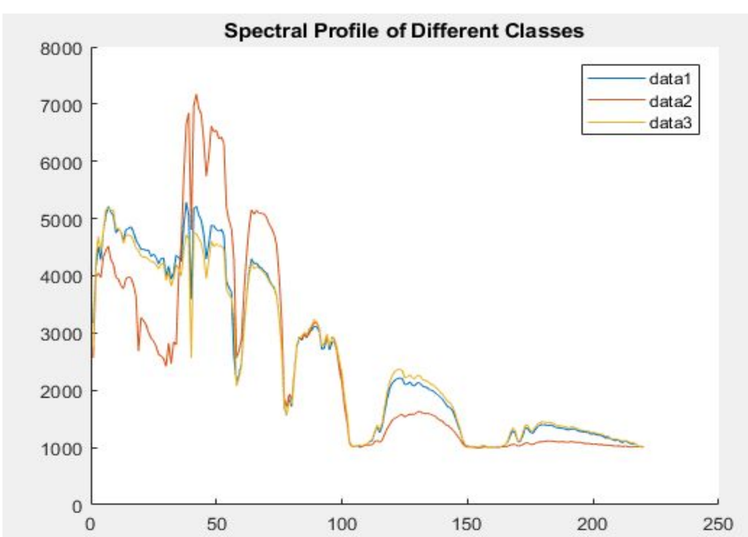
An Information - Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image

-Kshitij Srivastava
-1510110200, 4th Year ECE

Introduction:

Humans have always wanted to gain insights about the data present around them. RGB image is one such medium for getting information. These images are being used in variety of places ranging from image classification to extensive image analysis. However there are lots of data which are not available to the human eye. A spectral image is one such useful data which is not visible to naked eye. A multispectral image is one that captures image data within specific wavelength ranges across the electromagnetic spectrum. Spectral imaging can allow extraction of additional information the human eye fails to capture with its receptors for red, green and blue. This was originally developed for space-based imaging, and has also found use in document and painting analysis. Multi-spectral data captures the data for few spectral bands but later however there was a need to increase the amount of spectral bands and hence increase the amount of data. So hyper-spectral data was used for extracting data for hundreds of bands.

A hyperspectral image is a three dimensional matrix where the third dimension is the spectral domain represented by hundreds of spectral bands for a particular point. A hyperspectral image pixel will contain spectral information in its spectral bands which can be used for pixel variability, similarity and discrimination.



Because of the atmospheric effects, the spectral information of the pixel varies during image acquisition. Therefore an information theoretical approach referred as Spectral Information Measure (SIM) is made to take into account the spectral variability, similarity, and discrimination. SIM models the spectral band-to-band variability as result of uncertainty caused by randomness. It considers each pixel as a random variable and defines the desired probability distribution by normalizing its spectral histogram to unity.

SIM can describe the randomness of the pixel and can also generate high order statistics of pixel based on spectral profile. Spectral Information provided is very valuable in material discrimination. SIM is designed to capture uncertainty and unpredictability. So higher the data dimensionality the more effective is the SIM.

In order to determine the spectral similarity between two pixels a SIM-based Spectral Information Divergence (SID) is developed, based on the concept of divergence to measure the discrepancy of probabilistic behaviors between the spectral signatures of the two pixels. In comparison to Spectral angle mapper (SAM), SID is more effective in preserving spectral properties.

In order to compare the relative discriminatory power between two spectral measures, another criterion is proposed, based on the ratio of the spectral similarities of a pixel to another pixel relative to a reference pixel.

Spectral Information Measure (SIM)

For a given hyperspectral pixel (vector) $x = (x_1, ..., x_L)^T$, each component x_l is a pixel of band B_l acquired at a particular wavelength λ_l . $\{\lambda_l\}_{l=1}^{L}$ is the set of L wavelengths each of it corresponds to the spectral band channel.

So the hyperspectral pixel x can be modeled as a random variable by an probability space (Ω, Σ, P) where Ω is sample space, Σ is an event space and P is the probability measure.

