“Soft Classification Approaches for Identification of Agricultural Crops: Problems & Prospects using Indian Remote Sensing Data”

By

Anand N. Khobragade
Resources Scientist
Maharashtra Remote Sensing Applications Centre, Nagpur, Maharashtra State, India

April 3, 2017
State Agriculture Department plays vital role on predicting forecast for crop production & acreage estimation in Maharashtra.

Commodity market estimates crop production on the basis of crop mass arrival in market and field prediction from their sources like Crop Advisory Boards.

The timely and accurate acreage estimation of crop is the prime requisite for the purpose of better management upon crop production estimation.

The crop acreage statistics proves more crucial in event of natural calamity for taking strategic decisions like compensations to farmers based on losses they put up with.

In a nutshell, non-availability of accurate and finely estimated forecast necessitates the formation of coherent policy on fixing up agricultural commodity prices.

For this reason, Ministry of Agriculture and Cooperation, Government of India (GOI) initiated project known as FASAL (Forecasting of Agriculture outputs through Satellite, Agrometeorology and Land based observations), which will make extensive use of state of the art Remote Sensing and GIS technologies for achieving this mandate.

The State Remote Sensing Applications Centers and State Agriculture Department are PIA (project Implementation Agency) and hence MRSAC entrusted this work.
SAC, Ahmadabad used multi-date single sensor remote sensing data for estimating specific crop, e.g. RADARSAT- Rice, RESOURCESAT- Cotton & Sugarcane.

It is obvious that MRSAC were following methodology/model proposed by Space Application Centre (SAC, DOS), Ahmadabad, which still using manual mapping procedures for crop acreage estimation and hence prone to errors so often.

Multi-date single sensor data used for identification of single crop, whereas no estimation provided due to non-availability of optical data in monsoon for kharip crops.

Besides manual errors, SAC model still uses pure statistical Maximum Likelihood Classifier and NDVI algorithm for estimating crop production. In brief, hard classification approach was adapted under this model.

Lack of applicability of any advance sub-pixel classification as well as learning based classifier, appeal me to look for improvement in this implementation.

Apart, It’s obvious that crop estimates from Crop Advisory Boards and Government are often remains unmatched due to non-qualitative and unreliable approach.

The conventional methods of gathering information on crop acreage are cumbersome, costly, and protracted, especially when the extent of work is entire Maharashtra State.
Remote sensed data is an ideal source for mapping land cover and land uses at a variety of spatial and temporal scales.

Due to its multi-spectral nature and repetitive coverage, remote sensing data is the most suitable for rapid assessment of crops.

Consequently, a Remote Sensing technology has potential in estimating crop acreage at Regional/Command/District/Taluka/Village level due to its multi-spatial and multi-temporal nature.

By now, very few attempts were made towards incorporating contextual information into this knowledge engine. Its really a matter of investigation that how model behave with the fusion of contextual information in this soft classification approach.

Few Indian researchers started research on statistical learning method (like SVM) based sub pixel classification for land cover, few of them may be working in Agriculture domain too.

It was indeed motivation to extend his research by intermingling contextual sensitivity and other evolution theory concepts,

This research is worth towards serving mandate of GoM, in return Government of India.
Conventional Classification

Sub-Pixel Classification

Statistical based Soft Classification

Learning based Soft Classification

Statistical Learning based Soft Classification

Multi-Source Approach

Contextual Statistical Learning based Soft Classification

R-Tools

Big data – R HADOOP

Evolutionary Algorithms

GPU Approach

THE CONCEPTUAL JOURNEY – TOWARDS PROBLEM SOLUTIONS
Observations:

1. The finding of following research illustrate that out of three approaches using SVM, PCM and FCM when evaluated with respect to a fuzzy weighted matrix, SVM yields the best accuracy.

<table>
<thead>
<tr>
<th>User's Accuracy (%)</th>
<th>SVM</th>
<th>PCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>99.2</td>
<td>99.6</td>
</tr>
<tr>
<td>Barren Land</td>
<td>75.2</td>
<td>73.8</td>
</tr>
<tr>
<td>Forest</td>
<td>98.4</td>
<td>98.9</td>
</tr>
<tr>
<td>Sand</td>
<td>91.5</td>
<td>82.9</td>
</tr>
<tr>
<td>Water</td>
<td>74.4</td>
<td>82.9</td>
</tr>
</tbody>
</table>

2. From the studies, it has been found that SVM classification with Diagonal Norm is most suitable for Agriculture applications, whereas Mahalobonis algorithm performs poor due to the high dimensionality of satellite data.

<table>
<thead>
<tr>
<th>User's Accuracy (%)</th>
<th>SVM - Euclidean Norm</th>
<th>SVM - Diagonal Norm</th>
<th>SVM - Mahalanobis Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>88.4</td>
<td>100.0</td>
<td>76.1</td>
</tr>
<tr>
<td>Fallow Land</td>
<td>99.8</td>
<td>100.0</td>
<td>79.3</td>
</tr>
<tr>
<td>Forest</td>
<td>99.8</td>
<td>100.0</td>
<td>97.3</td>
</tr>
<tr>
<td>Sand</td>
<td>96.7</td>
<td>66.2</td>
<td>89.2</td>
</tr>
<tr>
<td>Water</td>
<td>99.9</td>
<td>84.9</td>
<td>77.4</td>
</tr>
</tbody>
</table>
• Supervised Classification Algorithm
• Binary Classifier
• Non-parametric Statistical Learning Based Algorithm.
• Overcomes drawbacks of traditional classification algorithms
  - Poor support for high dimensionality by Statistical Classifiers
  - Slow Computations by Neural Networks/Fuzzy Classifiers
  - Poor generalization by ANN classifiers.
• Features of SVM are
  - High Computational Efficiency
  - Robust in High Dimensionality
  - Good in Generalization and hence works well with small training data
  - Controls accuracy Vs complexity in function estimation
  - Superior performance in classifying hyperspectral images.

