithm which combines the best features of both X-means and G-means while reducing some of their limitations. In order to overcome the limitations discussed above, we modified the general process of the G-means algorithm in two ways. First, instead of the univariate statistical test (Anderson-Darling, AD) used in G-means, we use a multivariate test statistic (Shapiro-Wilk). This modification has the following advantages.

First, we don't have to project multivariate data into one dimension. This is important for earth science data mining as the geospatial data is often high-dimensional in nature. Finding a good projection can be as difficult as finding a good k. Second, the AD test is good for small samples; that is, when the number of samples ≤ 25. However, in earth science data sets typically we have a large number of samples (per cluster). Finally, the multivariate Shapiro-Wilk test exhibits good power against alternatives [13]. Second, at each split, we estimate the quality of new clusters as a criteria to back track the splits if needed. We use two tests to monitor the quality of the splits. The first test is the same as in X-means, where we monitor the local BIC values. If the BIC is not maximized, then we stop further splits and accept the parent node. In the second test, we use the KL Divergence measure after splitting the clusters to see if any pair of clusters is too close. If any two clusters are too close to each other, then it is better to combine them, even though such combination may violate significance testing. We also made two additional improvements. The first one is related to binary splits that are commonly used. Instead of binary splits, we increase the number of splits by one, based on how many clusters failed multivariate normality tests at a given iteration. The second one deals with the size of the cluster. In the following sections, we present our algorithm and experimental results. Experimental evaluation demonstrates that our algorithm can avoid the common problem of ending up with a large number of spurious clusters in the analysis of geospatial image data. In addition, our algorithm produced more clusters that conform to the model assumption (GMM) than any other method considered in this paper.

2. Clustering Framework

The basic statistical framework for our clustering approach is Gaussian Mixture Models (GMMs). Typically, model-based clustering approaches are not applied on entire data sets given the computational and data complexity. Instead, a subset of data samples is collected from the full data set, and the model parameters are estimated using these samples. Once a model is constructed, all of the samples (data points) in the full data set can then be assigned to one of the clusters based on some distance (or decision) criteria. Our algorithm is based on the assumption that the data samples are generated by a GMM. Then, the objective is to learn the GMM parameters from these samples. We now briefly describe an expectation maximization (EM) based algorithm to learn the GMM parameters.

2.1. Estimating GMM Parameters

Let us now assume that the sample data set $D = \{x_i\}_{i=1}^{n}$ is generated by the following mixture density.

$$p(x_i|\Theta) = \sum_{j=1}^{K} \alpha_j p_j(x_i|\theta_j) \qquad (1)$$

$$p(x|y_j) = \frac{1}{\sqrt{(2\pi)^{-N}|\Sigma_j|}} e^{\frac{-1}{2}(x-\mu_j)^t|\Sigma_j|^{-1}(x-\mu_j)} \qquad (2)$$

Here $p_j(x_i|\theta_j)$ is the pdf corresponding to the mixture j and parameterized by θ_j, and $\Theta = (\alpha_1, \ldots, \theta_K, \theta_1, \ldots, \theta_K)$ denotes all unknown parameters associated with the K-component mixture density. For a multivariate normal distribution (eq. 2), θ_j consists of elements of the mean vector μ_j and the distinct components of the covariance matrix Σ_j. The *log-likelihood* function for this mixture density can be defined as: $L(\Theta) = \sum_{i=1}^{n} \ln\left[\sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j)\right]$. In general, $L(\Theta)$ is difficult to optimize because it contains the ln of a sum term. However, this equation greatly simplifies in the presence of unobserved (or incomplete) samples. Normally, we treat the cluster labels as missing (unobserved) data, and use the expectation maximization technique to estimate the parameters (Θ). The EM algorithm consists of two steps, called the E-step and the M-step, as given below.