So the Probability measure P for x is given by

$$P(\{\lambda_j\}) = p_j = x_j / \sum_{l=1}^{L} x_l$$

The vector $p = (p_1, p_2, ..., p_L)^T$ is the desired probability vector obtained from pixel vector x. So pixel x can be seen as a single information source with its statistics governed by p which can be used to describe the spectral variability of pixel. Using this mean, variance, 3rd central moment etc can be calculated.

Using the probability vector, entropy of each hyperspectral image pixel x can be calculated. Entropy is useful in the describing the uncertainty and randomness present in the pixel x.

Entropy $H(x) = -\sum_{l=1}^{L} p_l log(p_l)$

Spectral Information Divergence (SID)

This SIM based approach measures spectral similarity measure to capture correlation between two pixels. $y = (y_1, \ldots.., y_L)^T$ is another pixel with probability vector $q = (q_1, q_2, \ldots q_L)^T$ where $q_i = y_i / \sum_{l=1}^{L} y_l$.

The lth band self-entropy of x and y is $I_l(x) = -log(p_l)$ and $I_l(y) = -log(q_l)$. The relative entropy of y with respect to x can be defined by

$$D(x \parallel y) = \sum_{l=1}^{L} p_l \left(\frac{p_l}{q_l}\right)$$

Where $D(x \parallel y)$ is known as Kullback-Leibler information measure, directed divergence, cross-entropy.

Using Kullback-Leibler Information measure, a symmetric hyperspectral measure Spectral Information Divergence (SID) can be defined which can be used to measure spectral similarity between two pixels x & y. SID offers a new look at spectral similarity by making use of relative entropy to account for the spectral information provided by each pixel.

$$SID(x, y) = D(x \parallel y) + D(y \parallel x)$$

Criteria For Spectral Discrimination

Spectral Discriminatory Probability

Many applications requires to identify a target pixel from an unknown image scene using an existing spectral library or database Δ. It is of interest to know that what is the likelihood of the pixel in question being identified as one of the spectra in Δ.

$\{s_j\}_{j=1}^{J}$ be the J spectral bands in database Δ and t be the target pixel to be identified using Δ. $m(.,.)$ be any hyperspectral measure. So the the spectral discriminatory probabilities of all s_j in Δ with respect to t is

$$p_{t,\Delta}^{m} = \frac{m(t, s_i)}{\sum_{j=1}^{J} m(t, s_j)} \qquad \text{for } i = 1, 2....J$$

The resulting probability vector is $p_{t,\Delta}^{m} = (p_{t,\Delta}^{m}(1), p_{t,\Delta}^{m}(2), \ldots, p_{t,\Delta}^{m}(J))^T$ and is called the spectral discriminatory probability vector of Δ w.r.t t.

Spectral Discriminatory Power

Spectral discriminatory power is used to identify which spectral similarity measure is more effective . It is designed based on the power of discriminating one pixel from another relative to a reference pixel d.

$m(.,.)$ is any hyperspectral measure. d be the spectral signature of a reference pixel and s_i, s_j be the spectral signature of two pixels.

$$PW^m(s_i, s_j; d) = max\left\{\frac{m(s_i, d)}{m(s_j, d)}, \frac{m(s_j, d)}{m(s_i, d)}\right\}$$

$PW^m(s_i, s_j; d)$ provides an index of spectral discrimination capability of a specific hyperspectral measure $m(.,.)$ between any two spectral signatures s_i, s_j relative to d. The higher the spectral discriminatory power, better the discriminatory power $m(.,.)$ has.

Using the spectral discriminatory probability vector spectral discriminatory entropy of Δ with respect to t can be defined.

$$H^m(t;\Delta) = -\sum_{j=1}^{J} p_{t,\Delta}^{m}(j)\, log(p_{t,\Delta}^{m}(j))$$

This provides the uncertainty measure of identifying t using the spectral signatures in Δ. A smaller $H^m(t;\Delta)$ indicates better chance to identify t.

