

# Classification

## Introduction

Multispectral classification is the process of sorting pixels into a finite number of individual classes, or categories of data, based on their data file values. If a pixel satisfies a certain set of criteria, the pixel is assigned to the class that corresponds to that criteria. This process is also referred to as image segmentation.

Depending on the type of information you want to extract from the original data, classes may be associated with known features on the ground or may simply represent areas that look different to the computer. An example of a classified image is a land cover map, showing vegetation, bare land, pasture, urban, etc.

## The Classification Process

### Pattern Recognition

Pattern recognition is the science—and art—of finding meaningful patterns in data, which can be extracted through classification. By spatially and spectrally enhancing an image, pattern recognition can be performed with the human eye; the human brain automatically sorts certain textures and colors into categories.

In a computer system, spectral pattern recognition can be more scientific. Statistics are derived from the spectral characteristics of all pixels in an image. Then, the pixels are sorted based on mathematical criteria. The classification process breaks down into two parts: training and classifying (using a decision rule).

### Training

First, the computer system must be trained to recognize patterns in the data. Training is the process of defining the criteria by which these patterns are recognized (Hord, 1982). Training can be performed with either a supervised or an unsupervised method, as explained below.

#### Supervised Training

Supervised training is closely controlled by the analyst. In this process, you select pixels that represent patterns or land cover features that you recognize, or that you can identify with help from other sources, such as aerial photos, ground truth data, or maps. Knowledge of the data, and of the classes desired, is required before classification.

By identifying patterns, you can instruct the computer system to identify pixels with similar characteristics. If the classification is accurate, the resulting classes represent the categories within the data that you originally identified.

## Unsupervised Training

Unsupervised training is more computer-automated. It enables you to specify some parameters that the computer uses to uncover statistical patterns that are inherent in the data. These patterns do not necessarily correspond to directly meaningful characteristics of the scene, such as contiguous, easily recognized areas of a particular soil type or land use. They are simply clusters of pixels with similar spectral characteristics. In some cases, it may be more important to identify groups of pixels with similar spectral characteristics than it is to sort pixels into recognizable categories.

Unsupervised training is dependent upon the data itself for the definition of classes. This method is usually used when less is known about the data before classification. It is then the analyst's responsibility, after classification, to attach meaning to the resulting classes (Jensen, 1996). Unsupervised classification is useful only if the classes can be appropriately interpreted.

## Signatures

The result of training is a set of signatures that defines a training sample or cluster. Each signature corresponds to a class, and is used with a decision rule (explained below) to assign the pixels in the image file to a class. Signatures in ERDAS IMAGINE can be parametric or nonparametric.

A parametric signature is based on statistical parameters (e.g., mean and covariance matrix) of the pixels that are in the training sample or cluster. Supervised and unsupervised training can generate parametric signatures. A set of parametric signatures can be used to train a statistically-based classifier (e.g., maximum likelihood) to define the classes.

A nonparametric signature is not based on statistics, but on discrete objects (polygons or rectangles) in a feature space image. These feature space objects are used to define the boundaries for the classes. A nonparametric classifier uses a set of nonparametric signatures to assign pixels to a class based on their location either inside or outside the area in the feature space image. Supervised training is used to generate nonparametric signatures (Kloer, 1994).

ERDAS IMAGINE enables you to generate statistics for a nonparametric signature. This function allows a feature space object to be used to create a parametric signature from the image being classified. However, since a parametric classifier requires a normal distribution of data, the only feature space object for which this would be mathematically valid would be an ellipse (Kloer, 1994).

When both parametric and nonparametric signatures are used to classify an image, you are more able to analyze and visualize the class definitions than either type of signature provides independently (Kloer, 1994).



---

See *"Math Topics"* for information on feature space images and how they are created.

## Decision Rule

After the signatures are defined, the pixels of the image are sorted into classes based on the signatures by use of a classification decision rule. The decision rule is a mathematical algorithm that, using data contained in the signature, performs the actual sorting of pixels into distinct class values.

### Parametric Decision Rule

A parametric decision rule is trained by the parametric signatures. These signatures are defined by the mean vector and covariance matrix for the data file values of the pixels in the signatures. When a parametric decision rule is used, every pixel is assigned to a class since the parametric decision space is continuous (Kloer, 1994).

### Nonparametric Decision Rule

A nonparametric decision rule is not based on statistics; therefore, it is independent of the properties of the data. If a pixel is located within the boundary of a nonparametric signature, then this decision rule assigns the pixel to the signature's class. Basically, a nonparametric decision rule determines whether or not the pixel is located inside of nonparametric signature boundary.

## Output File

When classifying an image file, the output file is an image file with a thematic raster layer. This file automatically contains the following data:

- class values
- class names
- color table
- statistics
- histogram

The image file also contains any signature attributes that were selected in the ERDAS IMAGINE Supervised Classification utility.



---

*The class names, values, and colors can be set with the Signature Editor or the Raster Attribute Editor.*

## Classification Tips

### Classification Scheme

Usually, classification is performed with a set of target classes in mind. Such a set is called a classification scheme (or classification system). The purpose of such a scheme is to provide a framework for organizing and categorizing the information that can be extracted from the data (Jensen et al, 1983). The proper classification scheme includes classes that are both important to the study and discernible from the data on hand. Most schemes have a hierarchical structure, which can describe a study area in several levels of detail.

A number of classification schemes have been developed by specialists who have inventoried a geographic region. Some references for professionally-developed schemes are listed below:

- Anderson, J.R., et al. 1976. "A Land Use and Land Cover Classification System for Use with Remote Sensor Data." *U.S. Geological Survey Professional Paper 964*.
- Cowardin, Lewis M., et al. 1979. *Classification of Wetlands and Deepwater Habitats of the United States*. Washington, D.C.: U.S. Fish and Wildlife Service.
- Florida Topographic Bureau, Thematic Mapping Section. 1985. *Florida Land Use, Cover and Forms Classification System*. Florida Department of Transportation, Procedure No. 550-010-001-a.
- Michigan Land Use Classification and Reference Committee. 1975. *Michigan Land Cover/Use Classification System*. Lansing, Michigan: State of Michigan Office of Land Use.

Other states or government agencies may also have specialized land use/cover studies.

It is recommended that the classification process is begun by defining a classification scheme for the application, using previously developed schemes, like those above, as a general framework.

### Iterative Classification

A process is iterative when it repeats an action. The objective of the ERDAS IMAGINE system is to enable you to iteratively create and refine signatures and classified image files to arrive at a desired final classification. The ERDAS IMAGINE classification utilities are tools to be used as needed, not a numbered list of steps that must always be followed in order.

The total classification can be achieved with either the supervised or unsupervised methods, or a combination of both. Some examples are below:

- Signatures created from both supervised and unsupervised training can be merged and appended together.

- Signature evaluation tools can be used to indicate which signatures are spectrally similar. This helps to determine which signatures should be merged or deleted. These tools also help define optimum band combinations for classification. Using the optimum band combination may reduce the time required to run a classification process.
- Since classifications (supervised or unsupervised) can be based on a particular area of interest (either defined in a raster layer or an .aoi layer), signatures and classifications can be generated from previous classification results.

### **Supervised vs. Unsupervised Training**

In supervised training, it is important to have a set of desired classes in mind, and then create the appropriate signatures from the data. You must also have some way of recognizing pixels that represent the classes that you want to extract.

Supervised classification is usually appropriate when you want to identify relatively few classes, when you have selected training sites that can be verified with ground truth data, or when you can identify distinct, homogeneous regions that represent each class.

On the other hand, if you want the classes to be determined by spectral distinctions that are inherent in the data so that you can define the classes later, then the application is better suited to unsupervised training. Unsupervised training enables you to define many classes easily, and identify classes that are not in contiguous, easily recognized regions.

*NOTE: Supervised classification also includes using a set of classes that is generated from an unsupervised classification. Using a combination of supervised and unsupervised classification may yield optimum results, especially with large data sets (e.g., multiple Landsat scenes). For example, unsupervised classification may be useful for generating a basic set of classes, then supervised classification can be used for further definition of the classes.*

### **Classifying Enhanced Data**

For many specialized applications, classifying data that have been merged, spectrally merged or enhanced—with principal components, image algebra, or other transformations—can produce very specific and meaningful results. However, without understanding the data and the enhancements used, it is recommended that only the original, remotely-sensed data be classified.

### **Dimensionality**

Dimensionality refers to the number of layers being classified. For example, a data file with 3 layers is said to be 3-dimensional, since 3-dimensional feature space is plotted to analyze the data.




---

*Feature space and dimensionality are discussed in ["Math Topics"](#).*

## Adding Dimensions

Using programs in ERDAS IMAGINE, you can add layers to existing image files. Therefore, you can incorporate data (called ancillary data) other than remotely-sensed data into the classification. Using ancillary data enables you to incorporate variables into the classification from, for example, vector layers, previously classified data, or elevation data. The data file values of the ancillary data become an additional feature of each pixel, thus influencing the classification (Jensen, 1996).

## Limiting Dimensions

Although ERDAS IMAGINE allows an unlimited number of layers of data to be used for one classification, it is usually wise to reduce the dimensionality of the data as much as possible. Often, certain layers of data are redundant or extraneous to the task at hand. Unnecessary data take up valuable disk space, and cause the computer system to perform more arduous calculations, which slows down processing.



---

*Use the Signature Editor to evaluate separability to calculate the best subset of layer combinations. Use the Image Interpreter functions to merge or subset layers. Use the Image Information tool (on the Viewer's tool bar) to delete a layer(s).*

## Supervised Training

Supervised training requires *a priori* (already known) information about the data, such as:

- What type of classes need to be extracted? Soil type? Land use? Vegetation?
- What classes are most likely to be present in the data? That is, which types of land cover, soil, or vegetation (or whatever) are represented by the data?

In supervised training, you rely on your own pattern recognition skills and *a priori* knowledge of the data to help the system determine the statistical criteria (signatures) for data classification.

To select reliable samples, you should know some information—either spatial or spectral—about the pixels that you want to classify.

The location of a specific characteristic, such as a land cover type, may be known through ground truthing. Ground truthing refers to the acquisition of knowledge about the study area from field work, analysis of aerial photography, personal experience, etc. Ground truth data are considered to be the most accurate (true) data available about the area of study. They should be collected at the same time as the remotely sensed data, so that the data correspond as much as possible (Star and Estes, 1990). However, some ground data may not be very accurate due to a number of errors and inaccuracies.

## Training Samples and Feature Space Objects

Training samples (also called samples) are sets of pixels that represent what is recognized as a discernible pattern, or potential class. The system calculates statistics from the sample pixels to create a parametric signature for the class.

The following terms are sometimes used interchangeably in reference to training samples. For clarity, they are used in this documentation as follows:

- Training sample, or sample, is a set of pixels selected to represent a potential class. The data file values for these pixels are used to generate a parametric signature.
- Training field, or training site, is the geographical AOI in the image represented by the pixels in a sample. Usually, it is previously identified with the use of ground truth data.

