Multi Temporal Classification Of Satellite Images

Sanket Shahane, Dhananjay Sathe, Pranav Nawathe, Kushal Nawalakha, Ankur Garg
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
svshahan@ncsu.edu, dssathe@ncsu.edu, ppnawath@ncsu.edu, kbnawala@ncsu.edu, agarg12@ncsu.edu

***Abstract*—**The project aims to use the spatio-temporal changes in the Satellite images, over a period of time, to better classify the various parts of an image, for example vegetation, open land, buildings, water bodies etc. Objective of the project is to make use this variation in spectral bands over time for a particular class to improve the prediction of a class for any pixel in the image.

Keywords-component; Multi Temporal Classification; Bayesian Model Averaging; Maximum likelihood classification; Feature selection;

I. INTRODUCTION

Studying spatial changes using multi temporal remotely sensed images has been an active field of research. This type of information has varied uses from monitoring the agricultural land use, natural resource management such as water resource monitoring to study the changes in metropolitan areas in temporal sense [1]. These type of studies, which are generally known as Land Use and Land Cover (LULC) are of great interest to governments to make strategic decisions about the land use. Landsat classification can be used to produce accurate landscape change analysis and statistics [5]. In this study, our focus is on analyzing LULC changes in urban settings, particularly in Bangalore city, India.

Thematic classification is the most widely used technique to analyze spatio-temporal changes on any land. There are numerous classification methods available such as neural networks, support vector machines (SVM), maximum likelihood classification (MLC) etc. We are using MLC as our classification algorithm. But, most of the times, due to changes in reflectance of earth's surface during various seasons and other factors like moisture content, terrain, classification result poses many errors [2]. Hence, it becomes necessary to analyze data from multiple times to improve the classification results as it enables to consider changing spectral signatures of same class over different images.

II. BACKGROUND AND RELATED WORK

Maximum likelihood classification has been in use for analyzing remotely sensed data has been in use for considerable time. Strahler [7] showed that using prior information about the expected class distribution can be used effectively for this type of data. Richards [6] also noted that maximum likelihood classification is the most widely used technique for LULC applications for remotely sensed data.

Most of the time we have multiple models giving us proper descriptions of the distributions of the observed data, in such case it is common practice to choose a model among these based on some criteria, like model fit to the observed database, predictive likelihood or predictive capabilities etc. The selected model is assumed to be a true model. But selecting a model may lead to overconfident inference and riskier decision making, because when we choose a model we ignore the uncertainty in favor of very particular distribution and assumptions on the selected model.

Most desirable way would be to model this source of uncertainty by appropriately selecting or combining multiple models. Edward E. Leamer,1978 [9] suggested using Bayesian inference as a framework capable of achieving this goal.

Bayesian Model Averaging (BMA) is an extension of the usual Bayesian inference methods. In Bayesian Model Averaging we model parameter uncertainty through the prior distribution, uncertainty obtaining posterior parameter and model posteriors using Bayes' theorem Which allows us to do direct model selection, combined estimation and prediction. Bayesian model averaging adds a layer to hierarchical modeling present in Bayesian inference by assuming a prior distribution over the set of all considered models describing the prior uncertainty over each model's capability to accurately describe the data**.**

Raftery, 1996 [10] provided a straightforward approximation for the evidence in generalized linear models, which enabled large no of applications to use BMA. There were also further advancements in implementation of BMA in large model spaces, from a preliminary filtering based on posterior probability ratios called Occam's Window proposed by Madigan and Raftery,1994 [11] to a stochastic search algorithm inspired in the Reversible Chain Markov Chain Monte Carlo proposed by Green, 1995 [12] with trans-dimensional jumps based on posterior model probabilities, the MC3 algorithm proposed by Madigan 1995 [13].

III. NOTATION

Here is the description of the notations that we will use throughout this paper. Labelled training examples are denoted as $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ such that each $\boldsymbol{x}$ is a vector consisting of v components corresponding to v views. As we are concerned about multi-class problem, we denote the class variable by $y \in \{c_1, c_2, \dots, c_k\}$. Also the validation training samples, that are unlabeled are denoted as $\{\dot{\boldsymbol{x}}\}_{i=1}^{u}$.