SIM based spectral similarity measure

There are two commonly used spectral similarity measures called Euclidean Distance and SAM. These spectral similarity criteria will be used to compare with the performances of (SID) Spectral Information Divergence similarity criterion.

$$ED\,(s_i,\ s_j)\ =\ \|\, s_i - s_j\|\ =\ \left[\sum_{l=1}^{L}(s_{il} - s_{jl})^2\right]^2$$

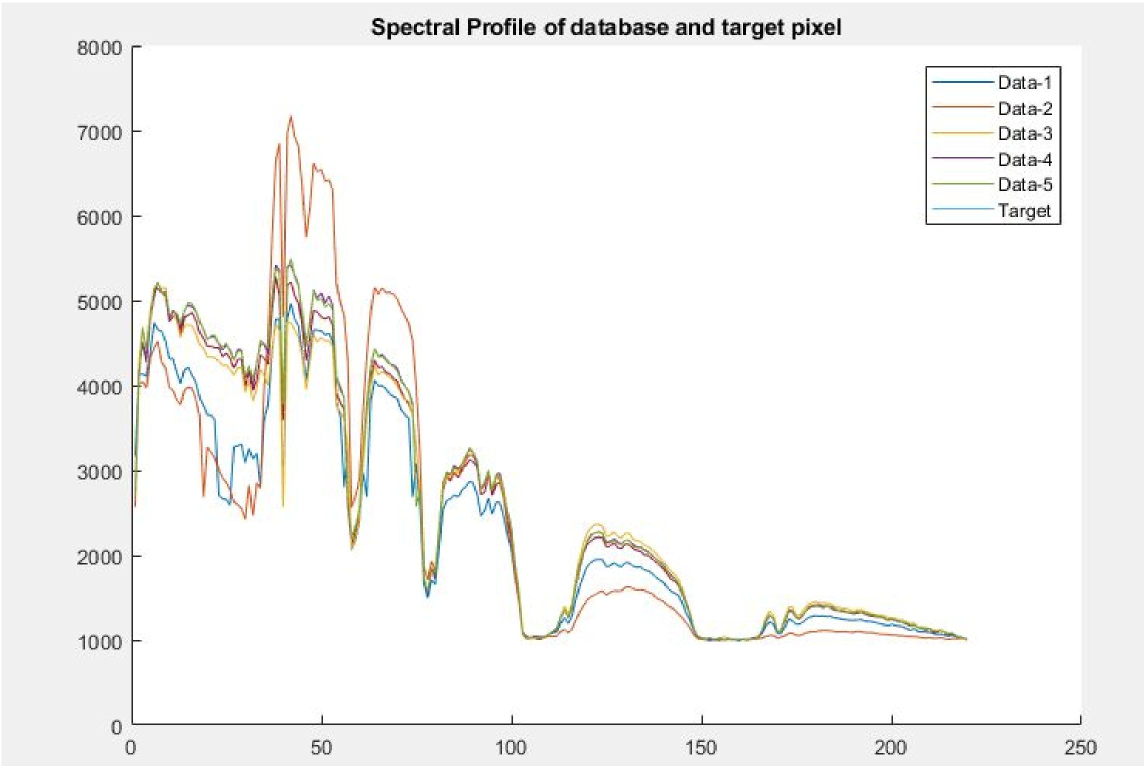
$$SAM(s_i,\ s_j)\ =\ cos^{-1}\left(\frac{<s_i, s_j>}{\|s_i\|\ \|s_j\|}\right)\ (\text{in radians})$$

Results

The data used in this project is the Indian pines dataset which is the one of the famous and widely used dataset on hyperspectral data. It contains 220 bands of data with 145 x 145 pixel image. And for each pixel it contains corresponding ground verification data which is indicated by the class. This dataset contains 17 unique classes.

For class 3 , SID was found for two separate spectral bands. And it was found it be 0.00129. For class 3 and class 15, SID was found for two spectral bands of the respective class and it was found to be 0.0247. Hence it tells that class 15 and class 3 are not similar to each other and they have some information in them. But in the first case due to the low value of SID, not much information was contained in them.

For spectral discriminatory probability, it is needed to identify the target pixel using a library. So a library was created which had 5 different classes of spectral bands and target pixel was to be classified on the basis of the database. This was implemented using three different criteria like Euclidean distance, Spectral angle mapper and spectral information divergence (SID). All the three criteria predicted the correct class of the target pixel.