Feature space objects are user-defined AOIs in a feature space image. The feature space signature is based on these objects.

## Selecting Training Samples

It is important that training samples be representative of the class that you are trying to identify. This does not necessarily mean that they must contain a large number of pixels or be dispersed across a wide region of the data. The selection of training samples depends largely upon your knowledge of the data, of the study area, and of the classes that you want to extract.

ERDAS IMAGINE enables you to identify training samples using one or more of the following methods:

- using a vector layer
- defining a polygon in the image
- identifying a training sample of contiguous pixels with similar spectral characteristics
- identifying a training sample of contiguous pixels within a certain area, with or without similar spectral characteristics
- using a class from a thematic raster layer from an image file of the same area (i.e., the result of an unsupervised classification)

## Digitized Polygon

Training samples can be identified by their geographical location (training sites, using maps, ground truth data). The locations of the training sites can be digitized from maps with the ERDAS IMAGINE Vector or AOI tools. Polygons representing these areas are then stored as vector layers. The vector layers can then be used as input to the AOI tools and used as training samples to create signatures.



---

*Use the Vector and AOI tools to digitize training samples from a map. Use the Signature Editor to create signatures from training samples that are identified with digitized polygons.*

## User-defined Polygon

Using your pattern recognition skills (with or without supplemental ground truth information), you can identify samples by examining a displayed image of the data and drawing a polygon around the training site(s) of interest. For example, if it is known that oak trees reflect certain frequencies of green and infrared light according to ground truth data, you may be able to base your sample selections on the data (taking atmospheric conditions, sun angle, time, date, and other variations into account). The area within the polygon(s) would be used to create a signature.



---

*Use the AOI tools to define the polygon(s) to be used as the training sample. Use the Signature Editor to create signatures from training samples that are identified with the polygons.*

## Identify Seed Pixel

With the Seed Properties dialog and AOI tools, the cursor (crosshair) can be used to identify a single pixel (seed pixel) that is representative of the training sample. This seed pixel is used as a model pixel, against which the pixels that are contiguous to it are compared based on parameters specified by you.

When one or more of the contiguous pixels is accepted, the mean of the sample is calculated from the accepted pixels. Then, the pixels contiguous to the sample are compared in the same way. This process repeats until no pixels that are contiguous to the sample satisfy the spectral parameters. In effect, the sample grows outward from the model pixel with each iteration. These homogenous pixels are converted from individual raster pixels to a polygon and used as an AOI layer.



---

*Select the Seed Properties option in the Viewer to identify training samples with a seed pixel.*



## Seed Pixel Method with Spatial Limits

The training sample identified with the seed pixel method can be limited to a particular region by defining the geographic distance and area.



*Vector layers (polygons or lines) can be displayed as the top layer in the Viewer, and the boundaries can then be used as an AOI for training samples defined under Seed Properties.*

## Thematic Raster Layer

A training sample can be defined by using class values from a thematic raster layer (see [Table 38](#)). The data file values in the training sample are used to create a signature. The training sample can be defined by as many class values as desired.

*NOTE: The thematic raster layer must have the same coordinate system as the image file being classified.*

**Table 38: Training Sample Comparison**

Method	Advantages	Disadvantages
Digitized Polygon	precise map coordinates, represents known ground information	may overestimate class variance, time-consuming
User-defined Polygon	high degree of user control	may overestimate class variance, time-consuming
Seed Pixel	auto-assisted, less time	may underestimate class variance
Thematic Raster Layer	allows iterative classifying	must have previously defined thematic layer

## Evaluating Training Samples

Selecting training samples is often an iterative process. To generate signatures that accurately represent the classes to be identified, you may have to repeatedly select training samples, evaluate the signatures that are generated from the samples, and then either take new samples or manipulate the signatures as necessary. Signature manipulation may involve merging, deleting, or appending from one file to another. It is also possible to perform a classification using the known signatures, then mask out areas that are not classified to use in gathering more signatures.



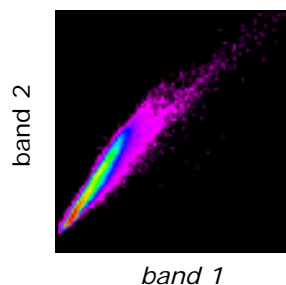
---

See *"Evaluating Signatures"* for methods of determining the accuracy of the signatures created from your training samples.

## Selecting Feature Space Objects

The ERDAS IMAGINE Feature Space tools enable you to interactively define feature space objects (AOIs) in the feature space image(s). A feature space image is simply a graph of the data file values of one band of data against the values of another band (often called a scatterplot). In ERDAS IMAGINE, a feature space image has the same data structure as a raster image; therefore, feature space images can be used with other ERDAS IMAGINE utilities, including zoom, color level slicing, virtual roam, Spatial Modeler, and Map Composer.

**Figure 93: Example of a Feature Space Image**



The transformation of a multilayer raster image into a feature space image is done by mapping the input pixel values to a position in the feature space image. This transformation defines only the pixel position in the feature space image. It does not define the pixel's value.

The pixel values in the feature space image can be the accumulated frequency, which is calculated when the feature space image is defined. The pixel values can also be provided by a thematic raster layer of the same geometry as the source multilayer image. Mapping a thematic layer into a feature space image can be useful for evaluating the validity of the parametric and nonparametric decision boundaries of a classification (Kloer, 1994).



---

*When you display a feature space image file (.fsp.img) in a Viewer, the colors reflect the density of points for both bands. The bright tones represent a high density and the dark tones represent a low density.*

## Create Nonparametric Signature

You can define a feature space object (AOI) in the feature space image and use it directly as a nonparametric signature. Since the Viewers for the feature space image and the image being classified are both linked to the ERDAS IMAGINE Signature Editor, it is possible to mask AOIs from the image being classified to the feature space image, and vice versa. You can also directly link a cursor in the image Viewer to the feature space Viewer. These functions help determine a location for the AOI in the feature space image.

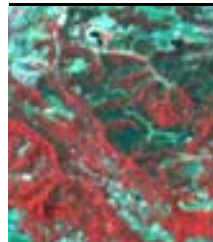
A single feature space image, but multiple AOIs, can be used to define the signature. This signature is taken within the feature space image, not the image being classified. The pixels in the image that correspond to the data file values in the signature (i.e., feature space object) are assigned to that class.

One fundamental difference between using the feature space image to define a training sample and the other traditional methods is that it is a nonparametric signature. The decisions made in the classification process have no dependency on the statistics of the pixels. This helps improve classification accuracies for specific nonnormal classes, such as urban and exposed rock (Faust et al, 1991).

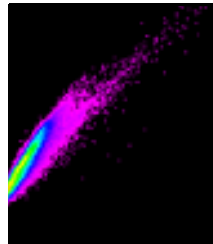


See "[Math Topics](#)" for information on feature space images.

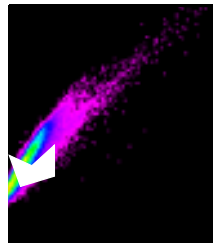
### Figure 94: Process for Defining a Feature Space Object



Display the image file to be classified in a Viewer (layers 3, 2, 1).



Create feature space image from the image file being (layer 1 vs. layer 2).



Draw an AOI (feature space object) around the desired area in the feature space image. Once you have a desired AOI, it can be used as a signature.

A decision rule is used to analyze each pixel in the image file being classified, and the pixels with the corresponding data file values are assigned to the feature space class.

## Evaluate Feature Space Signatures

Using the Feature Space tools, it is also possible to use a feature space signature to generate a mask. Once it is defined as a mask, the pixels under the mask are identified in the image file and highlighted in the Viewer. The image displayed in the Viewer must be the image from which the feature space image was created. This process helps you to visually analyze the correlations between various spectral bands to determine which combination of bands brings out the desired features in the image.

You can have as many feature space images with different band combinations as desired. Any polygon or rectangle in these feature space images can be used as a nonparametric signature. However, only one feature space image can be used per signature. The polygons in the feature space image can be easily modified and/or masked until the desired regions of the image have been identified.



---

*Use the Feature Space tools in the Signature Editor to create a feature space image and mask the signature. Use the AOI tools to draw polygons.*

**Table 39: Feature Space Signatures**

Advantages	Disadvantages
Provide an accurate way to classify a class with a nonnormal distribution (e.g., residential and urban).	The classification decision process allows overlap and unclassified pixels.
Certain features may be more visually identifiable in a feature space image.	The feature space image may be difficult to interpret.
The classification decision process is fast.	

## ■ Unsupervised Training

Unsupervised training requires only minimal initial input from you. However, you have the task of interpreting the classes that are created by the unsupervised training algorithm.

Unsupervised training is also called clustering, because it is based on the natural groupings of pixels in image data when they are plotted in feature space. According to the specified parameters, these groups can later be merged, disregarded, otherwise manipulated, or used as the basis of a signature.



---

*Feature space is explained in ["Math Topics"](#).*

## Clusters

Clusters are defined with a clustering algorithm, which often uses all or many of the pixels in the input data file for its analysis. The clustering algorithm has no regard for the contiguity of the pixels that define each cluster.

- The Iterative Self-Organizing Data Analysis Technique (ISODATA) (Tou and Gonzalez, 1974) clustering method uses spectral distance as in the sequential method, but iteratively classifies the pixels, redefines the criteria for each class, and classifies again, so that the spectral distance patterns in the data gradually emerge.
- The RGB clustering method is more specialized than the ISODATA method. It applies to three-band, 8-bit data. RGB clustering plots pixels in three-dimensional feature space, and divides that space into sections that are used to define clusters.

Each of these methods is explained below, along with its advantages and disadvantages.



---

*Some of the statistics terms used in this section are explained in "Math Topics".*

### ISODATA Clustering

ISODATA is iterative in that it repeatedly performs an entire classification (outputting a thematic raster layer) and recalculates statistics. Self-Organizing refers to the way in which it locates clusters with minimum user input.

The ISODATA method uses minimum spectral distance to assign a cluster for each candidate pixel. The process begins with a specified number of arbitrary cluster means or the means of existing signatures, and then it processes repetitively, so that those means shift to the means of the clusters in the data.

Because the ISODATA method is iterative, it is not biased to the top of the data file, as are the one-pass clustering algorithms.



---

*Use the Unsupervised Classification utility in the Signature Editor to perform ISODATA clustering.*

### ISODATA Clustering Parameters

To perform ISODATA clustering, you specify:

- $N$  - the maximum number of clusters to be considered. Since each cluster is the basis for a class, this number becomes the maximum number of classes to be formed. The ISODATA process begins by determining  $N$  arbitrary cluster means. Some clusters with too few pixels can be eliminated, leaving less than  $N$  clusters.

- $T$  - a convergence threshold, which is the maximum percentage of pixels whose class values are allowed to be unchanged between iterations.
- $M$  - the maximum number of iterations to be performed.