IV. Methodology

A. Data Collection and Exploration

Landsat 8 ETM 30m X 30m images of Bangalore city, India.

We have identified 4 classes in these views, open land, buildings, vegetation and water bodies by visual identification. We manually extracted 229 points from one image and by visual inspection, tagged classes for each of them. We then created point shape file using QGIS [4] and applied it on all the images to tag corresponding points in them to the same class. Here we assumed that these classes did not change over the time duration these images are captured. Out of these 229 points, we randomly selected 184 points as training data and kept remaining data as accuracy testing data to compute accuracy for all the models (assuming these points can represent test data).

We observed that the probability distribution for all the bands with respect to each class follows Gaussian distribution. Therefore, we could use this fact to estimate likelihood probabilities for building the maximum likelihood classifier. We also observed some bands were not contributing to distinguish between some classes and also a fact that band 8 has very less classification capability.

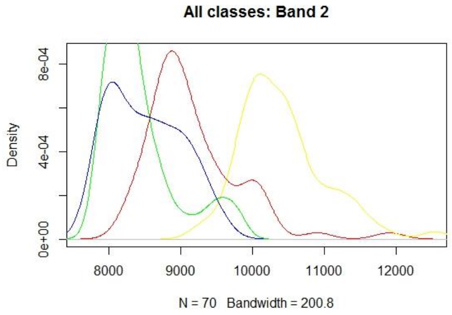
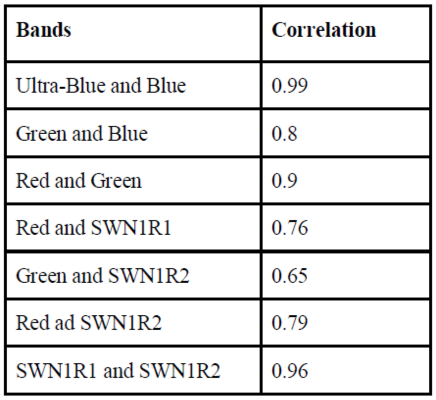


Figure 1: Band 2 Distribution for all classes

To confirm this fact, we calculated correlation between every pair of the bands to check how much dependency they exhibit among each other. Following is the table of correlation values for pairs which are more than 50 % similar.



Bands	Correlation
Ultra-Blue and Blue	0.99
Green and Blue	0.8
Red and Green	0.9
Red and SWN1R1	0.76
Green and SWN1R2	0.65
Red ad SWN1R2	0.79
SWN1R1 and SWN1R2	0.96

Table 1: Correlated features

B. MLC

Maximum likelihood classification is one of the most widely used classification algorithm for classifying remote sensing images for LULC analysis because of its simplicity and efficiency [6]. This is the motivation for us using maximum likelihood classification algorithm as our base classification algorithm.

In Bayes decision theory context, the decision rule adopted by a Maximum Likelihood classifier will be denoted as,

$$x^{*\cdot} \in \ c_k,\ if\ c_k = \arg \overset{max}{c_i \in \gamma} \{P(c_i)P(x^*/c_i)\} \quad (1)$$

where x^* is the unseen point in an image and $P(c_i)$ & $P(x^*/c_i)$ are the estimates of the a priori probability and of the conditional density function of the class c_i [7].

Now, when training data is added for construction of the model, the above equation becomes [1],

$$P(y^* = c_i/x_v^*, \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{l}) \ \propto \ P(c_i)P(\boldsymbol{x_v^*}/y^* = c_i) \quad (2)$$

Here, in x_v^*,v is particularly mentioned because we want to stress the fact that this classification is dependent on that particular image only.

In the above equation, $P(c_i)$ that is priori probability of that class can be calculated by taking the fraction of occurrence of that class in training data set. And, another term, $P(x_1^*/y^* = c_i)$ which is likelihood parameter can be calculated from the probability density function of $\boldsymbol{x_v^*}$ with respect to class c_i.