E-Step: For a multivariate normal distribution, the expectation $E[.]$, which is denoted by p_{ij}, is the probability that Gaussian mixture j generated data point i, and is given by:

$$p_{ij} = \frac{\left|\hat{\Sigma}_j\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_j)^t\hat{\Sigma}_j^{-1}(x_i-\hat{\mu}_j)\right\}}}{\sum_{l=1}^{M}\left|\hat{\Sigma}_l\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_l)^t\hat{\Sigma}_l^{-1}(x_i-\hat{\mu}_l)\right\}}} \qquad (3)$$

M-Step: The new estimates (at the kth iteration) of the model parameters in terms of the old parameters are computed using the following update equations:

$$\hat{\alpha}_j^k = \frac{1}{n}\sum_{i=1}^{n} p_{ij} \quad (4) \qquad \hat{\mu}_j^k = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \quad (5) \qquad \hat{\Sigma}_j^k = \frac{\sum_{i=1}^{n} p_{ij}(x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t}{\sum_{i=1}^{n} p_{ij}} \quad (6)$$

The EM algorithm iterates over these two steps until convergence is reached. We can now put together these individual pieces into the following algorithm (Table 1) which computes the parameters for each component in the finite GMM that generated our sample data D (without any cluster labels).

Inputs: D, sample data set; K, the number of clusters.
Initial Estimates: Do clustering by k-means, and estimate initial parameters using Maximum Likelihood Estimation (MLE) technique to find $\hat{\theta}$.
Loop: While the complete data *log-likelihood* improves:
E-step: Use current model (GMM) to estimate the class membership of each data sample, i.e., the probability that each Gaussian mixture component generated the given sample point, p_{ij} (see Equation 3).
M-step: Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each data sample (see Equations 4, 5, 6)
Output: Parameter vector Θ.

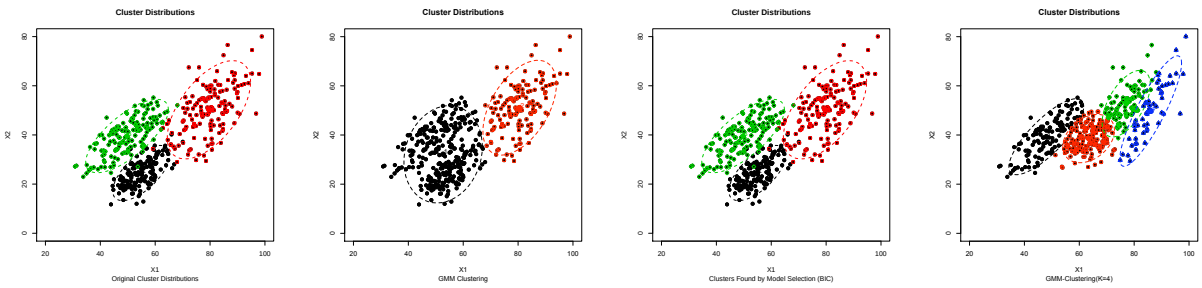
Table 1: Algorithm for estimating parameter of a GMM

	x	y
C1	55.00	25.00
C2	80.00	50.00
C3	50.00	40.00

Table 2: Simulation Parameters (μ)

	C1		C2		C3	
	x	y	x	y	x	y
x	30.00	25.00	60.00	40.00	60.00	50.00
y	25.00	40.00	40.00	90.00	50.00	70.00

Table 3: Simulation Parameters (Σ)



(a) Original Distribution (b) GMM(K=2) (c) GMM(K=3) (d) GMM(K=4)

Figure 1: Simulated vs. Estimated Using GMM Clustering (For different K values)

2.2. Simulation Example 1

We now demonstrate the GMM clustering algorithm (1) on a simulated data set. We generated a GMM with three components. The parameters are given in Table 2 and Table 3. We generated 150 bivariate Gaussian samples from each component density. We applied the GMM clustering algorithm on this sample data set by assuming different K values and the results are summarized in Figure 1. From the figure it can be seen that when the assumed K value is correct (that is, K=3) we have very good estimates (compare subfigures (a) and (c)). However for other values of K (subfigures (b) and (d)) the estimates are very different from the original distribution. This simulation emphasizes the need to estimate a good K value (if possible automatically from the data).