### Initial Cluster Means

On the first iteration of the ISODATA algorithm, the means of  $N$  clusters can be arbitrarily determined. After each iteration, a new mean for each cluster is calculated, based on the actual spectral locations of the pixels in the cluster, instead of the initial arbitrary calculation. Then, these new means are used for defining clusters in the next iteration. The process continues until there is little change between iterations (Swain, 1973).

The initial cluster means are distributed in feature space along a vector that runs between the point at spectral coordinates

$$(\mu_1 - \sigma_1, \mu_2 - \sigma_2, \mu_3 - \sigma_3, \dots, \mu_n - \sigma_n)$$

and the coordinates

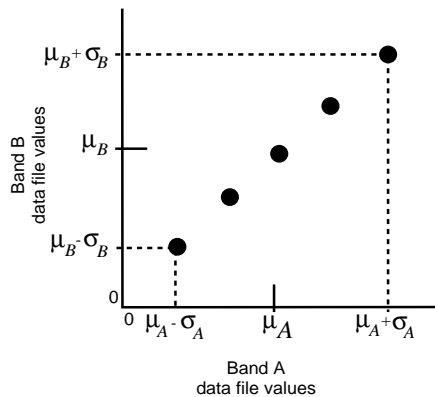
$$(\mu_1 + \sigma_1, \mu_2 + \sigma_2, \mu_3 + \sigma_3, \dots, \mu_n + \sigma_n)$$

Such a vector in two dimensions is illustrated in Figure 95. The initial cluster means are evenly distributed between

$$(\mu_A - \sigma_A, \mu_B - \sigma_B) \text{ and } (\mu_A + \sigma_A, \mu_B + \sigma_B)$$

**Figure 95: ISODATA Arbitrary Clusters**

5 arbitrary cluster means in two-dimensional spectral space



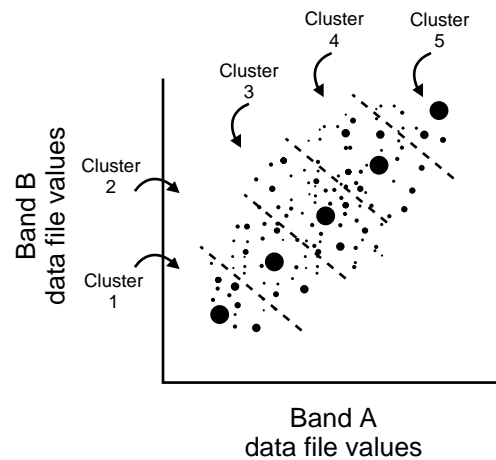
### Pixel Analysis

Pixels are analyzed beginning with the upper left corner of the image and going left to right, block by block.

The spectral distance between the candidate pixel and each cluster mean is calculated. The pixel is assigned to the cluster whose mean is the closest. The ISODATA function creates an output image file with a thematic raster layer and/or a signature file (.sig) as a result of the clustering. At the end of each iteration, an image file exists that shows the assignments of the pixels to the clusters.

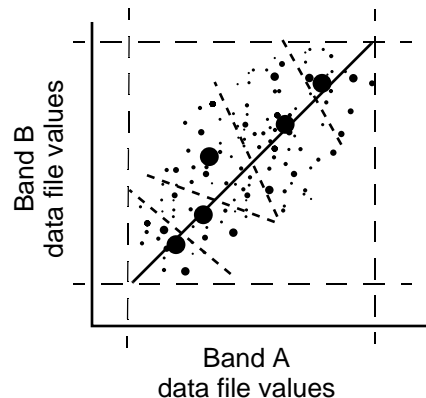
Considering the regular, arbitrary assignment of the initial cluster means, the first iteration of the ISODATA algorithm always gives results similar to those in [Figure 96](#).

**Figure 96: ISODATA First Pass**



For the second iteration, the means of all clusters are recalculated, causing them to shift in feature space. The entire process is repeated—each candidate pixel is compared to the new cluster means and assigned to the closest cluster mean.

**Figure 97: ISODATA Second Pass**



### Percentage Unchanged

After each iteration, the normalized percentage of pixels whose assignments are unchanged since the last iteration is displayed in the dialog. When this number reaches  $T$  (the convergence threshold), the program terminates.

It is possible for the percentage of unchanged pixels to never converge or reach  $T$  (the convergence threshold). Therefore, it may be beneficial to monitor the percentage, or specify a reasonable maximum number of iterations,  $M$ , so that the program does not run indefinitely.

**Table 40: ISODATA Clustering**

Advantages	Disadvantages
Because it is iterative, clustering is not geographically biased to the top or bottom pixels of the data file.	The clustering process is time-consuming, because it can repeat many times.
This algorithm is highly successful at finding the spectral clusters that are inherent in the data. It does not matter where the initial arbitrary cluster means are located, as long as enough iterations are allowed.	Does not account for pixel spatial homogeneity.
A preliminary thematic raster layer is created, which gives results similar to using a minimum distance classifier (as explained below) on the signatures that are created. This thematic raster layer can be used for analyzing and manipulating the signatures before actual classification takes place.	

**Principal Component Method**

Whereas clustering creates signatures depending on pixels' spectral reflectance by adding pixels together, the principal component method actually subtracts pixels. Principal Components Analysis (PCA) is a method of data compression. With it, you can eliminate data that is redundant by compacting it into fewer bands.

The resulting bands are noncorrelated and independent. You may find these bands more interpretable than the source data. PCA can be performed on up to 256 bands with ERDAS IMAGINE. As a type of spectral enhancement, you are required to specify the number of components you want output from the original data.

**Recommended Decision Rule**

Although the ISODATA algorithm is the most similar to the minimum distance decision rule, the signatures can produce good results with any type of classification. Therefore, no particular decision rule is recommended over others.

In most cases, the signatures created by ISODATA are merged, deleted, or appended to other signature sets. The image file created by ISODATA is the same as the image file that is created by a minimum distance classification, except for the nonconvergent pixels (100-7% of the pixels).





---

*Use the Merge and Delete options in the Signature Editor to manipulate signatures.*



---

*Use the Unsupervised Classification utility in the Signature Editor to perform ISODATA clustering, generate signatures, and classify the resulting signatures.*

## RGB Clustering



---

*The RGB Clustering and Advanced RGB Clustering functions in Image Interpreter create a thematic raster layer. However, no signature file is created and no other classification decision rule is used. In practice, RGB Clustering differs greatly from the other clustering methods, but it does employ a clustering algorithm.*

RGB clustering is a simple classification and data compression technique for three bands of data. It is a fast and simple algorithm that quickly compresses a three-band image into a single band pseudocolor image, without necessarily classifying any particular features.

The algorithm plots all pixels in 3-dimensional feature space and then partitions this space into clusters on a grid. In the more simplistic version of this function, each of these clusters becomes a class in the output thematic raster layer.

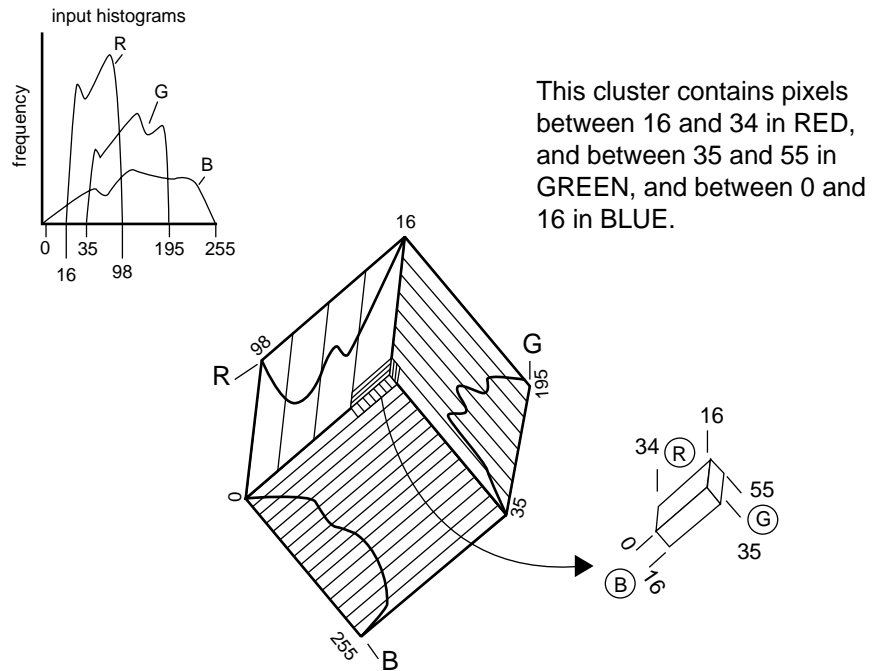
The advanced version requires that a minimum threshold on the clusters be set so that only clusters at least as large as the threshold become output classes. This allows for more color variation in the output file. Pixels that do not fall into any of the remaining clusters are assigned to the cluster with the smallest city-block distance from the pixel. In this case, the city-block distance is calculated as the sum of the distances in the red, green, and blue directions in 3-dimensional space.

Along each axis of the three-dimensional scatterplot, each input histogram is scaled so that the partitions divide the histograms between specified limits—either a specified number of standard deviations above and below the mean, or between the minimum and maximum data values for each band.

The default number of divisions per band is listed below:

- Red is divided into 7 sections (32 for advanced version)
- Green is divided into 6 sections (32 for advanced version)
- Blue is divided into 6 sections (32 for advanced version)

**Figure 98: RGB Clustering**



### Partitioning Parameters

It is necessary to specify the number of R, G, and B sections in each dimension of the 3-dimensional scatterplot. The number of sections should vary according to the histograms of each band. Broad histograms should be divided into more sections, and narrow histograms should be divided into fewer sections (see [Figure 98](#)).



*It is possible to interactively change these parameters in the RGB Clustering function in the Image Interpreter. The number of classes is calculated based on the current parameters, and it displays on the command screen.*

**Table 41: RGB Clustering**

Advantages	Disadvantages
The fastest classification method. It is designed to provide a fast, simple classification for applications that do not require specific classes.	Exactly three bands must be input, which is not suitable for all applications.
Not biased to the top or bottom of the data file. The order in which the pixels are examined does not influence the outcome.	Does not always create thematic classes that can be analyzed for informational purposes.

**Table 41: RGB Clustering**

<b>Advantages</b>	<b>Disadvantages</b>
(Advanced version only) A highly interactive function, allowing an iterative adjustment of the parameters until the number of clusters and the thresholds are satisfactory for analysis.	

### Tips

Some starting values that usually produce good results with the simple RGB clustering are:

$$R = 7$$

$$G = 6$$

$$B = 6$$

which results in  $7 \times 6 \times 6 = 252$  classes.

To decrease the number of output colors/classes or to darken the output, decrease these values.

For the Advanced RGB clustering function, start with higher values for R, G, and B. Adjust by raising the threshold parameter and/or decreasing the R, G, and B parameter values until the desired number of output classes is obtained.