Generally, for continuous variables the probability density function is described by Gaussian distribution and if there are more than one vector components in one data point, then by Multivariate Gaussian distribution. In our case, we will be using multiple band values per pixel, the density function will be [8],

$$P\left(x = b_1, b_2, \dots, b_q\right) = \frac{e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)}}{\sqrt{(2\pi)\Sigma}} \quad (3)$$

where q is the number of bands in consideration in every pixel.

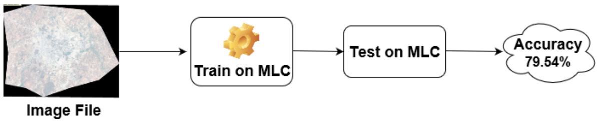


Figure 2: MLC Pipeline

C. *Bayesian Model Averaging.*

BMA is a method to combine the knowledge of two or more same models that have been trained on different datasets. Ensemble approaches are different than BMA in the sense that ensemble approaches combine the results of two or more different models trained on the same dataset. In multi-temporal image classification BMA is a useful method when no two models can correctly separate out all the classes. Consider the case: We have 4 classes; A, B, C, and D. In one of the images Classes A and B are separable and Class C and D are not. Using only one image yields poor results. However, an image of the same region taken at different time may be able to separate out Classes C and D but not A and B. Therefore, using any of the models built on individual image will yield poor results. BMA benefits by combining the knowledge of both the models and improving the overall accuracy per class. Approach for BMA is the probabilities given by every model are multiplied by their respective weights and then averaged out over all the models. Finally, the class for which the maximum probability has been assigned is outputted as the final class label.

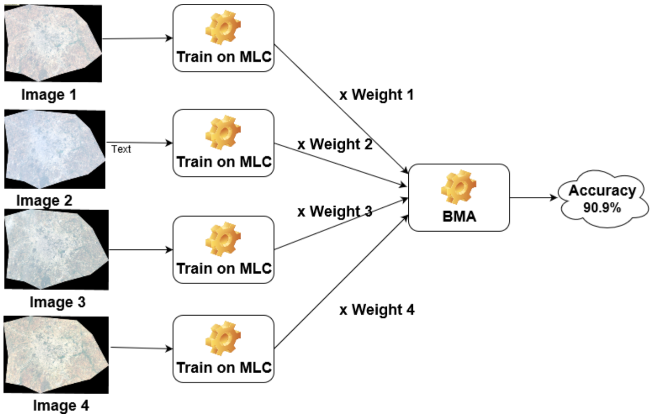


Figure 3: BMA pipeline

Method for weight calculation:

1. Fit a model for each image you plan to use in BMA.
2. weight = 1
3. For each value in the training data:
    a. Get the raw predictions vector given by the model { p[nClasses] }
    b. Observe the true class {i}; i can be the class number
    c. weight = weight * p[i]
4. Associate value in weight with that model.

D. *Feature Selection*

Using only a subset of all available features has the possibility of improving the generalization error of the model. The satellite images that have been used in the project contain 8 bands in total, as mentioned in the data collection and exploration section. Using exploratory data analysis, it was observed that some of those 8 features were not much useful as the distribution of some of these features was such that it would not be able to differentiate between the various classes in our data. We had also observed the presence of some highly-correlated features (correlation value > 0.8). Another reason that motivated us to explore feature selection was the fact that, the maximum likelihood classifier fits a multi-variate Gaussian distribution on the data. In our case, we didn’t have a lot of training samples. In the absence of large amounts of data, fitting a multi-variate Gaussian distribution leads to an unstable covariance matrix.

These observations led us explore feature selection which could possibly improve the accuracy of the model. Since the number of features was not very high, we decided to use Best Subset Selection to identify a subset of the 8 features which would perform best on validation data set. This wasn’t computationally expensive as there were only 2^8 = 256 possible subsets to be checked. We observed that, using only a subset of features instead of the complete 8, lead to significant improvement in the test accuracy. The detailed results are presented in the results section.

