CROP MONITORING AND RECOMMENDATION SYSTEM USING MACHINE LEARNING TECHNIQUES

Guided by:
Dr. P. Varalakshmi
Associate professor
Department of Computer Technology
Madras Institute of Technology

TEAM:
S. KRISHNA PRASAD (2013503514)
B. SIVA SREEDHARAN (2013503525)
S. JAISHANTH (2013503565)
Remote sensing is "the acquisition of information about an object, without being in physical contact with that object"

- Passive: The sensor records energy that is reflected or emitted from the source, such as light from the sun. This is also the most common type of system.

- Active: where the object is illuminated by radiation produced by the sensors, such as radar or microwaves.
AGRICULTURE

Scope

• Crop acreage estimation
• Crop modeling for yield & production forecast / estimation
• Crop & Orchard monitoring

Benefits

• Timely availability of crop statistics for decision making & planning
• Crop growth monitoring
• Soil status monitoring
• Regular reports regarding total area under cultivation
OBJECTIVE

- Collect satellite images for agricultural crop monitoring.

- Classify the image based on Soil type, moisture content, weather conditions, pH value, organic nitrogen etc.

- Perform satellite image processing with respect to textural and spatial features.

- Analyse crop patterns with the help of past records and map them with calculated data.

- Monitor crop yield and find ways for increasing it.

- Recommend profitable crops for each land type.
PROBLEM SCENARIO

* In the existing systems, limited number of parameters are taken into account in crop monitoring.

* This system collaborates multiple morphological components to arrive at the best crop even in the case of mixed soils.

* It leverages collaborative filtering wherein modifying the existing CF approach to arrive at advantageous recommendations.
PROBLEM STATEMENT

* To design and develop a crop recommendation system to assist farmers in choosing the best profitable crop for each soil type.

* The system uses remote sensing satellite images to arrive at the prediction.
Architecture

- Image Pre-processing
  - Crop field image from Farmer

- Image Segmentation-SLIC

- SVM Prefixed by SSSE

- Collaborative Filtering

- Crop Data Set

**LEARNING PHASE**

- Image Classifier Model

- Recommending Model

**SYNTHESIS PHASE**

- Test Images
- Image Classifier Evaluator
- Recommendation Model Evaluator
- Output

- Farmer’s GPS data
  - Land Statistical Data

- Soil Class
SLIC SEGMENTATION
Images are represented as a grid of pixels, in either single or multiple channels.

We take these $M \times N$ pixel grids and then apply algorithms to them, such as face detection and recognition, template matching, and even deep learning applied directly to the raw pixel intensities.

The problem is that our pixel grid is by no means a natural representation of an image.
If I were to take the *single pixel* pointed to by the red arrow (left) and show it to you on the right at an obviously zoomed-in scale, would we be able to tell *anything* about the image based solely on that pixel.
Why?

* Computational efficiency

* Perceptual meaningfulness

* Over Segmentation

* Graphs over Super pixels
SLIC is a simple and efficient method to decompose an image in visually homogeneous regions.

It is based on a spatially localized version of k-means clustering in which each pixel is associated to a feature vector $\Psi(x, y)$ of image $I(x, y)$.

\[
\Psi(x, y) = \begin{bmatrix}
\lambda_x \\
\lambda_y \\
I(x, y)
\end{bmatrix}
\]

\[
\lambda = \frac{\text{regularizer}}{\text{regionsize}}
\]
* $M = \left\lfloor \frac{\text{image width}}{\text{image size}} \right\rfloor$

* $N = \left\lfloor \frac{\text{image height}}{\text{region size}} \right\rfloor$

* K-means uses the standard LLOYd algorithm alternating assigning pixels to the closest centers a re-estimating the centers as the average of the corresponding feature vectors of the pixel assigned to them

* After k-means has converged, SLIC eliminates any connected region whose area is less than minRegionSize pixels.
If a region has already been visited, it is skipped; if not, its area is computed and if this is less than minRegionSize its label is changed to the one of a neighbour region at p that has already been visited.
ALGORITHM

* Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps $S$.
* Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient position
* repeat
* for each cluster center $C_k$ do
* Assign the best matching pixels from a $2S \times 2S$ square neighborhood around
* the cluster center according to the distance measure
* end for
* Compute new cluster centers and residual error $E \{L1$ distance between previous centers and recomputed centers $\}$
* until $E \leq$ threshold.
* Enforce connectivity.
SPATIAL-SPECTRAL CLUSTERING OF HYPERSPECTRAL IMAGERY
LE algorithm, which involves constructing a graph representing the high dimensional data and then using generalized eigenvectors of the graph Laplacian matrix as the basis for a lower-dimensional space in which local properties of the data are preserved.

Spatial-Spectral Schrodinger Eigenmaps (SSSE) algorithm, is based on adding non-diagonal potentials encoding spatial proximity to the Laplacian matrix of the original graph (which contains spectral proximity information).

SE was proposed for semi-supervised dimensionality reduction and learning; in SSSE, the ”semi-supervision” refers to knowledge of spatial proximity between pixels instead of knowledge of particular class labels.
A key benefit of SE is that the potential matrix $V$ enables semi-supervised clustering. If a subset of points in $X$ has a known label, defining $V$ to be a cluster potential will pull the corresponding points in $Y$ towards each other. This same behaviour extends to multiple labels.

\[ V^{(i,j)} = \begin{cases} 
1 & (k, l) \in \{(i, i), (i, i)\} \\
-1 & (k, l) \in \{(i, j), (j, i)\} \\
0 & \text{otherwise}
\end{cases} \]

\[ x_i^T = \begin{bmatrix} x_i f_i^T & x_i p_i^T \end{bmatrix}^T \]
Construct an undirected graph $G = (X,E)$ whose vertices are the points in $X$ and whose edges $E$ are defined based on proximity between the spectral components of the vertices.

Define weights for the edges in $E$ based on spectral information.

Define a cluster potential matrix $V$ that encodes proximity between the spatial components of the vertices.

Compute the smallest $m + 1$ eigenvalues and eigenvectors of $(L + \alpha V)f = \lambda Df$, where $D$ is the diagonal weighted degree matrix defined by $D_{i,i} = \sum_j W_{i,j}$, and $L = D - W$ is the Laplacian matrix.

Additional texture based features are added as must-link constraint.
Evaluate the soil class obtained from the image classifier and data and use the historical data to recommend suitable crops for each crop field.

**Input**: Feature vector containing features like soil class, terrain, slope, elevation, etc

**Output**: a list of recommended crops

1. $X \leftarrow$ input feature vector
2. AdaBoost $\leftarrow$ one vs rest Booster for class $i$
3. return $\text{argmax}_i(\text{AdaBoost} \cdot \text{predict}(X))$
Given the training data \((x_1, y_1), \ldots, (x_m, y_m)\),

\[ y_i \in \{-1, +1\}, x_i \in X \] is the object or instance, \(y_i\) is the classification.

For \(t=1, \ldots, T\):

Form a distribution \(D_T\) on \(\{1, \ldots, m\}\).

Select weaker classifier with smallest error \(\epsilon_t\) on \(D_t\):

\[ \epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i] \]

\[ h_t : X \rightarrow \{-1, +1\} \]

Output single classifier \(H_{\text{final}}(x)\)
The benefit of SSSE is that since coefficients are applied to the cluster potentials (and not applied as edge weights on the graph G), the spatial neighbourhood Np can be chosen to be quite small while still allowing G to contain edges corresponding to spectrally similar points that may be spatially distant.

Once a graph is constructed, any changes made with respect to $\sigma_f / \sigma_p$ can be achieved solely by modifying the cluster potential matrix.

