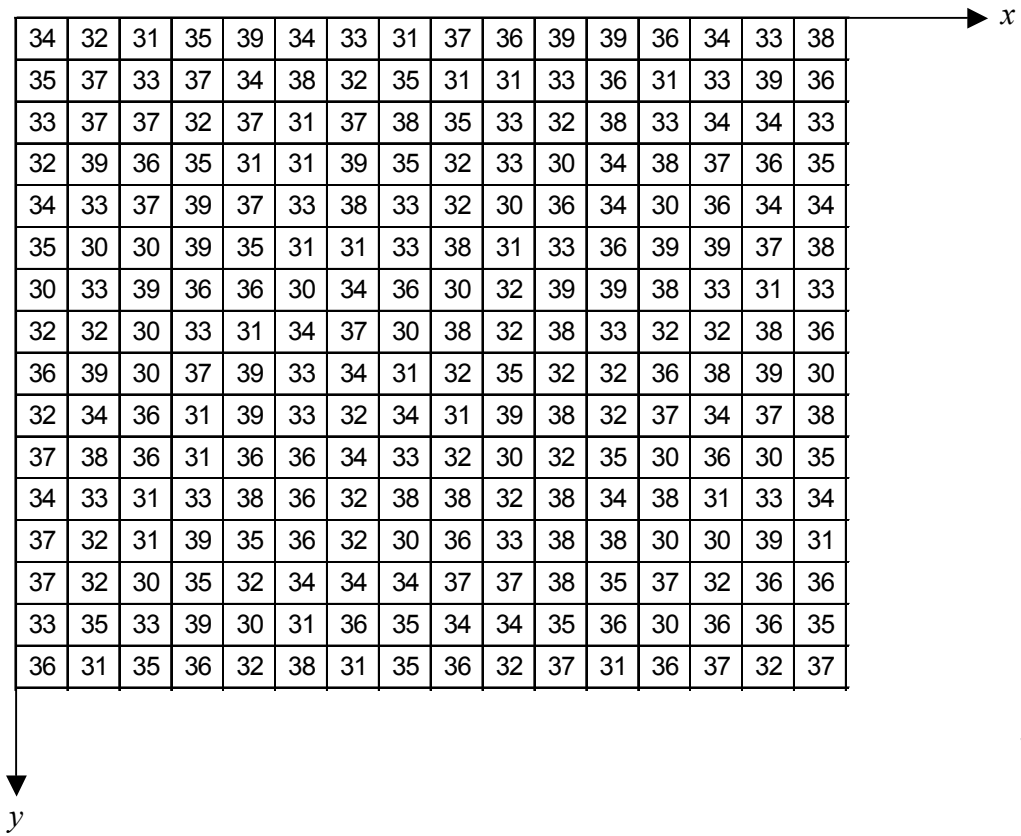
Multispectral Digital Image Processing

Single Band Imagery



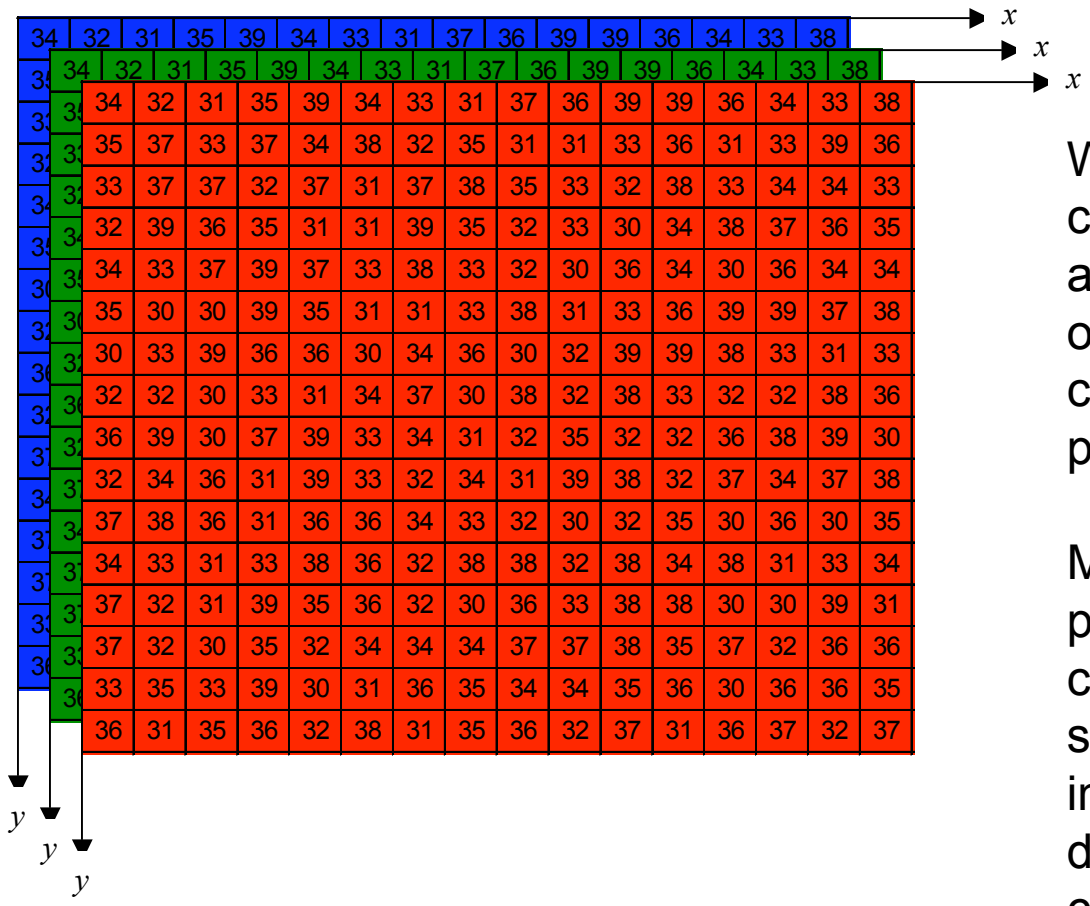
34	32	31	35	39	34	33	31	37	36	39	39	36	34	33	38
35	37	33	37	34	38	32	35	31	31	33	36	31	33	39	36
33	37	37	32	37	31	37	38	35	33	32	38	33	34	34	33
32	39	36	35	31	31	39	35	32	33	30	34	38	37	36	35
34	33	37	39	37	33	38	33	32	30	36	34	30	36	34	34
35	30	30	39	35	31	31	33	38	31	33	36	39	39	37	38
30	33	39	36	36	30	34	36	30	32	39	39	38	33	31	33
32	32	30	33	31	34	37	30	38	32	38	33	32	32	38	36
36	39	30	37	39	33	34	31	32	35	32	32	36	38	39	30
32	34	36	31	39	33	32	34	31	39	38	32	37	34	37	38
37	38	36	31	36	36	34	33	32	30	32	35	30	36	30	35
34	33	31	33	38	36	32	38	38	32	38	34	38	31	33	34
37	32	31	39	35	36	32	30	36	33	38	38	30	30	39	31
37	32	30	35	32	34	34	34	37	37	38	35	37	32	36	36
33	35	33	39	30	31	36	35	34	34	35	36	30	36	36	35
36	31	35	36	32	38	31	35	36	32	37	31	36	37	32	37

To this point, we have thought about imagery in one way, namely that in single band imagery, the brightness at any pixel location is equal to a digital count number that represents the brightness of the image at that point

The only variant on this theme has been color imagery that was composed of three such images, one representing the red, green and blue brightness values at these locations.

Multispectral Digital Image Processing

Color Imagery

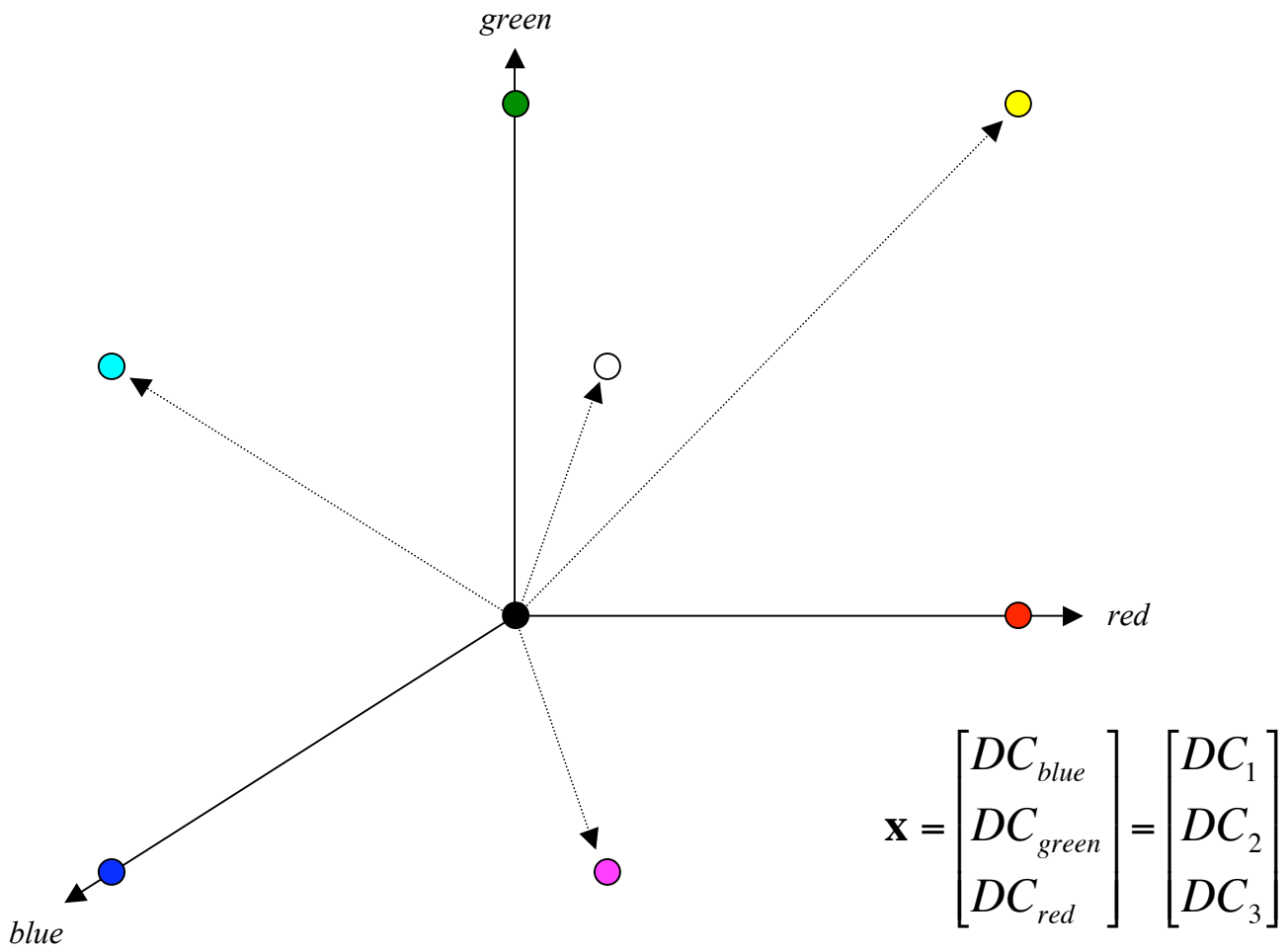


We have treated these three collections of brightness values as independent sets of data, only combining them to form color images for display purposes.

Multispectral digital image processing uses these three channels (and many more) simultaneously to generate information that can be used determine quantitative characteristics of the original scene.

Multispectral Digital Image Processing

Vector Notation

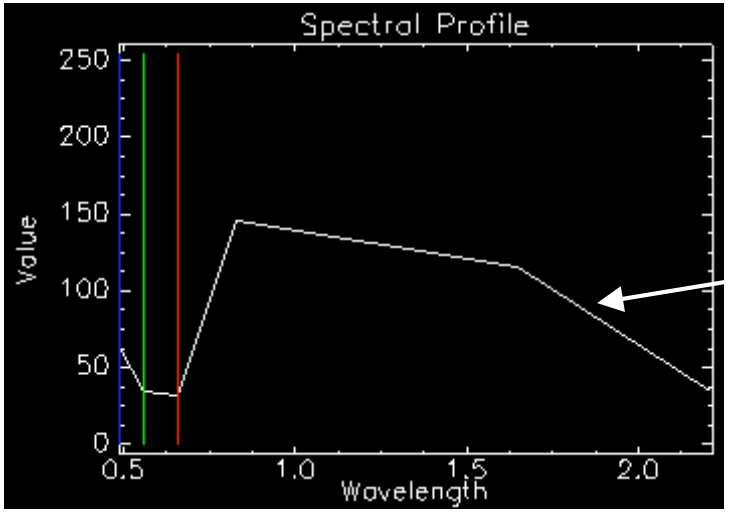


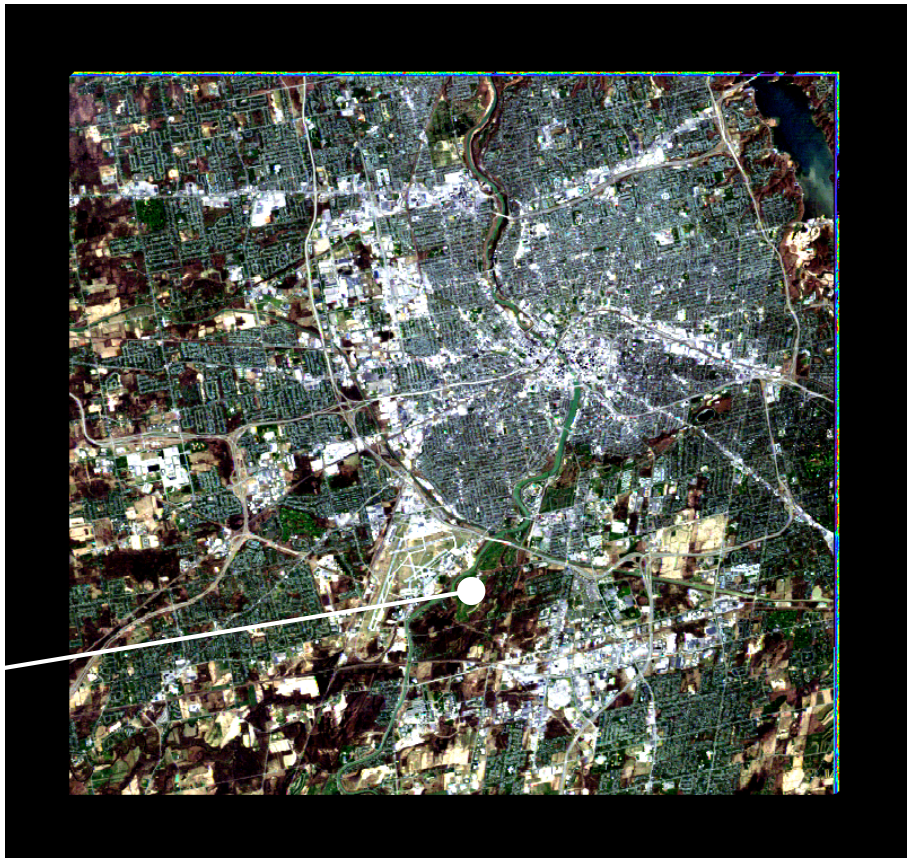
$$\mathbf{x} = \begin{bmatrix} DC_{blue} \\ DC_{green} \\ DC_{red} \end{bmatrix} = \begin{bmatrix} DC_1 \\ DC_2 \\ DC_3 \end{bmatrix}$$

Multispectral Digital Image Processing

Multispectral Imagery

$$\mathbf{x} = \begin{bmatrix} DC_1 \\ DC_2 \\ DC_3 \\ DC_4 \\ DC_{5^*} \\ DC_7 \end{bmatrix}$$





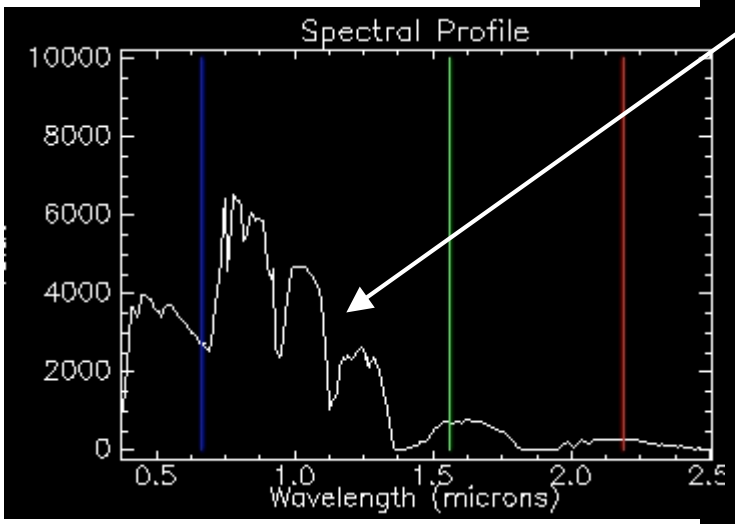
Landsat 5 Thematic Mapper
Bands 3, 2, 1

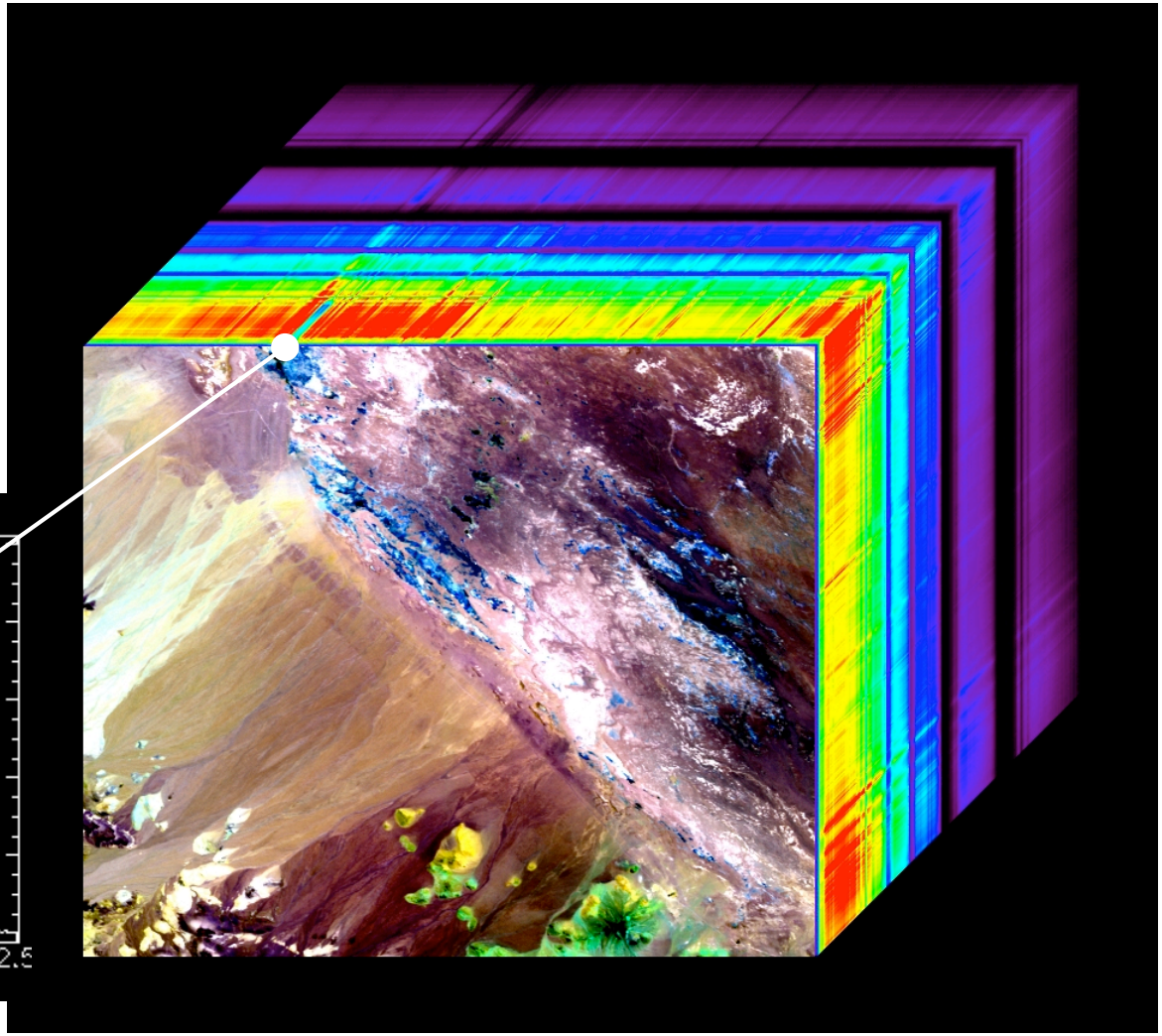
* NOTE: Band 6 has been omitted as it is a thermal infrared channel

Multispectral Digital Image Processing

Hyperspectral Imagery

$$\mathbf{x} = \begin{bmatrix} DC_1 \\ DC_2 \\ DC_3 \\ DC_4 \\ \vdots \\ DC_{224} \end{bmatrix}$$





AVIRIS - Lunar Lake Calibration Image
512 samples x 614 lines x 224 spectral channels
Bands 192, 128, 33

Review of Vector and Matrix Algebra

Transformation of a Vector

There will be times when we wish to create another vector $\mathbf{y}$ from the existing vector $\mathbf{x}$. As an example, the transformation of a two-dimensional vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

can be carried out via the following pair of equations*

$$y_1 = m_{11}x_1 + m_{12}x_2$$
$$y_2 = m_{21}x_1 + m_{22}x_2$$

* NOTE: row,column notation is being used for matrices which should not be confused with the column,row notation used by IDL

which states that each components of $\mathbf{y}$ is simply a linear combination of all of the components of $\mathbf{x}$. This can be expressed in matrix notation as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

Review of Vector and Matrix Algebra

Inverse of a Matrix

The inverse of a matrix $\mathbf{M}$ is denoted by $\mathbf{M}^{-1}$ and is defined by

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{I}$$

where $\mathbf{I}$ is known as the identity matrix and is defined as

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The identity matrix, $\mathbf{I}$, if used as a transformation matrix for $\mathbf{x}$ will leave this matrix unchanged

$$\mathbf{y} = \mathbf{I}\mathbf{x} = \mathbf{x}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{x}$$

Review of Vector and Matrix Algebra

Inverse of a Matrix

The inverse of a matrix is not a trivial computation for matrices larger than 2x2 in size. The inverse can always be expressed as

$$\mathbf{M}^{-1} = \frac{\mathbf{M}^*}{|\mathbf{M}|}$$

where $\mathbf{M}^*$ is called the *adjoint* and $|\mathbf{M}|$ is called the *determinant* of $\mathbf{M}$.