E. *Dimensionality Reduction*

As mentioned in the previous section, we had limited number of training samples. Fitting a Gaussian distribution on a small size data could lead to an unstable covariance matrix. To improve the Gaussian fit, reducing dimensionality was important.

For reducing dimensionality, we used Principal Component Analysis (PCA). We were able to reduce the dimensions from 8 to 4 and still retaining around 97% of the variance. We observed that using these components instead of the original 8 features helped significantly improve the accuracy. The results for the test accuracy of each maximum likelihood classifier on these principal components has been detailed in results section.

V. IMPLEMENTATION ENHANCEMENTS

A. *Vectorization of Maximum Likelihood Classifier*

Prediction using maximum likelihood classifier requires the 2 matrix multiplications for each test sample. For each test sample, transpose of the difference from the mean is multiplied by the covariance matrix which is then multiplied by the difference of the test sample from the mean.

This operation must be performed for each test sample. For classification of the complete LANDSAT image, it would involve a lot of matrix multiplications. To reduce the time required for prediction, we implemented the MLC prediction using python numpy vector operations for matrix multiplications. This helps calculate prediction probabilities for multiple samples using only one set of matrix multiplications. This helped improve runtime for the classification of the complete LANDSAT image.

B. Multithreading for faster classification

We Started using single thread, the time taken was 2143 secs for predicting the class for each pixel and generating its respective image, but each pixel value can be treated independent, Even with multithreading we gave each thread the same array for predicting its class, the problem with this was locking on array due to which we were not able to reduce the overall time to a great extent due to locking on the common data, so we separated the data, and each thread now has its own data each thread can work independently on the given data, with 2 threads the time taken reduced by ~70%. The reason for not achieving full scalability is python global interpreter lock, which must be held before accessing objects or call API functions.

VI. EXPERIMENTS AND RESULTS

A. MLC

On our initial image we got 79.54% accuracy, which was less than satisfactory. Hence, we decided to use multitemporal images for achieving better accuracy. Below are the results of MLC on all the images.

Image Used	MLC
Image 1	88.63%
Image 2	79.54%
Image 3	86.36%
Image 4	95.45%

Table 2: MLC Results

B. BMA

While calculating the weights for the models we faced a problem of precision error. To handle it we used log of the probabilities and converted the multiplication to summation. However, another challenge we faced was log(0) is -inf and the weights could not be calculated. To solve this, we rescaled the values to [1,2]. Result of BMA with above models was 90.90%

C. PCA

Using Principal component analysis, only the top 4 principal components were used to train maximum likelihood classifier instead of the complete 4. The results obtained improved for some images. The test accuracies are mentioned in the table below:

Image Used	MLC
Image 1	88.63%
Image 2	81.82%
Image 3	90.91%
Image 4	90.91%
BMA	90.90%

Table 3: PCA Results

D. Best Feature Subset

Best feature subset selection was used to find a subset of features that would perform best on the validation dataset. MLC models were trained on these features. Results are as mentioned below, along with the best feature subset obtained.

Image Used	Features Selected	MLC
Image 1	[Ultra_Blue, Blue, NIR, SWNIR_2, Cirrus]	93.18%
Image 2	[Ultra_Blue, Blue, Green, SWNIR_1, Cirrus]	88.84%
Image 3	[Ultra_Blue, SWNIR_2]	93.18%
Image 4	[Ultra_Blue, Green, Red, NIR, SWNIR_1, SWNIR_2, Cirrus]	97.72%
BMA		93.18%

Table 4: Best feature subset Results

E. Multithreading for faster classification.

Results for multithreaded Classification for complete image. Expected scalable time and actual time taken by the implementation.