Knowledge propogation:

Situation arise when cluster potentials which are extracted from small set of manually provided labels may have a crucial impact on the dimensionality reduction method. This can be achieved by replacing the cluster potential matrix $M$

$$M = \sum_{(x_{i_k}, x_{j_k}) \in M} \eta_{i_k,j_k} \cdot WD^{-1}V^{(i_k, j_k)}$$
Image pre-processing

The pre-processor uses multiple morphological component analysis (MMCA) for dimensionality reduction by means of content, coarseness, directional vectors and contrast features. This works by finding the possible contours with the specification to identify the soil areas.

Image Segmentation - SLIC

This module leverages the creation of superpixels using Simple linear iterative clustering (SLIC). In this method, the preprocessed images are taken as input and segmented crop field image is produced as the result.

SVM prefixed by SSSE

This module is the primary analysis module of the project. The SVM classifier is the backend classifier which takes the input of the Spectral spatial Schrodinger Eigen maps. This involves considering both spectral and spatial parameters in addition to textural features for partial knowledge and knowledge propagation.
Calculating NDVI

The normalized difference vegetation index (NDVI) is a simple graphical indicator that can be used to analyse remote sensing measurements and assess whether the target being observed contains live green vegetation or not.

**Image classifier evaluator**

This module takes in the pre-processed input image at the end and uses the pickled historical data sets of whether and crop yield parameters to predict the soil class and yield values.

**Recommendation model evaluator**

This module takes in the soil class and GPS parameters as input and also uses the Recommendation Model which AdaBoost to recommend the crop suitable for the image and historical data set provided.
<table>
<thead>
<tr>
<th>Datasets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather</td>
<td></td>
</tr>
<tr>
<td>weather_all.csv</td>
<td>Instantaneous weather data</td>
</tr>
<tr>
<td>Yield</td>
<td></td>
</tr>
<tr>
<td>yield_proc_data.csv</td>
<td>Yield forecast text data</td>
</tr>
<tr>
<td>Pickle Feed</td>
<td></td>
</tr>
<tr>
<td>PickleFeed.csv</td>
<td>County wise crop data</td>
</tr>
<tr>
<td>Location</td>
<td></td>
</tr>
<tr>
<td>county_lat_lng.csv</td>
<td>Locational data for each county</td>
</tr>
<tr>
<td>Images</td>
<td></td>
</tr>
<tr>
<td>PaviaU.mat</td>
<td>Hyperspectral images</td>
</tr>
<tr>
<td>KSC.mat</td>
<td>Hyperspectral images</td>
</tr>
<tr>
<td>Salinas.mat</td>
<td>Hyperspectral images</td>
</tr>
<tr>
<td>Salinas_corrected.mat</td>
<td>Hyperspectral images</td>
</tr>
<tr>
<td>Botswana.mat</td>
<td>Hyperspectral images</td>
</tr>
<tr>
<td>Indian_pines_g.mat</td>
<td>Hyperspectral images</td>
</tr>
</tbody>
</table>
User Interface - Preprocessor
CONTINUED..

Image Data

Mid –Stretch Filter
CONTINUED
CONTINUED..

Noise Removal
Image Classifier - SSSE

Ground Truth Class Labels
Continued...

ML constrained / SVM
Crop yield estimation

Loading raw weather data....
Loading yield files........
combining the two files for analysis
Fixing resulting NAs and missing data

Running model on TRAIN?  True
{'n_estimators': 5, 'learning_rate': 3.2000000000000002}

TRAINING SET?  True

=================================
Adaboost  r2:  0.914146128796
Adaboost  MSE:  73.0880285453
Adaboost  RMSE:  8.54915367421
=================================
Training data
Continued....