The *adjoint* is a transposed matrix of cofactors; which for a 2x2 matrix

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad \text{the adjoint is} \quad \mathbf{M}^* = \begin{bmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{bmatrix}$$

The *determinant* for a 2x2 matrix is defined as

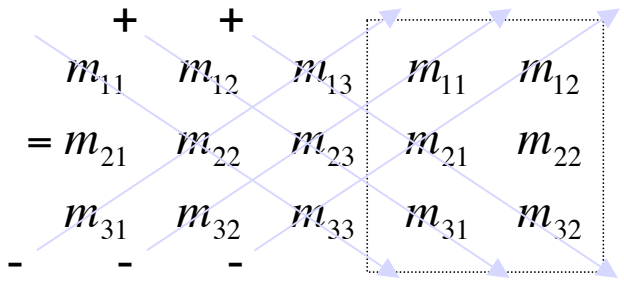
$$|\mathbf{M}| = \begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} = m_{11}m_{22} - m_{21}m_{12}$$

Review of Vector and Matrix Algebra

Inverse of a Matrix

for a 3x3 matrix, the determinant is defined as

$$|\mathbf{M}| = \begin{vmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{vmatrix}$$



$$= m_{11}m_{22}m_{33} + m_{12}m_{23}m_{31} + m_{13}m_{21}m_{32} - m_{31}m_{22}m_{13} - m_{32}m_{23}m_{11} - m_{33}m_{21}m_{12}$$

Review of Vector and Matrix Algebra

Inverse of a Matrix

Example

$$\mathbf{M} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\mathbf{M}^{-1} = \frac{\mathbf{M}^*}{|\mathbf{M}|}$$

$$= \frac{\begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}}{1 \cdot 4 - 3 \cdot 2}$$

$$= \frac{\begin{bmatrix} 4 & -2 \\ -3 & 1 \end{bmatrix}}{-2}$$

$$= \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

$$\mathbf{M}\mathbf{M}^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}$$

$$= \begin{bmatrix} (1)(-2) + (2)(1.5) & (1)(1) + (2)(-0.5) \\ (3)(-2) + (4)(1.5) & (3)(1) + (4)(-0.5) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \checkmark$$

```
IDL> a = [ [1,2], [3,4] ]
IDL> PRINT, DETERM(a)
   -2.00000
IDL> PRINT, INVERT(a)
   -2.00000      1.00000
    1.50000     -0.50000
IDL> PRINT, a # INVERT(a)
    1.00000      0.00000
    0.00000      1.00000
```

Review of Vector and Matrix Algebra

Inverse of a Matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 2 & 3 & 3 \end{bmatrix}$$

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{I}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} m_{11}^{-1} & m_{12}^{-1} & m_{13}^{-1} \\ m_{21}^{-1} & m_{22}^{-1} & m_{23}^{-1} \\ m_{31}^{-1} & m_{32}^{-1} & m_{33}^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 & 1 & 0 \\ 2 & 3 & 3 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc} m_{11}^{-1} & m_{12}^{-1} & m_{13}^{-1} \\ m_{21}^{-1} & m_{22}^{-1} & m_{23}^{-1} \\ m_{31}^{-1} & m_{32}^{-1} & m_{33}^{-1} \end{array}\right]$$

perform **GAUSSIAN ELIMINATION** on this concatenated matrix

to obtain this inverse

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 2 & 3 & 3 \end{bmatrix} \begin{bmatrix} -3 & -1\frac{1}{2} & 3\frac{1}{2} \\ 2 & 1\frac{1}{2} & -2\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 & 1 & 0 \\ 2 & 3 & 3 & 0 & 0 & 1 \end{array}\right]$$

Swap rows 1 & 2 (since the lead coefficient is larger)

$$\left[\begin{array}{ccc|ccc} 2 & 3 & 1 & 0 & 1 & 0 \\ 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 3 & 3 & 0 & 0 & 1 \end{array}\right]$$

Row 3 => Subtract row 1 from row 3

$$\left[\begin{array}{ccc|ccc} 2 & 3 & 1 & 0 & 1 & 0 \\ 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array}\right]$$

Row 2 => Subtract 1/2 * row 1 from row 2

$$\left[\begin{array}{ccc|ccc} 2 & 3 & 1 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 2\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array}\right]$$

Row 1 => Subtract 6 * row 2 from row 1

$$\left[\begin{array}{ccc|ccc} 2 & 0 & -14 & -6 & 4 & 0 \\ 0 & \frac{1}{2} & 2\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array}\right]$$

Row 2 => Add 7 * row 3 to row 1

$$\left[\begin{array}{ccc|ccc} 2 & 0 & 0 & -6 & -3 & 7 \\ 0 & \frac{1}{2} & 2\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array}\right]$$

Row 2 => Subtract 1.25 * row 3 from row 2

$$\left[\begin{array}{ccc|ccc} 2 & 0 & 0 & -6 & -3 & 7 \\ 0 & \frac{1}{2} & 0 & 1 & \frac{3}{4} & -\frac{5}{4} \\ 0 & 0 & 2 & 0 & -1 & 1 \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -3 & -1\frac{1}{2} & 3\frac{1}{2} \\ 0 & 1 & 0 & 2 & 1\frac{1}{2} & -2\frac{1}{2} \\ 0 & 0 & 1 & 0 & -\frac{1}{2} & \frac{1}{2} \end{array}\right]$$

Divide each row by the diagonal elements

Review of Vector and Matrix Algebra

Transpose of a Matrix

The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^t$ and is formed by exchanging the row and column dimensions of the matrix (or, in the case of a square matrix, it may be thought of as rotating the matrix around the major diagonal)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad \mathbf{A}^t = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \qquad \mathbf{A}^t = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \end{bmatrix}$$

Review of Vector and Matrix Algebra

Matrix Arithmetic

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} \\ a_{21} - b_{21} & a_{22} - b_{22} \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Column Vector

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$\mathbf{A}\mathbf{A}^t = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \begin{bmatrix} a_1 & a_2 \end{bmatrix} = \begin{bmatrix} a_1^2 & a_1 a_2 \\ a_1 a_2 & a_2^2 \end{bmatrix}$$

$$\mathbf{A}^t\mathbf{A} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = a_1^2 + a_2^2$$

dot product
scalar product
inner product

Review of Vector and Matrix Algebra

Property of an Orthogonal Matrix

The rows of an orthogonal matrix are an orthonormal basis. That is, each row has length one, and are mutually perpendicular. Similarly, the columns are also an orthonormal basis. In fact, given any orthonormal basis, the matrix whose rows are that basis is an orthogonal matrix. It is automatically the case that the columns are another orthonormal basis.

An important property of an orthogonal matrix is that its inverse is identical to it transpose, that is

$$\mathbf{M}^{-1} = \mathbf{M}^{t}$$

and as such is ALWAYS invertible.

Review of Vector and Matrix Algebra

Eigenvalues and Eigenvectors

Is there a vector, $\mathbf{x}$, that can be multiplied by a number, λ, and be transformed in exactly the same way as if that vector, $\mathbf{x}$, had been multiplied by a matrix $\mathbf{M}$? That is

$$\mathbf{Mx} = \lambda\mathbf{x}$$

This equality implies that

$$\mathbf{Mx} - \lambda\mathbf{x} = 0$$

$$\text{or} \quad (\mathbf{M} - \lambda\mathbf{I})\mathbf{x} = 0$$

This can be true if $\mathbf{x} = 0$ or if $|\mathbf{M} - \lambda\mathbf{I}| = 0$.

If the latter is solved for, λ will represent what are called the *eigenvalues* of $\mathbf{M}$ and $\mathbf{x}$ will be the corresponding *eigenvectors*.

Review of Vector and Matrix Algebra

Diagonalization of a Matrix

If you consider the transformation

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

and as previously defined the eigenvalues and eigenvectors of $\mathbf{M}$ are

$$\lambda_i \mathbf{x}_i = \mathbf{M}\mathbf{x}_i \qquad \text{for } i = 1, \ldots, n$$

These n different equations can be expressed in a compact form

$$\mathbf{X}\Lambda = \mathbf{M}\mathbf{X}$$

where Λ is the diagonal matrix of eigenvalues and $\mathbf{X}$ the matrix of eigenvectors $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \qquad \Lambda = \mathbf{X}^{-1}\mathbf{M}\mathbf{X}$$

Review of Probability and Statistics

Conditional Probability

$p(x)$ represents the probability that an event x occurs.

If $\mathbf{x}$ is an N-dimensional pixel vector, then $p(\mathbf{x})$ represents the probability that a pixel exists in N-dimensional space at the position designated by $\mathbf{x}$.

We often want to know the probability of an event occurring given some other event, this is known as *conditional probability* and is denoted as $p(x|y)$ namely, the probability of the event x occurring given y is specified. For example

$p(\mathbf{x} \mid \omega_i)$ for $i = 1, 2, \ldots M$ where M is the total number of pixel types (e.g. land cover classes) in the image.

represents the probability that a pixel of type ω_i exists at the position described by $\mathbf{x}$ in N-dimensional space.

Review of Probability and Statistics

Conditional Probability

If the complete set of $p(\mathbf{x}|\omega_i)$ are known (these are often referred to as the *class conditional probabilities*) then $p(\mathbf{x})$ can be found. Consider the product

$$p(\mathbf{x}\,|\,\omega_i)\,p(\omega_i)$$

where $p(\omega_i)$ is the probability that a pixel of type ω_i exists in the image. This product is the probability that a pixel at position $\mathbf{x}$ in N-dimensional space is a ω_i-type pixel.

The probability that a pixel of any type exists at position $\mathbf{x}$ is as stated previously and can be determined as

$$p(\mathbf{x}) = \sum_{i=1}^{M} p(\mathbf{x}\,|\,\omega_i)\,p(\omega_i)$$

Review of Probability and Statistics

Conditional Probability

The product in the previous summation is referred to as the *joint probability* of the events occurring and is written as

$$p(\mathbf{x},\omega_i) = p(\mathbf{x} \mid \omega_i)p(\omega_i)$$

which represents the probability that a pixel occurs at position $\mathbf{x}$ and that the pixel type is ω_i.

To summarize

$p(\omega_i)$	the probability that a ω_i-type of pixel occurs in the image
$p(\mathbf{x} \mid \omega_i)$	the probability of finding a pixel at position $\mathbf{x}$ given that we are interested in pixels of type ω_i
$p(\mathbf{x},\omega_i)$	the probability that a pixel occurs at position $\mathbf{x}$ and that the pixel is of type ω_i

Review of Probability and Statistics

Conditional Probability

We can also write

$$p(\omega_i, \mathbf{x}) = p(\omega_i \mid \mathbf{x}) p(\mathbf{x})$$

where $p(\omega_i|\mathbf{x})$ is the conditional probability that the pixel type is ω_i given that we are examining a pixel at position $\mathbf{x}$ in N-dimensional space. This is called the *posteriori probability* of pixel type ω_i.

The joint probabilities in these cases are identical, that is

$$p(\mathbf{x}, \omega_i) = p(\omega_i, \mathbf{x})$$

so

$$p(\mathbf{x} \mid \omega_i) p(\omega_i) = p(\omega_i \mid \mathbf{x}) p(\mathbf{x})$$

$$p(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i) p(\omega_i)}{p(\mathbf{x})}$$

which is known as Bayes' theorem.

Review of Probability and Statistics

Univariate Normal Probability Distribution

It is often assumed that the class conditional probabilities, representing the probability of finding a pixel at location $\mathbf{x}$ given we are interested in pixels of type ω_i, are distributed normally. In a one-dimensional space, this probability is represented as

$$p(x \mid \omega_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

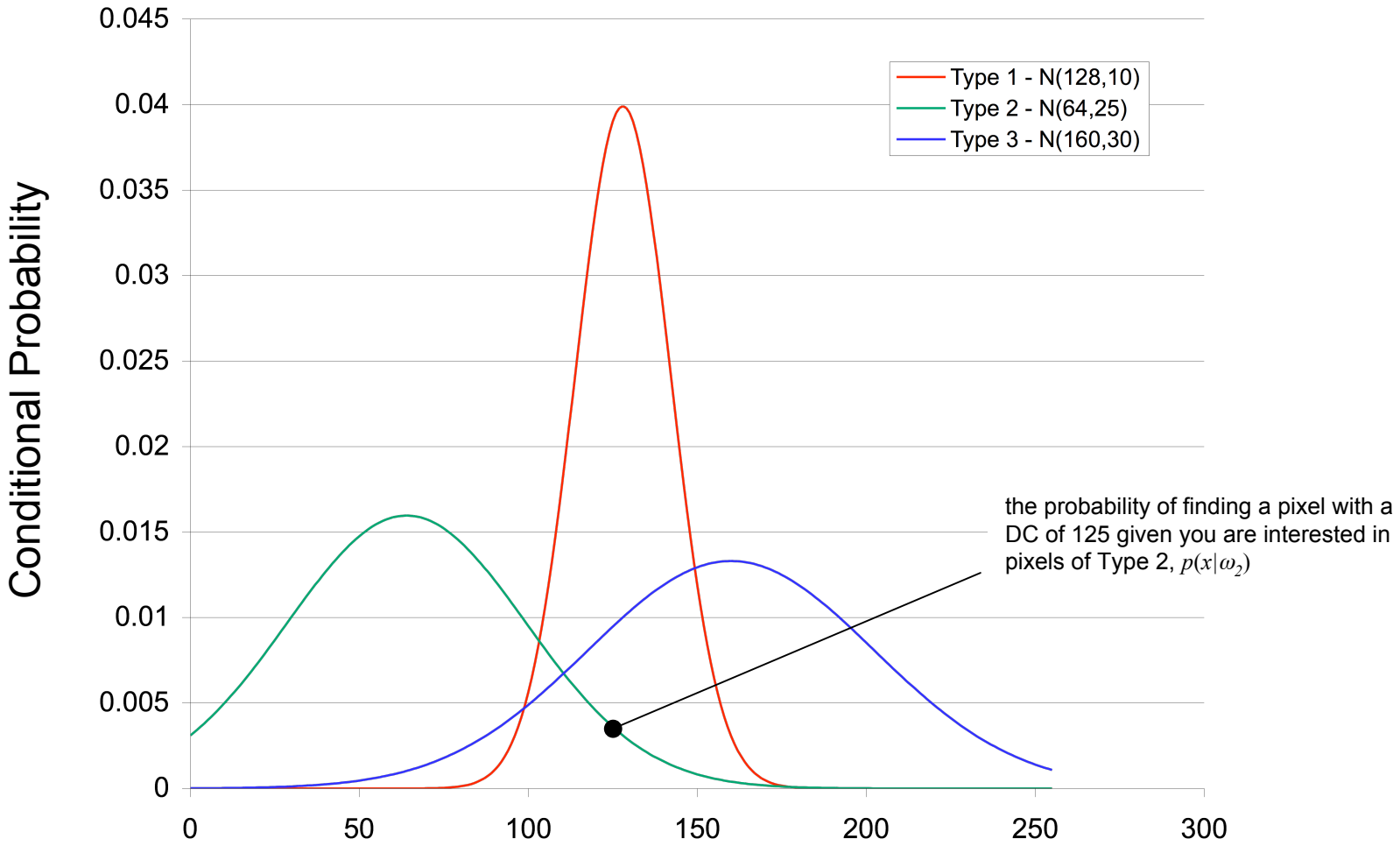
where μ_i and σ_i are the mean and standard deviation of x_i for pixels of type ω_i. These descriptive statistics should be computed based upon a large sample of x to assure that they are unbiased estimates (N_i samples).

$$\mu_i = \frac{1}{N_i} \sum_{j \in \omega_i} x_j \qquad \sigma_i^2 = \frac{1}{N_i - 1} \sum_{j \in \omega_i} (x_j - \mu_i)^2$$

NOTE: x_j is the j^{th} sample of n_i pixels representing pixels of type ω_i

Review of Probability and Statistics

Univariate Normal Probability Distribution



Multivariate Normal Distribution

The univariate case presented is seldom of use in image processing as separability of pixel types is difficult, if not impossible, to achieve with a single band image. It does, however, serve as a good starting point to imply the multivariate equivalent. The multivariate normal conditional probability distribution is given by

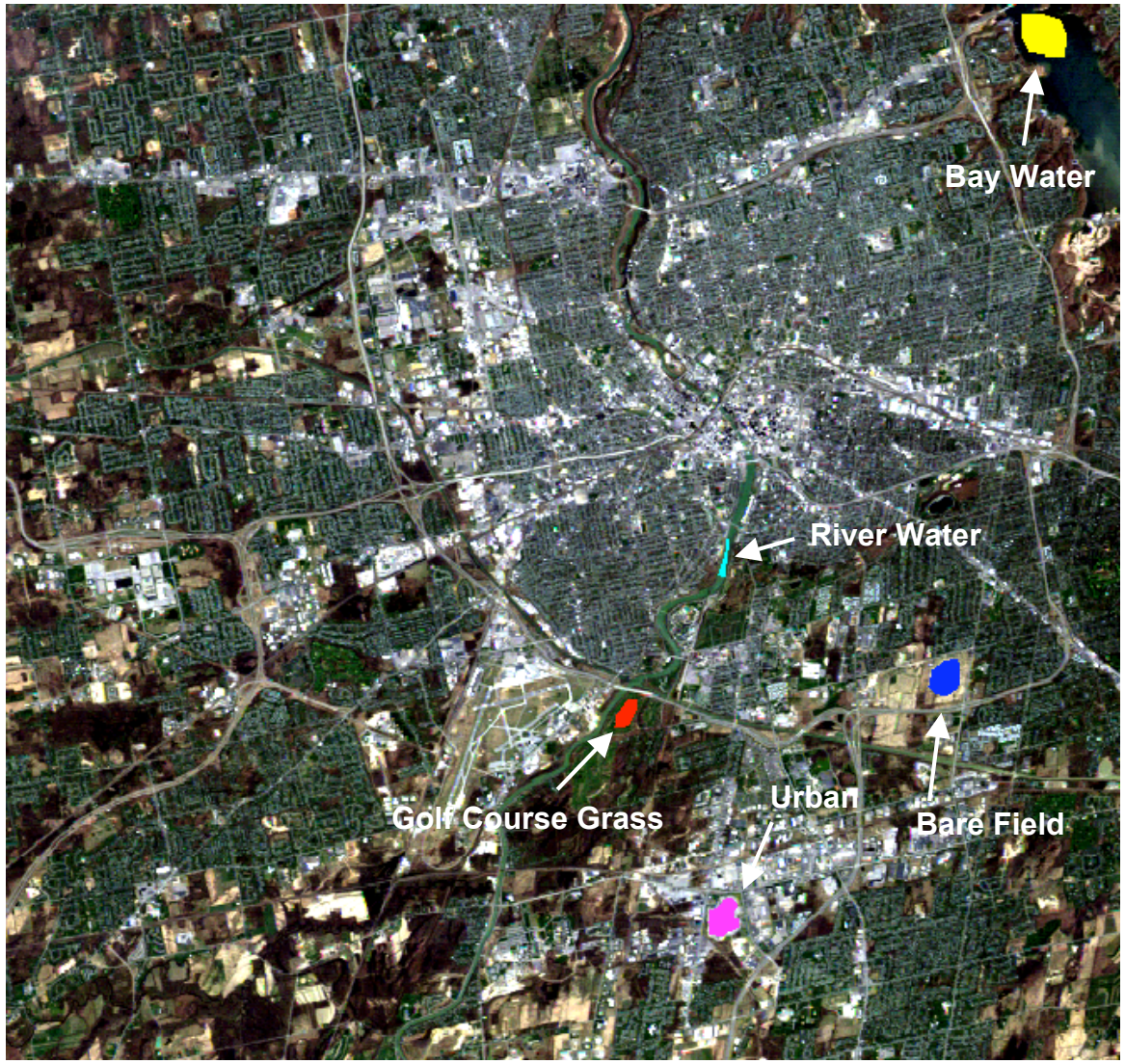
$$p(\mathbf{x}\,|\,\omega_i) = \frac{1}{2\pi^{\frac{N}{2}}\left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}$$

$$\boldsymbol{\mu}_i = \frac{1}{N_i}\sum_{j\in\omega_i}\mathbf{x}_{i,j} = \frac{1}{N_i}\begin{bmatrix}\sum_{j\in\omega_i} x_{i,j,1}\\ \sum_{j\in\omega_i} x_{i,j,2}\\ \vdots \\ \sum_{j\in\omega_i} x_{i,j,N}\end{bmatrix}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{N_i - 1}\sum_{j\in\omega_i}(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^t$$