## Signature Files

A signature is a set of data that defines a training sample, feature space object (AOI), or cluster. The signature is used in a classification process. Each classification decision rule (algorithm) requires some signature attributes as input—these are stored in the signature file (.sig). Signatures in ERDAS IMAGINE can be parametric or nonparametric.

The following attributes are standard for all signatures (parametric and nonparametric):

- **name**—identifies the signature and is used as the class name in the output thematic raster layer. The default signature name is Class <number>.
- **color**—the color for the signature and the color for the class in the output thematic raster layer. This color is also used with other signature visualization functions, such as alarms, masking, ellipses, etc.
- **value**—the output class value for the signature. The output class value does not necessarily need to be the class number of the signature. This value should be a positive integer.

- order—the order to process the signatures for order-dependent processes, such as signature alarms and parallelepiped classifications.
- parallelepiped limits—the limits used in the parallelepiped classification.

### Parametric Signature

A parametric signature is based on statistical parameters (e.g., mean and covariance matrix) of the pixels that are in the training sample or cluster. A parametric signature includes the following attributes in addition to the standard attributes for signatures:

- the number of bands in the input image (as processed by the training program)
- the minimum and maximum data file value in each band for each sample or cluster (minimum vector and maximum vector)
- the mean data file value in each band for each sample or cluster (mean vector)
- the covariance matrix for each sample or cluster
- the number of pixels in the sample or cluster

### Nonparametric Signature

A nonparametric signature is based on an AOI that you define in the feature space image for the image file being classified. A nonparametric classifier uses a set of nonparametric signatures to assign pixels to a class based on their location, either inside or outside the area in the feature space image.




---

*The format of the .sig file is described in the On-Line Help. Information on these statistics can be found in ["Math Topics"](#).*

## Evaluating Signatures

Once signatures are created, they can be evaluated, deleted, renamed, and merged with signatures from other files. Merging signatures enables you to perform complex classifications with signatures that are derived from more than one training method (supervised and/or unsupervised, parametric and/or nonparametric).




---

*Use the Signature Editor to view the contents of each signature, manipulate signatures, and perform your own mathematical tests on the statistics.*

## Using Signature Data

There are tests to perform that can help determine whether the signature data are a true representation of the pixels to be classified for each class. You can evaluate signatures that were created either from supervised or unsupervised training. The evaluation methods in ERDAS IMAGINE include:

- Alarm—using your own pattern recognition ability, you view the estimated classified area for a signature (using the parallelepiped decision rule) against a display of the original image.
- Ellipse—view ellipse diagrams and scatterplots of data file values for every pair of bands.
- Contingency matrix—do a quick classification of the pixels in a set of training samples to see what percentage of the sample pixels are actually classified as expected. These percentages are presented in a contingency matrix. This method is for supervised training only, for which polygons of the training samples exist.
- Divergence—measure the divergence (statistical distance) between signatures and determine band subsets that maximize the classification.
- Statistics and histograms—analyze statistics and histograms of the signatures to make evaluations and comparisons.

*NOTE: If the signature is nonparametric (i.e., a feature space signature), you can use only the alarm evaluation method.*

After analyzing the signatures, it would be beneficial to merge or delete them, eliminate redundant bands from the data, add new bands of data, or perform any other operations to improve the classification.

### Alarm

The alarm evaluation enables you to compare an estimated classification of one or more signatures against the original data, as it appears in the Viewer. According to the parallelepiped decision rule, the pixels that fit the classification criteria are highlighted in the displayed image. You also have the option to indicate an overlap by having it appear in a different color.

With this test, you can use your own pattern recognition skills, or some ground truth data, to determine the accuracy of a signature.



---

*Use the Signature Alarm utility in the Signature Editor to perform n-dimensional alarms on the image in the Viewer, using the parallelepiped decision rule. The alarm utility creates a functional layer, and the Viewer allows you to toggle between the image layer and the functional layer.*

## Ellipse

In this evaluation, ellipses of concentration are calculated with the means and standard deviations stored in the signature file. It is also possible to generate parallelepiped rectangles, means, and labels.

In this evaluation, the mean and the standard deviation of every signature are used to represent the ellipse in 2-dimensional feature space. The ellipse is displayed in a feature space image.

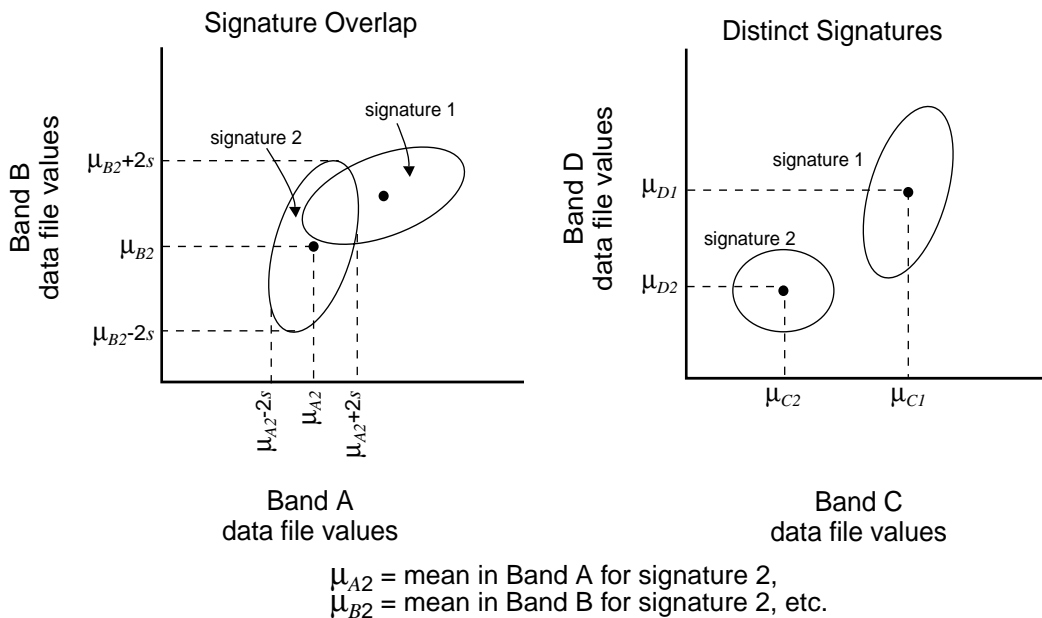


*Ellipses are explained and illustrated in "Math Topics" under the discussion of Scatterplots.*

When the ellipses in the feature space image show extensive overlap, then the spectral characteristics of the pixels represented by the signatures cannot be distinguished in the two bands that are graphed. In the best case, there is no overlap. Some overlap, however, is expected.

Figure 99 shows how ellipses are plotted and how they can overlap. The first graph shows how the ellipses are plotted based on the range of 2 standard deviations from the mean. This range can be altered, changing the ellipse plots. Analyzing the plots with differing numbers of standard deviations is useful for determining the limits of a parallelepiped classification.

**Figure 99: Ellipse Evaluation of Signatures**



By analyzing the ellipse graphs for all band pairs, you can determine which signatures and which bands provide accurate classification results.



---

*Use the Signature Editor to create a feature space image and to view an ellipse(s) of signature data.*

## Contingency Matrix

*NOTE: This evaluation classifies all of the pixels in the selected AOIs and compares the results to the pixels of a training sample.*

The pixels of each training sample are not always so homogeneous that every pixel in a sample is actually classified to its corresponding class. Each sample pixel only weights the statistics that determine the classes. However, if the signature statistics for each sample are distinct from those of the other samples, then a high percentage of each sample's pixels is classified as expected.

In this evaluation, a quick classification of the sample pixels is performed using the minimum distance, maximum likelihood, or Mahalanobis distance decision rule. Then, a contingency matrix is presented, which contains the number and percentages of pixels that are classified as expected.



---

*Use the Signature Editor to perform the contingency matrix evaluation.*

## Separability

Signature separability is a statistical measure of distance between two signatures. Separability can be calculated for any combination of bands that is used in the classification, enabling you to rule out any bands that are not useful in the results of the classification.

For the distance (Euclidean) evaluation, the spectral distance between the mean vectors of each pair of signatures is computed. If the spectral distance between two samples is not significant for any pair of bands, then they may not be distinct enough to produce a successful classification.

The spectral distance is also the basis of the minimum distance classification (as explained below). Therefore, computing the distances between signatures helps you predict the results of a minimum distance classification.



---

*Use the Signature Editor to compute signature separability and distance and automatically generate the report.*

The formulas used to calculate separability are related to the maximum likelihood decision rule. Therefore, evaluating signature separability helps you predict the results of a maximum likelihood classification. The maximum likelihood decision rule is explained below.

There are three options for calculating the separability. All of these formulas take into account the covariances of the signatures in the bands being compared, as well as the mean vectors of the signatures.



Refer to “[Math Topics](#)” for information on the mean vector and covariance matrix.

## Divergence

The formula for computing Divergence ( $D_{ij}$ ) is as follows:

$$D_{ij} = \frac{1}{2}\text{tr}((C_i - C_j)(C_i^{-1} - C_j^{-1})) + \frac{1}{2}\text{tr}((C_i^{-1} - C_j^{-1})(\mu_i - \mu_j)(\mu_i - \mu_j)^T)$$

Where:

- $i$  and  $j$  = the two signatures (classes) being compared
- $C_i$  = the covariance matrix of signature  $i$
- $\mu_i$  = the mean vector of signature  $i$
- $tr$  = the trace function (matrix algebra)
- $T$  = the transposition function

Source: [Swain and Davis, 1978](#)

## Transformed Divergence

The formula for computing Transformed Divergence ( $TD$ ) is as follows:

$$D_{ij} = \frac{1}{2}\text{tr}((C_i - C_j)(C_i^{-1} - C_j^{-1})) + \frac{1}{2}\text{tr}((C_i^{-1} - C_j^{-1})(\mu_i - \mu_j)(\mu_i - \mu_j)^T)$$

$$TD_{ij} = 2000\left(1 - \exp\left(\frac{-D_{ij}}{8}\right)\right)$$

Where:

- $i$  and  $j$  = the two signatures (classes) being compared
- $C_i$  = the covariance matrix of signature  $i$
- $\mu_i$  = the mean vector of signature  $i$
- $tr$  = the trace function (matrix algebra)
- $T$  = the transposition function

Source: [Swain and Davis, 1978](#)



According to Jensen, the transformed divergence “gives an exponentially decreasing weight to increasing distances between the classes.” The scale of the divergence values can range from 0 to 2,000. Interpreting your results after applying transformed divergence requires you to analyze those numerical divergence values. As a general rule, if the result is greater than 1,900, then the classes can be separated. Between 1,700 and 1,900, the separation is fairly good. Below 1,700, the separation is poor (Jensen, 1996).