Your forecast for this season in:
NORTH portion of ID is:
Prediction is [ 57.75416667 ] bushels/acre

Yield Prediction
Relevance prediction of crops

<table>
<thead>
<tr>
<th>CROP</th>
<th>SOIL</th>
<th>LONDD</th>
<th>LATDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>cotton</td>
<td>2</td>
<td>69.16667</td>
<td>34.5</td>
</tr>
<tr>
<td>cotton</td>
<td>2</td>
<td>69.16667</td>
<td>34.5</td>
</tr>
<tr>
<td>cotton</td>
<td>2</td>
<td>69.16667</td>
<td>34.5</td>
</tr>
<tr>
<td>cotton</td>
<td>2</td>
<td>61.43333</td>
<td>34.33333</td>
</tr>
<tr>
<td>cotton</td>
<td>2</td>
<td>62.13333</td>
<td>32.38333</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.75</td>
<td>40.61667</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.30056</td>
<td>39.635</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.38611</td>
<td>42.22222</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.56111</td>
<td>41.07778</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.51111</td>
<td>41.42361</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.54083</td>
<td>41.17667</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.99556</td>
<td>41.06944</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.64722</td>
<td>40.215</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.78444</td>
<td>40.63722</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.48361</td>
<td>40.34528</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.77028</td>
<td>41.37778</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.95889</td>
<td>40.65528</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.7325</td>
<td>40.56444</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.40639</td>
<td>41.69278</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>19.7625</td>
<td>41.37</td>
</tr>
<tr>
<td>ground nut</td>
<td>2</td>
<td>20.71083</td>
<td>40.72167</td>
</tr>
</tbody>
</table>
Predicted crop -Relevance

train features:

[[ 2. 69.166667  34.5 ]
 [ 2. 69.166667  34.5 ]
 [ 2. 69.166667  34.5 ]
...,            
 [ 5. -38.58  -8.  ]
 [ 5. -37.84972 -7.63 ]
 [ 5. -38.41972 -7.30972]]
Yield prediction accuracy

ALL regression performance

- $y_{true}$ (bu/acre)
- $y_{pred}$ (bu/acre)

state
Accuracy Parameters

\[ \text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} = \frac{TP + TN}{P + N} \]

\[ \text{Sensitivity} = \frac{TP}{TP + FN} = \frac{TP}{P} \]

\[ \text{Specificity} = \frac{TN}{TN + FP} = \frac{TN}{N} \]

\[ \text{Precision} = \frac{TP}{TP + FP} \]

\[ \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \]

Where \( n \) is the total number of tuples and \( y_i \) are the data points.
## Classification Accuracy

<table>
<thead>
<tr>
<th>=&gt;</th>
<th>SM</th>
<th>SSSE</th>
<th>GB</th>
<th>BM</th>
<th>BE</th>
<th>HZY</th>
</tr>
</thead>
<tbody>
<tr>
<td>=&gt;</td>
<td>UN</td>
<td>ML</td>
<td>UN</td>
<td>ML</td>
<td>UN</td>
<td>ML</td>
</tr>
<tr>
<td>Overall Accuracy:</td>
<td>0.9689</td>
<td>0.9786</td>
<td>0.9760</td>
<td>0.9831</td>
<td>0.9799</td>
<td>0.9800</td>
</tr>
<tr>
<td>Average Accuracy:</td>
<td>0.9961</td>
<td>0.9973</td>
<td>0.9975</td>
<td>0.9979</td>
<td>0.9966</td>
<td>0.9975</td>
</tr>
<tr>
<td>Average Precision:</td>
<td>0.9112</td>
<td>0.9124</td>
<td>0.9812</td>
<td>0.9815</td>
<td>0.9341</td>
<td>0.9188</td>
</tr>
<tr>
<td>Average Sensitivity:</td>
<td>0.9800</td>
<td>0.9808</td>
<td>0.9851</td>
<td>0.9863</td>
<td>0.9842</td>
<td>0.9831</td>
</tr>
<tr>
<td>Average Specificity:</td>
<td>0.9978</td>
<td>0.9985</td>
<td>0.9981</td>
<td>0.9988</td>
<td>0.9986</td>
<td>0.9986</td>
</tr>
</tbody>
</table>
Overall Accuracy