Conclusion!!!

SVM is th best classification method as compare to others.
Observations:

1. For problems with more than two classes, where a classifier is typically constructed by combining several binary SVMs, most researchers simply select all binary SVM models simultaneously in one hyper-parameter space.

2. Though this all-in-one method works well, there is another choice – the one-in-one method where each binary SVM model is selected independently.

3. As per literature survey, pair-wise coupling (PLC) multi-class approaches works in two steps: First the original pairwise probabilities are converted into a new set of pairwise probabilities, then pairwise coupling is employed to construct the global posterior probabilities. Experimental results show that PLC algorithm is effective and efficient.

4. It is learnt that fuzzy based multi-class approach is also adapted for converting Binary SVM problem into multi-class problem.
A Comparison of Methods for Multiclass Support Vector Machines
Chih-Wei Hsu and Chih-Jen Lin

TABLE IV
A COMPARISON USING THE LINEAR KERNEL (BEST RATES BOLD-FACED)

<table>
<thead>
<tr>
<th>Problem</th>
<th>One-against-one</th>
<th>DAG</th>
<th>One-against-all</th>
<th>[25], [27]</th>
<th>C&amp;S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
<td>rate</td>
<td>C</td>
<td>rate</td>
<td>C</td>
</tr>
<tr>
<td>iris</td>
<td>2^4</td>
<td>97.333</td>
<td>2^8</td>
<td>97.333</td>
<td>2^12</td>
</tr>
<tr>
<td>wine</td>
<td>2^-2</td>
<td>99.438</td>
<td>2^-2</td>
<td>98.315</td>
<td>2^-1</td>
</tr>
<tr>
<td>glass</td>
<td>2^8</td>
<td>66.355</td>
<td>2^4</td>
<td>63.551</td>
<td>2^-2</td>
</tr>
<tr>
<td>vowel</td>
<td>2^5</td>
<td>82.954</td>
<td>2^4</td>
<td>81.439</td>
<td>2^11</td>
</tr>
<tr>
<td>vehicle</td>
<td>2^12</td>
<td>80.615</td>
<td>2^5</td>
<td>80.851</td>
<td>2^12</td>
</tr>
<tr>
<td>segment</td>
<td>2^12</td>
<td>96.017</td>
<td>2^11</td>
<td>95.844</td>
<td>2^12</td>
</tr>
</tbody>
</table>

TABLE II
A COMPARISON USING THE RBF KERNEL (BEST RATES BOLD-FACED)

<table>
<thead>
<tr>
<th>Problem</th>
<th>One-against-one</th>
<th>DAG</th>
<th>One-against-all</th>
<th>[25], [27]</th>
<th>C&amp;S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(C, γ)</td>
<td>rate</td>
<td>(C, γ)</td>
<td>rate</td>
<td>(C, γ)</td>
</tr>
<tr>
<td>iris</td>
<td>(2^12, 2^-9)</td>
<td>97.333</td>
<td>(2^12, 2^-8)</td>
<td>96.667</td>
<td>(2^9, 2^-3)</td>
</tr>
<tr>
<td>wine</td>
<td>(2^-7, 2^-10)</td>
<td>99.438</td>
<td>(2^-6, 2^-9)</td>
<td>98.876</td>
<td>(2^-7, 2^-6)</td>
</tr>
<tr>
<td>glass</td>
<td>(2^11, 2^-3)</td>
<td>71.495</td>
<td>(2^12, 2^-3)</td>
<td>73.832</td>
<td>(2^11, 2^-2)</td>
</tr>
<tr>
<td>vowel</td>
<td>(2^4, 2^0)</td>
<td>99.053</td>
<td>(2^2, 2^2)</td>
<td>98.674</td>
<td>(2^-4, 2^1)</td>
</tr>
<tr>
<td>vehicle</td>
<td>(2^-9, 2^-3)</td>
<td>80.615</td>
<td>(2^11, 2^-3)</td>
<td>86.052</td>
<td>(2^11, 2^-4)</td>
</tr>
<tr>
<td>segment</td>
<td>(2^-6, 2^0)</td>
<td>97.403</td>
<td>(2^11, 2^-3)</td>
<td>97.359</td>
<td>(2^-7, 2^-6)</td>
</tr>
<tr>
<td>dna</td>
<td>(2^-3, 2^-6)</td>
<td>95.447</td>
<td>(2^3, 2^-6)</td>
<td>95.447</td>
<td>(2^2, 2^-6)</td>
</tr>
<tr>
<td>satimage</td>
<td>(2^4, 2^0)</td>
<td>91.3</td>
<td>(2^4, 2^0)</td>
<td>91.25</td>
<td>(2^2, 2^-1)</td>
</tr>
<tr>
<td>letter</td>
<td>(2^4, 2^2)</td>
<td>97.98</td>
<td>(2^4, 2^2)</td>
<td>97.98</td>
<td>(2^3, 2^-2)</td>
</tr>
<tr>
<td>shuttle</td>
<td>(2^-1, 2^3)</td>
<td>99.924</td>
<td>(2^-1, 2^3)</td>
<td>99.924</td>
<td>(2^2, 2^-4)</td>
</tr>
</tbody>
</table>
**Contextual Classifiers**

As per literature survey, following are the popular contextual classifiers.