3. Learning to estimate K

In this section we address the problem of estimating K automatically from the data. Our work is closely related to three well-known techniques that deal with automatic determination of optimal K; namely, model-based clustering [14], X-means clustering [8], and G-means clustering [2]. These three approaches can be conceptually described using a generic divide and merge algorithm (see Figure 2, which is adopted from [15]). As shown in this figure, the divide and merge clustering algorithms have two phases: a top-down "divide" phase and a bottom-up "merge" phase.

3.1. Model-based clustering

As with the estimation of the model parameters for a finite Gaussian mixture model, we assume that the training dataset D is generated by a finite Gaussian mixture model, but we don't know either the number of components or the labels for any of the mixture components. In the previous section, we outlined an algorithm to find parameters by assuming a K-component finite Gaussian mixture model. In general, we can estimate parameters for any arbitrary K-component model, as long as there are a sufficient number of samples available for each component and the covariance matrix does not become singular. Then the question remains, which K-component model is better? This question is addressed in the area of model selection, where the objective is to choose a model that maximizes a cost function. There are several cost functions available in the literature. The most commonly used measures are Akaike's information criterion (AIC), Bayesian information criteria (BIC), and minimum description length (MDL). The common criteria behind these models is to penalize models that have additional parameters, so BIC and AIC based model selection criteria follow the principal of parsimony. In this study we analyzed BIC as a model selection criterion, which also takes the same form as MDL. We chose BIC because it has been found to be very useful in model based clustering [14], and because it is defined in terms of maximized log-likelihood, which we are computing in our parameter estimation procedure defined in the previous section. BIC can be defined as, $BIC = MDL = -2\log L(\Theta) + m\log(N)$, where N is the number of samples and m is the number of parameters. Typically, the algorithm is run 'K number of times, where at each k the data is "divided" into k clusters. BIC (or any other suitable cost function) is computed for each k, and the best model is chosen for which the BIC value is maximum. Typically, these approaches do not have a "merge" step.

3.2. X-means clustering

The X-means [8] clustering algorithm is a variation of BIC based model selection. Instead of running the model selection algorithm K times, the X-means algorithm starts with a minimum number of k centroids and continually adds new centroids by splitting the existing centroids until an upper bound K is reached. At each split, the new BIC value of the resulting (local) model is compared against the parent BIC value. A split is kept if the local score (BIC) is better, otherwise this new split is discarded and the previous (parent) structure is retained.

3.3. G-means clustering

G-means [2] clustering is similar to X-means clustering. G-means is initialized by k-means clustering for a suitable initial K value. Each cluster is then tested for normality using a univariate test, the Anderson-Darling (AD) statistic. For a user given p-value, if the AD test fails, then the cluster is split into two clusters. k-means clustering is performed again with the new K, and the process is repeated until no more clusters (new splits) can be found. Multivariate data is projected onto a single dimension to permit the AD test. In [2], the authors argue that BIC has a tendency to find too many clusters. In our experiments, we found that G-means clustering also tends to find too many clusters as there is no good stopping/merging criteria. We demonstrate this through an example simulated data set (see subsection 3.6).

3.4. Our approach (GX-means)

First, we start with a simple extension to G-means. We initiate the algorithm with a minimum number of clusters. Each cluster is recursively split into two new clusters if it fails a statistical significance test criteria. Up to this point the algorithm is very similar to the G-means algorithm. However, we have additional test criteria to stop the splits so that smaller clusters with fewer data points can be avoided.