<table>
<thead>
<tr>
<th></th>
<th>SM</th>
<th>GB</th>
<th>SSSE</th>
<th>BM</th>
<th>BE</th>
<th>HZYF</th>
</tr>
</thead>
<tbody>
<tr>
<td>UN</td>
<td>0.9689</td>
<td>0.9797</td>
<td>0.976</td>
<td>0.9669</td>
<td>0.9714</td>
<td>0.9742</td>
</tr>
<tr>
<td>ML</td>
<td>0.9786</td>
<td>0.9805</td>
<td>0.9831</td>
<td>0.9768</td>
<td>0.9752</td>
<td>0.9739</td>
</tr>
</tbody>
</table>
Average Specificity

|       | UN   | ML   | GB   | SSSE | BM   | BE   | HYZ |
|-------|------|------|------|------|------|------|-----|------|
| SM    | 0.9808 | 0.9808 | 0.9842 | 0.9851 | 0.9760 | 0.9818 | 0.9767 | 0.9767 |
| GB    | 0.9842 | 0.9831 | 0.9863 | 0.9851 | 0.9760 | 0.9818 | 0.9797 | 0.9751 |
| SSSE  | 0.9851 | 0.9863 | 0.9851 | 0.9760 | 0.9818 | 0.9797 | 0.9797 | 0.9767 |
| BM    | 0.9760 | 0.9810 | 0.9851 | 0.9760 | 0.9760 | 0.9797 | 0.9797 | 0.9751 |
| BE    | 0.9818 | 0.9818 | 0.9863 | 0.9810 | 0.9797 | 0.9797 | 0.9797 | 0.9751 |
| HYZ   | 0.9767 | 0.9751 | 0.9797 | 0.9797 | 0.9797 | 0.9797 | 0.9797 | 0.9751 |
Average precision

<table>
<thead>
<tr>
<th></th>
<th>SM</th>
<th>GB</th>
<th>SSSE</th>
<th>BM</th>
<th>BE</th>
<th>HYZY</th>
</tr>
</thead>
<tbody>
<tr>
<td>UN</td>
<td>0.98</td>
<td>0.9842</td>
<td>0.9851</td>
<td>0.976</td>
<td>0.9818</td>
<td>0.9767</td>
</tr>
<tr>
<td>ML</td>
<td>0.9808</td>
<td>0.9831</td>
<td>0.9863</td>
<td>0.981</td>
<td>0.9797</td>
<td>0.9751</td>
</tr>
</tbody>
</table>
Average accuracy

<table>
<thead>
<tr>
<th></th>
<th>UN</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>0.98</td>
<td>0.9808</td>
</tr>
<tr>
<td>GB</td>
<td>0.9842</td>
<td>0.9831</td>
</tr>
<tr>
<td>SSSE</td>
<td>0.9851</td>
<td>0.9863</td>
</tr>
<tr>
<td>BM</td>
<td>0.976</td>
<td>0.981</td>
</tr>
<tr>
<td>BE</td>
<td>0.9818</td>
<td>0.9797</td>
</tr>
<tr>
<td>HYZY</td>
<td>0.9767</td>
<td>0.9751</td>
</tr>
</tbody>
</table>
### Average Sensitivity

<table>
<thead>
<tr>
<th></th>
<th>UN</th>
<th>GB</th>
<th>SSSE</th>
<th>BM</th>
<th>BE</th>
<th>HZYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>0.98</td>
<td>0.9842</td>
<td>0.9851</td>
<td>0.976</td>
<td>0.9818</td>
<td>0.9767</td>
</tr>
<tr>
<td>ML</td>
<td>0.9808</td>
<td>0.9831</td>
<td>0.9863</td>
<td>0.981</td>
<td>0.9797</td>
<td>0.9751</td>
</tr>
</tbody>
</table>
SYSTEM INFORMATION

TOOLS REQUIRED

- MATLAB
- GIS
- Beam Raster Toolbox

LANGUAGE

- Python
The output is to predict the best suitable crop for each soil variety.

Further, provide assistance by continuous monitoring of crops to increase the yield.
This project helps in assisting farmers by recommending suitable crops for different land types. It greatly helps in reducing the monetary losses due to improper selection of crops.