- **Markov Random Field (MRF classifiers)**
- Discriminative Random Field (DRF classifiers)
- **Spatial AdaBoost classifier**
- Spatial Auto-regression (SAR Classifiers)
- Expectation Maximization (EM classifier)
- Maximum a Posterior (MAP classifier)

**Observations:**

1. Out of the above, Spatial AdaBoost and MRF-based classifier shows an excellent performance.

2. Markov random fields (MRF) is a popular model for incorporating spatial class dependencies (spatial context) between neighboring pixels in an image, and temporal class dependencies between different images of the same scene.

3. AdaBoost shows an excellent performance similar to that of the MRF-based classifier with much less computational cost.
**Study Area**

**Process Flow**

- **Multispectral/PAN Data**
- **Microwave Data**

1. **Geo-rectification of Data**
2. **Creation of Training Data**
3. **Classification (Soft/Hard)**
4. **Multi-Source Approach**
5. **Accuracy Assessment**

**Experiments:**
1. Identification of unambiguous training samples for specific crop.
2. Investigate parametric/non-parametric soft computing approach using multi-source and multi-temporal data.
3. Evaluate effect of contextual parameter on finally classified image as a result of multi-source approach.

Finally, outcomes of this research could be validated using Land Suitability Model of National Bureau Soil Survey & Land Use Planning, (NBSS & LUP, GOI). Thus, one can crosscheck the feasibility of remote sensing results with ground condition of the area under study.
Data Preparation for Experimentations

ISSUES AND SOLUTIONS

• Considering my problem definition, it is needed to check generalization nature of Image Processing Algorithms
• Hence, input data is been prepared to evaluate algorithms from lower to higher dimensionality of remote sensing data.
• Total 15 Data Sets were prepared for addressing issues related with Multi-Sensory data and Multi-Resolution data.

ABOUT MULTI-SENSORY DATA

RESOURCESAT Satellite – LISS IV sensor (5.8 m Resolution)
CARTOSAT Satellite – CARTOSAT-I sensor (2.5 m Resolution)
Ortho-rectified Merged Product of LISS IV MX + CARTOSAT-I

Data Set #1 – Input Image for 1-Class Problem
Data Set #2 – Input Image for 2-Class Problem
Data Set #3 – Input Image for 3-Class Problem
Data Set #4 – Input Image for 4-Class Problem
Data Set #5 – Input Image for Multi-Class Problem
Orthorectfied LISS IV
Orthorectfied CARTOSAT-I
Omerged Product (L+C)
Data Samples for 1-Class Problem

Ortho-rectified LISS IV

Ortho-rectified CARTOSAT-I

Merged Product LISSIV MX + CARTOSAT
Data Samples for 2-Class Problem

Ortho-rectified LISS IV

Ortho-rectified CARTOSAT-I

Merged Product (LISSIV MX + CARTOSAT)
Data Samples for 3-Class Problem

Merged Product (LISSIV MX + CARTOSAT)

Ortho-rectified LISS IV

Ortho-rectified CARTOSAT-I
METHODOLOGY – Analytical Approach (Phase-II)

Data Samples for Multi-Class Problem

Ortho-rectified CARTOSAT-I

Merged Product LISSIV+CARTOSAT

Ortho-rectified L4
EXPERIMENTAL RESULTS
## Experimentations Outcomes: 1-Class LISS IV

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ISODATA</th>
<th>Minimum Distance</th>
<th>Parallelepiped</th>
<th>Maximum Likelihood</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>80 %</td>
<td>86.67 %</td>
<td>100 %</td>
<td>100 %</td>
<td>100 %</td>
</tr>
<tr>
<td><strong>Kappa Coefficient</strong></td>
<td>0.6538</td>
<td>0.7273</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### METHODOLOGY – Analytical Approach (Phase-II)
Experimentations Outcomes: 1-Class CARTOSAT

**Algorithm**

- **Minimum Distance**
  - Accuracy: 100%
  - Kappa Coefficient: 1.0

- **Parallelepiped**
  - Accuracy: 100%
  - Kappa Coefficient: 1.0

- **Maximum Likelihood**
  - Accuracy: 53.19%
  - Kappa Coefficient: 0.3517

- **K-means**
  - Accuracy: 80.92%
  - Kappa Coefficient: 0.6532

- **SVM**
  - Accuracy: 100%
  - Kappa Coefficient: 1.0
## Experimentations Outcomes: 1- Class Merged Product

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ISODATA</th>
<th>Minimum Distance</th>
<th>Parallele-piped</th>
<th>K-means</th>
<th>Maximum Likelihood</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>80 %</td>
<td>93.33%</td>
<td>80.00%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Kappa Coefficient</td>
<td>0.6538</td>
<td>0.8421</td>
<td>0.6591</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Kappa Accuracies of Individual Classes - Taluka level image with ground truth