A conceptual framework of our approach is shown in Figure 3. In this figure, H_o represents the null hypothesis, and T implies acceptance and F implies rejection. As compared to the G-means algorithm, we made two modifications in our approach. First, instead of a univariate test statistic, we adopted a multivariate test statistic, known as the

Inputs: D, sample data set; significance (default p-value = 0.001), initial K=0, clusters2process (default = 2)
Clustering: GMM-Clustering (see Algorithm 1)
Loop 1: While (TRUE):
Loop 2: For all clusters2process
BIC test: If BIC is improved accept the split, otherwise accept the root cluster
Statistical test: Shapiro-Wilk test (see Eq. 7).
Check: If a cluster fails statistical test, **then** split that cluster into two clusters; increment number of cluster2process, **else** accept the cluster, remove (cluster) data from D, decrement number of clusters2process, and increment K
Clustering: GMM-Clustering(cluster2process)
Merge Compute pair-wise KL-Divergence, **If** two-clusters are closer than threshold value, **then** merge these clusters, decrement clusters2process
Check: If clusters2process = 0 **then** break (from Loop 1)
Output: Parameter vector Θ.

Table 4: GX-means Algorithm

Shapiro-Wilk statistic (see subsection 3.5). We accept the split if BIC is improved as in the X-means algorithm. We also used the KL divergence measure to check if the new clusters (meaning their distributions) are highly overlapping. If two distributions are highly overlapping, then those clusters are merged. After a split is accepted, we apply statistical significance tests on each of the new clusters resulting from the splits. Statistical tests are sensitive to noise, so it is likely that the splitting process (increasing K) will continue beyond optimal K as many times as the statistical significance test fails (even though clusters are close to multivariate normal). As a check to prevent this from happening, we added additional criteria to check for the quality of splits. In our experiments with both simulated and real data, it is observed that binary splits lead to the discovery of false clusters. This happens because samples from some Gaussians are distributed among different nodes of the split. The merge step is not helpful in those cases, and this in turn leads to failure of the Shapiro-Wilk test. The overall impact is that sometimes these splits are continued until the samples in the resulting clusters are too few to meaningfully test for the statistical significance. To deal with such instances, we incorporated two improvements into the GX-means algorithm. First, instead of binary splits, we resort to a different split criterion based on the number of clusters that fail the statistical significance test. For example, at a given iteration, if two clusters fail the test, then we pool all the samples from the two clusters and split them into three. Let us now assume that the pooled samples are described by a three component GMM. Binary splits would have led to finding a four component GMM, with samples from one (or more) of the Gaussians spread across these clusters. Second, we stop further splitting a cluster if the size of the cluster (number of data points in a cluster) is less than a predetermined threshold. This criterion is particularly important in geospatial image clustering. Due to the nature of the spatial distribution of some objects (e.g., water), it may frequently happen that the sampling scheme picks up very few samples from some of these objects. And though these samples may be distinct from other classes, they may not pass the statistical significance test when there are not enough samples to accurately estimate the correct covariance structure. Small clusters may also result due to noise.

These small clusters may be treated in the following three ways. First, the obvious choice is to discard samples from such clusters and select the model using only the GMM components that passed the Shapiro-Wilk test. However, many times it may be useful to keep the full model and evaluate the clustered image to see if the small clusters produce any meaningful spatial objects in the real world. The third option is to merge the small clusters into the closest cluster (e.g., based on KL divergence). In our experiments we observed that some of the small clusters actually produced meaningful spatial objects (e.g. in the case of water). The GX-means algorithm is summarized in Table 4.

Conceptually we can think of this new approach as fusing the best features from all three approaches (model-based, X-means, and G-means) while at the same time avoiding some of the disadvantages of these methods. The advantages of this new approach are clearly evident from our experiments.

