Gaussian Mixture Model + Rule-Based Classifier for Settlement Mapping

Omkar Acharya, Sridutt Bhalachandra, Felix Kim, Sean Mahaffey, Jonathan Raynor, Amit Watve

Department of Computer Science, North Carolina State University

***Abstract*—A combination of unsupervised learning (soft clustering through a Gaussian Mixture Model) and supervised learning (classification via a Rule-Based Classifier using the GMM results as attributes) is used to automate the process of classifying patches of satellite images as one of five area types: commercial, residential 1, residential 2, water, and vegetation. Both the GMM and RBC were built from scratch. The final results of our Rule-Based Classifier are of comparable quality to the Rule-Based Classifier used in the Weka package.**

Keywords - gaussian mixture model; rule based classifier; classification; settlement mapping; supervised learning; unsupervised learning

I. Introduction and Problem Statement

Settlement Mapping is vital to explore the settlement activity in an area and can give insights into the dynamic relationship between the people and places. It facilitates understanding of land and resource use that can help in urban planning and disaster recovery. Conventionally, settlement mapping used survey data from government or other agencies collected through physical undertakings like a census among others. However, with the advent of the remote sensing era, satellites or other aerial vehicles have become invaluable sources of survey data helping to map unexplored terrain [1], [2], [3].

In the present era, the abundance of data from remote sensing has shifted the challenges from data collection to data mining, which is required to extract useful information. Likewise, many additional challenges now exist in mapping settlements from the remote sensing data. How does one know what each of the components for the pixels in the image represent? How do you distinguish an Urban settlement from Rural? How do you differentiate commercial buildings from housing or vegetation from water? We attempt to address these challenges using Gaussian Mixture Models with Rule-Based Classifiers.

The first objective is to do data preprocessing by taking a raw satellite image in .tif format and extracting appropriate-sized sample patches to both train and test a supervised learner that can classify a patch as a certain area type whether urban vs. rural or more fine-grained categories such as commercial vs. residential type 1 vs. residential type 2 vs. vegetation vs. water. One question to answer is how large the patch size should be (e.g. 50x50 pixels or 75x75 pixels). Another challenge is how to extract the patches in an automated fashion as opposed to manually.

The second objective is to take the raw pixel RGB band values and convert these to component responsibilities (where each component represents an object in the image such as a tree or building) that can be used as features for the supervised learner classifying the patch as a certain area type. In this case, a Gaussian Mixture Model is used to get the component responsibilities and it shall be implemented from scratch. Questions to answer include how many components to use for the GMM, how to determine what the components represent, and why to use a GMM in the first place.

The third and final objective is to implement a supervised learner that will determine the area type of a patch. In this case, a Rule-Based Classifier is used and shall be implemented from scratch. Questions to answer include what kind of rule-generation method to use, what makes an RBC a suitable classifier for this task, and if there are any kinds of classification in particular that the RBC struggles more with.

II. Related Work

A Gaussian Mixture Model (GMM) is a probabilistic model for soft clustering that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [4]. These unknown parameters can be learned through the Expectation Maximization algorithm. The Gaussian Mixture Model may be seen as a generalization of the K-means algorithm since each Gaussian not only requires a mean parameter (as does K-means) but also a variance parameter. Advantages that a GMM has over K-means include: it better handles clusters of various shapes and sizes [5] and also instead of providing hard cluster assignments it provides soft assignments that are more informative and that can later on be converted to hard assignments. In addition, a GMM can use K-means for initialization and therefore can leverage the benefits of K-means clustering.

In contrast, a Rule-Based Classifier is a technique for classifying records using a collection of if...then... rules [5]. It offers several advantages: it is relatively easy to interpret (especially compared to a neural network) while being as highly expressive as a decision tree, can handle cases where classes are imbalanced, and can handle attributes that are repetitive without depending so much on feature selection to remove them. For the specific domain of settlement mapping, the ease of model interpretability specifically could be very valuable.

Currently, there seems to be a lack of published research on the problem of performing settlement mapping using an approach of clustering pixels combined with supervised learning and classification without using neural networks. Therefore, this paper explores relatively uncharted territory. Similar work has been done using neural networks to classify rural areas in Europe [6]. There has also been work done involving monitoring change in topology and land-use [7], [8]. These projects share our use of satellite images to draw conclusions about an area but differ in either approach, goal, or both.