**Agriculture Land Class**

- Total Ground Reference Point : 36
- Total reference check points : 150
- Area Calculated : 861.66 sq. km.
- Overall Accuracy : 59.14 %
- Kappa Accuracy : 38.62 %

**Wasteland Land Class**

- Total Ground Reference Point : 88
- Total reference check points : 150
- Area Calculated : 441.52 sq. km.
- Overall Accuracy : 76.54 %
- Kappa Accuracy : 44.54 %

**Kappa Coefficient:**
It is coefficient of Agreement as a Measure of Map Accuracy. i.e. it is a measure between interpretation and classification.

\[
k = \frac{\theta_1 - \theta_2}{1 - \theta_2}
\]

\[
\theta_1 = \frac{\sum x_{ii}}{N}, \text{ where } i = 1 \text{ to } r
\]

\[
\theta_2 = \frac{\sum x_i + x + i}{N^2}, \text{ where } i = 1 \text{ to } r
\]

Where,
- \( r \) is the number of rows in the error matrix
- \( x_{ii} \) is the \( i \)th diagonal element
- \( x_i \) is the marginal total of row \( i \)
- \( x_{i} \) is the marginal total of column \( i \)
- \( N \) is the total number of observations

Individual class accuracy = \( \frac{\text{No of polygons correctly interpreted as that class}}{\text{No of polygons checked in the field for the same class}} \) X 100
Results: LISS IV Multi-Class Classification for complete image

Even after keeping my machine running for apx. 12 hours continuously from 10pm to 10 am.

Need to check all possible reasons for implementing SVM on whole image...

Concepts of data scaling need to study in deep.
LISS IV
11 Dec 2013

CARTOSAT
5 Feb 2014
NRSC, Hyderabad provides FTP link through Email
User Name & Password is also provided for download data
Generates Invoice for the same.

Procedure Followed
- Get lat/long for AOI
- Check availability on NRSC website
- Note down all scenes names falling into AOI
- Fill up NRSC Data Order Form
- Deposit money via DD
- Get FTP link for download satellite data
- Get Data with Invoice.
Field Survey was conducted on 3rd March, 2014 using GPS enabled digital camera.
MAPPING LOCATIONS ON SATELLITE IMAGE

Legend
- Field_GPS_Locations
- Nagpur_Villages

L4_Satellite_Image
RGB
- Red:  Band_3
- Green: Band_2
- Blue:  Band_1

GPS field locations of the crop fields in and around SAONER Taluka as Georeference on Satellite Image with Village Boundaries of Nagpur District
GPS field locations of the crop fields in and around SAVNER Taluka as Georeference on Satellite Image with Village Boundaries

ZOOMED VIEW
MAPPING LOCATIONS ON SATELLITE IMAGE

GPS field locations of the crop fields
Showing clear Village Boundaries

ZOOMED VIEW
MAPPING LOCATIONS ON SATELLITE IMAGE

ZOOMED VIEW

COTTON

GRAM

COTTON
WHEAT

SUGARCANE

ORCHARDS

COTTON
WHEAT

GRAM
TOMMATO

ZOOMED VIEW
Selection of training sets

Cotton

Gram

Wheat
# MATLAB IMPLEMENTATION FOR IMAGE CLASSIFICATION

<table>
<thead>
<tr>
<th>Input Images</th>
<th>Original Test Image 1</th>
<th>SVM Classification Output</th>
<th>Optimized SVM Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Site 1</td>
<td><img src="image1.png" alt="Image 1" /></td>
<td><img src="output1.png" alt="Output" /></td>
<td><img src="optimized1.png" alt="Optimized Output" /></td>
</tr>
<tr>
<td>Test Site 2</td>
<td><img src="image2.png" alt="Image 2" /></td>
<td><img src="output2.png" alt="Output" /></td>
<td><img src="optimized2.png" alt="Optimized Output" /></td>
</tr>
</tbody>
</table>
## MATLAB IMPLEMENTATION FOR IMAGE CLASSIFICATION

### SVM Classification Accuracies

<table>
<thead>
<tr>
<th>Crop Types</th>
<th>Kappa Coefficient</th>
<th>Overall Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cotton</td>
<td>0.607</td>
<td>90%</td>
</tr>
<tr>
<td>Gram</td>
<td>0.6153</td>
<td>80%</td>
</tr>
</tbody>
</table>

### Optimized SVM Classification Accuracies

<table>
<thead>
<tr>
<th>Crop Types</th>
<th>Kappa Coefficient</th>
<th>Overall Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cotton</td>
<td>0.607</td>
<td>90%</td>
</tr>
<tr>
<td>Gram</td>
<td>0.6153</td>
<td>80%</td>
</tr>
</tbody>
</table>

### Time Taken

<table>
<thead>
<tr>
<th>Classification Algorithm</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>1.9694 Sec</td>
</tr>
<tr>
<td>Optimized SVM</td>
<td>2.2527 Sec</td>
</tr>
</tbody>
</table>
Over to MATLAB implementation (New)
## NEW MATLAB IMPLEMENTATION : RESULTS