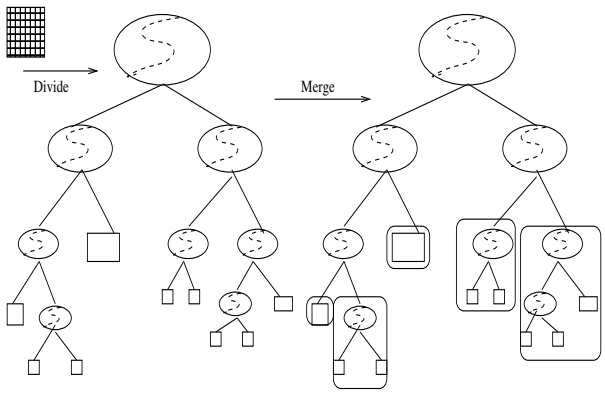


Figure 2: Generic divide and merge clustering approach

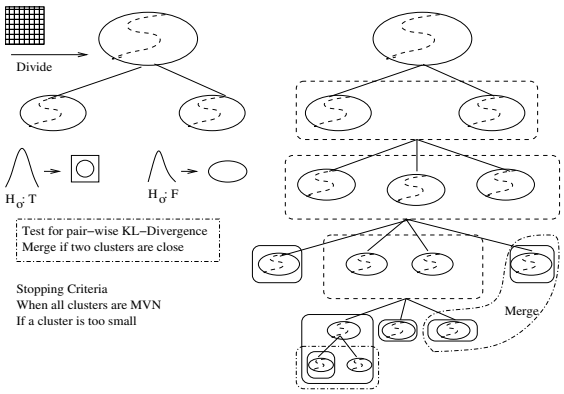


Figure 3: GX-means clustering approach

3.5. Testing for multivariate Gaussian fit

We now briefly explain the multivariate Shapiro-Wilk test. The Shapiro-Wilk test was originally proposed in [16], and it is commonly used in testing the null hypothesis that a multivariate sample $x_{i=1}^{n}$ comes from a normally distributed population. The Shapiro-Wilk test statistic (W) is calculated as follows:

$$W = \frac{(\Sigma_{i=1}^{n} a_i x_{(i)})^2}{\Sigma_{i=1}^{n}(x_i - \mu)^2} \tag{7}$$

where $[x_{(i)}]$ is the ith order statistic, that is, the ith smallest number in the sample; $[\mu] = \frac{(x_1+...+x_n)}{n}$ is the sample mean; $[a_i]$ are the constants generated from the means, variances and covariances of the order statistics of a sample of size n from a normal distribution. The constants a_i are computed using $(a_1, \ldots, a_n) = \frac{\mu^T \Sigma^{-1}}{\sqrt{(\mu^T \Sigma^{-1} \Sigma^{-1} \mu)}}$, where $\mu = (\mu_1, \ldots, \mu_n)^T$ are the expected values of the order statistics.

The null hypothesis that the samples from a given cluster are normally distributed for a chosen significance level α is rejected if the p-value for the test is less than the selected α. The method for computing the p-value depends on the size of the cluster (population). For $n = 3$, the probability distribution of W is known and the p-value can be directly determined. For $n > 4$, the p-value is computed using the following transformation:

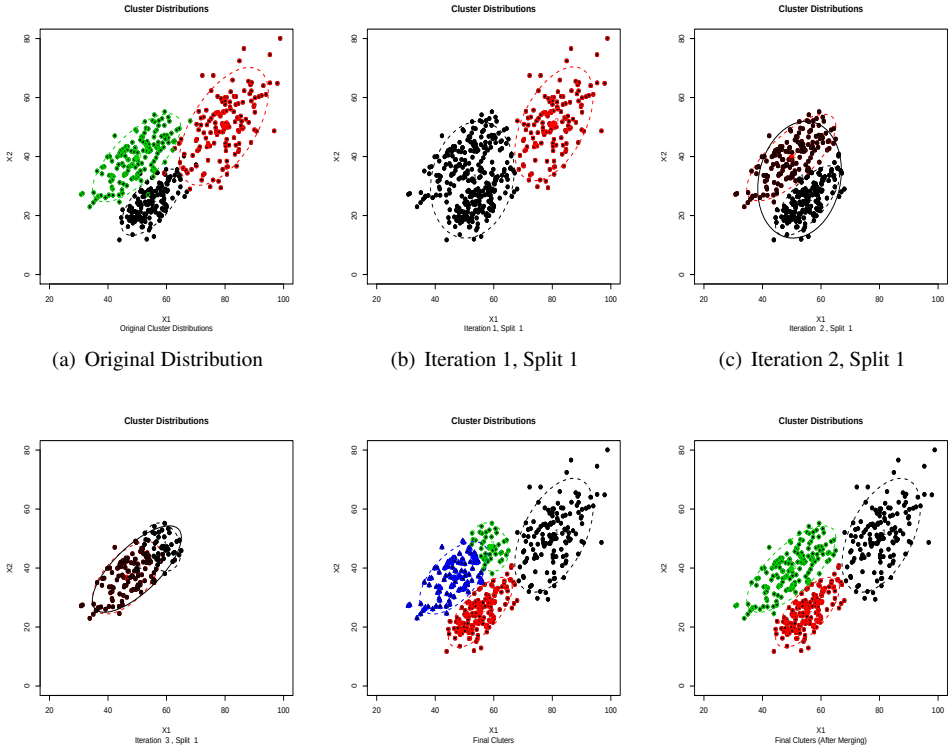
$$Z_n = \begin{cases} (-\log(\gamma - \log(1 - W_n)) - \mu)/\Sigma & : \quad 4 \leq n \leq 11 \\ (\log(1 - W_n) - \mu)/\Sigma & : \quad 12 \leq n \leq 2000 \end{cases} \tag{8}$$

3.6. Simulation Example 2

We now demonstrate the our GX-means clustering algorithm (4) on the simulated data set (Table 3). The results are summarized in Figure 4. The first iteration found two clusters (Figure 4(b)). The red cluster passes the Shapiro-Wilk test. As a result, only the 2nd cluster (black) is split into two clusters (c). In the next iteration, the red cluster failed the Shapiro-Wilk test, and as a result it was split into two clusters (d). The G-means cluster algorithm would have resulted in the final solution shown in Figure 4(e). On the other hand, the additional step introduced in our algorithm finds that these two clusters are very close (KL-Divergence), and thus decrements the number of clusters to 3. The final solution is shown in Figure 4(f). Compare Figure 4(f) with the original distribution in Figure 4(a).

4. Experimental Results

We applied all four clustering algorithms (Model-based, G-means, X-means, and GX-means) on the two real data sets described below.



(a) Original Distribution (b) Iteration 1, Split 1 (c) Iteration 2, Split 1

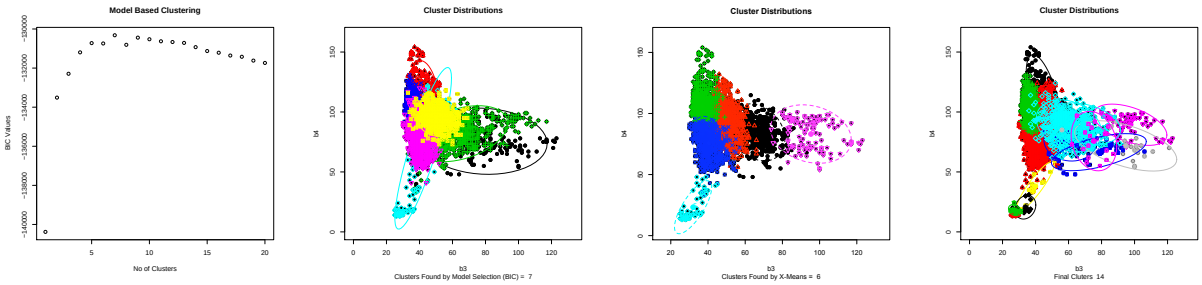
(d) Iteration 3, Split 1 (e) Without Merge Step (f) With Merge Step