Fig. 1. The 3-band (red/green/blue) raster GeoTiff test image with a spatial resolution of 5898x7696 and a pixel size of 1 meter

		Predicted				
		Commercial	**Residential Type 1**	**Residential Type 2**	**Vegetation**	**Water**
Actual	**Commercial**	**3**	0	2	0	0
	Residential Type 1	0	**5**	0	0	0
	Residential Type 2	0	0	**3**	0	0
	Vegetation	0	0	0	**3**	0
	Water	0	1	0	0	**3**

TABLE I. CONFUSION MATRIX SHOWING THE CLASSIFICATION OF WEKA RIPPER ON THE TEST DATA WITH 20 SAMPLES. THE ACCURACY OBTAINED IS 85%

		Predicted				
		Commercial	**Residential Type 1**	**Residential Type 2**	**Vegetation**	**Water**
Actual	**Commercial**	**3**	1	0	1	0
	Residential Type 1	0	**5**	0	0	0
	Residential Type 2	0	0	**3**	0	0
	Vegetation	0	0	0	**3**	0
	Water	0	1	0	0	**3**

TABLE II. CONFUSION MATRIX SHOWING THE CLASSIFICATION OF OUR RBC ON THE TEST DATA WITH 20 SAMPLES. THE ACCURACY OBTAINED IS 85%

III. METHODOLOGY

Figure 3 shows the overall pipeline for data preprocessing, GMM for unsupervised attribute generation, and RBC for supervised learning and classification.

The starting point is a satellite image that is initially broken into a grid of patches, some of which are manually labeled as one of 5 classes: commercial, residential type 1, residential type 2, vegetation, and water. The 5 classes were predetermined using domain knowledge from an expert.

Each raw image was in the following file format: 3-band (red/green/blue) raster GeoTiff with a spatial resolution of 5898x7696 and a pixel size of 1 meter. Patches from the image of various sizes were tested and the most accurate classifications resulted from patches of size 50x50 pixels. This is because smaller patches were not able to encompass more than one element of a class. For example, it may cover only part of a roof and not reach the road in a residential type one patch. Larger patches were also problematic because they would contain components from multiple classes causing the models representation of that class to become distorted. It is also important to note that for the image dataset, there was no predetermined ground truth supplied, not only for the