<table>
<thead>
<tr>
<th>MATLAB</th>
<th>Training Image</th>
<th>Classification Output</th>
<th>SVM Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images Type</td>
<td><img src="image1.png" alt="Training Image" /></td>
<td><img src="image2.png" alt="Classification Output" /></td>
<td>Linear Kernel</td>
</tr>
<tr>
<td>Time taken</td>
<td>16sec</td>
<td>52sec</td>
<td>Linear Kernel</td>
</tr>
</tbody>
</table>
JAVA (OWN) IMPLEMENTATION : SVM CLASSIFICATION

Front End GUI
SUBSETTING IMAGES w.r.t. VILLAGE BOUNDARIES FOR TRAINING & TESTING

Satellite Images as cropped w.r.t. villages in Savner Taluka
Front-end GUI with Color Selection for training samples
Select features (RGB) from CSV file that prepared from input train images
JAVA (OWN) IMPLEMENTATION : SVM CLASSIFICATION

Press TEST button, confusion matrix shown in table & time taken is displayed

Classification Done in 15312ms
JAVA (OWN) IMPLEMENTATION: SVM CLASSIFICATION

8-bit Input image
Of Borgaon village
With training
Sites for selections of features

Borgaon village

Total 30 Training Samples are used
MATLAB IMPLEMENTATION FOR IMAGE CLASSIFICATION

Dataset Generator Tool

Testing in progress

Classified Output

GUI
MATLAB IMPLEMENTATION FOR IMAGE CLASSIFICATION

- Class 0: User Accuracy 63, Producer Accuracy 91.4%, Kappa Accuracy 0.9
- Class 1: User Accuracy 707, Producer Accuracy 0.9, Kappa Accuracy 0.9
- Total: User Accuracy 770, Producer Accuracy 91.4%, Kappa Accuracy 0.9

Algorithm: SVM
Kernel: LINEAR
C: 30
Degree: 0.0
Gamma: 0.0
Coef0: 0

Classification Done
**Total Geographic Area**: 605.55 ha

**Total Irrigated Area**: 204.85 ha

**Total Cotton sown Area**: 85 ha

Area estimated from

**SVM(Linear)**: 136.06 ha

Area estimated from

**SVM(RBF)**: 141 ha
<table>
<thead>
<tr>
<th>JAVA (Own)</th>
<th>Classified Output</th>
<th>SVM Training</th>
<th>SVM Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Kernel</td>
<td><img src="image1.png" alt="Image" /></td>
<td>0.386sec But MATLAB took 16sec</td>
<td>28sec But MATLAB took 52sec</td>
</tr>
<tr>
<td>RBF Kernel</td>
<td><img src="image2.png" alt="Image" /></td>
<td>0.846sec</td>
<td>1132sec</td>
</tr>
</tbody>
</table>
## RABBI CROPS: YEAR 2013-2014

<table>
<thead>
<tr>
<th>District</th>
<th>Year</th>
<th>Wheat</th>
<th>Gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>District 1</td>
<td>2013</td>
<td>10000</td>
<td>5000</td>
</tr>
<tr>
<td>District 2</td>
<td>2014</td>
<td>15000</td>
<td>7000</td>
</tr>
</tbody>
</table>

**Target:** Wheat & Gram Crop
## Kharip Crops: Year 2013-2014

### Government Records for Validations: Accuracy Assessment

The table below provides the data for Kharip crops in the year 2013-2014. The columns include various parameters such as TGA, Yield, and more. The data is used for the accuracy assessment of government records.

<table>
<thead>
<tr>
<th>T.A.</th>
<th>TGA</th>
<th>...</th>
<th>Yield</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Target: Cotton Crop
Points to Ponder

- On my request, Director, MRSAC wrote to General Manager, RRSC, ISRO
- Asking for help from ISRO Scientist for processing RISAT data
- This coordination will be extended for 1 month
- ISRO Scientist will guide on in-depth parameter processing of RISAT data.
- On based upon this knowledge, it is easy to write SVM implementation for RISAT data also.

1. We can perform analysis on agriculture area as its MRS images doesn’t support for fine resolutions.
**VALUE ADDITION TO SVM**

**Diagnosis for improvement**

- **Known features selection from Input Image** $(X : x_1, x_2, x_3)$
- **Training of Image** $(X, y, C, @\text{linear Kernel})$
- **Generation of Trained Model with optimum parameters**
- **Testing of Image** (model, $X$)
- **Classified Output Image** ($Y$)

**Scope for selecting more features i.e. contextual info**

**Scope for using any other optimization algo (may be EA)**

**SMO for Training**

**Freedom of designing our own kernel Function**

However, there are 42 types of kernels available for different applications.