Multivariate Normal Distribution

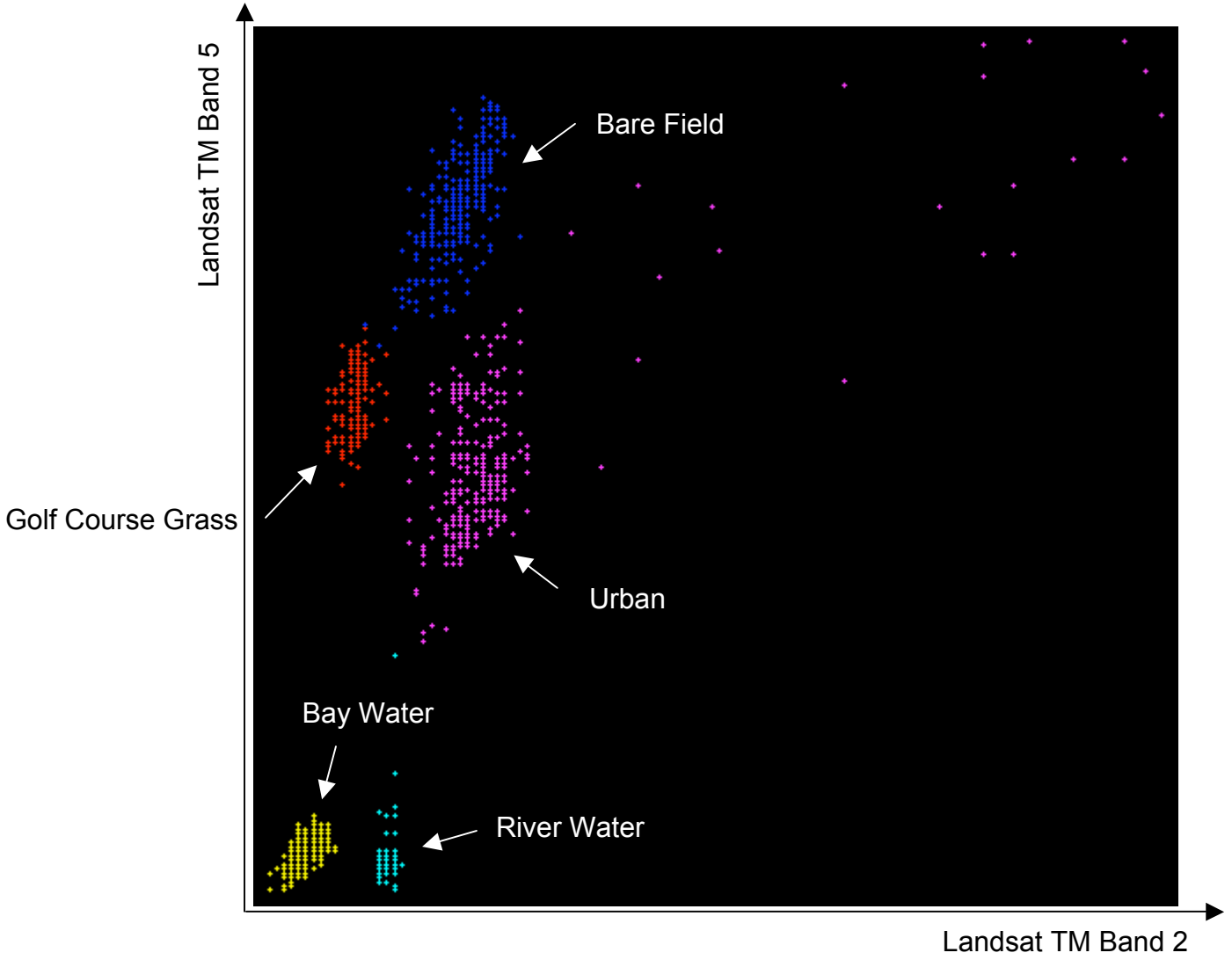
Region of Interest (ROI) Selection



Landsat TM
Bands 3,2,1 (R,G,B)

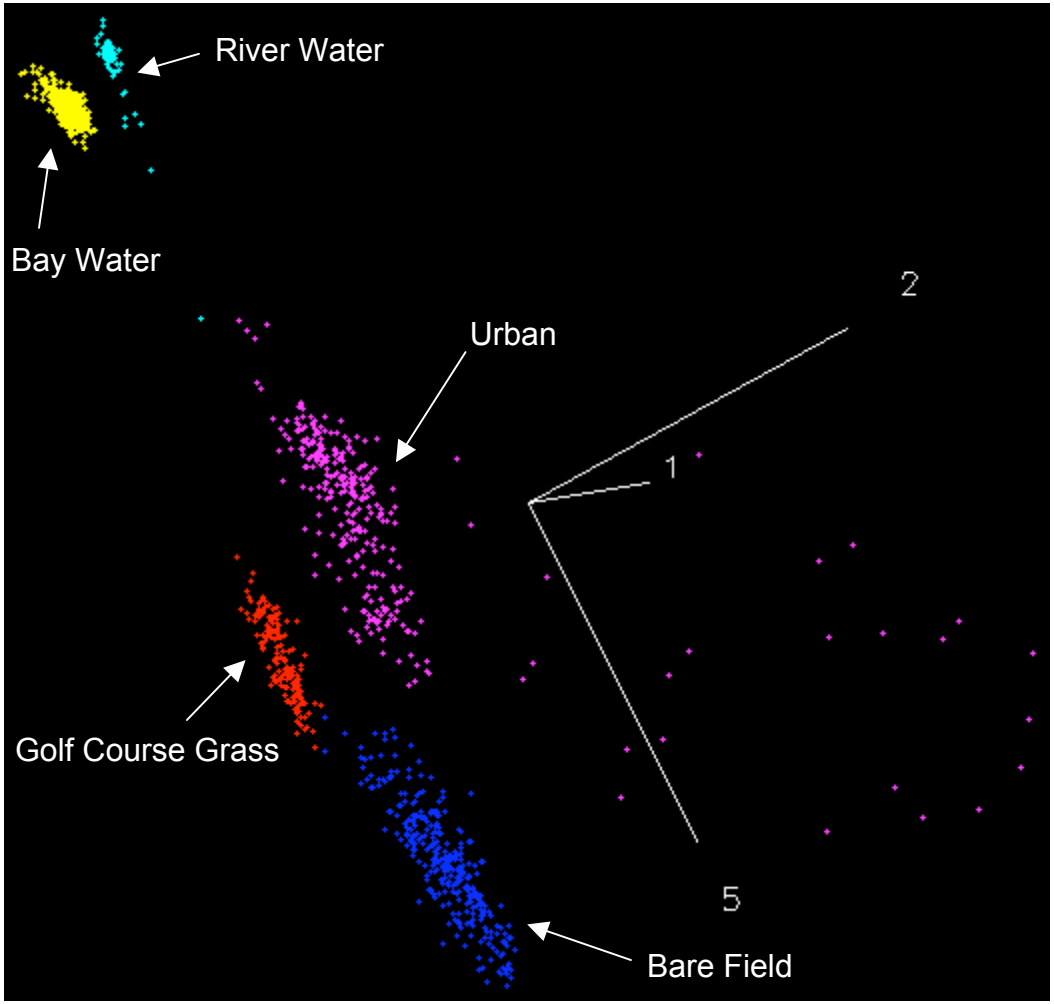
Multivariate Normal Distribution

Two-Channel Distribution



Multivariate Normal Distribution

Three-Channel Distribution



Landsat TM

Multivariate Normal Distribution

Descriptive Statistics

Band	Min	Max	Mean	Stdev
1	85	255	106.576052	29.194125
2	40	142	52.436893	16.343713
3	54	225	74.870550	24.257869
4	42	179	62.822006	21.294107
5	64	202	106.223301	22.236158
6	38	90	64.766990	9.048790

Urban

Band	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6
1	852.296957	446.968289	608.834552	557.852866	509.036534	99.456090
2	446.968289	267.116946	365.852194	294.938406	284.009267	66.449533
3	608.834552	365.852194	588.444227	468.499611	414.593935	101.531427
4	557.852866	294.938406	468.499611	453.438995	374.919745	80.370729
5	509.036534	284.009267	414.593935	374.919745	494.446728	151.844408
6	99.456090	66.449533	101.531427	80.370729	151.844408	81.880595

$$\left|\Sigma_{urban}\right| = 3.52516x10^{11}$$

Band	Min	Max	Mean	Stdev
1	57	72	66.238342	2.456933
2	21	30	26.466321	1.482361
3	19	32	27.129534	1.933713
4	8	19	14.537133	1.547378
5	7	24	16.863558	2.671852
6	4	15	9.949914	1.746746

Bay Water

Band	Band 1	Band 2	Band 3	Band 4	Band 5	Band 6
1	6.036520	2.672401	3.581530	2.954802	4.119081	2.486007
2	2.672401	2.197393	2.259560	1.634903	2.414947	1.431701
3	3.581530	2.259560	3.739247	2.323036	3.256459	1.930375
4	2.954802	1.634903	2.323036	2.394380	2.874452	1.679199
5	4.119081	2.414947	3.256459	2.874452	7.138791	3.065819
6	2.486007	1.431701	1.930375	1.679199	3.065819	3.051120

$$\left|\Sigma_{bay\ water}\right| = 27.7171$$

Statistical Distance

Univariate Case

The distance between two points, namely an observation, x, and a class mean, μ_i, can be measured in a number of ways. In a traditional sense, the Euclidean distance between these points in a univariate space is determined as

$$d_i = \sqrt{(x - \mu_i)^2}$$

and in an N-dimensional space as

$$d_i = \sqrt{\sum_{j=1}^{N} (x_j - \mu_{i,j})^2}$$

This distance is strictly the lineal distance between the observation and the mean of the population of which it is a member.

No regard is given to the distributional shape.

Statistical Distance

Univariate Case

In order to get a distance that is sensitive to the distributional characteristics of the population of which it is a member, you need to account for the uncertainty in the data. From statistical inference theory, we have the measure of distance known as the z-score, namely

$$z = \frac{x - \mu}{\sigma}$$

and to change this into a measure of statistical distance, we have

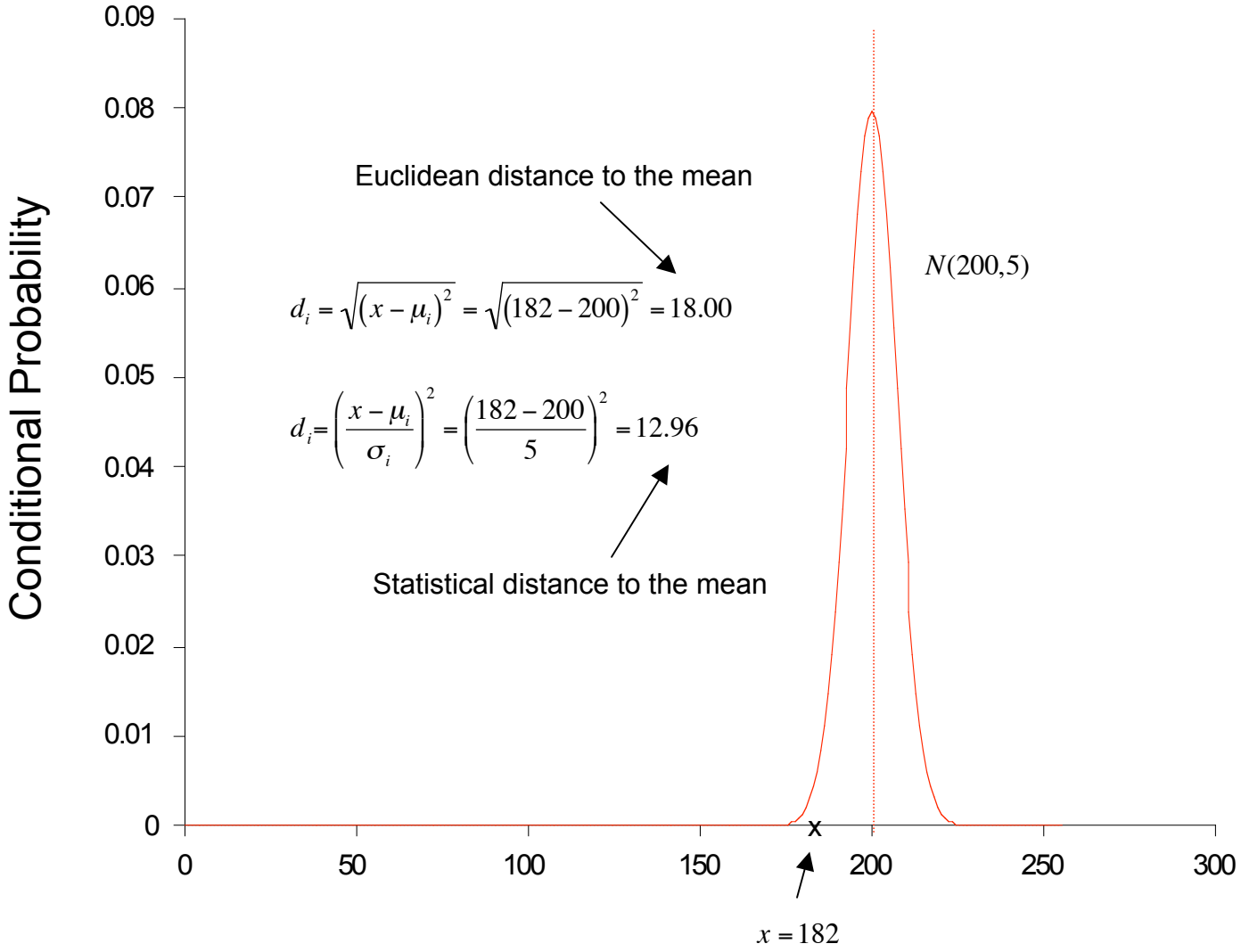
$$d_i = \left(\frac{x - \mu_i}{\sigma_i} \right)^2$$

which is found explicitly in the exponent of the probability distribution function for a normal distribution

$$p(x \mid \omega_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

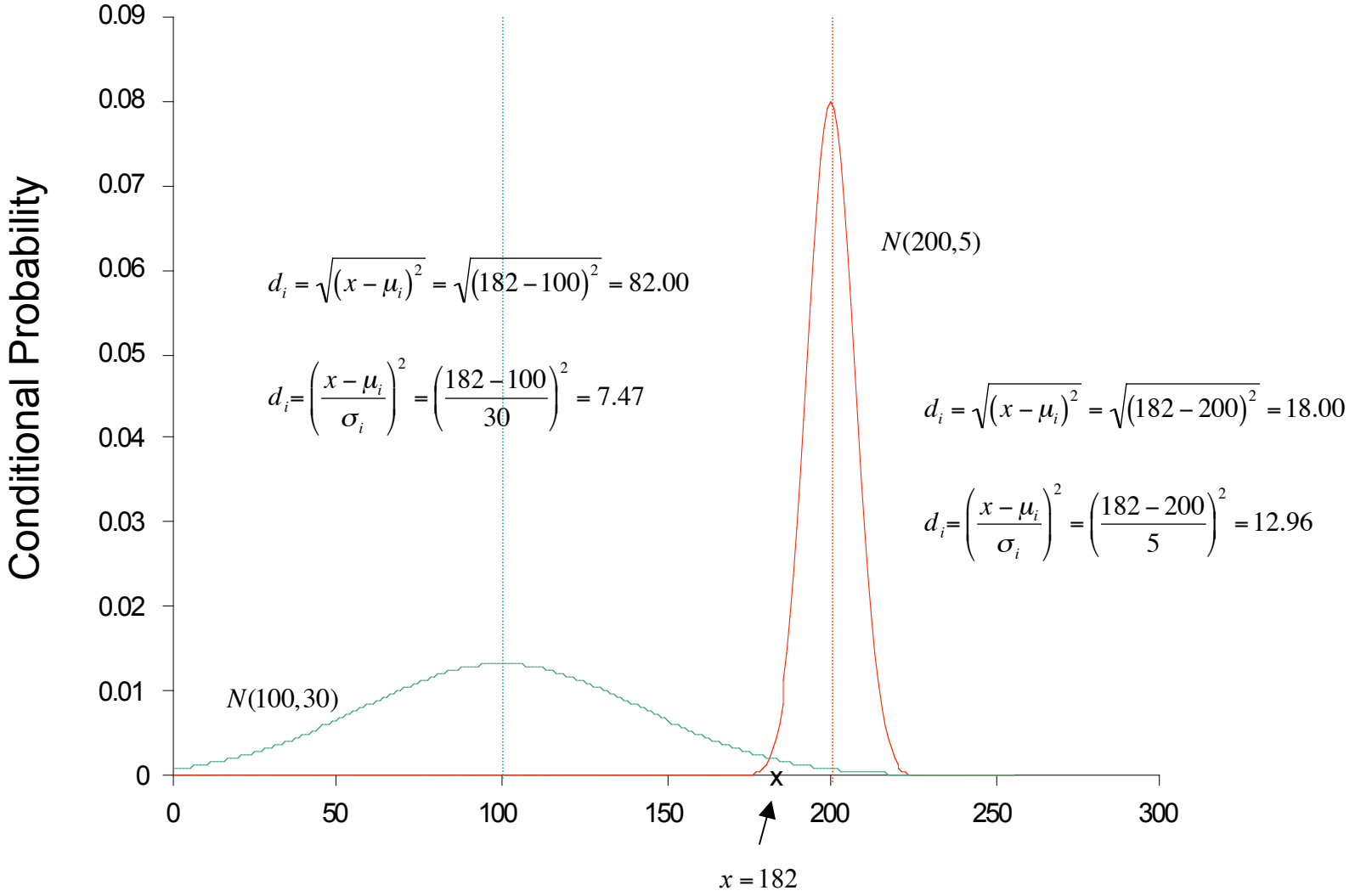
Statistical Distance

Univariate Case



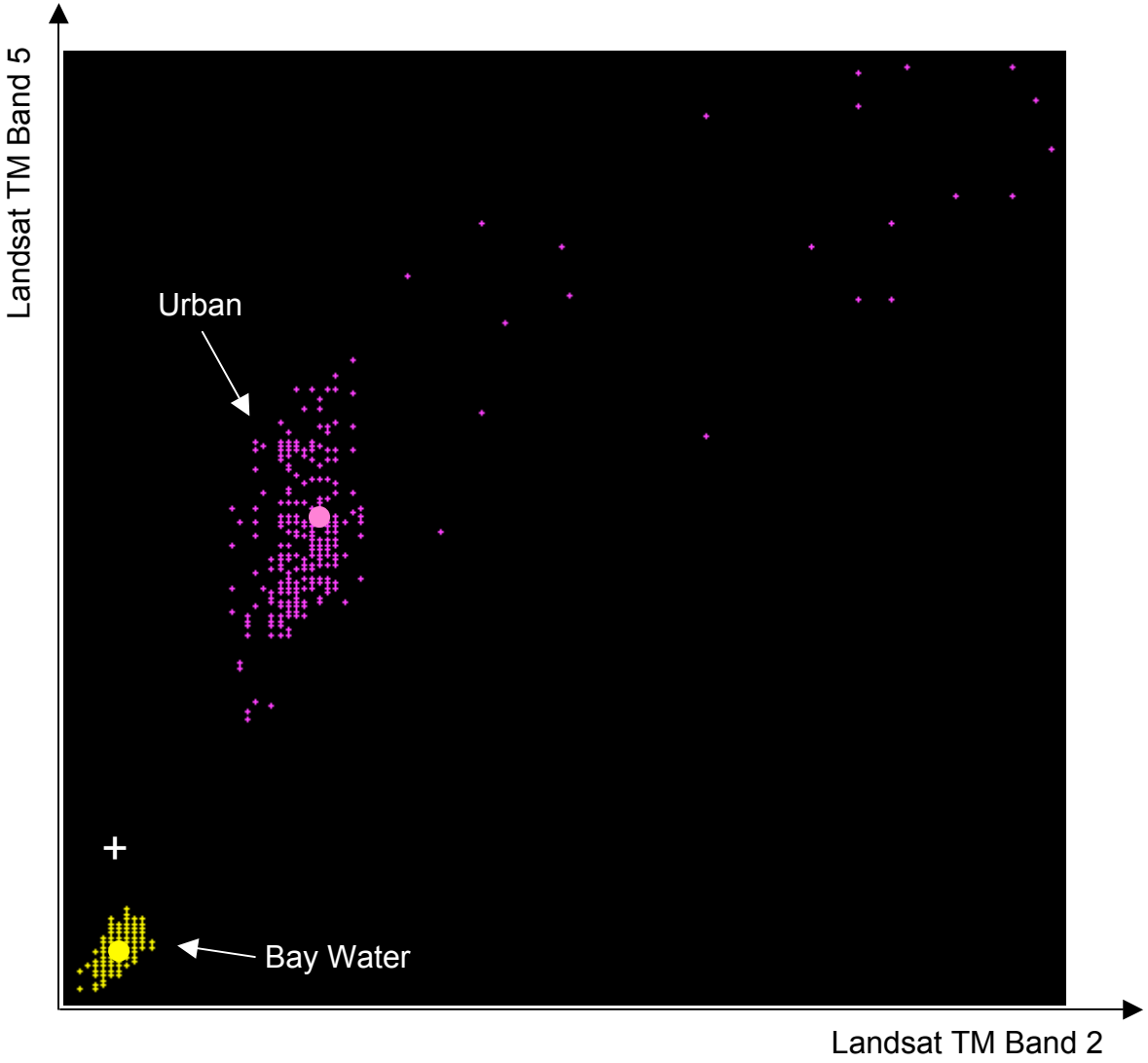
Statistical Distance

Univariate Case



Statistical Distance

Bi-variate Case



Statistical Distance

Multivariate Case

As the bi-variate case on the previous slide has exemplified, in addition to accounting for the spread of the data along the individual observation variable axes, there is a need to account for the correlation/covariance that exist between the variables on these axes. The equivalent multivariate measure of statistical distance is know as the *Mahalanobis distance*

$$d_i = \left(\mathbf{x} - \boldsymbol{\mu}_i\right)^t \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x} - \boldsymbol{\mu}_i\right)$$

which resembles the multivariate equivalent of the univariate squared z-score

$$d_i = \left(\frac{x - \mu_i}{\sigma_i}\right)^2$$

and which is also found explicitly in the exponent of the probability distribution function for a multivariate normal distribution

$$p(\mathbf{x} \mid \omega_i) = \frac{1}{2\pi^{\frac{N}{2}} \left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)}$$