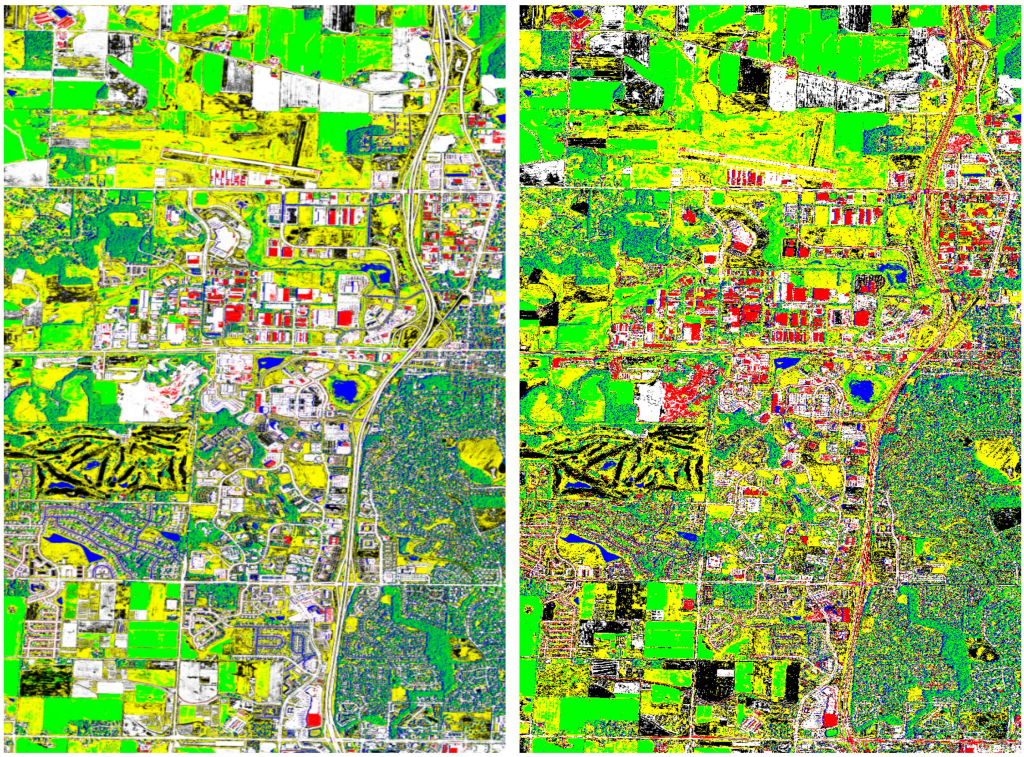

Fig. 2. The left half shows the output of Scikit-Learns GMM model. The right half shows the output of our GMM implementation on the test image. Green-vegetation, white-concrete/road, yellow-ground, red-building, black-miscellaneous, blue-water

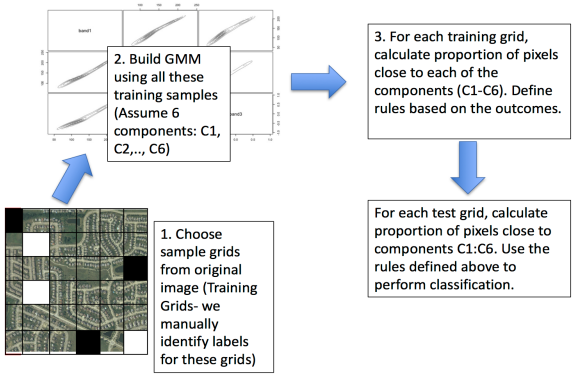

Fig. 3. Schematic explaining the procedure that uses GMM and Rule-Based Classifiers to detect components in an image [9]

appropriate patch sizes, but also the class labels for each patch as well. This means that patches for training and testing had to be manually extracted from the source data.

Multiple representative patches were identified and sampled for each class, exclusively for use in the training set. Random sampling was used to produce patches for the test set, but each random sample had to be manually labeled using domain knowledge in order to determine class and overall accuracy later in the process.

A single 800 x 800 pixel image generated by merging 16 200 x 200 pixel patches was used to build a GMM that assigns component responsibilities to each of the pixels where each component is a type of object in the image. Since the creation of the GMM is an unsupervised soft clustering problem, it was a challenge to determine how many components were to be used without any predetermined labels for the pixels. Through runs of the GMM on various numbers of components, it was determined that the optimal number of components was 6. Specifically, the soft clustering results of the GMM were converted to hard assignments and then the SSE was taken for all the data points. The SSE reached its low point with 6 components. Then the actual descriptions of the 6 components (vegetation, concrete/road, ground, building, miscellaneous, and water) were determined by comparisons to the original image and domain knowledge. Hard assignments from the K-means algorithm (which itself was initialized using K-means++ to spread out the initial centroids) were used as the initial component responsibilities for the Expectation step of the Expectation Maximization algorithm to avoid getting stuck in a local optimum.

For each training grid, the relative proportion of pixels close to each of the components was calculated. These 6 proportions became the features for the supervised learning problem of defining the rules for the rule-based classifier. Next, for each test grid, the proportion of pixels close to the training components is calculated using weighted similarity (Listing 1) and then the rule-based classifier is used to perform the classification for the test grid.