% Save the model
idx = alphas > 0;
model.X= X(idx,:);
model.y= Y(idx);
model.kernelFunction = kernelFunction;
model.b= b;
model.alphas= alphas(idx);
model.w = ((alphas.*Y)'*X)';
## Important Publications:

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Year</th>
<th>Published by</th>
<th>Description/Title</th>
</tr>
</thead>
</table>
CONCLUSION & FUTURE SCOPE

Vegetation Indices may be tested on SVM:
- NDVI (Normalized Differential Vegetation Index)
- TVI (Transformed Vegetation Index)
- SaVI (Soil-adjusted Vegetation Index)
- MSaVI (modified soil-adjusted vegetation index)

Use of Contextual Classifiers:
- Markov Random Field (MRF classifiers)
- Discriminative Random Field (DRF classifiers)
- Spatial AdaBoost classifier
- Spatial Auto-regression (SAR Classifiers)
- Expectation Maximization (EM classifier)
- Maximum a Posterior (MAP classifier)

Development of own kernel for SVM:
1. Study existing kernel & compare their performance specific to our problem.
2. Try to design own suitable kernel based on problem requirement (hint: color space as kernel space)

Optimization of SVM using EA (GA/DE):
1. If used GA, then try GA variants - ZCS, LCS, & XCS.
2. If used DE, then try for SaDE or NSADE
3. Both GA/DE may be used for
   a. Find best discriminant features
   b. Find optimal instances (optimal training selection)
   c. Find best parameter values for SVM

Big data implementation:
1. Considering limitation of existing image processing software, we need algorithm which can handled processing big size satellite image with less time.
2. Need to write module for Big data implementation for satellite image classification problem, which may improve SVM performance (i.e. parallel processing)
3. May design hardware implementation of SVM
   a. By writing m/c code (driver) for SVM hardware
   b. Designing h/w like GPU or Reusing GPU for svm computation
Overall SVM Implementation on Remote Sensing Images

**Known features selection from Input Image** ($X : x_1, x_2, x_3$)

**Training of Image** ($X, y, C, @\text{linear Kernel}$)

**Generation of Trained Model** with optimum parameters

**Testing of Image** (model, $X$)

**Classified Output Image** ($Y$)

Adding more features for better accuracies...

SMO for Training

Scope for optimization of SVM parameters

Scope for selecting more relevant features i.e. contextual info

Freedom of designing our own kernel Function

Scope for selecting more relevant features i.e. contextual info

```matlab
% Save the model
idx = alphas > 0;
model.X = X(idx,:);
model.y = Y(idx);
model.kernelFunction = kernelFunction;
model.b = b;
model.alphas = alphas(idx);
model.w = [(alphas.*Y)'*X]';
```
FEATURES SELECTION: Insight

FEATURES IN GENERAL:

• Spectral
• Size
• Shape
• Texture
• Pattern
• Entropy
• Color

FEATURES IN AGRICULTURAL:

• Vector mask
• Vegetation Indices
• Season
• Crop Cycle
• Dryness
• LAI

Properties of Satellite Image:

• Radiometry (bit depth)
• Number of band used
• Spatial Resolution
• Image heterogeneity

Examples in Agriculture:

• Gram: smooth pinkish color
• Wheat: dark red color
• Cotton: mixed pattern
• Sugarcane: magenta color
### Contextual Features Selection: Insight

<table>
<thead>
<tr>
<th>S.N.</th>
<th>Feature Name</th>
<th>Type</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Border Index</td>
<td>Shape</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Area in sq. m.</td>
<td>Size</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Roundness</td>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Brightness</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Shape Index</td>
<td>Shape</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Mean Green</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>Mean Red</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Mean NIR</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>Compactness</td>
<td>Shape</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>SD of Green</td>
<td>Texture</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>SD of Red</td>
<td>Texture</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>SD of NIR</td>
<td>Texture</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>Length</td>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Width</td>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>GLCM</td>
<td>Texture</td>
<td>Yes</td>
</tr>
<tr>
<td>S.N.</td>
<td>FEATURE NAME</td>
<td>TYPE</td>
<td>APPLICABILITY</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------</td>
<td>--------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>16</td>
<td>Rectangularity</td>
<td>Shape</td>
<td>Yes</td>
</tr>
<tr>
<td>17</td>
<td>Density</td>
<td>Shape</td>
<td>Yes</td>
</tr>
<tr>
<td>18</td>
<td>Asymmetry</td>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>NDVI</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>20</td>
<td>Border Length</td>
<td>Shape</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Entropy</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>22</td>
<td>Height</td>
<td>Elevation</td>
<td>Yes</td>
</tr>
<tr>
<td>23</td>
<td>Shadow</td>
<td>Elevation</td>
<td>Yes</td>
</tr>
<tr>
<td>24</td>
<td>Leaf Area Index</td>
<td>Spectral</td>
<td>Yes</td>
</tr>
<tr>
<td>25</td>
<td>Topological features (slopes)</td>
<td>Containment/Adjacency</td>
<td>Yes</td>
</tr>
<tr>
<td>26</td>
<td>Segmentation Scale</td>
<td>Temporal</td>
<td>Yes</td>
</tr>
<tr>
<td>27</td>
<td>Connectivity</td>
<td>Adjacency</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Roughness</td>
<td>Texture</td>
<td>Yes</td>
</tr>
<tr>
<td>29</td>
<td>Edge pixel</td>
<td>Adjacency</td>
<td></td>
</tr>
</tbody>
</table>
**Training Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of pixels for Cotton Crop</td>
<td>Varying with training sites</td>
</tr>
<tr>
<td>No. of pixels for other classes</td>
<td>Fixed at 7687 pixels</td>
</tr>
<tr>
<td>Average no. of pixels for Cotton Crop</td>
<td>apx. 450 to 500 pixels</td>
</tr>
<tr>
<td>Standard deviation of samples</td>
<td>2</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>2</td>
</tr>
<tr>
<td>Neighbourhood pixel pattern</td>
<td>8 neighbors</td>
</tr>
<tr>
<td>SVM kernel type</td>
<td>RBF Kernel</td>
</tr>
<tr>
<td>Gamma Value for RBF Kernel</td>
<td>0.333</td>
</tr>
<tr>
<td>Penalty parameter</td>
<td>100</td>
</tr>
</tbody>
</table>
## Total Cotton sown Area: 130 ha