Statistical Distance

A Closer Look at Covariance

The covariance matrix for class i is more clearly denoted as

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{11,i} & \sigma_{12,i} & \cdots & \sigma_{1n,i} \\ \sigma_{21,i} & \sigma_{22,i} & & \\ \vdots & & \ddots & \\ \sigma_{n1,i} & & & \sigma_{nn,i} \end{bmatrix} = \begin{bmatrix} \sigma_{1,i}^2 & \sigma_{12,i} & \cdots & \sigma_{1n,i} \\ \sigma_{21,i} & \sigma_{2,i}^2 & & \\ \vdots & & \ddots & \\ \sigma_{n1,i} & & & \sigma_{n,i}^2 \end{bmatrix}$$

$$\sigma_{kl,i} = \sum_{n=1}^{N_i} \frac{\left(x_{k,n} - \mu_{k,i}\right)\left(x_{l,n} - \mu_{l,i}\right)}{N_i - 1}$$

NOTE: This term reduces to variance when $k = l$

which can be converted to correlation between bands k and l by dividing by the product of standard deviations for the individual bands, k and l, for class i

$$\rho_{kl,i} = \frac{\sigma_{kl,i}}{\sqrt{\sigma_{k,i}\sigma_{l,i}}}$$

Statistical Distance

A Closer Look at Covariance

If the individual bands exhibit no correlation, the covariance matrix is diagonal (0 in all the off-diagonal terms), and if a constraint is imposed that the spread of the data is equal along all observation axes, we have

$$\Sigma_i = \begin{bmatrix} \sigma_{1,i}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{2,i}^2 & & \\ \vdots & & \ddots & \\ 0 & & & \sigma_{n,i}^2 \end{bmatrix} = \begin{bmatrix} \sigma_i^2 & 0 & \cdots & 0 \\ 0 & \sigma_i^2 & & \\ \vdots & & \ddots & \\ 0 & & & \sigma_i^2 \end{bmatrix}$$

and the Mahalanobis distance reduces to a scaled-squared Euclidean distance as follows

$$\begin{aligned} d_i &= \left(\mathbf{x} - \boldsymbol{\mu}_i\right)^t \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x} - \boldsymbol{\mu}_i\right) \\ &= \left(\mathbf{x} - \boldsymbol{\mu}_i\right)^t \boldsymbol{\Sigma}_i^{t} \left(\mathbf{x} - \boldsymbol{\mu}_i\right) \\ &= \left[(x_1 - \mu_{i,1})^2 \sigma_i^2 + (x_1 - \mu_{i,1})^2 \sigma_i^2 + \ldots + (x_N - \mu_{i,N})^2 \sigma_i^2\right] \\ &= \sigma_i^2 \sum_{n=1}^{N} \left(x_n - \mu_{i,n}\right)^2 \end{aligned}$$

Classification

Maximum Likelihood

Let all the classes (types) of pixels in an image be

$$\omega_i, \quad i = 1,\ 2,\ \ldots, M$$

where M is the total number of classes present. To determine which class a pixel at position $\mathbf{x}$ in N-dimensional space belongs to, we need only look at the conditional probabilities that the class is ω_i given that we are examining a pixel at position $\mathbf{x}$ for each class

$$p(\omega_i \mid \mathbf{x}), \quad i = 1,\ 2,\ \ldots, M$$

which give the probability that the correct class is ω_i for the current pixel. The classification of pixels follow the rule

$$\mathbf{x} \in \omega_i \quad \text{if} \quad p(\omega_i \mid \mathbf{x}) > p(\omega_j \mid \mathbf{x}) \quad \text{for all} \quad j \neq i$$

Classification

Maximum Likelihood

PROBLEM: The conditional probabilities $p(\omega_i \mid \mathbf{x})$ are unknown.

SOLUTION: The conditional probabilities $p(\mathbf{x} \mid \omega_i)$ can be estimated by deriving the descriptive statistics from specified regions of interest as we have seen and using Bayes' theorem

$$p(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i) p(\omega_i)}{p(\mathbf{x})}$$

The previous classification rule can be reformulated as

$$\frac{p(\mathbf{x} \mid \omega_i) p(\omega_i)}{p(\mathbf{x})} > \frac{p(\mathbf{x} \mid \omega_j) p(\omega_j)}{p(\mathbf{x})}$$

$$p(\mathbf{x} \mid \omega_i) p(\omega_i) > p(\mathbf{x} \mid \omega_j) p(\omega_j)$$

namely

$$\mathbf{x} \in \omega_i \quad \text{if} \quad p(\mathbf{x} \mid \omega_i) p(\omega_i) > p(\mathbf{x} \mid \omega_j) p(\omega_j) \quad \text{for all} \quad j \neq i$$

Classification

Maximum Likelihood

For mathematical convenience, we can rewrite the previous rule as

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i$$

where $g_i(\mathbf{x})$ is known as a *discriminant function* and is defined as

$$\begin{aligned} g_i(\mathbf{x}) &= \ln\{p(\mathbf{x} \mid \omega_i)\, p(\omega_i)\} \\ &= \ln\{p(\mathbf{x} \mid \omega_i)\} + \ln\{p(\omega_i)\} \end{aligned}$$

Classification

Gaussian Maximum Likelihood

In the case of the multivariate normal distribution, the discriminant function is written as

$$g_i(\mathbf{x}) = \ln\{p(\omega_i)\} + \ln\left\{\frac{1}{2\pi^{\frac{N}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}\right\}$$

$$= \ln\{p(\omega_i)\} + \ln\left\{2\pi^{-\frac{N}{2}}|\boldsymbol{\Sigma}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}\right\}$$

$$= \ln\{p(\omega_i)\} + \ln\left\{2\pi^{-\frac{N}{2}}\right\} + \ln\left\{|\boldsymbol{\Sigma}_i|^{-\frac{1}{2}}\right\} + \ln\left\{e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}\right\}$$

$$= \ln\{p(\omega_i)\} - \frac{N}{2}\ln\{2\pi\} - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$$

Since the term $-N/2 \ln\{2\pi\}$ is constant for all classes, this simplifies to

$$g_i(\mathbf{x}) = \ln\{p(\omega_i)\} - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$$

Classification

Gaussian Maximum Likelihood

The discriminant functions for each spectral class, ω_i, set up decision surfaces in N-dimensional space. The boundaries between these surfaces set up regions for each spectral class. If an unknown pixel falls within one of these regions, it is assumed to be part of the class represented by this region.

The boundaries occur where

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0$$

and since the discriminant function is dominated by the quadratic term

$$(\mathbf{x} - \boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)$$

these boundaries will tend to be circular, parabolic, and elliptical in shape.

ENVI/IDL Programming Tips

`result = ENVI_GET_ROI_IDS( ROI_NAMES=`*classNames* `)`

`result` will contain a list of id numbers for each of the regions of interest (ROI's) associated with the current display, *classNames* will be a string containing the names of the ROI's.

`result = ENVI_GET_ROI(` *roiID* `)`

`result` will contain a list of indices that indicate where in a one-dimensional string of image data the pixels exist for the ROI *roiID* indicated.

`ENVI_SELECT, FID=`*fileID*

give you the standard ENVI file selection dialog that allows you to choose one of the image files currently loaded in ENVI. *fileID* will contain the id number for the file containing the displayed data.

ENVI/IDL Programming Tips

`result = ENVI_GET_ROI_DATA(` *roiID,* `FID=`*fileId,* `POS=`*bandArray* `)`

`result` will contain an array with the same number of columns as the specified *bandArray* and `POS` keyword and a number of rows equal to the number of points in the ROI. *roiID* is the ROI id number obtained with `ENVI_GET_ROI_IDS` and `FID=`*fileId* is the fileID obtained from `ENVI_SELECT`.

`result = CORRELATE(` *x, y,* `/COVARIANCE )`

`result` will contain the covariance between the two specified data vectors

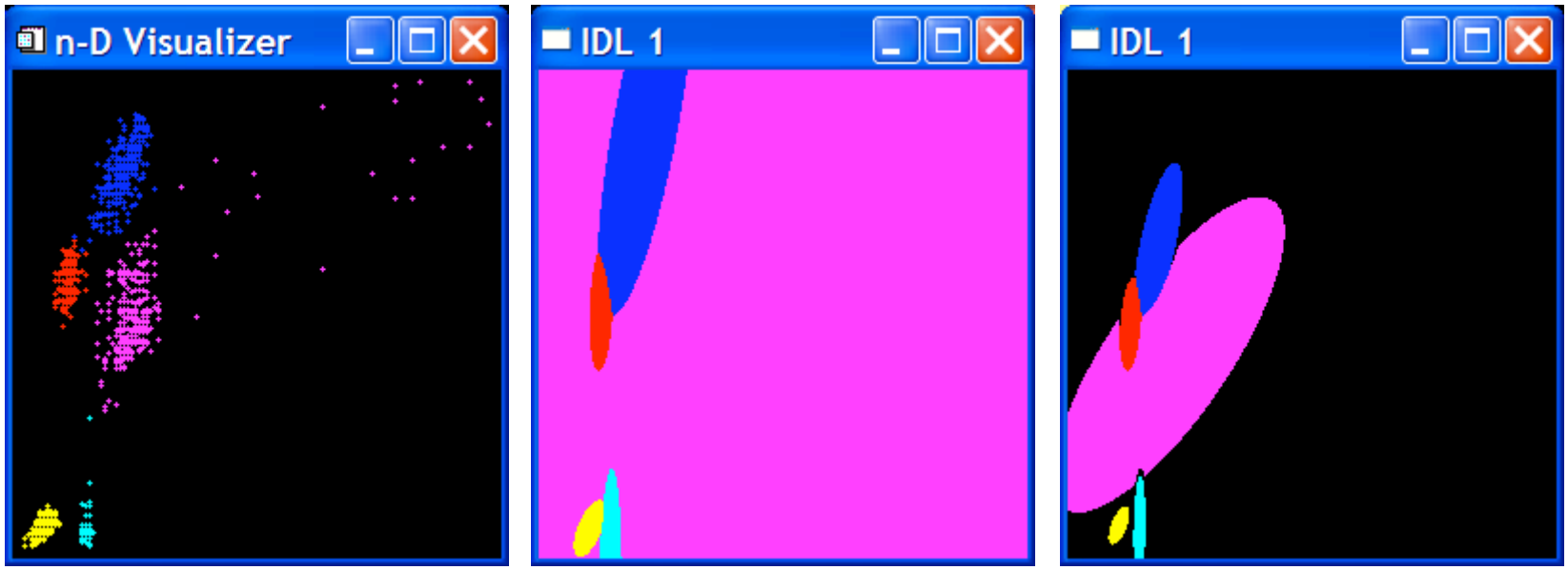
Classification

Gaussian Maximum Likelihood

No Threshold Applied

Threshold Applied

$\chi^2_{2,0.001} = 13.82$



Classification

Gaussian Maximum Likelihood

We have seen that there is overlap (sometimes significant) between the probability distribution functions defined by the training data, especially when it is assumed that the individual observations in the training data come from a multivariate normal distribution.

It is often useful to be able to reject pixels in these overlap regions, especially when the probability of class membership in either of the classes is low. For this purpose we would like to derive a threshold which the probability of class membership must exceed before assigning it to that class.

Using discriminant functions, this threshold-based decision rule would be as before

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i$$

with the added constraint

$$g_i(\mathbf{x}) > T_i$$

Classification

Gaussian Maximum Likelihood

If we expand and rearrange the decision rule as follows

$$g_i(\mathbf{x}) > T_i$$

$$\ln\{p(\omega_i)\} - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) > T_i$$

$$(\mathbf{x}-\boldsymbol{\mu}_i)^t \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) > -2T_i + 2\ln\{p(\omega_i)\} - \ln\{|\boldsymbol{\Sigma}_i|\}$$

The quadratic term on the left-hand side of the inequality (the Mahalanobis distance) has a χ^2 distribution with N degrees of freedom (where N is the dimensionality of the spectral space) if $\mathbf{x}$ is assumed to be distributed normally. Therefore, one can consult a χ^2 table to find the threshold value below which pixels will not be assigned to an individual class, namely

$$T_i = -\frac{1}{2}\chi^2_{N,\alpha} - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} + \ln\{p(\omega_i)\}$$

Classification

Gaussian Maximum Likelihood

EXAMPLE

For the reflective bands of the Landsat Thematic Mapper (6 bands), if you would like 95% of all pixels in a class to be assigned to that class, allowing 5% to remain unclassified, you would have the following

$$T_i = -\frac{1}{2}\chi^2_{6,0.95} - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} + \ln\{p(\omega_i)\}$$

$$= -\frac{1}{2}(12.59) - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} + \ln\{p(\omega_i)\}$$

$$= -6.295 - \frac{1}{2}\ln\{|\boldsymbol{\Sigma}_i|\} + \ln\{p(\omega_i)\}$$

which is a function of the prior probability and class covariance matrix for class ω_i.

Classification

Minimum Distance

It is often impossible to get an accurate estimation of the multivariate descriptive statistics, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, for a class of image pixels due to limited numbers of these pixels present in your data set or the limited ability of the user to identify a suitably large number of the pixels. As such, the underlying assumptions of Gaussian Maximum Likelihood classification is undermined.

When you can not obtain a large enough sample of data or you feel that your data set is not represented by a normally distributed probability distribution, you may be better off using a minimum distance to the mean classifier. This assigns an unknown pixel to the class, ω_i, to which it is closest (by some measure of distance).

Most commonly used is Euclidean distance represented as

$$\begin{aligned} d(\mathbf{x},\boldsymbol{\mu}_i)^2 &= (\mathbf{x}-\boldsymbol{\mu}_i)^t(\mathbf{x}-\boldsymbol{\mu}_i) \\ &= (\mathbf{x}-\boldsymbol{\mu}_i)\cdot(\mathbf{x}-\boldsymbol{\mu}_i) \\ &= \mathbf{x}\cdot\mathbf{x} - 2\boldsymbol{\mu}_i\cdot\mathbf{x} + \boldsymbol{\mu}_i\cdot\boldsymbol{\mu}_i \end{aligned}$$

Classification

Minimum Distance

In this Euclidean distance measure, the dot product of the unknown pixel with itself is common to all class mean distance terms, $d(\mathbf{x},\boldsymbol{\mu}_i)^2$, and as such, can be eliminated leaving you with the discriminant function

$$g_i(\mathbf{x}) = 2\boldsymbol{\mu}_i \cdot \mathbf{x} - \boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_i$$

with a reversal of sign to allow the same decision rule we have be using all along to be applied

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i$$

It is imperative that one notice that this minimum distance to the mean classifier relies only on the mean vector for the class, there is know distributional assumption, and as such, there is no mechanism by which the spread of the data can be taken into account.

A nice middle ground is the minimum distance to the mean approach using the Mahalanobis distance measure of distance.

Classification

Minimum Distance

Just as in the case of the Gaussian maximum likelihood classifier, the discriminant functions for the minimum distance to the mean classifier for each spectral class, ω_i, set up decision surfaces in N-dimensional space.

As before, the boundaries occur where

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0$$

and since this time, the discriminant functions are linear, these lines of demarcation between classes i and j will be lines or hyperplanes, and represented as

$$2(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot \mathbf{x} - \left(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_j\right) = 0$$

which severely limits the ability to separate classes with complicated distributional nature.

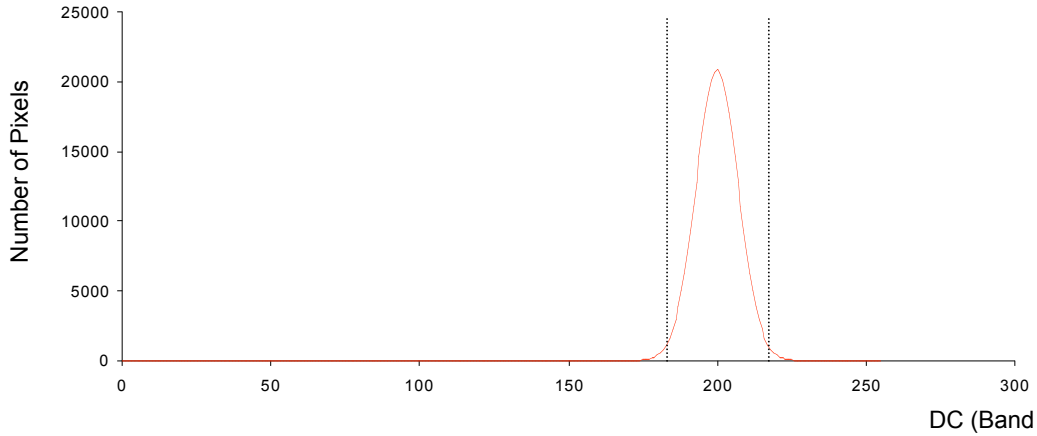
Classification

Parallelepiped

Another simple classification scheme involves the construction of hypercubes (parallelepipeds) in N-dimensional space that, if they contain an unknown pixel, are considered to be the class to which that pixel belongs.

The boundaries of these hypercubes are determined upon inspection of the histograms in each band of the data. Boundaries are placed at some point in the tails of these histogram within which a certain percentage of the data is contained.