Listing 1. To calculate weights of each component of the test grid for each rule.

```
for each i in rules
  for each j in components
    weight[i,j]=1/(1+abs(proportion[j]-rule proportion[i,j]))
```

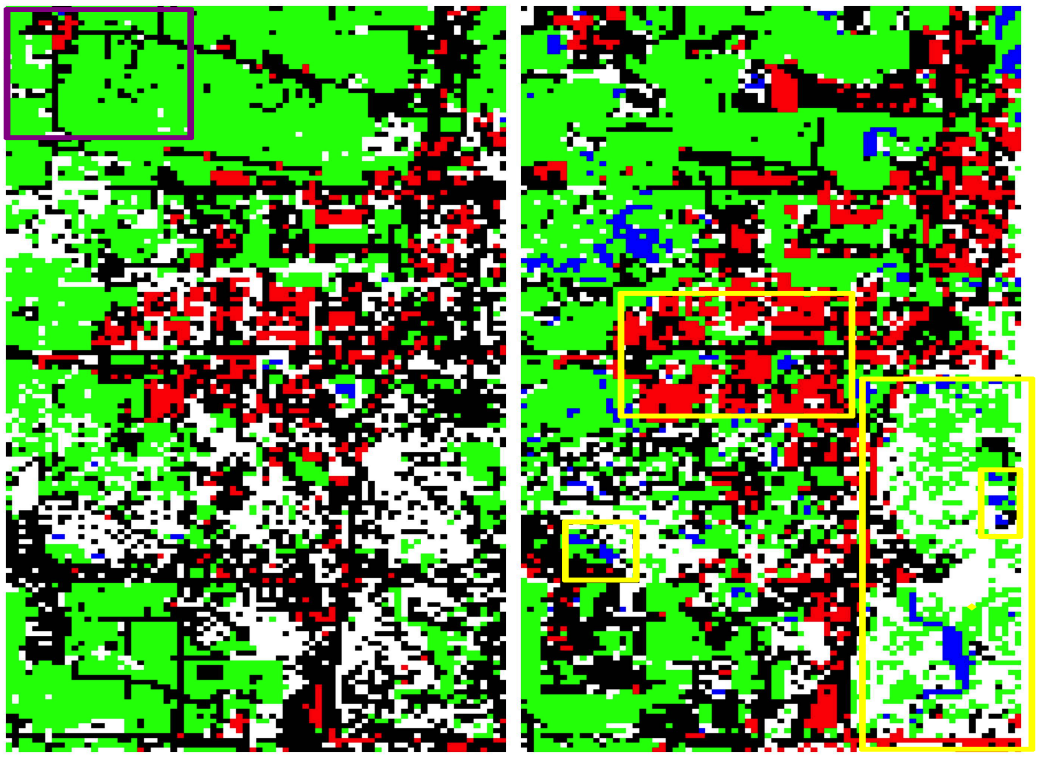

Fig. 4. The left half shows the output of using the rules from Weka RIPPER, while the right half shows the output from our Rule Based Classifier alg. Red-commercial, white-residential type 1, black-residential type 2, green-vegetation, blue-water

The rule-based classifier we chose to implement was a one-rule classifier with rules generated based on the training patch data. We added a majority vote based on different levels of variance of the actual values from the rule to ensure that variables such as the lighting quality of the image do not mutate the results. Each subpatch of the test image is classified into one of 5 classes- Commercial, Residential Type-1, Residential Type-2, Vegetation, Water. The comparison of results between rules generated by Weka RIPPER and our rule-based classifier are shown in Figure 4.

IV. Experiment and Results

The GMM was built on a single 800 x 800 pixel image generated by merging 16 200 x 200 pixel patches. As an experiment to determine the number of components to use, the GMM was also run on a different 91 x 116 pixel image using various numbers of components and the SSEs were calculated for 4, 5, 6, 7, and 8 components. The lowest SSE was 9.35E06 for 6 components. Therefore we decided to use 6 components on our actual experiment as well. The visual results of running both the scikit-learn implementation of GMM and our implementation of GMM on the entire test image are displayed in Figure 2. Our GMM performed better in distinguishing buildings from concrete than the sklearn implementation. The rest of the results are very similar.

After extraction and labeling of 45 patches, these labeled patches were partitioned into 2 disjoint groups: one group consisted of 25 training patches for the Rule-Based Classifier (5 for each class) and the other group consisted of 20 patches for the hold-out test set (Holdout Method). Both our RBC and the Weka RIPPER achieved the same accuracy: 85%. The confusion matrices for Weka RIPPER and our RBC are displayed in Table I and Table II. Any improvements past this point would be marginal due to the nature of patches encompassing multiple types of regions.