Threads	Actual time (sec)	Expected time (sec)
1	2143	2143
2	1493	1071
4	1329	535
8	1107	267
16	800	133
32	530	67
64	339	34

Table 5: Multithreading Results

And the graph for the results are as follows:

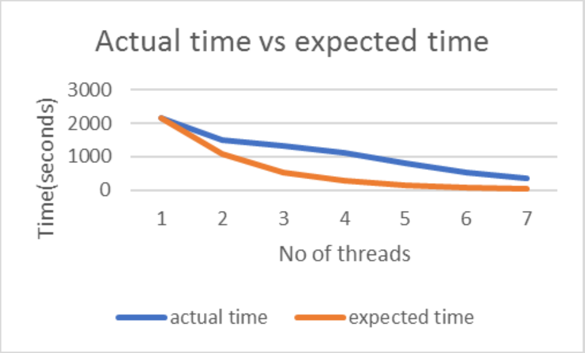


Figure 4: Multithreading actual vs expected time

VII. Conclusion and Future scope

Bayesian Model Averaging is a good technique for cases when you have a set of classifiers which are not able to separate out all the classes but perform well separating a subset of the classes. The data should have constant class labels across all the temporal images. We expect that BMA should improve results over individual models when all the classifiers have a similar range of accuracies. However, when there is a significant difference between the individual models it is better to go with the best individual model. This is evident from our experiments with individual models, and combining knowledge learnt by models using BMA. In our case, we have a model which is giving 97% accuracy and others around 90% and BMA gives 93.18%. Moreover, dimensionality reduction and feature subset selection helped in significantly improving the accuracy.

A. *Change detection*

Before assigning final class label by BMA, if the posterior probability of winner class is not sufficiently larger than the next highest class probability, then we can say that both the models are quite confident about their assignments and we should not directly assign the class label, it actually represents there is a possible change in class in the corresponding time period. We can use this technique to detect class change over the time.

VIII. Acknowledgement

We would like to acknowledge Krishna Karthik Gadiraju for his support. We would also like to thank Dr. Ranga Raju Vatsavai for guidance.

References

[1] Varun Chandola and Ranga Raju Vatsavai , “Multi-Temporal Remote Sensing Image Classification – A Multi-View Approach”.

[2] Bruzzone, L., and D. F. Prieto. “Unsupervised Retraining of a Maximum Likelihood Classifier for the Analysis of Multitemporal Remote Sensing Images.” *IEEE Transactions on Geoscience and Remote Sensing* 39, no. 2 (February 2001): 456–60. doi:10.1109/36.905255.

[3] Eason, G., B. Noble, and I. N. Sneddon. “On Certain Integrals of Lipschitz-Hankel Type Involving Products of Bessel Functions.” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 247, no. 935 (April 19, 1955): 529–51. doi:10.1098/rsta.1955.0005.

[4] Gorunescu, Florin. “Introduction to Data Mining.” In *Data Mining*, by Florin Gorunescu, 1–43. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-19721-5_1.

[5] “Introduction to Data Mining - Springer.” https://link.springer.com/chapter/10.1007%2F978-3-642-19721-5_1.

[6] “Remote Sensing Digital Image Analysis - Springer.” https://link.springer.com/book/10.1007/978-3-642-30062-2.

[7] Strahler, Alan H. “The Use of Prior Probabilities in Maximum Likelihood Classification of Remotely Sensed Data.” *Remote Sensing of Environment* 10, no. 2 (September 1, 1980): 135–63. doi:10.1016/0034-4257(80)90011-5.

[8] Yuan, Fei, Kali E. Sawaya, Brian C. Loeffelholz, and Marvin E. Bauer. “Land Cover Classification and Change Analysis of the Twin Cities (Minnesota) Metropolitan Area by Multitemporal Landsat Remote Sensing.” *Remote Sensing of Environment* 98, no. 2–3 (October 15, 2005): 317–28. doi:10.1016/j.rse.2005.08.006.

[9] Leamer, E. E. (1978). Specification searches, New York: Wiley.

[10] Raftery, A. E. (1996). Approximate bayes factors and accounting for model uncertainty in generalised linear models, Biometrika 83(2): 251–266.

[11] Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using occam’s window, Journal of the American Statistical Association 89(428): 1535–1546.

[12] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination, Biometrika 82(4): 711–732.

[13] Madigan, D., York, J. and Allard, D. (1995). Bayesian graphical models for discrete data, International Statistical Review/Revue Internationale de Statistique pp. 215–232.