For example, the boundaries can be drawn at the 95% point, where 95% of the data is contained within these boundaries, for example

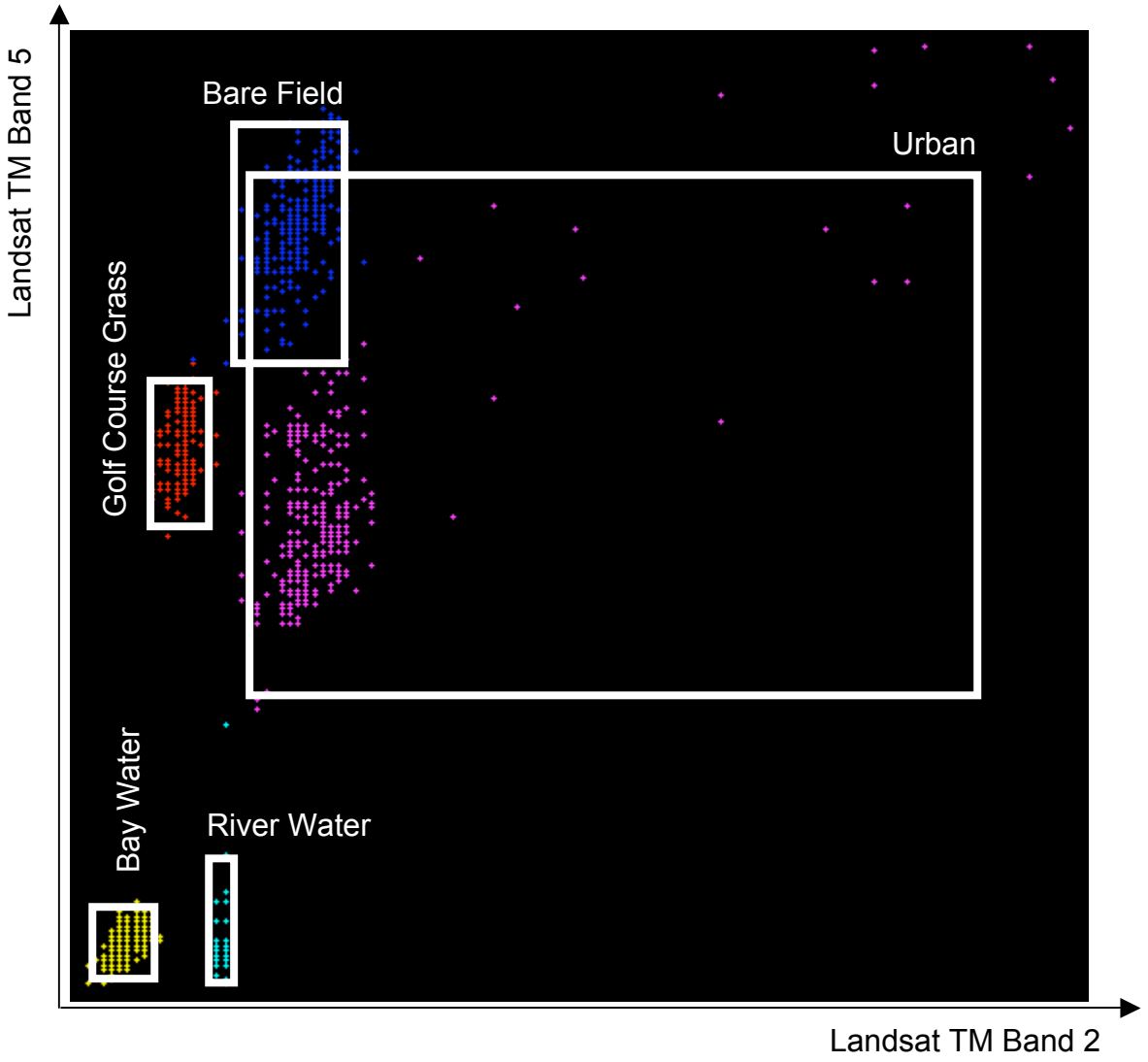


Classification

Parallelepiped

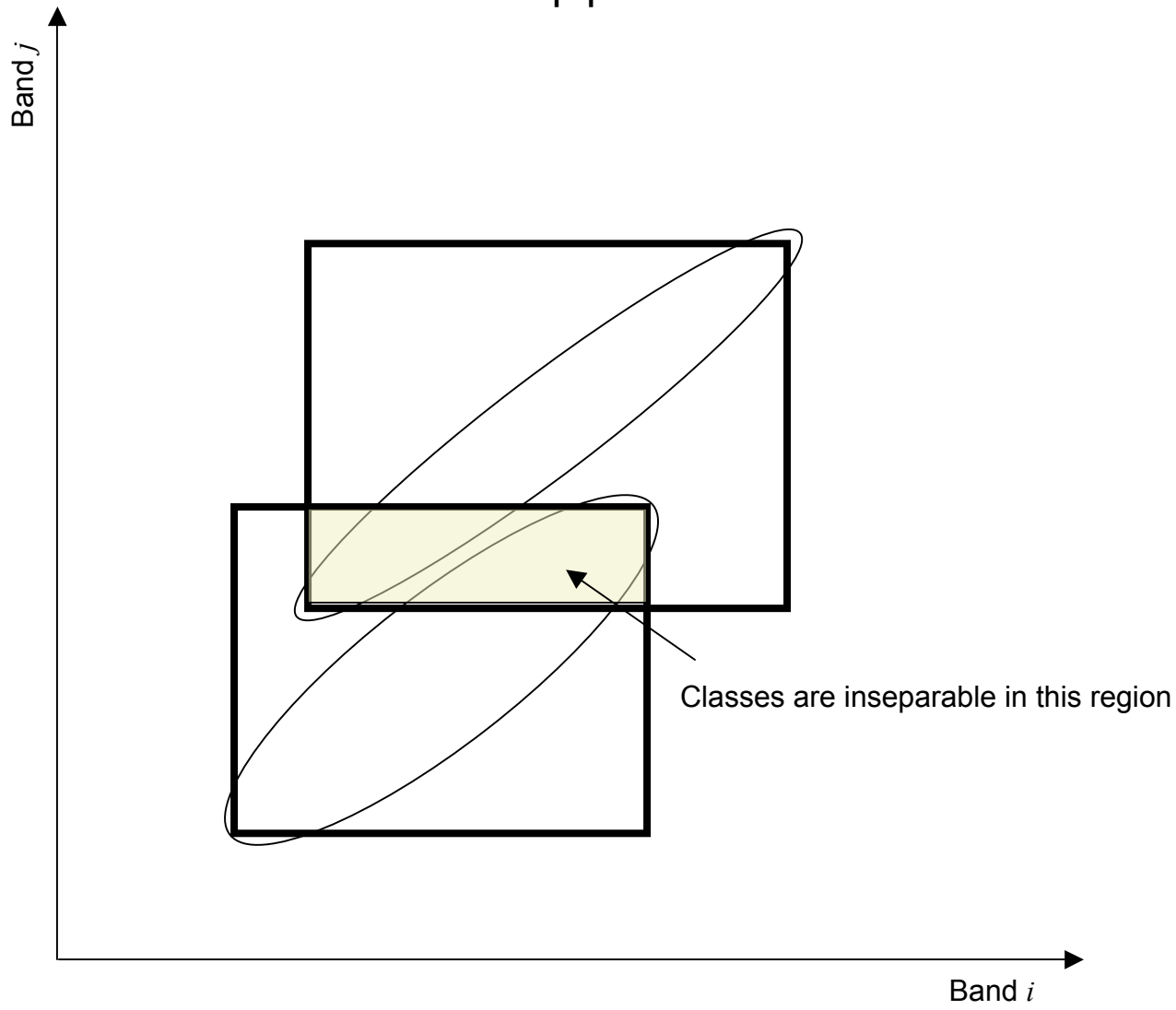
In a two dimensional space, these boundaries form boxes around the class scattergrams as shown to the left).

Note the significant overlap can occur between the boxes formed to denote classes.



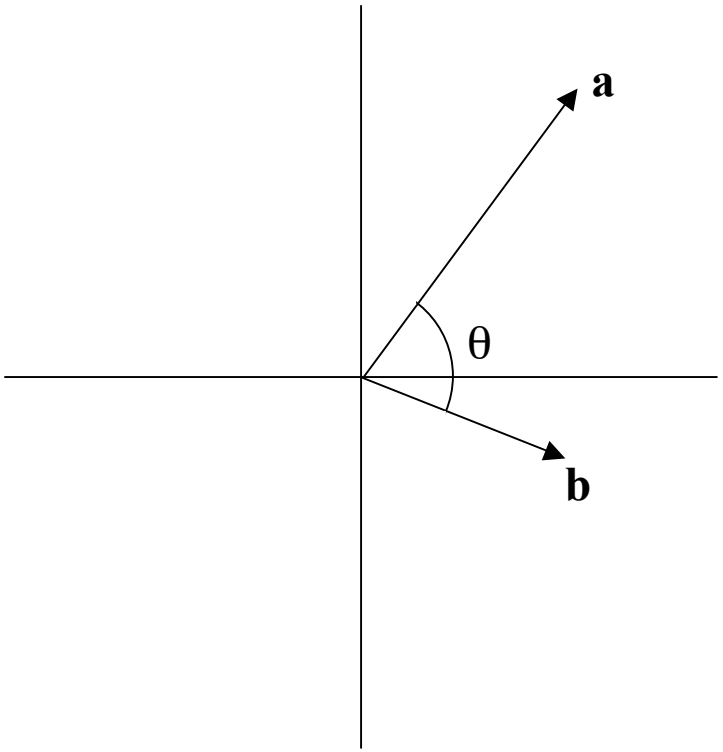
Classification

Parallelepiped



Classification

Spectral Angle Mapper (SAM)



$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

$$\bar{\mathbf{a}} = \sqrt{a_1^2 + a_2^2 + \ldots + a_N^2}$$

$$\bar{\mathbf{b}} = \sqrt{b_1^2 + b_2^2 + \ldots + b_N^2}$$

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \ldots + a_N b_N$$

$$\mathbf{a} \cdot \mathbf{b} = \bar{\mathbf{a}}\bar{\mathbf{b}} \cos\theta$$

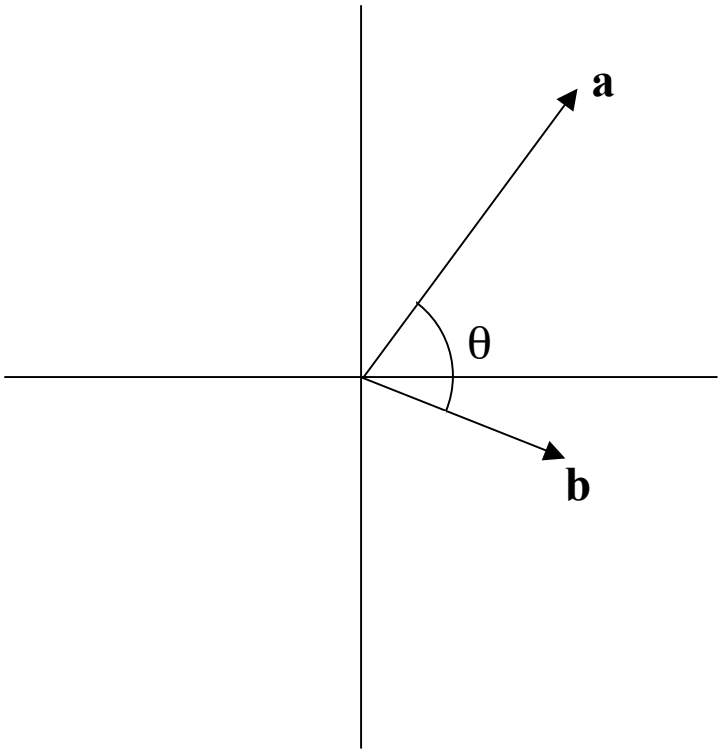
$$\cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\bar{\mathbf{a}}\bar{\mathbf{b}}}$$

$$\theta = \cos^{-1}\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\bar{\mathbf{a}}\bar{\mathbf{b}}}\right)$$

Classification

Spectral Angle Mapper (SAM)

EXAMPLE



$$\mathbf{a} = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

$$\overline{\mathbf{a}} = \sqrt{4^2 + 5^2} = \sqrt{16 + 25} = \sqrt{41} = 6.40$$

$$\overline{\mathbf{b}} = \sqrt{4^2 + (-1)^2} = \sqrt{16 + 1} = \sqrt{17} = 4.12$$

$$\mathbf{a} \cdot \mathbf{b} = (4)(4) + (5)(-1) = 16 - 5 = 11$$

$$\theta = \cos^{-1}\left(\frac{11}{(6.40)(4.12)}\right)$$

$$= \cos^{-1}(0.417172)$$

$$= 1.14046 \ \ radians$$

$$= 65.343^{\circ}$$

Classification

Spectral Angle Mapper (SAM)

The spectral angle mapper classification scheme assigns an unknown pixel $\mathbf{x}$ to the class ω_i if the angle in N-dimensional space between this unknown vector and the class mean vector is smaller than all other such angles. The decision rule is given by

$$\mathbf{x} \in \omega_i \quad \text{if} \quad \theta_i(\mathbf{x}) > \theta_j(\mathbf{x}) \quad \text{for all} \quad j \neq i$$

The power and weakness of this methodology stem from the same fact; the classification is based on the direction of the vector only, not the magnitude. The following points need to be made

1. The spectral character of a pixel defines its direction while the overall amount of reflected or emitted energy define its magnitude. This technique will allow for a sunlit grass and a shaded grass pixel to be assigned to a single grass category since the vector directions are the same (only the magnitudes of the vectors change).
2. If two categories exist in the same direction, with inherently different magnitudes, the spectral angle mapper will confuse their assignment. (*e.g.* Tyvek and Spectralon)

Unsupervised Classification

Some of the parametric classifiers that have been presented so far have the underlying assumption of multivariate normality.

Failure to satisfy these assumptions will result is severe degradation in performance of these methods, especially if a class distribution is multi-modal and the user has not resolved this condition during the training process.

Users tend to specify classes of pixels by function rather than by statistical uni-modality, and as such will tend to violate the assumptions quite often.

Clustering or unsupervised classification can help in the process of identifying "classes" that achieve both end.

Unsupervised Classification

Clustering and Similarity

Clustering implies grouping pixels in a multidimensional space.

Similarity is the quantitative measure that is used to say how alike pixels are to one another in this space and to assemble the clusters.

These similarity metrics are typically distance measures. The most frequently encountered are

Euclidean distance

$$d(\mathbf{x}_1,\mathbf{x}_2) = \left\|\mathbf{x}_1 - \mathbf{x}_2\right\| = \sqrt{\left(\mathbf{x}_1 - \mathbf{x}_2\right)^t\left(\mathbf{x}_1 - \mathbf{x}_2\right)} = \sqrt{\sum_{i=1}^{N}\left(x_{1i} - x_{2i}\right)^2}$$

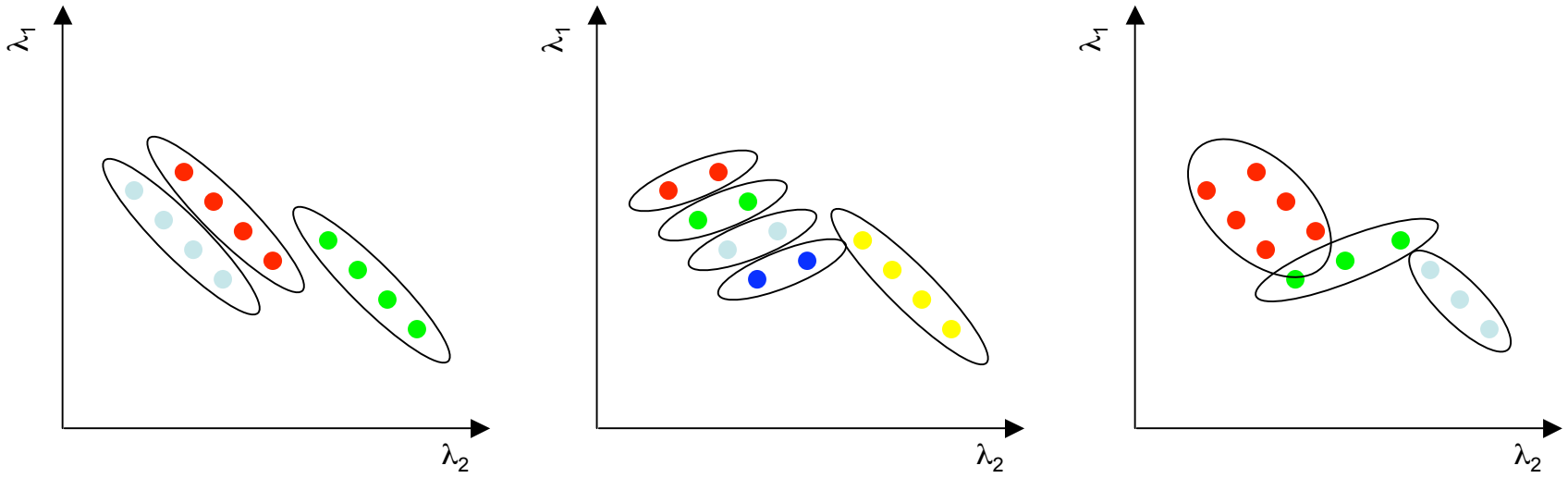
L1 distance

$$d(\mathbf{x}_1,\mathbf{x}_2) = \sum_{i=1}^{N}\left|x_{1i} - x_{2i}\right|$$

Unsupervised Classification

Clustering

Using distance measures it is possible to group the data into clusters, however, there are many clusters you could form from your data, which one is the best?



Unsupervised Classification

Clustering

A quality metric that is often used to determine which clustering of your data is best is sum of squared error (SSE) measure, namely

$$SSE = \sum_{C_i} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)^t (\mathbf{x} - \boldsymbol{\mu}_i)$$

where $\boldsymbol{\mu}_i$ is the mean of the i^{th} cluster, the outer sum is across all clusters in the current collection and the inner sum is across all point (pixels) in the i^{th} cluster of the current collection.

This represents the cumulative distance of each pattern (collection of pixels) from its cluster mean for each individual cluster summed over all clusters.

If this value is small, then the patterns are close to their respective means and the clustering is good. If it is not small, then perhaps there is a better cluster for the data.

Unsupervised Classification

ISODATA

1. Initialize the optimization with C points in the multivariate space to serve as candidate cluster means

$$\hat{\boldsymbol{\mu}}_i, \quad i = 1,2,\ldots,C$$

   No two candidate means can be the same, and, as a general rule, they should be spaced uniformly over N-dimensional space.

2. Every pixel (or subsampling of pixels) are assigned to a cluster based upon which cluster mean they are closest to (using some distance measure).

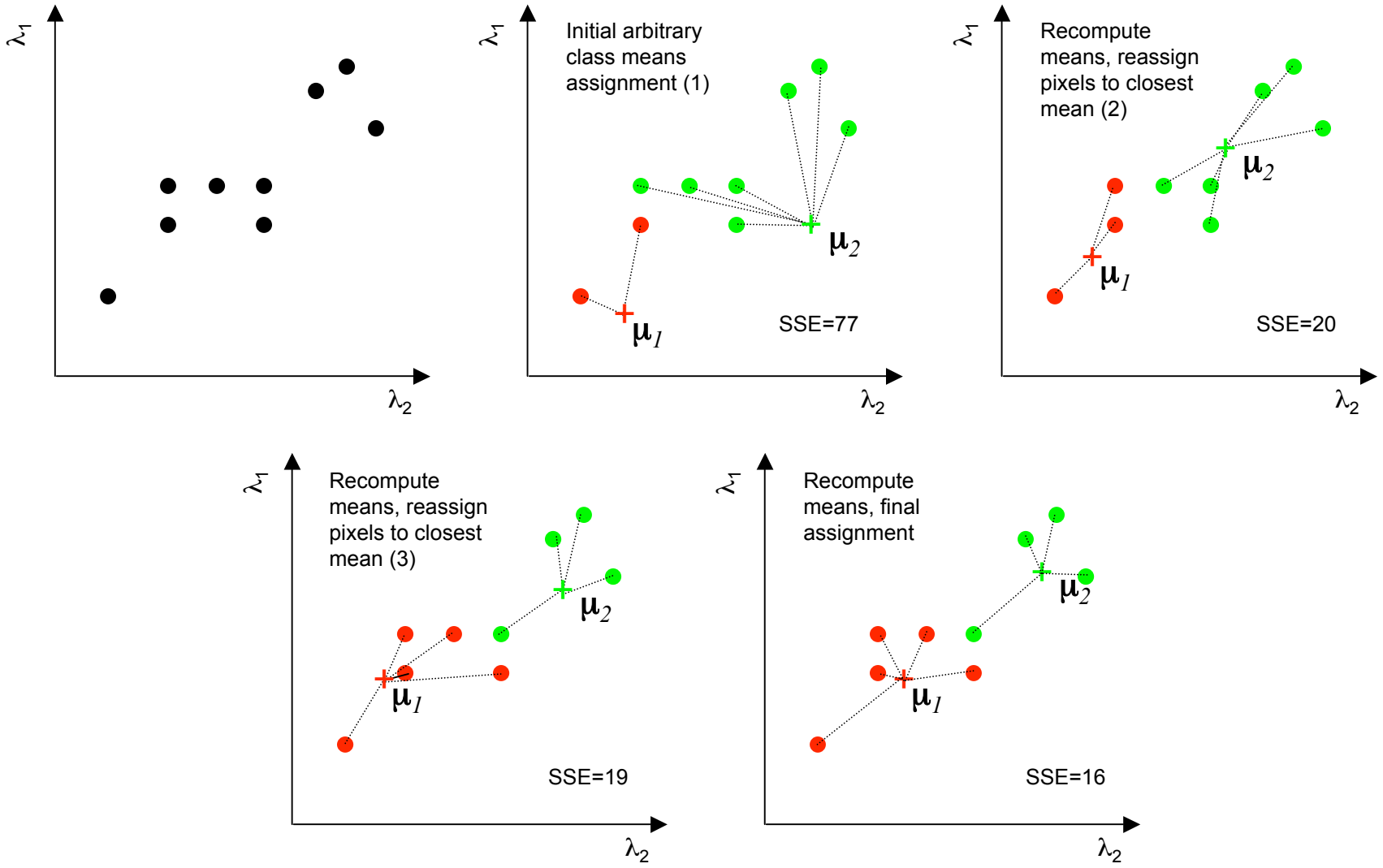
3. New cluster means are computed based upon the clusters formed in Step 2, namely

$$\boldsymbol{\mu}_i, \quad i = 1,2,\ldots,C$$

4. If $\boldsymbol{\mu}_i = \hat{\boldsymbol{\mu}}_i$ for all i, the search is over. If not, redefine the $\hat{\boldsymbol{\mu}}_i$ values to the current $\boldsymbol{\mu}_i$ values and repeats Steps 2 through 4. You could also track incremental improvements in SSE and continue as long as improvement is being made (but this is far less efficient).