In the final test, the Geotiff Image (Figure 1) is divided into 18172 patches each of size 50 x 50. The GMM+RBC models are then applied to each patch. As the patches are not annotated it is difficult to evaluate the accuracy of our RBC model using a confusion matrix. To overcome this and evaluate our RBC model, we use the output image from our RBC and visually compare its results with the output image from the Weka RIPPER, which is also run on each of the 18172 patches. The running time for getting an output image takes in excess of 4 hours for each of the RBC models (our model and Weka RIPPER).

The results for the evaluation are shown in Figure 4. In a visual comparison of the two outputs, we see that our RBC outperforms Weka RIPPER. Our RBC performed better than Weka Ripper on the following classes as seen in the regions marked yellow in Figure 4: Commercial, Residential Type 1, and Water. In contrast, the Weka Ripper had more success in classifying Vegetation as seen in the region marked purple in Figure 4. Both performed similarly in classifying Residential Type 2.

V. Conclusion & Future Work

Once the pixel proportions for each of the six components have been determined, these may be used as features for the supervised classification problem of determining whether a region type is commercial, residential type 1, residential type 2, vegetation, or water. Therefore, it is certainly possible

that a different type of supervised learner for a multi-class classification problem could have been used such as logistic regression with regularization or SVM. Either of these methods could by itself potentially outperform the rule-based classifier, but in addition to that it is possible to ensemble the results of these new classifiers with the rule-based classifier to generate a result that is more accurate than any single model.

One future consideration for potentially expanding on this process could be to use multispectral images for the input rather than just 3-band RGB images. This could increase the classification accuracy, because the process as is did sometimes struggle to uniquely distinguish between certain components having similar RGB values (eg. interpreting building shadows as water or certain kinds of roofing as ground). These additional embedded features could better equip GMM to make these distinctions.

The GMM performed very well at classifying each pixel of a satellite image into a discrete list of components. While our RBC output was not one hundred percent accurate, no RBC is perfect and the nature of classifying patches of varying contents prevents any system from achieving perfection. We believe this system will quickly and accurately classify satellite images with enough accuracy to allow researchers to spend more time analyzing data rather than processing it.

VI. Acknowledgement

We would like to thank Krishna Karthik Gadiraju who served as an advisor and mentor to our group during this undertaking. Also, we thank Prof. Raju Vatsavai for teaching us all the underlying principles of automated learning and data analysis that provided the necessary foundation for this project.

References

[1] Q. Weng and D. A. Quattrochi, *Urban remote sensing*. CRC Press, 2006.

[2] R. R. Vatsavai, A. Ganguly, V. Chandola, A. Stefanidis, S. Klasky, and S. Shekhar, "Spatiotemporal data mining in the era of big spatial data: algorithms and applications," in *Proceedings of the 1st ACM SIGSPATIAL international workshop on analytics for big geospatial data*. ACM, 2012, pp. 1–10.

[3] A. J. Tatem, A. M. Noor, and S. I. Hay, "Defining approaches to settlement mapping for public health management in kenya using medium spatial resolution satellite imagery," *Remote Sensing of Environment*, vol. 93, no. 1, pp. 42–52, 2004.

[4] "Gaussian Mixture Models," http://scikit-learn.org/stable/modules/mixture.html.

[5] P.-N. Tan, M. Steinbach, and V. Kumar, "Data mining cluster analysis: basic concepts and algorithms," *Introduction to data mining*, 2013.

[6] J. R. Blunden, W. Pryce, and P. Dreyer, "The classification of rural areas in the european context: an exploration of a typology using neural network applications," *Regional Studies*, vol. 32, no. 2, pp. 149–160, 1998.

[7] C. Almeida, J. Gleriani, E. F. Castejon, and B. Soares-Filho, "Using neural networks and cellular automata for modelling intra-urban land-use dynamics," *International Journal of Geographical Information Science*, vol. 22, no. 9, pp. 943–963, 2008.

[8] A. Schneider, "Monitoring land cover change in urban and peri-urban areas using dense time stacks of landsat satellite data and a data mining approach," *Remote Sensing of Environment*, vol. 124, pp. 689–704, 2012.

[9] "Methodology image courtesy of Krishna Karthik Gadiraju."