<table>
<thead>
<tr>
<th>S.N.</th>
<th>TRAINING SAMPLES</th>
<th>NO. OF PIXELS (COTTON)</th>
<th>CLASSIFIED NO. OF PIXEL (COTTON)</th>
<th>COTTON AREA CLASSIFIED (Ha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One Sample</td>
<td>470</td>
<td>74745</td>
<td>186.878</td>
</tr>
<tr>
<td>2</td>
<td>Two Samples</td>
<td>471</td>
<td>57324</td>
<td>143.322</td>
</tr>
<tr>
<td>3</td>
<td>Three Samples</td>
<td>490</td>
<td>55869</td>
<td>139.684</td>
</tr>
<tr>
<td>4</td>
<td>Four Samples</td>
<td>490</td>
<td>57099</td>
<td>142.759</td>
</tr>
<tr>
<td>5</td>
<td>Five Samples</td>
<td>476</td>
<td>56255</td>
<td>140.649</td>
</tr>
<tr>
<td>6</td>
<td>Six Samples</td>
<td>512</td>
<td>59372</td>
<td>148.442</td>
</tr>
</tbody>
</table>
Objective: Satellite Image Feature Selection using Genetic Algorithm & SVM

System Parameters

Set GA Parameters

Set SVM Parameters

Perform Experiment

Results:

1\textsuperscript{st} Experiment used – Linear Kernel
Experiment Started: Friday, December 04, 2015, 6:05:09 PM
Experiment Ended: Wednesday, December 09, 2015, 2:37:47 AM
Duration: 4 days, 8 hours, 32 minutes and 38 seconds
Total hours: 104 hours

2\textsuperscript{nd} Experiment used – RBF Kernel
Experiment Started: Saturday, December 12, 2015, 4:39:44 PM
Experiment Ended: Wednesday, December 14, 2015, 4:52:27 AM
Duration: 1 day, 12 hours, 12 minutes and 43 seconds
Total hours: 36 hours
Selection of best features for SVM(Linear) using GA

**OBSERVATIONS:**
1. Lower the fitness variance, closer will be the solution.
2. Closer the values of Median & Average Fitness, better will be possibilities of convergence.
3. For almost best fitness, all the features is been selected.
4. Majority of worst fitness have 4 number of bands (features).
5. This shows that data is almost Linear separable.

![Fig. – Scatter Plot of Best, Worst, Median, & Average fitness values](image)
Selection of best features for SVM(RBF) using GA

**OBSERVATIONS:**
1. Lowest Fitness variance found to be 0.021255918
2. Lowest best fitness value is 0.135494383, which is same for almost generation intervals.
3. Majority of worst fitness have fitness value 1.
4. Average fitness and Median fitness values are deviating by large difference.
5. Hence, GA based solutions for SVM RBF kernel not suggested.
6. May try different GA operators for better solution.

![Scatter Plot of Best, Worst, Median, & Average fitness values](image)
Selection of best features for SVM using GA and Heuristic

```java
// set svm parameters
final svm_parameter param = new svm_parameter();
// two class
param.svm_type = svm_parameter.C_SVC;
param.kernel_type = svm_parameter.LINEAR;
    param.degree = (int)genotype[0];
    param.gamma = genotype[1];
    param.coef0 = genotype[2];
    param.nu = genotype[3];
    param.cache_size = 100;  
    param.C = genotype[4];
    param Его = 1e-3;
    param.p = 0.1;
    param.shrinking = 1;
    param.probability = 0;
    param.nr_weight = 0;
    param.weight_label = new int[0];
    param.weight = new double[0];

// build model
svm_model model = svm.svm_train(prob, param);
// save model to file system
saveModel( model, "libsvm.model" );
// test on training set
int trueCount = 0, falseCount = 0;
for{ int index = 0; index < prob.l; index++ }
    double actual = prob.y[index];
    double prediction = svm.svm_predict(model, prob.x[index] );
    // warning! double equality test not trustworthy
    if( actual == prediction ){
        trueCount++;
    }else{
        falseCount++;
    }
}

    double accuracy = (falseCount * 1.0 )/(trueCount + falseCount);
    ind.setFitness(new SimpleValueFitness( accuracy ));
```
### Selection of best features for SVM using Heuristic Method

<table>
<thead>
<tr>
<th>obj</th>
<th>rho</th>
<th>nSV</th>
<th>vBSV</th>
<th>Total nSV</th>
<th>Fitness</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[0 1 0 0 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 1 0 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[0 0 1 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[0 1 0 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[0 1 0 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[0 1 0 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[1 1 0 0 0]</td>
</tr>
<tr>
<td>-1296.536155608003</td>
<td>-1.90830594795089</td>
<td>1308</td>
<td>1288</td>
<td></td>
<td>0.04095058271160278</td>
<td>[1 1 0 0 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
<tr>
<td>-4098.216836733862</td>
<td>-9.78571425803268</td>
<td>4100</td>
<td>4097</td>
<td></td>
<td>0.1354943829860218</td>
<td>[1 0 0 1 0]</td>
</tr>
</tbody>
</table>
### Selection of best features for SVM using Heuristic Method

<table>
<thead>
<tr>
<th>E9</th>
<th>F9</th>
<th>G9</th>
<th>H9</th>
<th>I9</th>
<th>J9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **GA based fitness:** 0.037818369
Selection of best features for SVM using Heuristic Method

OBSERVATIONS:

1. Can get best SVM accuracies even if we used mean features of any one of the bands in image, surprisingly even in the absence of main red, green and blue bands of the image.