Unsupervised Classification

ISODATA



Unsupervised Classification

ISODATA - Options

Deletion

Clusters might be deleted if there are two few points in the cluster from which to derive accurate descriptive statistics for the collection of points (a rule-of-thumb is $10N$ points for N-dimensional data)

Merging

Clusters may be merged if they are too close together in the multivariate space, forming one new cluster

Splitting

Elongated clusters can be split based upon a user defined set of maximum standard deviations for each spectral band, beyond which the cluster should be divided in two

ENVI/IDL Programming Tips

Clustering Program Shell

```
PRO CLUSTER_DATA

;*****************************************************************************************
; SET THE NUMBER OF CLUSTERS TO FIND
;*****************************************************************************************
   numClusters = 10

;*****************************************************************************************
; CHOOSE THE IMAGE FILE CURRENTLY LOADED INTO MEMORY TO WORK WITH
;*****************************************************************************************
   ENVI_SELECT, FID=fileID

;*****************************************************************************************
; FIND THE DIMENSIONALITY OF THE CHOSEN IMAGE
;*****************************************************************************************
   ENVI_FILE_QUERY, fileID, NS=numSamples, NL=numLines, NB=numBands

   dimensions = [ -1, 0, numSamples-1, 0, numLines-1 ]

;*****************************************************************************************
; CREATE AN IMAGE CUBE CONTAINING ALL THE BANDS OF THE CHOSEN IMAGE
;*****************************************************************************************
   image = BYTARR( numSamples, numLines, numBands )
   FOR band = 0, numBands-1 DO BEGIN
      image[*,*,band] = ENVI_GET_DATA( FID=fileID, DIMS=dimensions, POS=band )
   ENDFOR

;*****************************************************************************************
; CREATE AN INITIAL SET OF CLUSTER MEANS
;*****************************************************************************************
   randomSeed = 1000L
   clusterMean = DBLARR( numClusters, numBands )
   FOR clusterNumber = 0, numClusters-1 DO BEGIN
      clusterMean[clusterNumber,*] = RANDOMU( randomSeed, numBands ) * 255
   ENDFOR


END
```

Feature Reduction

The "cost" of classification increases with the number of spectral feature that represent each pixel. The cost of minimum distance to the mean and parallelepiped classifiers tend to increase linearly while maximum likelihood classifiers show quadratic increases in "cost".

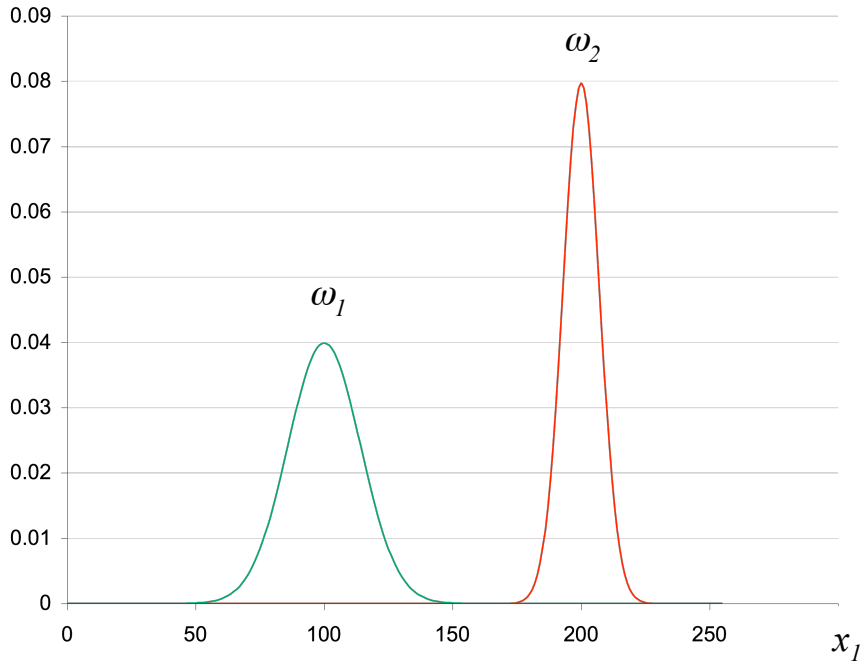
The concept of feature reduction is to eliminate features that do not contribute greatly to classification accuracy. There are two primary methods by which this is done;

1) direct elimination of features that do not contribute to the power of the classification scheme, and

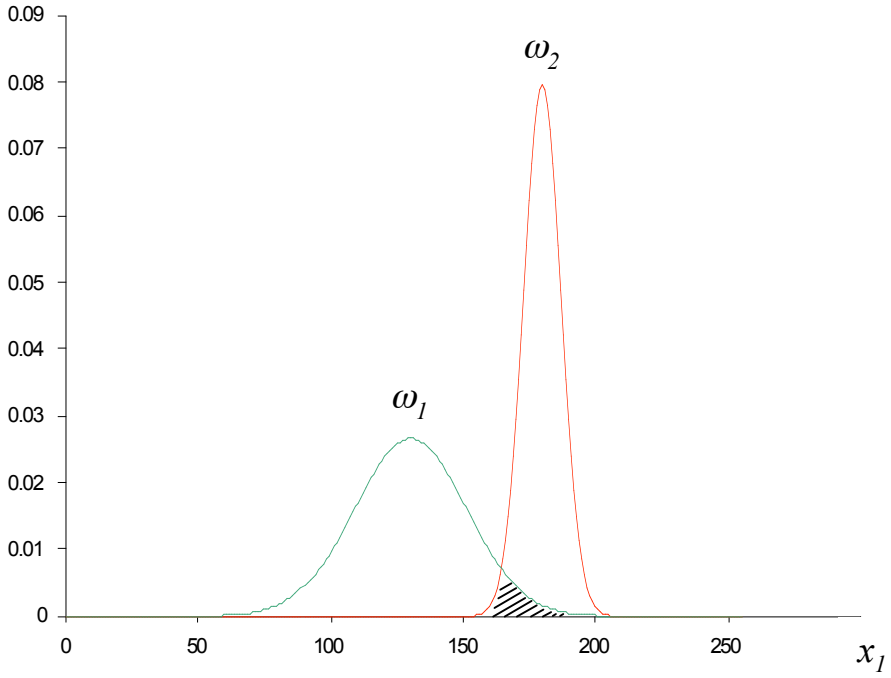
2) transformation of the spectral space associated with each pixel to form a new space in which the transformed spectral bands provide better separability for the classes of interest.

Feature Reduction

Separability



Relatively small degree of classification error would occur



Relatively high degree of classification error would occur in the overlap region

Feature Reduction

Separability

One needs to be able to quantify the degree of overlap (and therefore separability) between potential spectral classes.

The distance between means is obviously not enough on its own as this measure does not account for the overlap region. As with the statistical distance measures we've already spoken about, a combination of mean and standard deviation should work well.

Lets take a look at several measures of separability.

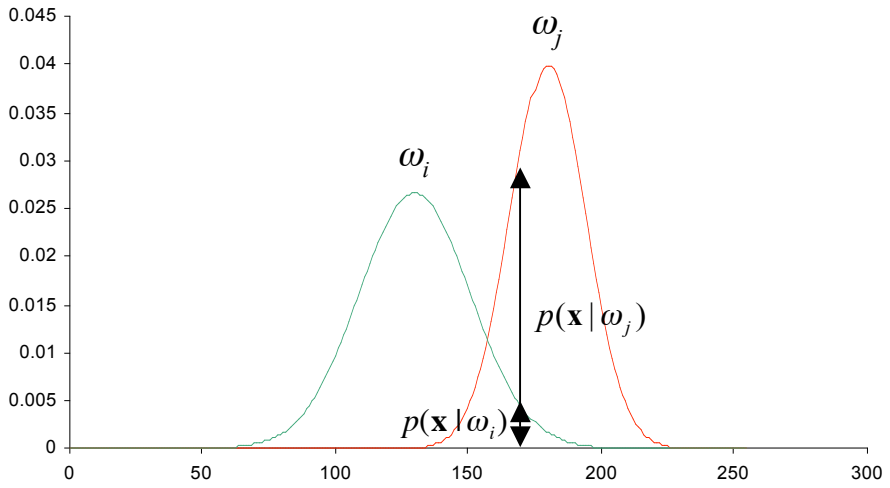
Feature Reduction

Measures of Separability - Divergence

Divergence is a measure of the degree of separability of a pair of probability distributions based on the degree of overlap. It is derived from the likelihood ratio

$$L_{ij}(\mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i)}{p(\mathbf{x} \mid \omega_j)}$$

where these represent the probabilities at position $\mathbf{x}$ of the i^{th} and j^{th} class probability distributions.



$L_{ij}(\mathbf{x}) = 0$ or ∞ for all $\mathbf{x}$ for very separable classes

Feature Reduction

Measures of Separability - Divergence

It is common to use the natural logarithm of the likelihood ratio (to avoid computational issues)

$$L_{ij}^{'}(\mathbf{x}) = \ln\left\{p(\mathbf{x} \mid \omega_i)\right\} - \ln\left\{p(\mathbf{x} \mid \omega_j)\right\}$$

and to define the *divergence* of a pair of probability distributions as

$$d_{ij} = \xi\left\{L_{ij}^{'}(\mathbf{x}) \mid \omega_i\right\} + \xi\left\{L_{ji}^{'}(\mathbf{x}) \mid \omega_j\right\}$$

where

$$\xi\left\{L_{ij}^{'}(\mathbf{x}) \mid \omega_i\right\} = \int_{\mathbf{x}} L_{ij}^{'}(\mathbf{x}) p(\mathbf{x} \mid \omega_i) d\mathbf{x}$$

which is the average likelihood ratio over all pixels in the i^{th} spectral class (the same hold true for the j^{th} spectral class and its constituents), so

Feature Reduction

Measures of Separability - Divergence

$$
\begin{aligned}
d_{ij} &= \xi\left\{L_{ij}^{'}(\mathbf{x}) \mid \omega_i\right\} + \xi\left\{L_{ji}^{'}(\mathbf{x}) \mid \omega_j\right\} \\
&= \int_{\mathbf{x}} L_{ij}^{'}(\mathbf{x}) p(\mathbf{x} \mid \omega_i) d\mathbf{x} + \int_{\mathbf{x}} L_{ji}^{'}(\mathbf{x}) p(\mathbf{x} \mid \omega_j) d\mathbf{x} \\
&= \int_{\mathbf{x}} \left[\ln\left\{p(x \mid \omega_i)\right\} - \ln\left\{p(x \mid \omega_j)\right\}\right] p(\mathbf{x} \mid \omega_i) d\mathbf{x} + \int_{\mathbf{x}} \left[\ln\left\{p(x \mid \omega_j)\right\} - \ln\left\{p(x \mid \omega_i)\right\}\right] p(\mathbf{x} \mid \omega_j) d\mathbf{x} \\
&= \int_{\mathbf{x}} \left[\ln\left\{p(x \mid \omega_i)\right\} - \ln\left\{p(x \mid \omega_j)\right\}\right] p(\mathbf{x} \mid \omega_i) d\mathbf{x} - \int_{\mathbf{x}} \left[\ln\left\{p(x \mid \omega_i)\right\} - \ln\left\{p(x \mid \omega_j)\right\}\right] p(\mathbf{x} \mid \omega_j) d\mathbf{x} \\
&= \int_{\mathbf{x}} \left[p(\mathbf{x} \mid \omega_i) - p(\mathbf{x} \mid \omega_j)\right]\left[\ln\left\{p(x \mid \omega_i)\right\} - \ln\left\{p(x \mid \omega_j)\right\}\right] d\mathbf{x} \\
&= \int_{\mathbf{x}} \left\{p(\mathbf{x} \mid \omega_i) - p(\mathbf{x} \mid \omega_j)\right\} \ln \frac{p(x \mid \omega_i)}{p(x \mid \omega_j)} d\mathbf{x}
\end{aligned}
$$

Feature Reduction

Measures of Separability - Divergence

Properties

1) Always positive
2) $d_{ij}=d_{ji}$
3) If $p(\mathbf{x}|\omega_i)=p(\mathbf{x}|\omega_j)$ for all $\mathbf{x}$ then $d_{ij}= d_{ji} = 0$; *i.e.* there is no divergence between the distribution and itself
4) For statistically independent features (spectral components)

$$x_1, x_2, \ldots, x_N \in \mathbf{x}$$

then

$$p(\mathbf{x} \mid \omega_i) = \prod_{n=1}^{N} p(x_n \mid \omega_i)$$

and

$$d_{ij}(\mathbf{x}) = \sum_{i=1}^{N} d_{ij}(x_n)$$

5) Since divergence is never negative

$$d_{ij}(x_1, x_2, \ldots, x_n, x_{n+1}) > d_{ij}(x_1, x_2, \ldots, x_n)$$

therefore, divergence never decreases as the number of features increase.

Feature Reduction

Divergence for a Pair of Normal Distributions

$$d_{ij} = \frac{1}{2} Tr\left\{\left(\Sigma_i - \Sigma_j\right)\left(\Sigma_j^{-1} - \Sigma_i^{-1}\right)\right\} + \frac{1}{2} Tr\left\{\left(\Sigma_i^{-1} + \Sigma_j^{-1}\right)\left(\mu_i - \mu_j\right)\left(\mu_i - \mu_j\right)^t\right\}$$

involves the covariance of the data only

involves the normalized distance between class means

For the case where there are more than 2 classes, all pairwise divergences need to be checked to see whether a particular feature (spectral) subset gives separable data.

An average indication of separability is the average divergence, namely

$$d_{average} = \sum_{i=1}^{M} \sum_{j=i+1}^{M} p(\omega_i) p(\omega_j) d_{ij}$$

which is weighted by the *a priori* class probabilities for the M spectral classes.

Feature Reduction

Feature Selection Using Divergence

If you have M spectral classes and N total features (or bands) and wish to pick the best n feature subset to maximize separability of these classes, you must compute all of the possible pairwise divergence values between classes for each feature combination.

There are ${}^{N}C_{n}$ possible combinations of n features from N total and for each combination there are ${}^{M}C_{2}$ pairwise divergence measures. In order to pick the best n feature combination, you must perform the following number of pairwise divergence computations

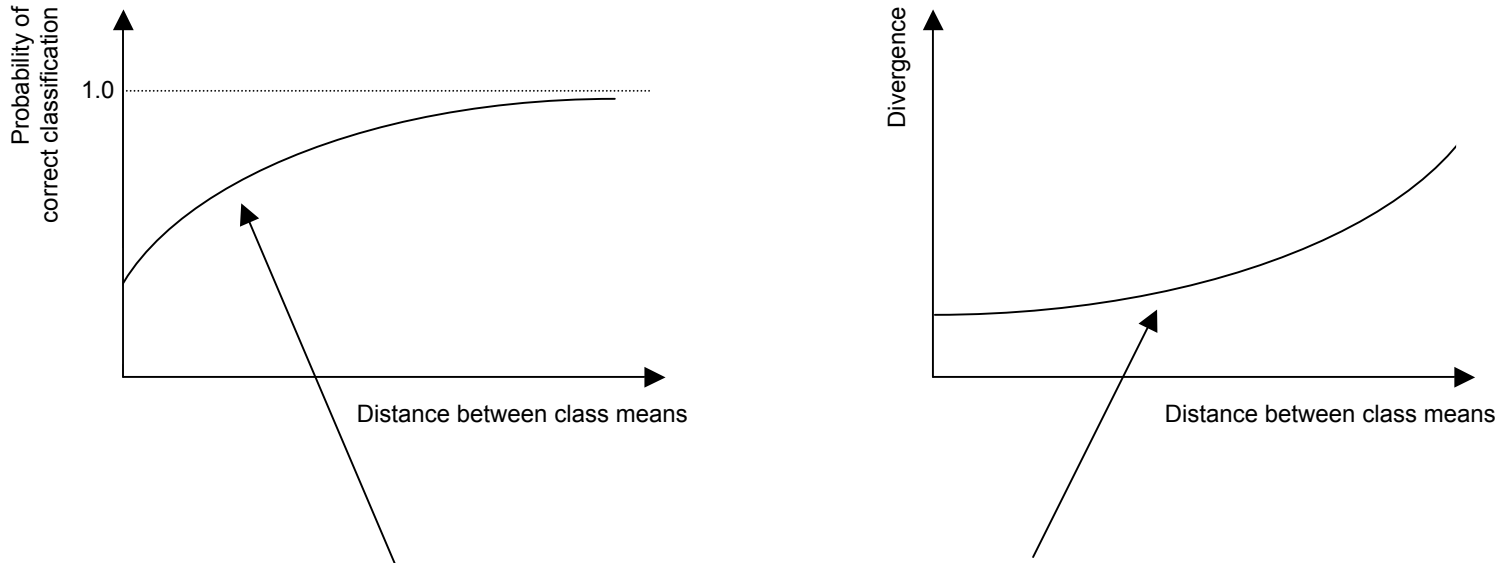
$$n_{divergences} = {}^{N}C_{n}\,{}^{M}C_{2}$$

For example, the best 4 feature subset from the 7 band Landsat TM sensor for a problem that is attempting to separate 5 classes requires the following number of pairwise divergence computations

$$n_{divergences} = {}^{7}C_{4}\,{}^{5}C_{2} = \frac{7!}{4!(7-4)!}\cdot\frac{5!}{2!(5-2)!} = \frac{7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2\cdot 1}{(4\cdot 3\cdot 2\cdot 1)(3\cdot 2\cdot 1)}\cdot\frac{5\cdot 4\cdot 3\cdot 2\cdot 1}{(2\cdot 1)(3\cdot 2\cdot 1)} = 35\cdot 10 = 350$$

Feature Reduction

Problems With Divergence



The quadratic increase in divergence with linear increase in the distance between means for normal probability distributions, is misleading since the probability of correct classification asymptotically approaches unity

Feature Reduction

Jeffries-Matusita (JM) Distance

A non-saturating measure of separability is desired. The Jeffries-Matusita distance is defined as

$$J_{ij} = \int_{\mathbf{x}} \left(\sqrt{p(\mathbf{x} \mid \omega_i)} - \sqrt{p(\mathbf{x} \mid \omega_j)} \right)^2 d\mathbf{x}$$

which is a measure of the average distance between class density functions.

For normal class distribution functions

$$J_{ij} = 2\left(1 - e^{-B}\right)$$

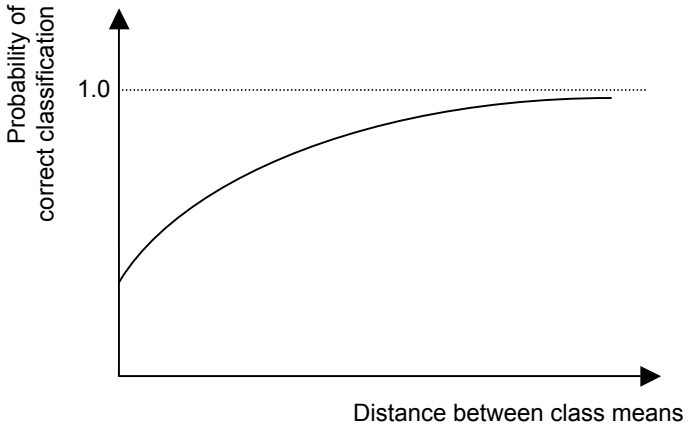
where

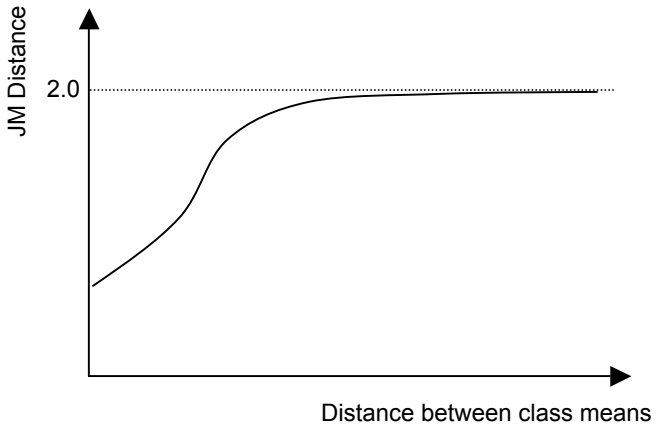
$$B = \frac{1}{8}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^t \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2} \right)^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) + \frac{1}{2} \ln \left(\frac{\left| \frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2} \right|}{\left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}} \left|\boldsymbol{\Sigma}_j\right|^{\frac{1}{2}}} \right)$$

(known as the Bhattacharyya distance)

Feature Reduction

Jeffries-Matusita (JM) Distance





Feature Reduction

Transformed Divergence

As the divergence and Bhattacharyya distance are similar in form to one another, one term based on covariance alone and the other based on a normalized distance-like measure,

$$d_{ij} = \frac{1}{2} Tr\left\{\left(\boldsymbol{\Sigma}_i - \boldsymbol{\Sigma}_j\right)\left(\boldsymbol{\Sigma}_j^{-1} - \boldsymbol{\Sigma}_i^{-1}\right)\right\} + \frac{1}{2} Tr\left\{\left(\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{-1}\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^t\right\}$$

$$B = \frac{1}{8}\left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right)^t \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right)^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\right) + \frac{1}{2} \ln\left(\frac{\left|\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right|}{\left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}} \left|\boldsymbol{\Sigma}_j\right|^{\frac{1}{2}}}\right)$$

Swain and Davis (1978) proposed a saturating term based upon divergence, transformed divergence, with the form

$$d_{ij}^T = 2\left(1 - e^{-\frac{1}{8} d_{ij}}\right)$$

P.H. Swain and S.M. Davis, (Eds.), 1978: Remote Sensing: The Quantitative Approach, New York, McGraw-Hill

Feature Reduction

Principal Components

Correlation in a data set is equivalent to redundancy.

If you can predict the value of one observation from another, then the individual observations do not carry as much information as they might.

This concept, which we first saw when examining image compression techniques, is even more evident in multispectral and hyperspectral data sources.

If we can reduce the dimensionality of a hyperspectral data set, then not only may many of the techniques that we have looked at for image classification run faster, but they would run better since the quality of the data sets are improved.

Feature Reduction

Principal Components

As we mentioned earlier, the question arises as to whether there exists an alternate N-dimensional data space in which an original data set may exist without exhibiting correlation.

If this alternate multi-dimensional data space does exist, the covariance matrix derived for this linearly-transformed data would be diagonal.