For Spectral discriminating power, the reference pixel was chosen to be of class 3, S_i pixel of class 0 and S_j pixel of class 11. The spectral discriminating power was found using Euclidean distance, Spectral Angle Mapper (SAM) and Spectral Information Divergence (SID). Euclidean distance had the power of 1.53, SAM had the power of 2.3 and the power of SID is 4.8. It shows that SID is twice as more effective than SAM. Hence it concludes that the SID has the best discriminating power over all.

Conclusion

The Spectral Information Measure (SIM) is introduced to model a hyperspectral image pixel as a random variable to capture spectral variability in the pixel. Then the Spectral Information Divergence (SID) was further developed to measure the spectral similarity between two pixels. It is demonstrated that the SIM-based criteria performed more effectively than the traditional spectral measure SAM.

Code Snippets

```
%% Loading the Indian Pines Data
load Indian_pines;      %Loading the Indian Pines Spectral Data
load Indian_pines_gt;   %Loading the Indian Pines Ground_Verification Data

%% Converting the 3D matrix data to 2D tabular form

[length , breadth ] = size(indian_pines_gt);    %Finding the length and breadth
[ ~ ,  ~ , bands]=size(indian_pines);   %Finding the number of bands in the Indian Pines spectral
data

input = zeros(length*breadth, bands);   %Making empty vector for input spectral bands
output = zeros(length*breadth, 1);      %Making empty vector for output ground data

%Extracting the spectral data for all the pixels and putting it into input
%vector
index=1;
for i=1:length
   for j=1:breadth
      y = squeeze(indian_pines(i,j,:)); % Extracting the spectral bands for i,j index
      input(index,:) = y;
      output(index,1) = indian_pines_gt(i,j); %Ground truth data for i,j index
      index = index + 1;
   end
end

%% Plotting Different Spectral Bands

%Some randomly selected spectral bands
index1 = 1;
index2 = 75;
index3 = 100;

figure;
hold on;
plot(input(index1,:))
plot(input(index2,:))
plot(input(index3,:))
title('Spectral Profile of Different Classes')
hold off;
legend;


%% Probability Vector
```

```
prob_vector = zeros(length*breadth, bands); %Probability vector for all the input spectral bands
[input_len , ~] = size(input);             %Total Number of Spectral Bands

%Finding the probability vector for all the spectral bands
for i=1:input_len
   sums = sum(input(i,:));
   answer = (input(i,:)/sums);
   prob_vector(i,:) = answer ;
end

%% Entropy of the Hyper-Spectral Image Pixel
entropy_vector = zeros(length*breadth, 1); %Entropy vector for all the spectral bands

for i=1:input_len
   entropy_vector(i,1) = -1*sum(log10(prob_vector(i,:)).* prob_vector(i,:));
   %Entropy for each pixel
end

%% Spectral Information Divergence

% Selected pixels
pixel_index1 = 1;
pixel_index2 = 75;
%Relative Entropy of y with respect to x
D_xy =
sum(prob_vector(pixel_index1,:).*(log10(prob_vector(pixel_index1,:)./prob_vector(pixel_index2,:
))));
%Relative Entropy of x with respect to y
D_yx =
sum(prob_vector(pixel_index2,:).*(log10(prob_vector(pixel_index2,:)./prob_vector(pixel_index1,:
))));
SID_value = D_xy +D_yx;

%% Spectral Discriminatory Probability

% 5 Spectral Signatures
input_database = input([1,25,75,100,150],:);    %Spectral Database
output_database = output([1,25,75,100,150],:);  %Spectral database output
[len_input_database , ~] = size(input_database);

%Target Pixel for identification
target_pixel_index = 5;
```

```
target_pixel = input(target_pixel_index,:);
target_pixel_prob = (input(i,:)/sum(input(i,:)));

%Spectral Similarity measures calculation Vector
%Zero vector for Euclidian Distance Measurement