### Jeffries-Matusita Distance

The formula for computing Jeffries-Matusita Distance ( $JM$ ) is as follows:

$$\alpha = \frac{1}{8}(\mu_i - \mu_j)^T \left( \frac{C_i + C_j}{2} \right)^{-1} (\mu_i - \mu_j) + \frac{1}{2} \ln \left( \frac{|(C_i + C_j)/2|}{\sqrt{|C_i| \times |C_j|}} \right)$$

$$JM_{ij} = \sqrt{2(1 - e^{-\alpha})}$$

Where:

- $i$  and  $j$  = the two signatures (classes) being compared
- $C_i$  = the covariance matrix of signature  $i$
- $\mu_i$  = the mean vector of signature  $i$
- $\ln$  = the natural logarithm function
- $|C_i|$  = the determinant of  $C_i$  (matrix algebra)

Source: Swain and Davis, 1978

According to Jensen, “The JM distance has a saturating behavior with increasing class separation like transformed divergence. However, it is not as computationally efficient as transformed divergence” (Jensen, 1996).

### Separability

Both transformed divergence and Jeffries-Matusita distance have upper and lower bounds. If the calculated divergence is equal to the appropriate upper bound, then the signatures can be said to be totally separable in the bands being studied. A calculated divergence of zero means that the signatures are inseparable.

- $TD$  is between 0 and 2000.
- $JM$  is between 0 and 1414.

A separability listing is a report of the computed divergence for every class pair and one band combination. The listing contains every divergence value for the bands studied for every possible pair of signatures.

The separability listing also contains the average divergence and the minimum divergence for the band set. These numbers can be compared to other separability listings (for other band combinations), to determine which set of bands is the most useful for classification.

### Weight Factors

As with the Bayesian classifier (explained below with maximum likelihood), weight factors may be specified for each signature. These weight factors are based on a priori probabilities that any given pixel is assigned to each class. For example, if you know that twice as many pixels should be assigned to Class A as to Class B, then Class A should receive a weight factor that is twice that of Class B.

*NOTE: The weight factors do not influence the divergence equations (for TD or JM), but they do influence the report of the best average and best minimum separability.*

The weight factors for each signature are used to compute a weighted divergence with the following calculation:

$$W_{ij} = \frac{\sum_{i=1}^{c-1} \left( \sum_{j=i+1}^c f_i f_j U_{ij} \right)}{\frac{1}{2} \left[ \left( \sum_{i=1}^c f_i \right)^2 - \sum_{i=1}^c f_i^2 \right]}$$

Where:

- $i$  and  $j$  = the two signatures (classes) being compared
- $U_{ij}$  = the unweighted divergence between  $i$  and  $j$
- $W_{ij}$  = the weighted divergence between  $i$  and  $j$
- $c$  = the number of signatures (classes)
- $f_i$  = the weight factor for signature  $i$

### Probability of Error

The Jeffries-Matusita distance is related to the pairwise probability of error, which is the probability that a pixel assigned to class  $i$  is actually in class  $j$ . Within a range, this probability can be estimated according to the expression below:

$$\frac{1}{16}(2 - JM_{ij}^2)^2 \leq P_e \leq 1 - \frac{1}{2}\left(1 + \frac{1}{2}JM_{ij}^2\right)$$

Where:

$i$  and  $j$  = the signatures (classes) being compared

$JM_{ij}$  = the Jeffries-Matusita distance between  $i$  and  $j$

$P_e$  = the probability that a pixel is misclassified from  $i$  to  $j$

Source: [Swain and Davis, 1978](#)

## Signature Manipulation

In many cases, training must be repeated several times before the desired signatures are produced. Signatures can be gathered from different sources—different training samples, feature space images, and different clustering programs—all using different techniques. After each signature file is evaluated, you may merge, delete, or create new signatures. The desired signatures can finally be moved to one signature file to be used in the classification.

The following operations upon signatures and signature files are possible with ERDAS IMAGINE:

- View the contents of the signature statistics
- View histograms of the samples or clusters that were used to derive the signatures
- Delete unwanted signatures
- Merge signatures together, so that they form one larger class when classified
- Append signatures from other files. You can combine signatures that are derived from different training methods for use in one classification.



---

*Use the Signature Editor to view statistics and histogram listings and to delete, merge, append, and rename signatures within a signature file.*

## Classification Decision Rules

Once a set of reliable signatures has been created and evaluated, the next step is to perform a classification of the data. Each pixel is analyzed independently. The measurement vector for each pixel is compared to each signature, according to a decision rule, or algorithm. Pixels that pass the criteria that are established by the decision rule are then assigned to the class for that signature. ERDAS IMAGINE enables you to classify the data both parametrically with statistical representation, and nonparametrically as objects in feature space. [Figure 100](#) shows the flow of an image pixel through the classification decision making process in ERDAS IMAGINE ([Kloer, 1994](#)).

If a nonparametric rule is not set, then the pixel is classified using only the parametric rule. All of the parametric signatures are tested. If a nonparametric rule is set, the pixel is tested against all of the signatures with nonparametric definitions. This rule results in the following conditions:

- If the nonparametric test results in one unique class, the pixel is assigned to that class.
- If the nonparametric test results in zero classes (i.e., the pixel lies outside all the nonparametric decision boundaries), then the unclassified rule is applied. With this rule, the pixel is either classified by the parametric rule or left unclassified.
- If the pixel falls into more than one class as a result of the nonparametric test, the overlap rule is applied. With this rule, the pixel is either classified by the parametric rule, processing order, or left unclassified.

### **Nonparametric Rules**

ERDAS IMAGINE provides these decision rules for nonparametric signatures:

- parallelepiped
- feature space

### **Unclassified Options**

ERDAS IMAGINE provides these options if the pixel is not classified by the nonparametric rule:

- parametric rule
- unclassified

### **Overlap Options**

ERDAS IMAGINE provides these options if the pixel falls into more than one feature space object:

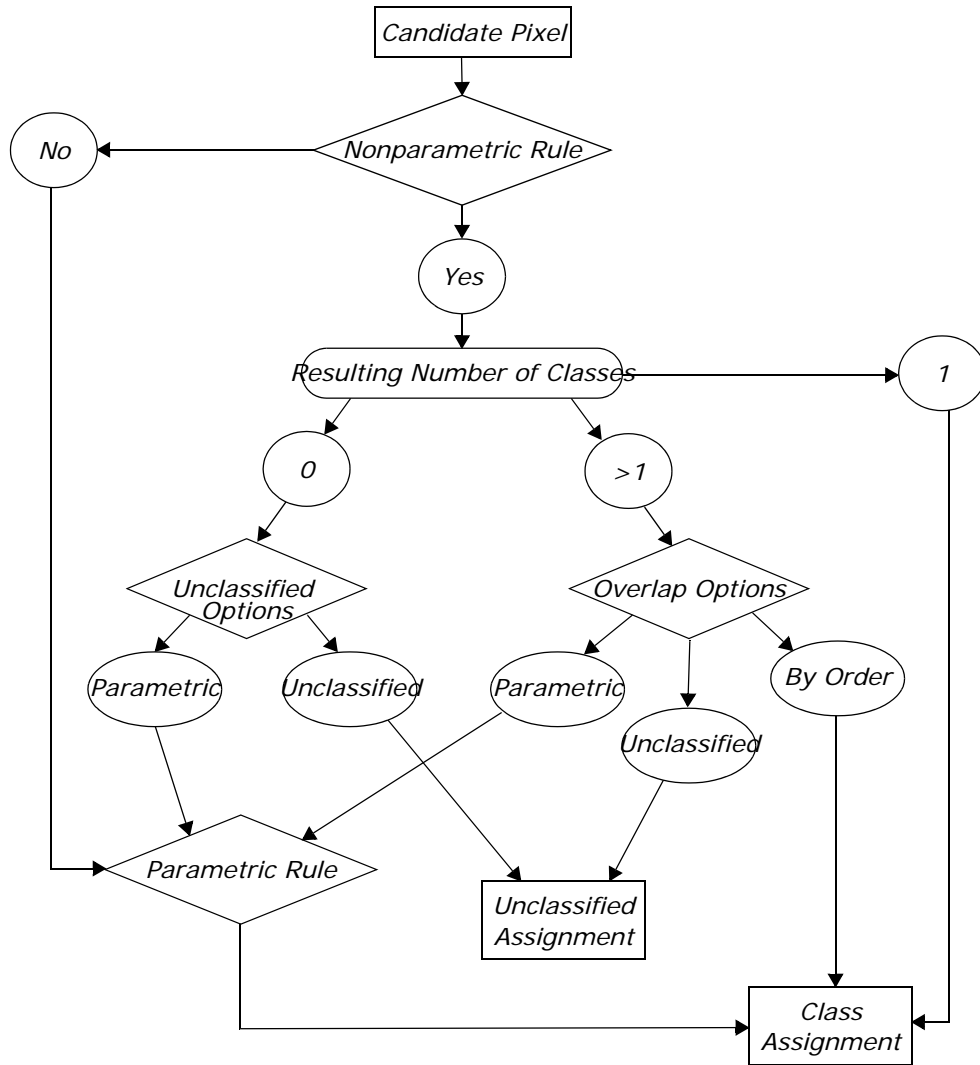
- parametric rule
- by order
- unclassified

### **Parametric Rules**

ERDAS IMAGINE provides these commonly-used decision rules for parametric signatures:

- minimum distance
- Mahalanobis distance
- maximum likelihood (with Bayesian variation)

**Figure 100: Classification Flow Diagram**



**Parallelepiped**

In the parallelepiped decision rule, the data file values of the candidate pixel are compared to upper and lower limits. These limits can be either:

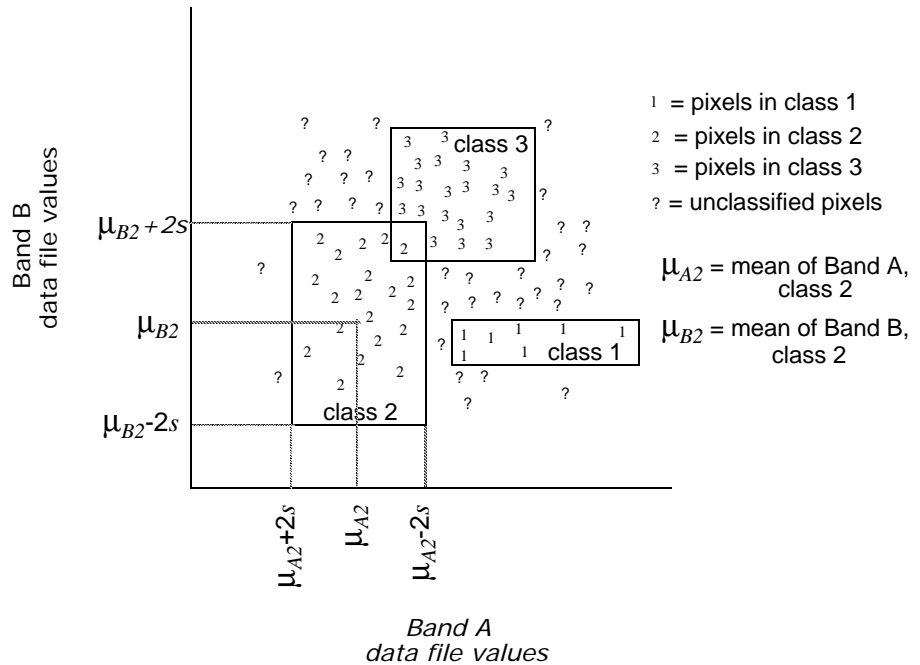
- the minimum and maximum data file values of each band in the signature,
- the mean of each band, plus and minus a number of standard deviations, or
- any limits that you specify, based on your knowledge of the data and signatures. This knowledge may come from the signature evaluation techniques discussed above.