If the data points (or pixels) in this new data space were represented by y and the linear transform applied to generate the points was G, we would have the relationship

$$\mathbf{y} = \mathbf{G}\mathbf{x}$$

with the constraint that the covariance matrix of the data (pixels) in y-space is diagonal.

Feature Reduction

Principal Components

In y-space, the covariance matrix is

$$\boldsymbol{\Sigma}_y = \xi\left\{\left(\mathbf{y}-\boldsymbol{\mu}_y\right)\left(\mathbf{y}-\boldsymbol{\mu}_y\right)^t\right\}$$

where $\boldsymbol{\mu}_y$ is the mean of the transformed data set. We can express this mean value in terms of the original data as

$$\begin{aligned}
\boldsymbol{\mu}_y &= \xi\{\mathbf{y}\} \\
&= \xi\{\mathbf{G}\mathbf{x}\} \\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbf{G}\mathbf{x}_k \\
&= \mathbf{G}\frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k \qquad \text{since } \mathbf{G} \text{ is a matrix of constants} \\
&= \mathbf{G}\xi\{\mathbf{x}\} \\
&= \mathbf{G}\boldsymbol{\mu}_x
\end{aligned}$$

where $\boldsymbol{\mu}_x$ is the mean of original data.

Feature Reduction

Principal Components

and the covariance matrix can be re-written as

$$\begin{aligned}\boldsymbol{\Sigma}_y &= \xi\left\{(\mathbf{Gx}-\mathbf{G}\boldsymbol{\mu}_x)(\mathbf{Gx}-\mathbf{G}\boldsymbol{\mu}_x)^t\right\} \\ &= \xi\left\{\mathbf{G}(\mathbf{x}-\boldsymbol{\mu}_x)(\mathbf{x}-\boldsymbol{\mu}_x)^t\mathbf{G}^t\right\} \qquad \text{since } [\mathbf{Gx}]^t = \mathbf{x}^t\mathbf{G}^t \\ &= \mathbf{G}\xi\left\{(\mathbf{x}-\boldsymbol{\mu}_x)(\mathbf{x}-\boldsymbol{\mu}_x)^t\right\}\mathbf{G}^t \\ &= \mathbf{G}\boldsymbol{\Sigma}_x\mathbf{G}^t\end{aligned}$$

where $\boldsymbol{\Sigma}_x$ is the covariance matrix for the original data in x-space.

Since the requirement on $\boldsymbol{\Sigma}_y$ is that it be diagonal, then $\mathbf{G}$ is the transposed matrix of eigenvectors provided that $\mathbf{G}$ is orthogonal. If this is true, then $\boldsymbol{\Sigma}_y$ is the diagonal matrix of eigenvalues of $\boldsymbol{\Sigma}_x$.

$$\boldsymbol{\Sigma}_y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_N \end{bmatrix}$$

Feature Reduction

Principal Components

$$\Sigma_y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \lambda_N \end{bmatrix}$$

As this is a covariance matrix, it can be seen that the eigenvalues are the variances of the individual components of the transformed data set (*i.e.* the new "bands" in y-space).

Covariance matrix for principal components transform of Landsat TM image of Rochester

Principal Component Band	1	2	3	4	5	6	% Σ^2	% total
1	999.08	0.00	0.00	0.00	0.00	0.00	56.5%	56.5%
2	0.00	506.97	0.00	0.00	0.00	0.00	28.6%	85.1%
3	0.00	0.00	220.53	0.00	0.00	0.00	12.4%	97.5%
4	0.00	0.00	0.00	27.34	0.00	0.00	1.5%	99.0%
5	0.00	0.00	0.00	0.00	14.42	0.00	0.8%	99.8%
6	0.00	0.00	0.00	0.00	0.00	3.43	0.2%	100.0%

Most of the variance is contained within the first three bands

Feature Reduction

Principal Components (Example)

$\mathbf{x}$	$\mathbf{x}-\boldsymbol{\mu}_x$	$(\mathbf{x}-\boldsymbol{\mu}_x)(\mathbf{x}-\boldsymbol{\mu}_x)^t$
$\begin{bmatrix}2\\2\end{bmatrix}$	$\begin{bmatrix}-1.5\\-1.5\end{bmatrix}$	$\begin{bmatrix}2.25 & 2.25\\2.25 & 2.25\end{bmatrix}$
$\begin{bmatrix}2\\3\end{bmatrix}$	$\begin{bmatrix}-1.5\\-0.5\end{bmatrix}$	$\begin{bmatrix}2.25 & 0.75\\0.75 & 0.25\end{bmatrix}$
$\begin{bmatrix}3\\4\end{bmatrix}$	$\begin{bmatrix}-0.5\\0.5\end{bmatrix}$	$\begin{bmatrix}0.25 & -0.25\\-0.25 & 0.25\end{bmatrix}$
$\begin{bmatrix}4\\3\end{bmatrix}$	$\begin{bmatrix}0.5\\-0.5\end{bmatrix}$	$\begin{bmatrix}0.25 & -0.25\\-0.25 & 0.25\end{bmatrix}$
$\begin{bmatrix}5\\5\end{bmatrix}$	$\begin{bmatrix}1.5\\1.5\end{bmatrix}$	$\begin{bmatrix}2.25 & 2.25\\2.25 & 2.25\end{bmatrix}$
$\begin{bmatrix}5\\4\end{bmatrix}$	$\begin{bmatrix}1.5\\0.5\end{bmatrix}$	$\begin{bmatrix}2.25 & 0.75\\0.75 & 0.25\end{bmatrix}$

$$\begin{bmatrix}3.5\\3.5\end{bmatrix}$$

$$\boldsymbol{\Sigma}_x = \begin{bmatrix}1.90 & 1.10\\1.10 & 1.10\end{bmatrix}$$

$$\boldsymbol{\rho}_x = \begin{bmatrix}1.00 & 0.76\\0.76 & 1.00\end{bmatrix}$$

$$|\boldsymbol{\Sigma}_x - \lambda\mathbf{I}| = 0$$

$$\left|\begin{bmatrix}1.90 & 1.10\\1.10 & 1.10\end{bmatrix} - \lambda\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right| = 0$$

$$\begin{vmatrix}1.90-\lambda & 1.10\\1.10 & 1.10-\lambda\end{vmatrix} = 0$$

$$(1.90-\lambda)(1.10-\lambda) - (1.10)(1.10) = 0$$

$$2.09 - 1.90\lambda - 1.10\lambda + \lambda^2 - 1.21 = 0$$

$$\lambda^2 - 3.0\lambda + 0.88 = 0$$

$$\lambda = \{2.67, 0.33\}$$

$$\boldsymbol{\Sigma}_y = \begin{bmatrix}2.67 & 0\\0 & 0.33\end{bmatrix}$$

$$\sum_{i=1}^{2}\lambda_i = 2.67 + 0.33 = 3.0 = Tr\begin{pmatrix}1.90 & 1.10\\1.10 & 1.10\end{pmatrix} = Tr(\boldsymbol{\Sigma}_x) \quad QED$$

The sum of the eigenvalues is always equal to the trace of the original covariance matrix

Feature Reduction

Principal Components (Example)

It is now of interest to find the eigenvectors (*i.e.* the transformation matrix G) for this data set

$$[\boldsymbol{\Sigma}_x - \lambda_1 \mathbf{I}]\mathbf{g}_1 = 0$$

$$\begin{bmatrix} 1.90 & 1.10 \\ 1.10 & 1.10 \end{bmatrix}\begin{bmatrix} g_{1,1} \\ g_{2,1} \end{bmatrix} - \begin{bmatrix} 2.67 & 0 \\ 0 & 2.67 \end{bmatrix}\begin{bmatrix} g_{1,1} \\ g_{2,1} \end{bmatrix} = 0$$

$$\begin{bmatrix} 1.90g_{1,1} + 1.10g_{2,1} \\ 1.10g_{1,1} + 1.10g_{2,1} \end{bmatrix} - \begin{bmatrix} 2.67g_{1,1} \\ 2.67g_{2,1} \end{bmatrix} = 0$$

$$-0.77g_{1,1} + 1.10g_{2,1} = 0$$

$$1.10g_{1,1} - 1.57g_{2,1} = 0$$

$$g_{1,1} = 1.43g_{2,1} \quad \longleftarrow \text{ from either equation}$$

The resulting matrix G has to be orthogonal so that

$$G^{-1} \text{ is defined as } G^t$$

which requires the eigenvector to be normalized, namely

$$g_{1,1}^2 + g_{2,1}^2 = 1$$

Feature Reduction

Principal Components (Example)

so we have

$$(1.43g_{2,1})^2 + g_{2,1}^2 = 1$$

$$3.0449g_{2,1}^2 = 1$$

$$g_{2,1} = \sqrt{\frac{1}{3.0449}} = 0.57$$

$$g_{1,1} = 1.43(0.57) = 0.82$$

$$\mathbf{g}_1 = \begin{bmatrix} 0.82 \\ 0.57 \end{bmatrix}$$

and similarly

$$[\boldsymbol{\Sigma}_x - \lambda_2 \mathbf{I}]\mathbf{g}_2 = 0$$

$$\begin{bmatrix} 1.90 & 1.10 \\ 1.10 & 1.10 \end{bmatrix}\begin{bmatrix} g_{1,2} \\ g_{2,2} \end{bmatrix} - \begin{bmatrix} 0.33 & 0 \\ 0 & 0.33 \end{bmatrix}\begin{bmatrix} g_{1,2} \\ g_{2,2} \end{bmatrix} = 0$$

$$\begin{bmatrix} 1.90g_{1,2} + 1.10g_{2,2} \\ 1.10g_{1,2} + 1.10g_{2,2} \end{bmatrix} - \begin{bmatrix} 0.33g_{1,2} \\ 0.33g_{2,2} \end{bmatrix} = 0$$

$$1.57g_{1,2} + 1.10g_{2,2} = 0$$

$$1.10g_{1,2} + 0.77g_{2,2} = 0$$

$$g_{1,2} = -0.70g_{2,2}$$

$$(-0.70g_{2,2})^2 + g_{2,2}^2 = 1$$

$$1.49g_{2,2}^2 = 1$$

$$g_{2,2} = \sqrt{\frac{1}{1.49}} = 0.82$$

$$g_{1,2} = -0.70(0.82) = -0.57$$

$$\mathbf{g}_2 = \begin{bmatrix} -0.57 \\ 0.82 \end{bmatrix}$$

Feature Reduction

Principal Components (Example)

and finally, the principal components transformation matrix is

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 \end{bmatrix}^t = \begin{bmatrix} 0.82 & -0.57 \\ 0.57 & 0.82 \end{bmatrix}^t = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix}$$

that leads to the orthogonal data set

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.78 \\ 0.50 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3.35 \\ 1.32 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4.74 \\ 1.57 \end{bmatrix}$$

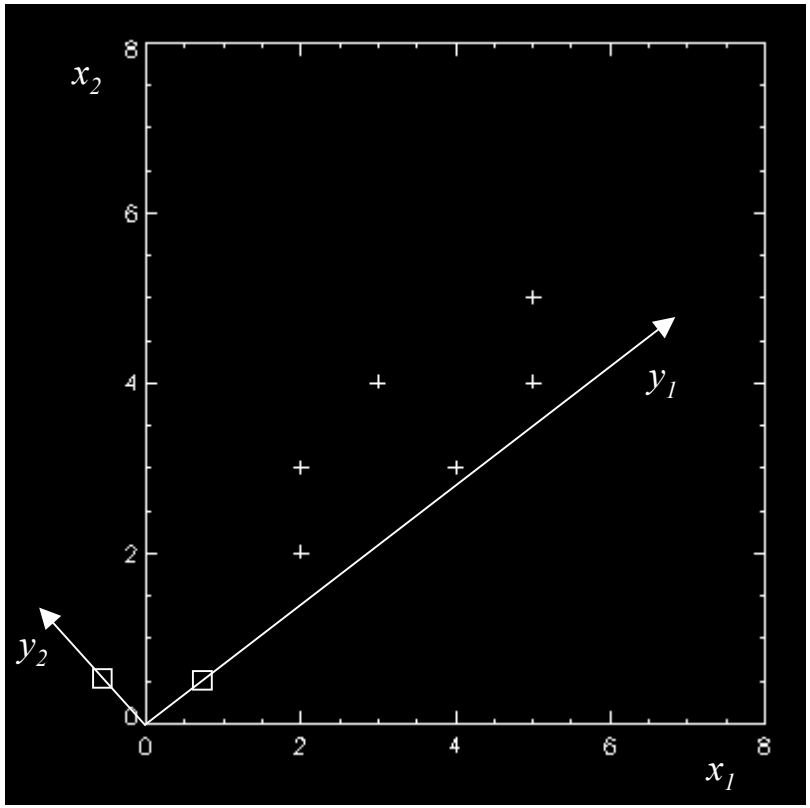
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 4.99 \\ 0.18 \end{bmatrix}$$

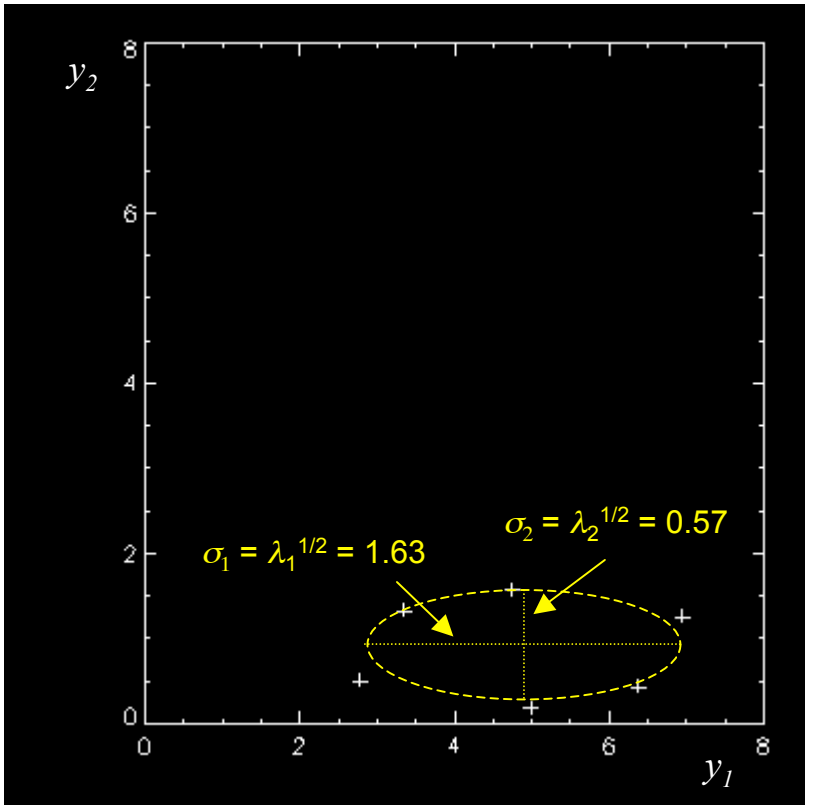
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 6.95 \\ 1.25 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 6.38 \\ 0.43 \end{bmatrix}$$

Feature Reduction

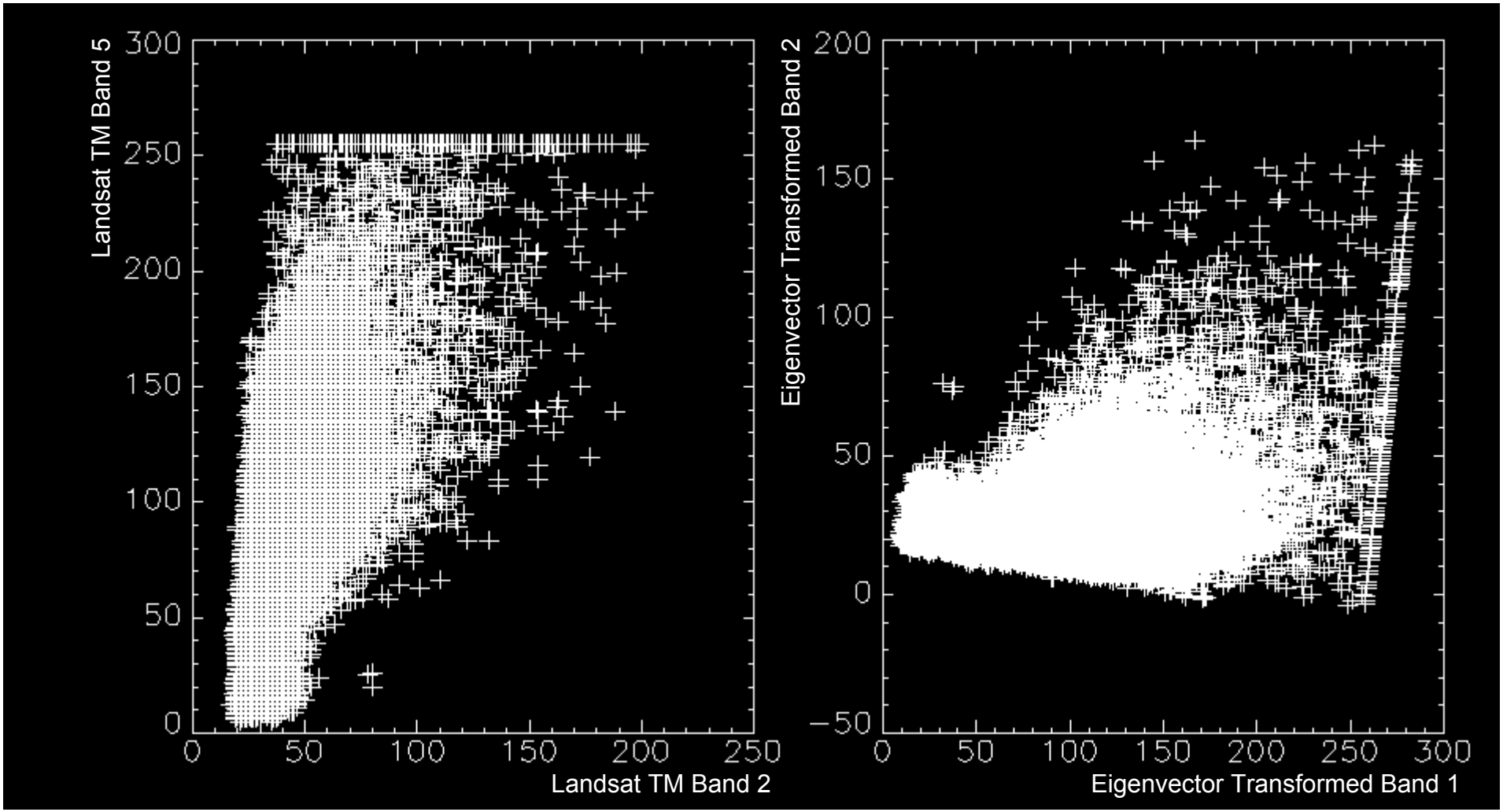
Principal Components (Example)





Feature Reduction

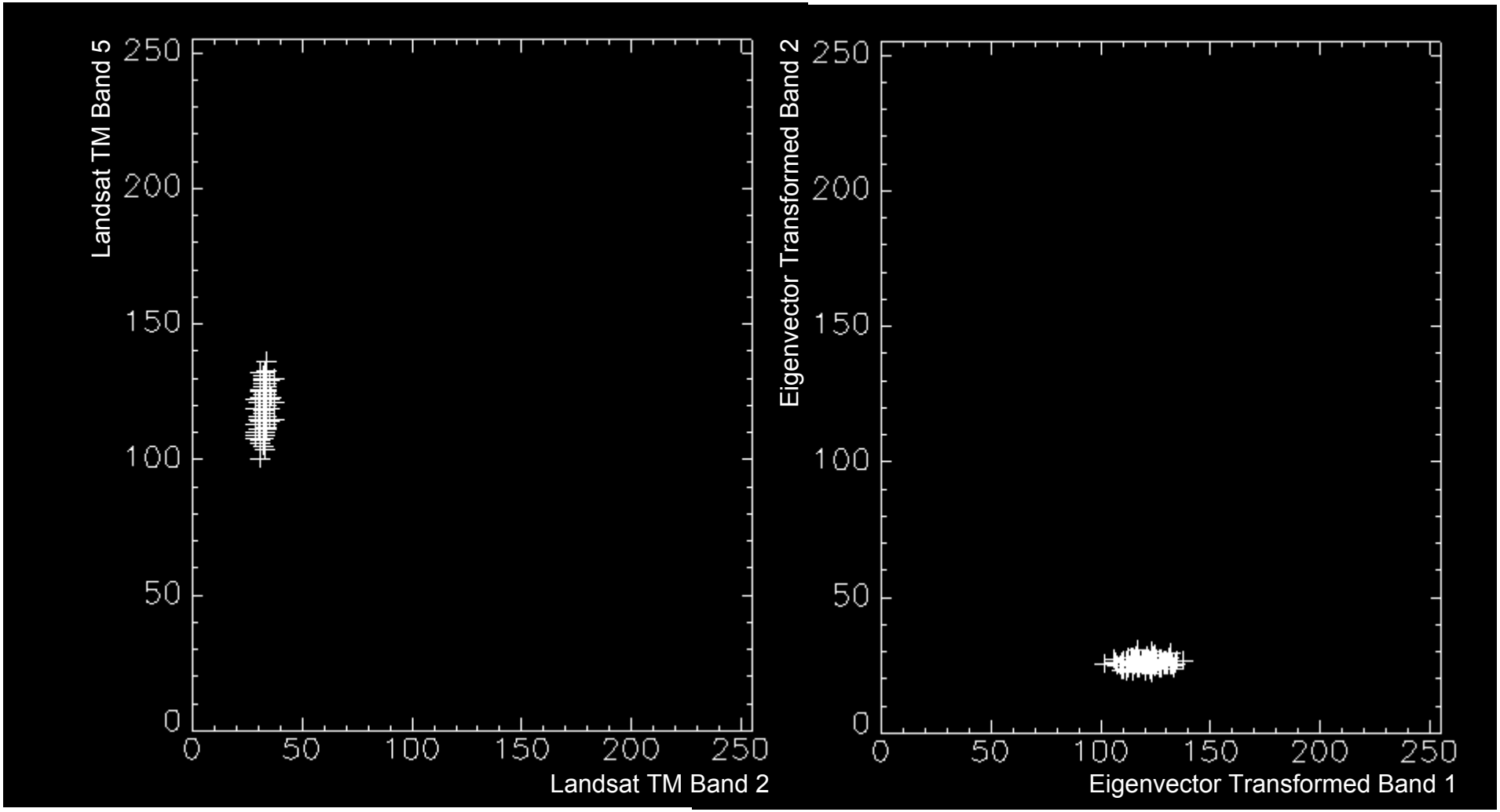
Principal Components (Example)