prob_vector_euclid = zeros(target_pixel_index,1);
%Zero vector for SAM
prob_vector_sam = zeros(target_pixel_index,1);
%Zero vector for SID
prob_vector_sid = zeros(target_pixel_index,1);

%Finding SAM and Euclid Distance for all the spectral profiles in Database
for i=1:len_input_database
    prob_vector_euclid(i,1) = Euclid_Distance( target_pixel,input_database(i,:) );
    prob_vector_sam(i,1) = SAM( target_pixel,input_database(i,:) );
    input_database_prob = input_database(i,:)/ (sum(input_database(i,:)));
    prob_vector_sid(i,1) = SID( target_pixel_prob, input_database_prob );
end

% Check the prediction by Euclidian Distance
[~,predicted_index_euclid] = min(prob_vector_euclid);
predicted_output_euclid = output_database(predicted_index_euclid);

% Check the prediction by SAM
[~,predicted_index_sam] = min(prob_vector_sam);
predicted_output_sam = output_database(predicted_index_sam);

% Check the prediction by SAM
[~,predicted_index_sid] = min(prob_vector_sid);
predicted_output_sid = output_database(predicted_index_sid);

%% For showing the figure of the database and the target pixel

figure;
hold on;
for i=1:len_input_database
    plot(input_database(i,:));
end
plot(target_pixel, '-')
title('Spectral Profile of database and target pixel');
hold off;
legend('Data-1','Data-2', 'Data-3', 'Data-4', 'Data-5', 'Target' );
```

```
%% Spectral Discriminating Power

reference_pixel_index = 5;                              %Index of reference pixel
reference_pixel = input(reference_pixel_index,:);          %Reference pixel spectral bands
reference_pixel_output = output(reference_pixel_index,1);   %Reference pixel output
reference_pixel_prob = reference_pixel/ (sum(reference_pixel)); %Reference pixel Prob

Si_index = 25;                                          %Index of Si pixel
Si_pixel = input(Si_index,:);                           %Si pixel spectral bands
Si_pixel_output = output(Si_index,1);                   %Si pixel output
Si_pixel_prob = Si_pixel/ (sum(Si_pixel));              %Si pixel probability

Sj_index = 75;                              %Index of Sj pixel
Sj_pixel = input(Sj_index,:);                 %Sj pixel spectral bands
Sj_pixel_output = output(Sj_index,1);           %Sj pixel spectral bands
Sj_pixel_prob = Sj_pixel/ (sum(Sj_pixel));       %Sj pixel probability

% When Euclidian is used for Spectral Similarity Measure
arg1 = Euclid_Distance(Si_pixel,reference_pixel )/Euclid_Distance(Sj_pixel,reference_pixel);
arg2 = Euclid_Distance(Sj_pixel,reference_pixel )/Euclid_Distance(Si_pixel,reference_pixel);
PW_euclid = max(arg1, arg2 );

% When SAM is used for Spectral Similarity Measure
arg1 = SAM(Si_pixel,reference_pixel )/SAM(Sj_pixel,reference_pixel);
arg2 = SAM(Sj_pixel,reference_pixel )/SAM(Si_pixel,reference_pixel);
PW_sam = max(arg1, arg2 );

% When SID is used for Spectral Similarity Measure

arg1 = SID(Si_pixel_prob,reference_pixel_prob )/SID(Sj_pixel_prob,reference_pixel_prob);
arg2 = SID(Sj_pixel_prob,reference_pixel_prob )/SID(Si_pixel_prob,reference_pixel_prob);
PW_sid = max(arg1, arg2 );

%% Spectral Discriminatory Entropy

%Spectral discriminatory entropy of database with respect to reference pixel for Euclid Distance
spectral_discriminatory_entropy_euclid = -1*sum(log10(prob_vector_euclid).*
prob_vector_euclid);
%Spectral discriminatory entropy of database with respect to reference
%pixel for SAM
spectral_discriminatory_entropy_sam = -1*sum(log10(prob_vector_sam).* prob_vector_sam);

%% SAM Vs SID
```

```
index1 = 5;
index2 = 75;

sid_dist = SID(prob_vector(index1,:), prob_vector(index2,:));
eucid_dist = Euclid_Distance(input(index1,:),input(index2,:));
sam_dist = SAM(input(index1,:),input(index2,:));
```