*These limits can be set using the Parallelepiped Limits utility in the Signature Editor.*

There are high and low limits for every signature in every band. When a pixel's data file values are between the limits for every band in a signature, then the pixel is assigned to that signature's class. [Figure 101](#) is a two-dimensional example of a parallelepiped classification.

**Figure 101: Parallelepiped Classification Using  $\pm$  Two Standard Deviations as Limits**



The large rectangles in [Figure 101](#) are called parallelepipeds. They are the regions within the limits for each signature.

### Overlap Region

In cases where a pixel may fall into the overlap region of two or more parallelepipeds, you must define how the pixel can be classified.

- The pixel can be classified by the order of the signatures. If one of the signatures is first and the other signature is fourth, the pixel is assigned to the first signature's class. This order can be set in the Signature Editor.
- The pixel can be classified by the defined parametric decision rule. The pixel is tested against the overlapping signatures only. If neither of these signatures is parametric, then the pixel is left unclassified. If only one of the signatures is parametric, then the pixel is automatically assigned to that signature's class.
- The pixel can be left unclassified.

## Regions Outside of the Boundaries

If the pixel does not fall into one of the parallelepipeds, then you must define how the pixel can be classified.

- The pixel can be classified by the defined parametric decision rule. The pixel is tested against all of the parametric signatures. If none of the signatures is parametric, then the pixel is left unclassified.
- The pixel can be left unclassified.

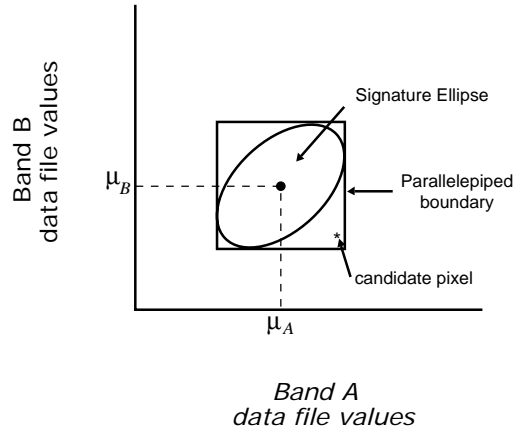


*Use the Supervised Classification utility in the Signature Editor to perform a parallelepiped classification.*

**Table 42: Parallelepiped Decision Rule**

Advantages	Disadvantages
Fast and simple, since the data file values are compared to limits that remain constant for each band in each signature.	Since parallelepipeds have corners, pixels that are actually quite far, spectrally, from the mean of the signature may be classified. An example of this is shown in <a href="#">Figure 102</a> .
Often useful for a first-pass, broad classification, this decision rule quickly narrows down the number of possible classes to which each pixel can be assigned before the more time-consuming calculations are made, thus cutting processing time (e.g., minimum distance, Mahalanobis distance, or maximum likelihood).	
Not dependent on normal distributions.	

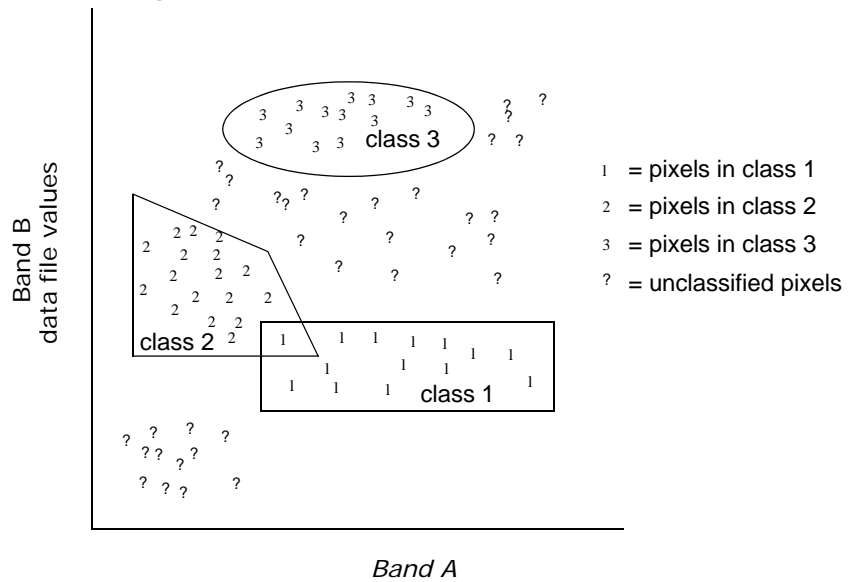
**Figure 102: Parallelepiped Corners Compared to the Signature Ellipse**



**Feature Space**

The feature space decision rule determines whether or not a candidate pixel lies within the nonparametric signature in the feature space image. When a pixel's data file values are in the feature space signature, then the pixel is assigned to that signature's class. Figure 103 is a two-dimensional example of a feature space classification. The polygons in this figure are AOIs used to define the feature space signatures.

**Figure 103: Feature Space Classification**



**Overlap Region**

In cases where a pixel may fall into the overlap region of two or more AOIs, you must define how the pixel can be classified.

- The pixel can be classified by the order of the feature space signatures. If one of the signatures is first and the other signature is fourth, the pixel is assigned to the first signature's class. This order can be set in the Signature Editor.



- The pixel can be classified by the defined parametric decision rule. The pixel is tested against the overlapping signatures only. If neither of these feature space signatures is parametric, then the pixel is left unclassified. If only one of the signatures is parametric, then the pixel is assigned automatically to that signature's class.
- The pixel can be left unclassified.

### Regions Outside of the AOIs

If the pixel does not fall into one of the AOIs for the feature space signatures, then you must define how the pixel can be classified.

- The pixel can be classified by the defined parametric decision rule. The pixel is tested against all of the parametric signatures. If none of the signatures is parametric, then the pixel is left unclassified.
- The pixel can be left unclassified.

**Table 43: Feature Space Decision Rule**

Advantages	Disadvantages
Often useful for a first-pass, broad classification.	The feature space decision rule allows overlap and unclassified pixels.
Provides an accurate way to classify a class with a nonnormal distribution (e.g., residential and urban).	The feature space image may be difficult to interpret.
Certain features may be more visually identifiable, which can help discriminate between classes that are spectrally similar and hard to differentiate with parametric information.	
The feature space method is fast.	

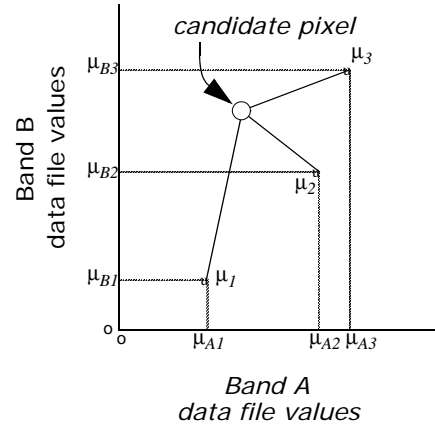


*Use the Decision Rules utility in the Signature Editor to perform a feature space classification.*

### Minimum Distance

The minimum distance decision rule (also called spectral distance) calculates the spectral distance between the measurement vector for the candidate pixel and the mean vector for each signature.

**Figure 104: Minimum Spectral Distance**



In [Figure 104](#), spectral distance is illustrated by the lines from the candidate pixel to the means of the three signatures. The candidate pixel is assigned to the class with the closest mean.

The equation for classifying by spectral distance is based on the equation for Euclidean distance:

$$SD_{xyc} = \sqrt{\sum_{i=1}^n (\mu_{ci} - X_{xyi})^2}$$

Where:

- $n$  = number of bands (dimensions)
- $i$  = a particular band
- $c$  = a particular class
- $X_{xyi}$  = data file value of pixel  $x,y$  in band  $i$
- $\mu_{ci}$  = mean of data file values in band  $i$  for the sample for class  $c$
- $SD_{xyc}$  = spectral distance from pixel  $x,y$  to the mean of class  $c$

Source: [Swain and Davis, 1978](#)

When spectral distance is computed for all possible values of  $c$  (all possible classes), the class of the candidate pixel is assigned to the class for which  $SD$  is the lowest.

**Table 44: Minimum Distance Decision Rule**

Advantages	Disadvantages
<p>Since every pixel is spectrally closer to either one sample mean or another, there are no unclassified pixels.</p>	<p>Pixels that should be unclassified (i.e., they are not spectrally close to the mean of any sample, within limits that are reasonable to you) become classified. However, this problem is alleviated by thresholding out the pixels that are farthest from the means of their classes. (See the discussion on "Thresholding".)</p>
<p>The fastest decision rule to compute, except for parallelepiped.</p>	<p>Does not consider class variability. For example, a class like an urban land cover class is made up of pixels with a high variance, which may tend to be farther from the mean of the signature. Using this decision rule, outlying urban pixels may be improperly classified. Inversely, a class with less variance, like water, may tend to overclassify (that is, classify more pixels than are appropriate to the class), because the pixels that belong to the class are usually spectrally closer to their mean than those of other classes to their means.</p>

**Mahalanobis Distance**



*The Mahalanobis distance algorithm assumes that the histograms of the bands have normal distributions. If this is not the case, you may have better results with the parallelepiped or minimum distance decision rule, or by performing a first-pass parallelepiped classification.*

Mahalanobis distance is similar to minimum distance, except that the covariance matrix is used in the equation. Variance and covariance are figured in so that clusters that are highly varied lead to similarly varied classes, and vice versa. For example, when classifying urban areas—typically a class whose pixels vary widely—correctly classified pixels may be farther from the mean than those of a class for water, which is usually not a highly varied class (Swain and Davis, 1978).

The equation for the Mahalanobis distance classifier is as follows:

$$D = (X - M_c)^T (Cov_c^{-1}) (X - M_c)$$

Where:

- $D$  = Mahalanobis distance
- $c$  = a particular class
- $X$  = the measurement vector of the candidate pixel
- $M_c$  = the mean vector of the signature of class  $c$
- $Cov_c$  = the covariance matrix of the pixels in the signature of class  $c$
- $Cov_c^{-1}$  = inverse of  $Cov_c$
- $T$  = transposition function

The pixel is assigned to the class,  $c$ , for which  $D$  is the lowest.