2. There will be no effect on accuracies, even if, RGB bands added to its combination of means RGB.

3. The support vectors ranges from 1172 to 4100, whereas average numbers of support vector found to be 1553.

4. When no features selected, the algorithms found minimum support vectors(1172), however number of support vectors toll up to its maximum capacity(4100) in event almost 2 features has been selected for training & testing if satellite image
Finding Optimal SVM parameter using SGE Algorithm(GA)

Used JCLEC library: Java Class Library for Evolutionary Computation

- Can compare the values of SVM penalty parameter C in each generation for Best, Worst, Median, and Average fitness. With this best C values, other SVM Kernel (RBF) parameters can be found.

```
<table>
<thead>
<tr>
<th>Generation Report</th>
<th>degree</th>
<th>gamma</th>
<th>coef0</th>
<th>nu</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best individual</td>
<td>3</td>
<td>0.4251349</td>
<td>3.7509267</td>
<td>3.0183974</td>
<td>4.88114877</td>
</tr>
<tr>
<td>Worst individual</td>
<td>3</td>
<td>0.178972</td>
<td>1.2901003</td>
<td>0.0738615</td>
<td>4.8963807</td>
</tr>
<tr>
<td>Median individual</td>
<td>3</td>
<td>0.3509891</td>
<td>4.0241228</td>
<td>4.2732299</td>
<td>1.7396915</td>
</tr>
<tr>
<td>Average fitness</td>
<td>0.0378097933281879</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fitness variance</td>
<td>8.03801053852765E-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generation Report</th>
<th>degree</th>
<th>gamma</th>
<th>coef0</th>
<th>nu</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best individual</td>
<td>3</td>
<td>0.3161505</td>
<td>0.3195486</td>
<td>1.0685974</td>
<td>4.34696924</td>
</tr>
<tr>
<td>Worst individual</td>
<td>3</td>
<td>0.8036663</td>
<td>0.3823735</td>
<td>1.5112623</td>
<td>3.98304174</td>
</tr>
<tr>
<td>Median individual</td>
<td>3</td>
<td>0.7866476</td>
<td>3.0749181</td>
<td>3.9197762</td>
<td>0.78134388</td>
</tr>
<tr>
<td>Average fitness</td>
<td>0.03772149647289</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fitness variance</td>
<td>3.32421610843482E-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generation Report</th>
<th>degree</th>
<th>gamma</th>
<th>coef0</th>
<th>nu</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best individual</td>
<td>3</td>
<td>0.3161505</td>
<td>0.3195486</td>
<td>1.0685974</td>
<td>4.34696924</td>
</tr>
<tr>
<td>Worst individual</td>
<td>3</td>
<td>0.2896498</td>
<td>2.817049</td>
<td>3.6310044</td>
<td>3.03461814</td>
</tr>
<tr>
<td>Median individual</td>
<td>3</td>
<td>0.2909372</td>
<td>2.8159427</td>
<td>1.1544891</td>
<td>0.57952041</td>
</tr>
<tr>
<td>Average fitness</td>
<td>0.0376247264800506</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Fig. – Scatter Plot of SVM Penalty parameter C in each generation for Best, Worst, Median, & average fitness case

Optimal value of SVM Penalty Parameter C = 4.956419
Finding Optimal SVM parameter using SGE Algorithm (GA)

Fig. – Scatter Plot of Number of Iterations in GA Based SVM parameter optimization

Minimum Iterations – 18,79,147
Maximum Iterations – 1,00,000,00
Average Iterations - 9447056

Fig. – Scatter Plot of Number of Support Vectors (nSV) In GA Based SVM parameter optimization

Minimum nSV – 1171
Maximum nSV – 1185
Average nSV - 1176
Future Roadmap:

- Use other EA based approaches for SVM parameter optimization & feature selection
- Explore the possibility of using GPU accelerated approach for GA based SVM
- Can use more number of contextual features and check SVM accuracies for crop discrimination.
- Can test such optimized performance of SVM on HADOOP for better time complexity.

Future Agricultural Applications for Research

- **GPS enabled Mobile based Online Crop Monitoring System**
  - Mobiles for Crop Survey will be provided to Talathis at GP level by Agriculture Dept, GoM
  - Talathi will capture crop location along with farm photograph using Mobile application
  - Geo-tagged photographs with few details will be send to central server from each farm fields across Maharashtra State and Analysis will be carried for estimation crop productions in area.
  - figures will be utmost important for framing policies for commodities market and fixing up crop prices in India.

- **Crop Insurance System**
- **Sugarcane Mapping**
- **Horticultural Mapping**
The study may encourage soil scientist, agro-economist, agriculturist, and decision makers in entire agriculture domain towards effective use of Remote Sensing and GIS technologies in their monitoring, planning, and management process.