Figure 4: GX-means Algorithm Trace

Multispectral geospatial image data: The Cloquet study site encompasses Carlton County, Minnesota, which is approximately 20 miles southwest of Duluth, Minnesota. The region is predominantly forested, composed mostly of upland hardwoods and lowland conifers. There is a scattering of agriculture throughout. The topography is relatively flat, with the exception of the eastern portion of the county containing the St. Louis River. Wetlands, both forested and non-forested, are common throughout the area. The largest city in the area is Cloquet, a town of about 10,000. We used a spring Landsat 7 scene taken May 31, 2000 and cropped it to the study region. The final rectified and cropped image size is 1343 lines x 2019 columns x 6 bands. We selected 400 random plots. From each plot, we extracted 9 feature vectors (6-dimensional) by placing a 3 x 3 window at the center of each plot. In other words, the sample data set consisted of of 3600 feature vectors.

Hyperspectral geospatial image data: We used Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) image data collected over the "Jasper Ridge" region during 1997. AVIRIS data is widely used in many earth science related projects, including vegetation mapping, soil studies, mineral exploration, and atmospheric pollution studies. It is a unique optical sensor that captures calibrated images with 224 spectral channels (bands) with wavelengths from 400 to 2500 nanometers. Hyperspectral data poses several challenges to remote sensing scientists. Analyzing 224 spectral bands for interesting patterns or spatial distributions of objects is next to impossible for any human analyst. Given the fine spectral resolution, it is expected that AVIRIS can allow discrimination of many more classes than multispectral images (e.g., Landsat 7 image described above) can allow. Therefore, clustering offers many advantages in exploring hyperspectral data. Due to the high computational complexity and issues involved with statistical significance tests in large dimensional spaces, we applied principal component analysis and reduced the number of channels to 6. Figure 5 shows the bivariate density plots of the clusters found by the various methods.

Table 5 shows the number of clusters found by each of the clustering methods. It also gives the number of clusters that actually passed the Shapiro-Wilk test. From the above table, it is clear that our algorithm finds a good



(a) Model-based (BIC-Values) (b) Model-based (K=7) (c) X-means (K=6) (d) GX-means(K=14)

Figure 5: Clusters found on Carlton Landsat data

Method	Total Clusters	Gaussian Clusters	Small Clusters
Model-based	7	2	-
X-means	6	4	-
G-means	19	5	-
GX-means	14	9	5

Table 5: Carlton Multispectral Image Clustering Results

Method	Total Clusters	Gaussian Clusters	Small Clusters
Model-based	31	14	-
X-means	22	15	-
G-means	36	12	-
GX-means	27	26	1

Table 6: Jasper Ridge Hyperspectral Image Clustering Results

number of Gaussian clusters. It is surprising to see the conformance of clusters found by the model-based clustering to the Gaussian mixture model assumption, as only a few clusters passed the statistical significance test. We forced G-means to stop splitting if the cluster size is less than 20 data points; otherwise, we end up with far too many clusters (which were not Gaussian anyway). Another interesting fact to observe is that if we remove the small clusters then our solution is very close (both in terms of the number of clusters and pair-wise KL Divergence) to the model-based clustering. However, our solution found more Gaussian clusters. We also compared our clustering results to a supervised classification experiment. In previous studies using the Carlton dataset, remote sensing analysts identified 10 classes for the site. With the removal of small clusters, the GX-means produced 9 clusters. We visually compared our clustering results with that of the supervised classification image. There appears to be a good correspondence between the clusters and the thematic classes identified by the analysts. However, further analysis and experimentation is needed to quantitatively verify this correspondence. The raw and clustered images are shown in Figure 6.