**Table 45: Mahalanobis Decision Rule**

Advantages	Disadvantages
Takes the variability of classes into account, unlike minimum distance or parallelepiped.	Tends to overclassify signatures with relatively large values in the covariance matrix. If there is a large dispersion of the pixels in a cluster or training sample, then the covariance matrix of that signature contains large values.
May be more useful than minimum distance in cases where statistical criteria (as expressed in the covariance matrix) must be taken into account, but the weighting factors that are available with the maximum likelihood/Bayesian option are not needed.	Slower to compute than parallelepiped or minimum distance.
	Mahalanobis distance is parametric, meaning that it relies heavily on a normal distribution of the data in each input band.

**Maximum Likelihood/Bayesian**



*The maximum likelihood algorithm assumes that the histograms of the bands of data have normal distributions. If this is not the case, you may have better results with the parallelepiped or minimum distance decision rule, or by performing a first-pass parallelepiped classification.*

The maximum likelihood decision rule is based on the probability that a pixel belongs to a particular class. The basic equation assumes that these probabilities are equal for all classes, and that the input bands have normal distributions.

## Bayesian Classifier

If you have a priori knowledge that the probabilities are not equal for all classes, you can specify weight factors for particular classes. This variation of the maximum likelihood decision rule is known as the Bayesian decision rule (Hord, 1982). Unless you have a priori knowledge of the probabilities, it is recommended that they not be specified. In this case, these weights default to 1.0 in the equation.

The equation for the maximum likelihood/Bayesian classifier is as follows:

$$D = \ln(a_c) - [0.5 \ln(|Cov_c|)] - [0.5 (X - M_c)T (Cov_c^{-1}) (X - M_c)]$$

Where:

$D$  = weighted distance (likelihood)

$c$  = a particular class

$X$  = the measurement vector of the candidate pixel

$M_c$  = the mean vector of the sample of class  $c$

$a_c$  = percent probability that any candidate pixel is a member of class  $c$  (defaults to 1.0, or is entered from *a priori* knowledge)

$Cov_c$  = the covariance matrix of the pixels in the sample of class  $c$

$|Cov_c|$  = determinant of  $Cov_c$  (matrix algebra)

$Cov_c^{-1}$  = inverse of  $Cov_c$  (matrix algebra)

$\ln$  = natural logarithm function

$T$  = transposition function (matrix algebra)

The inverse and determinant of a matrix, along with the difference and transposition of vectors, would be explained in a textbook of matrix algebra.

The pixel is assigned to the class,  $c$ , for which  $D$  is the lowest.

**Table 46: Maximum Likelihood/Bayesian Decision Rule**

Advantages	Disadvantages
The most accurate of the classifiers in the ERDAS IMAGINE system (if the input samples/clusters have a normal distribution), because it takes the most variables into consideration.	An extensive equation that takes a long time to compute. The computation time increases with the number of input bands.
Takes the variability of classes into account by using the covariance matrix, as does Mahalanobis distance.	Maximum likelihood is parametric, meaning that it relies heavily on a normal distribution of the data in each input band.

**Table 46: Maximum Likelihood/Bayesian Decision Rule**

Advantages	Disadvantages
	Tends to overclassify signatures with relatively large values in the covariance matrix. If there is a large dispersion of the pixels in a cluster or training sample, then the covariance matrix of that signature contains large values.

## Fuzzy Methodology

### Fuzzy Classification

The Fuzzy Classification method takes into account that there are pixels of mixed make-up, that is, a pixel cannot be definitively assigned to one category. Jensen notes that, “Clearly, there needs to be a way to make the classification algorithms more sensitive to the imprecise (fuzzy) nature of the real world” (Jensen, 1996).

Fuzzy classification is designed to help you work with data that may not fall into exactly one category or another. Fuzzy classification works using a membership function, wherein a pixel’s value is determined by whether it is closer to one class than another. A fuzzy classification does not have definite boundaries, and each pixel can belong to several different classes (Jensen, 1996).

Like traditional classification, fuzzy classification still uses training, but the biggest difference is that “it is also possible to obtain information on the various constituent classes found in a mixed pixel. . .” (Jensen, 1996). Jensen goes on to explain that the process of collecting training sites in a fuzzy classification is not as strict as a traditional classification. In the fuzzy method, the training sites do not have to have pixels that are exactly the same.

Once you have a fuzzy classification, the fuzzy convolution utility allows you to perform a moving window convolution on a fuzzy classification with multiple output class assignments. Using the multilayer classification and distance file, the convolution creates a new single class output file by computing a total weighted distance for all classes in the window.

### Fuzzy Convolution

The Fuzzy Convolution operation creates a single classification layer by calculating the total weighted inverse distance of all the classes in a window of pixels. Then, it assigns the center pixel in the class with the largest total inverse distance summed over the entire set of fuzzy classification layers.

This has the effect of creating a context-based classification to reduce the speckle or salt and pepper in the classification. Classes with a very small distance value remain unchanged while classes with higher distance values may change to a neighboring value if there is a sufficient number of neighboring pixels with class values and small corresponding distance values. The following equation is used in the calculation:

$$T[k] = \sum_{i=0}^{s-1} \sum_{j=0}^{s-1} \sum_{l=0}^{n-1} \frac{W_{ijl}}{D_{ijl}[k]}$$

Where:

- $i$  = row index of window
- $j$  = column index of window
- $s$  = size of window (3, 5, or 7)
- $l$  = layer index of fuzzy set
- $n$  = number of fuzzy layers used
- $W$  = weight table for window
- $k$  = class value
- $D[k]$  = distance file value for class  $k$
- $T[k]$  = total weighted distance of window for class  $k$

The center pixel is assigned the class with the maximum  $T[k]$ .

## Expert Classification

Expert classification can be performed using the IMAGINE Expert Classifier™. The expert classification software provides a rules-based approach to multispectral image classification, post-classification refinement, and GIS modeling. In essence, an expert classification system is a hierarchy of rules, or a decision tree, that describes the conditions under which a set of low level constituent information gets abstracted into a set of high level informational classes. The constituent information consists of user-defined variables and includes raster imagery, vector coverages, spatial models, external programs, and simple scalars.

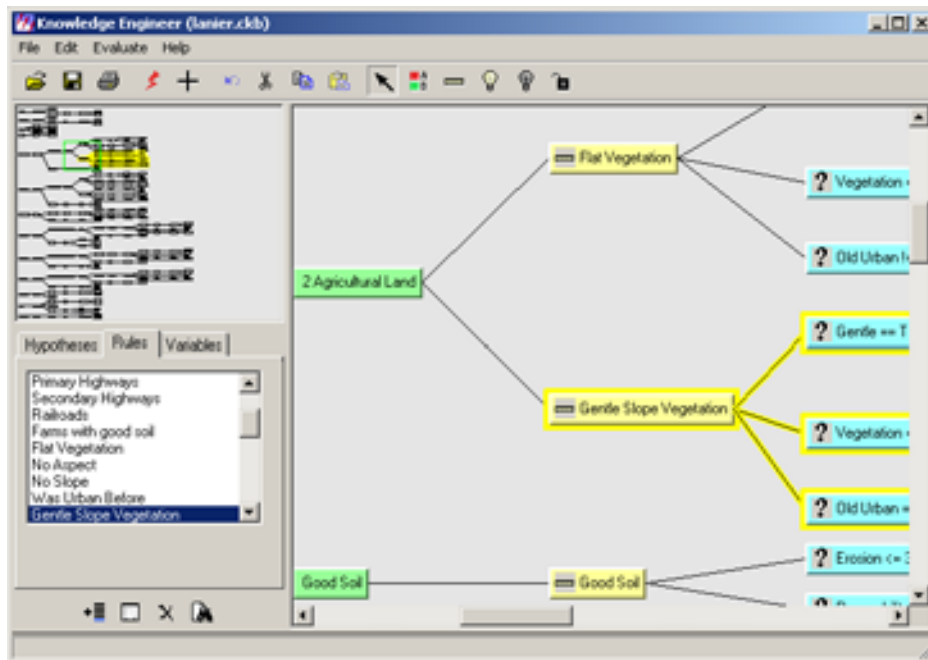
A rule is a conditional statement, or list of conditional statements, about the variable's data values and/or attributes that determine an informational component or hypotheses. Multiple rules and hypotheses can be linked together into a hierarchy that ultimately describes a final set of target informational classes or terminal hypotheses. Confidence values associated with each condition are also combined to provide a confidence image corresponding to the final output classified image.

The IMAGINE Expert Classifier is composed of two parts: the Knowledge Engineer and the Knowledge Classifier. The Knowledge Engineer provides the interface for an expert with first-hand knowledge of the data and the application to identify the variables, rules, and output classes of interest and create the hierarchical decision tree. The Knowledge Classifier provides an interface for a nonexpert to apply the knowledge base and create the output classification.

## Knowledge Engineer

With the Knowledge Engineer, you can open knowledge bases, which are presented as decision trees in editing windows.

**Figure 105: Knowledge Engineer Editing Window**



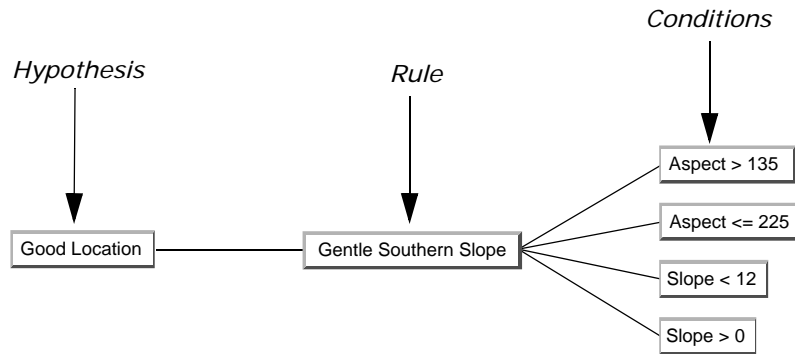
In [Figure 105](#), the upper left corner of the editing window is an overview of the entire decision tree with a green box indicating the position within the knowledge base of the currently displayed portion of the decision tree. This box can be dragged to change the view of the decision tree graphic in the display window on the right. The branch containing the currently selected hypotheses, rule, or condition is highlighted in the overview.

The decision tree grows in depth when the hypothesis of one rule is referred to by a condition of another rule. The terminal hypotheses of the decision tree represent the final classes of interest. Intermediate hypotheses may also be flagged as being a class of interest. This may occur when there is an association between classes.

[Figure 106](#) represents a single branch of a decision tree depicting a hypothesis, its rule, and conditions.



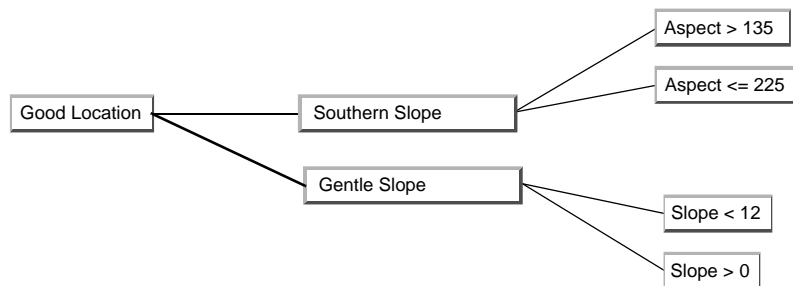
**Figure 106: Example of a Decision Tree Branch**



In this example, the rule, which is Gentle Southern Slope, determines the hypothesis, Good Location. The rule has four conditions depicted on the right side, all of which must be satisfied for the rule to be true.

However, the rule may be split if either Southern or Gentle slope defines the Good Location hypothesis. While both conditions must still be true to fire a rule, only one rule must be true to satisfy the hypothesis.

**Figure 107: Split Rule Decision Tree Branch**



### Variable Editor

The Knowledge Engineer also makes use of a Variable Editor when classifying images. The Variable editor provides for the definition of the variable objects to be used in the rules conditions.

The two types of variables are raster and scalar. Raster variables may be defined by imagery, feature layers (including vector layers), graphic spatial models, or by running other programs. Scalar variables may be defined with an explicit value, or defined as the output from a model or external program.

## Evaluating the Output of the Knowledge Engineer

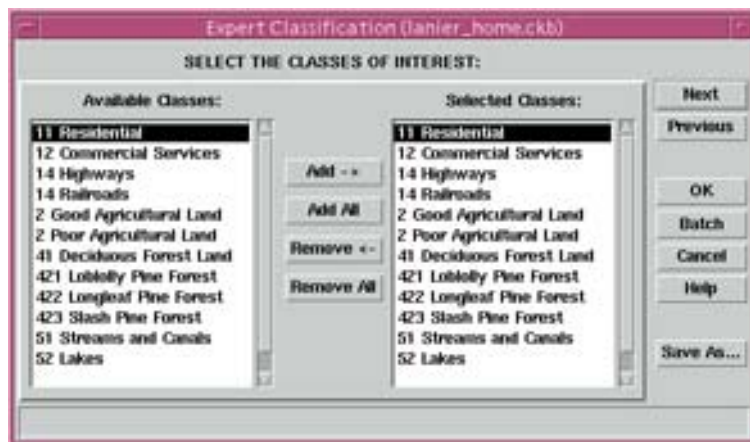
The task of creating a useful, well-constructed knowledge base requires numerous iterations of trial, evaluation, and refinement. To facilitate this process, two options are provided. First, you can use the Test Classification to produce a test classification using the current knowledge base. Second, you can use the Classification Pathway Cursor to evaluate the results. This tool allows you to move a crosshair over the image in a Viewer to establish a confidence level for areas in the image.

### Knowledge Classifier

The Knowledge Classifier is composed of two parts: an application with a user interface, and a command line executable. The user interface application allows you to input a limited set of parameters to control the use of the knowledge base. The user interface is designed as a wizard to lead you through pages of input parameters.

After selecting a knowledge base, you are prompted to select classes. The following is an example classes dialog:

**Figure 108: Knowledge Classifier Classes of Interest**



After you select the input data for classification, the classification output options, output files, output area, output cell size, and output map projection, the Knowledge Classifier process can begin. An inference engine then evaluates all hypotheses at each location (calculating variable values, if required), and assigns the hypothesis with the highest confidence. The output of the Knowledge Classifier is a thematic image, and optionally, a confidence image.

## Evaluating Classification

After a classification is performed, these methods are available for testing the accuracy of the classification:

- **Thresholding**—Use a probability image file to screen out misclassified pixels.
- **Accuracy Assessment**—Compare the classification to ground truth or other data.

## Thresholding

Thresholding is the process of identifying the pixels in a classified image that are the most likely to be classified incorrectly. These pixels are put into another class (usually class 0). These pixels are identified statistically, based upon the distance measures that were used in the classification decision rule.

### Distance File

When a minimum distance, Mahalanobis distance, or maximum likelihood classification is performed, a distance image file can be produced in addition to the output thematic raster layer. A distance image file is a one-band, 32-bit offset continuous raster layer in which each data file value represents the result of a spectral distance equation, depending upon the decision rule used.

- In a minimum distance classification, each distance value is the Euclidean spectral distance between the measurement vector of the pixel and the mean vector of the pixel's class.
- In a Mahalanobis distance or maximum likelihood classification, the distance value is the Mahalanobis distance between the measurement vector of the pixel and the mean vector of the pixel's class.

The brighter pixels (with the higher distance file values) are spectrally farther from the signature means for the classes to which they re assigned. They are more likely to be misclassified.

The darker pixels are spectrally nearer, and more likely to be classified correctly. If supervised training was used, the darkest pixels are usually the training samples.

**Figure 109: Histogram of a Distance Image**

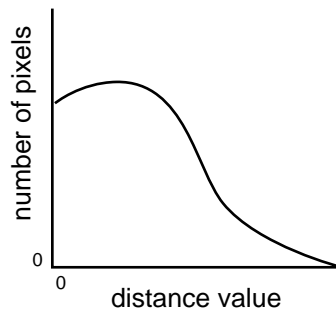


Figure 109 shows how the histogram of the distance image usually appears. This distribution is called a chi-square distribution, as opposed to a normal distribution, which is a symmetrical bell curve.

### Threshold

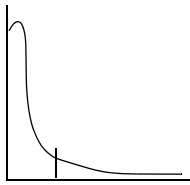
The pixels that are the most likely to be misclassified have the higher distance file values at the tail of this histogram. At some point that you define—either mathematically or visually—the tail of this histogram is cut off. The cutoff point is the threshold.

To determine the threshold:

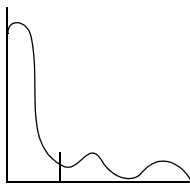
- interactively change the threshold with the mouse, when a distance histogram is displayed while using the threshold function. This option enables you to select a chi-square value by selecting the cut-off value in the distance histogram, or
- input a chi-square parameter or distance measurement, so that the threshold can be calculated statistically.

In both cases, thresholding has the effect of cutting the tail off of the histogram of the distance image file, representing the pixels with the highest distance values.

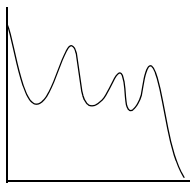
**Figure 110: Interactive Thresholding Tips**



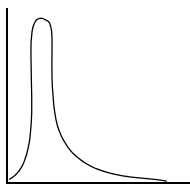
*Smooth chi-square shape—try to find the breakpoint where the curve becomes more horizontal, and cut off the tail.*



*Minor mode(s) (peaks) in the curve probably indicate that the class picked up other features that were not represented in the signature. You probably want to threshold these features out.*



*Not a good class. The signature for this class probably represented a polymodal (multi-peaked) distribution.*



*Peak of the curve is shifted from 0. Indicates that the signature mean is off-center from the pixels it represents. You may need to take a new signature and reclassify.*

Figure 110 shows some example distance histograms. With each example is an explanation of what the curve might mean, and how to threshold it.

## Chi-square Statistics

If the minimum distance classifier is used, then the threshold is simply a certain spectral distance. However, if Mahalanobis or maximum likelihood are used, then chi-square statistics are used to compare probabilities (Swain and Davis, 1978).

When statistics are used to calculate the threshold, the threshold is more clearly defined as follows:

$T$  is the distance value at which  $C\%$  of the pixels in a class have a distance value greater than or equal to  $T$ .

Where:

$T$  = the threshold for a class

$C\%$  = the percentage of pixels that are believed to be misclassified, known as the confidence level

$T$  is related to the distance values by means of chi-square statistics. The value  $X^2$  (chi-squared) is used in the equation.  $X^2$  is a function of:

- the number of bands of data used—known in chi-square statistics as the number of degrees of freedom
- the confidence level

When classifying an image in ERDAS IMAGINE, the classified image automatically has the degrees of freedom (i.e., number of bands) used for the classification. The chi-square table is built into the threshold application.

*NOTE: In this application of chi-square statistics, the value of  $X^2$  is an approximation. Chi-square statistics are generally applied to independent variables (having no covariance), which is not usually true of image data.*



---

*A further discussion of chi-square statistics can be found in a statistics text.*



---

*Use the Classification Threshold utility to perform the thresholding.*

## Accuracy Assessment

Accuracy assessment is a general term for comparing the classification to geographical data that are assumed to be true, in order to determine the accuracy of the classification process. Usually, the assumed-true data are derived from ground truth data.

It is usually not practical to ground truth or otherwise test every pixel of a classified image. Therefore, a set of reference pixels is usually used. Reference pixels are points on the classified image for which actual data are (or will be) known. The reference pixels are randomly selected (Congalton, 1991).

*NOTE: You can use the ERDAS IMAGINE Accuracy Assessment utility to perform an accuracy assessment for any thematic layer. This layer does not have to be classified by ERDAS IMAGINE (e.g., you can run an accuracy assessment on a thematic layer that was classified in ERDAS Version 7.5 and imported into ERDAS IMAGINE).*

### Random Reference Pixels

When reference pixels are selected by the analyst, it is often tempting to select the same pixels for testing the classification that were used in the training samples. This biases the test, since the training samples are the basis of the classification. By allowing the reference pixels to be selected at random, the possibility of bias is lessened or eliminated (Congalton, 1991).

The number of reference pixels is an important factor in determining the accuracy of the classification. It has been shown that more than 250 reference pixels are needed to estimate the mean accuracy of a class to within plus or minus five percent (Congalton, 1991).

ERDAS IMAGINE uses a square window to select the reference pixels. The size of the window can be defined by you. Three different types of distribution are offered for selecting the random pixels:

- random—no rules are used
- stratified random—the number of points is stratified to the distribution of thematic layer classes
- equalized random—each class has an equal number of random points



---

*Use the Accuracy Assessment utility to generate random reference points.*

### Accuracy Assessment CellArray

An Accuracy Assessment CellArray is created to compare the classified image with reference data. This CellArray is simply a list of class values for the pixels in the classified image file and the class values for the corresponding reference pixels. The class values for the reference pixels are input by you. The CellArray data reside in an image file.



---

*Use the Accuracy Assessment CellArray to enter reference pixels for the class values.*

## Error Reports

From the Accuracy Assessment CellArray, two kinds of reports can be derived.

- The error matrix simply compares the reference points to the classified points in a  $c \times c$  matrix, where  $c$  is the number of classes (including class 0).
- The accuracy report calculates statistics of the percentages of accuracy, based upon the results of the error matrix.

When interpreting the reports, it is important to observe the percentage of correctly classified pixels and to determine the nature of errors of the producer and yourself.



---

*Use the Accuracy Assessment utility to generate the error matrix and accuracy reports.*

## Kappa Coefficient

The Kappa coefficient expresses the proportionate reduction in error generated by a classification process compared with the error of a completely random classification. For example, a value of .82 implies that the classification process is avoiding 82 percent of the errors that a completely random classification generates ([Congalton, 1991](#)).