5. Conclusions

We developed a divide and merge clustering algorithm based on a GMM distribution and expectation maximization (EM) parameter estimation. The algorithm combines good features of three well-known clustering schemes. At the same time, our algorithm avoids certain important limitations of these approaches. Experimental evaluation on simulated data shows that our algorithm produces parameters which are very close to the original distribution. Clustering on real data sets shows good conformance of clusters to the model assumption, in contrast to the results obtained using the existing algorithms. The results on the Carlton image also demonstrated a good correspondence between the clusters and thematic (information) classes chosen by remote sensing analysts in supervised classification projects. Further analysis and experimentation is needed to understand the performance and utility of this algorithm in earth science data mining applications.

Our analysis also brought forth a few challenges that need further research considerations. For large datasets, such as hyperspectral images, model-based clustering methods tend to be limited by memory and computational requirements. The same applies to many feature selection methods. For example, attempts to apply Laplacian Eigenmap-based dimensionality reduction [17] on the hyperspectral data quickly became infeasible beyond 10 percent of image samples. Therefore, we resorted to the use of PCA to reduce the dimensionality of the hyperspectral image before

(a) Jasper Ridge AVIRIS Image (b) Jasper Ridge GX-means clustering (K=27)

Figure 6: Jasper Ridge hyperspectral image data along with clustered image

applying various clustering techniques. A second problem associated with clustering geospatial images occurs due to sampling the image space to generate data for clustering. In our experiments we noticed that no two samplings produced exactly the same results. Many times even the number of clusters is not exactly the same. This happens mainly due to the complexity of geographic phenomena captured by the images and the spatial distribution of objects (some are small and some are linear, etc). There is a need to combine models generated from different data samples in hopes that the combination will result in a better overall clustering and also to avoid huge memory requirements. Our future research will focus on some of these challenges.

6. References

[1] A. R. C.A. Coburn, A multiscale texture analysis procedure for improved forest stand classification, International Journal of Remote Sensing 25 (20) (2004) 4287–4308.

[2] G. Hamerly, C. Elkan, Learning the k in k-means, in: In Neural Information Processing Systems, MIT Press, 2003.

[3] A. K. Jain, R. C. Dubes, Algorithms for clustering data, Prentice-Hall, Inc., USA., 1988.

[4] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.

[5] J. Han, M. Kamber, A. K. H. Tung, Spatial Clustering Methods in Data Mining: A Survey, in: Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.

[6] G. J. Mclachlan, D. Peel, On a resampling approach to choosing the number of components in normal mixture models, in: Proceedings of Interface 96, 28th Symposium on the Interface, 1997, pp. 260–266.

[7] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, Journal of the Royal Statistical Society: Series B 63 (2) (2001) 411–423.

[8] D. Pelleg, A. W. Moore, X-means: Extending k-means with efficient estimation of the number of clusters, in: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 727–734.

[9] n. Miguel Á. Carreira-Perpi Mode-finding for mixtures of gaussian distributions, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1318–1323.

[10] C. Fraley, A. Raftery, R. Wehrens, Incremental model-based clustering for large datasets with small clusters, Journal of Computational and Graphical Statistics 14 (2005) 2005.

[11] Y. Feng, G. Hamerly, Pg-means: learning the number of clusters in data, in: Advances in Neural Information Processing Systems 19, MIT Press, 2007, pp. 393–400.

[12] R. R. Vatsavai, Incremental clustering algorithm for earth science data mining, in: 9th International Conference on Computational Science (ICCS), Vol. 5545 of Lecture Notes in Computer Science, Springer, 2009, pp. 375–384.

[13] S. S. Shapiro, M. B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 3 (52).

[14] C. Fraley, A. E. Raftery, Model-based clustering, discriminant analysis, and density estimation, Journal of the American Statistical Association 97 (2002) 611–631.

[15] D. Cheng, R. Kannan, S. Vempala, G. Wang, A divide-and-merge methodology for clustering, ACM Trans. Database Syst. 31 (4) (2006) 1499–1525.

[16] S. S. Shapiro, M. B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (3) (1965) 591–611.

[17] M. Belkin, P. Niyogi, Semi-supervised learning on riemannian manifolds, Mach. Learn. 56 (1-3) (2004) 209–239.