Entire Image

Feature Reduction

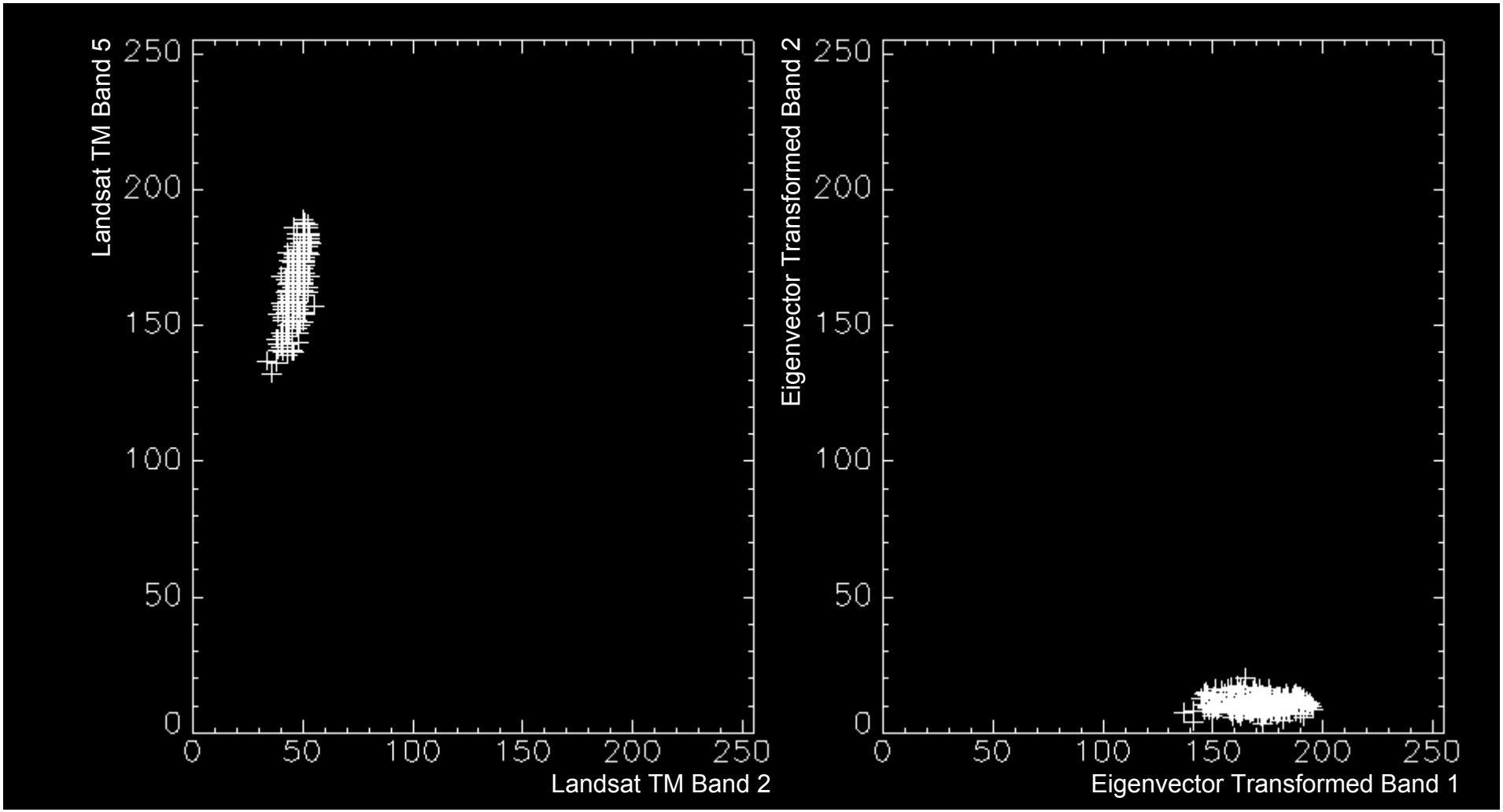
Principal Components (Example)



Golf Course Grass

Feature Reduction

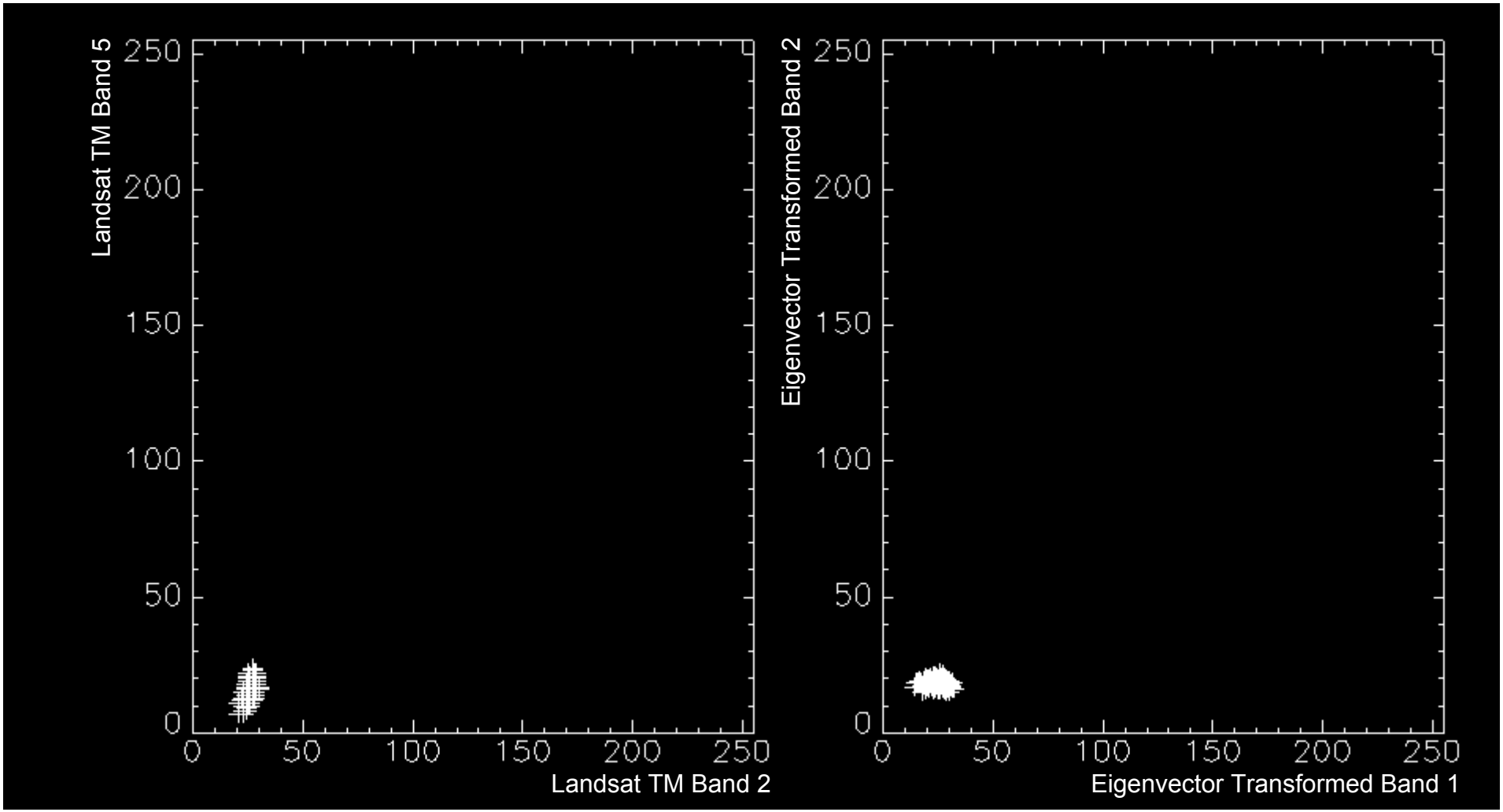
Principal Components (Example)



Bare Field

Feature Reduction

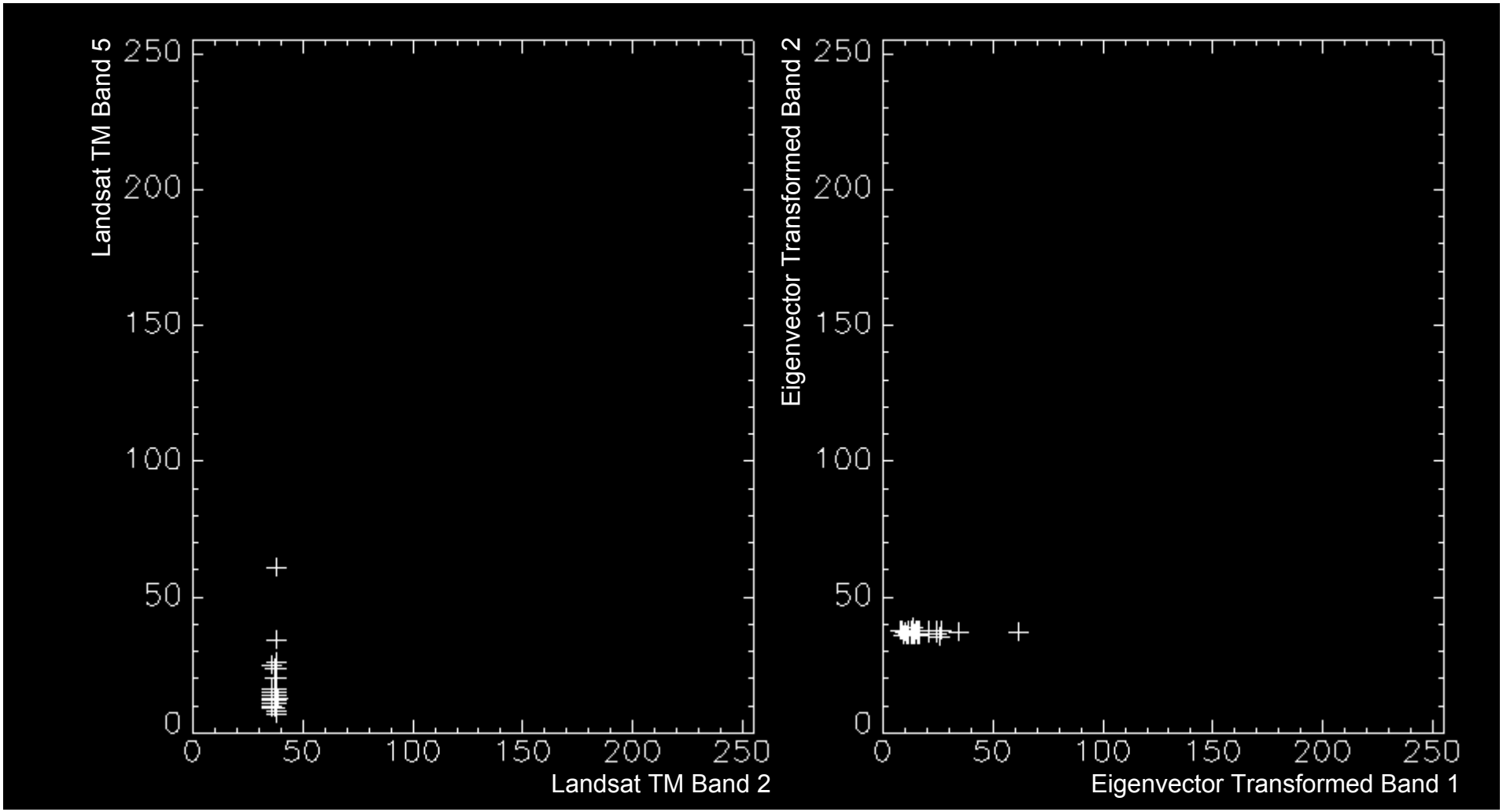
Principal Components (Example)



Bay Water

Feature Reduction

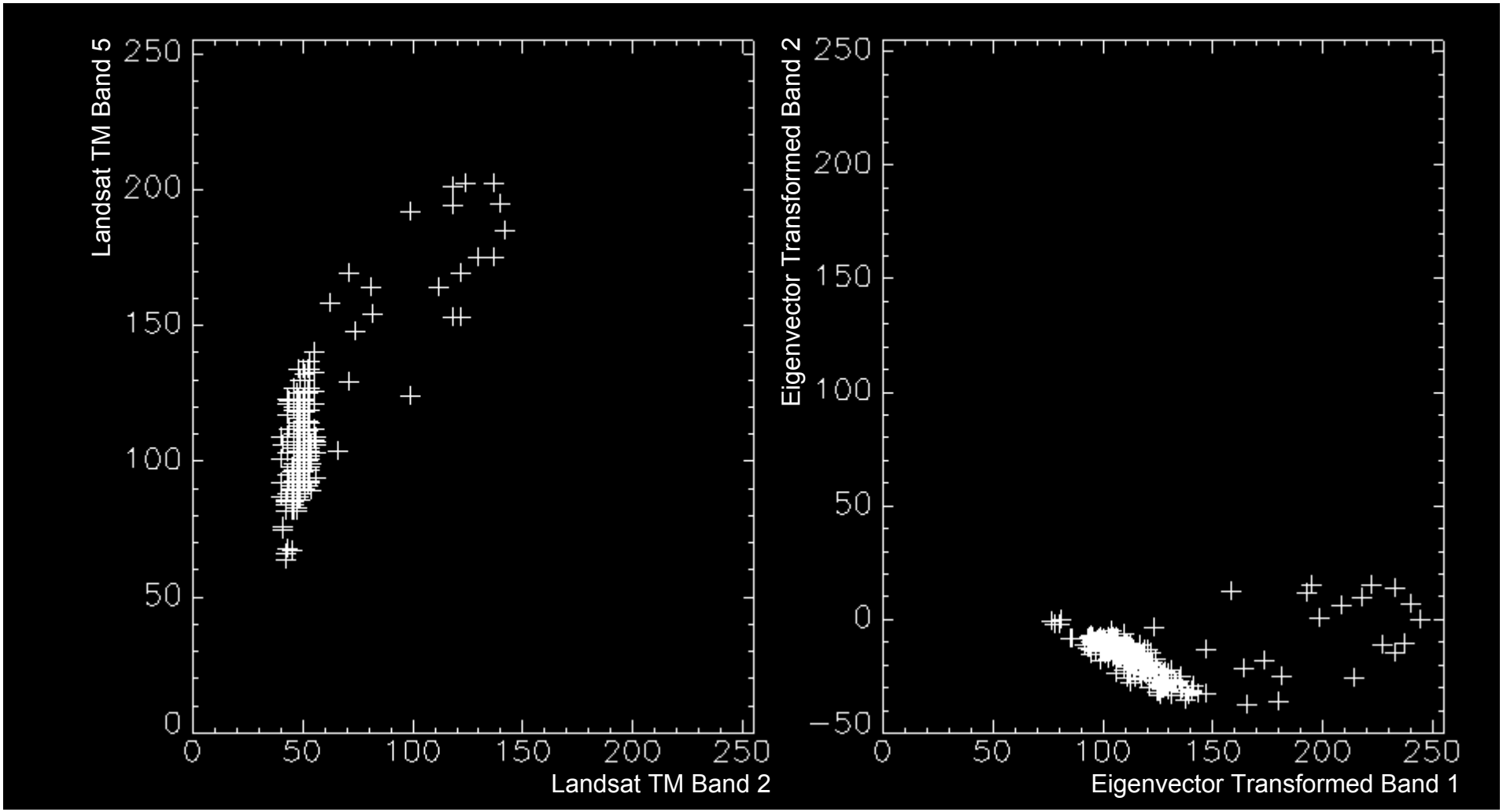
Principal Components (Example)



River Water

Feature Reduction

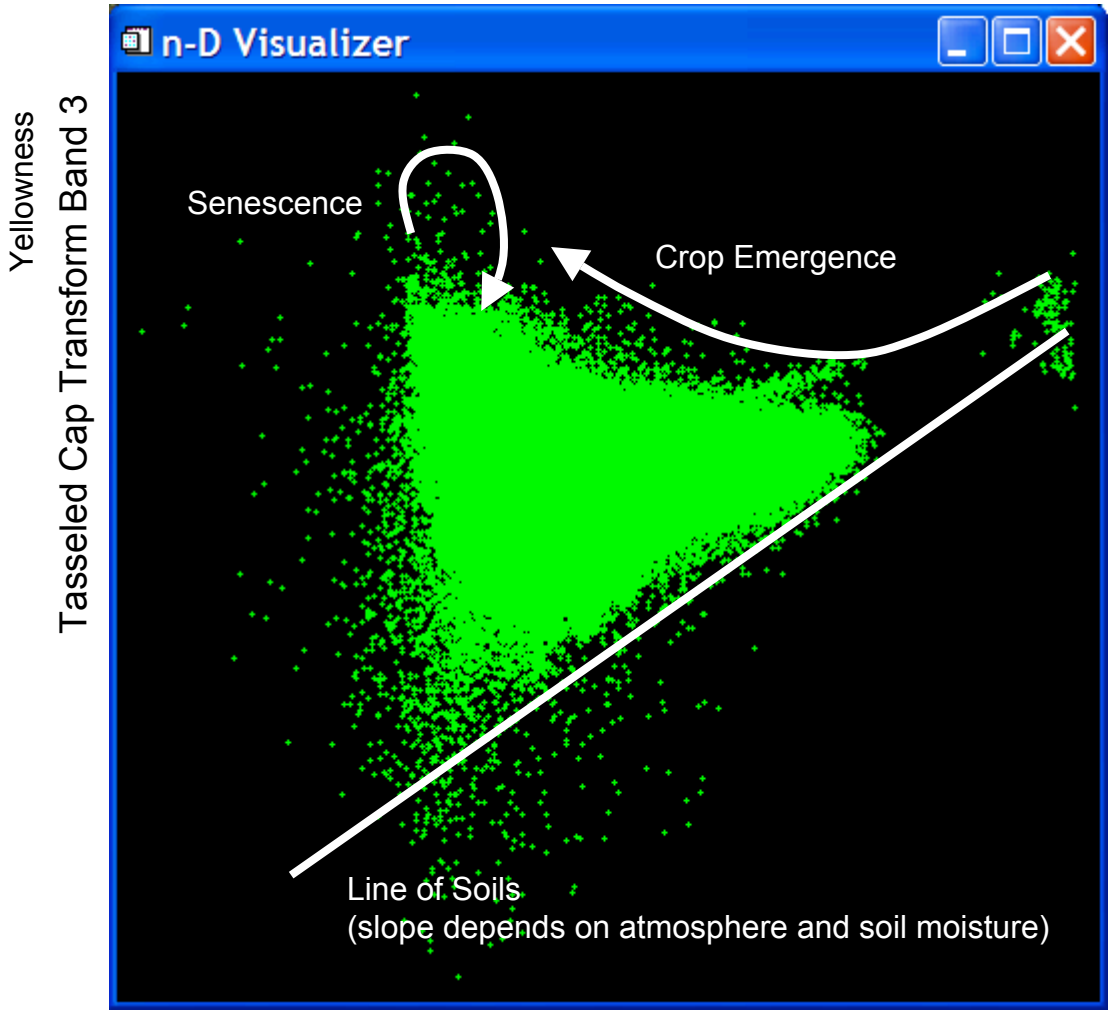
Principal Components (Example)



Urban

Feature Reduction

Tasseled Cap Transform (Kauth-Thomas Transform)



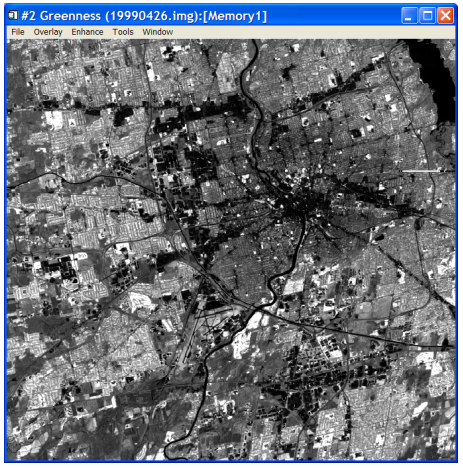
applied to Landsat TM

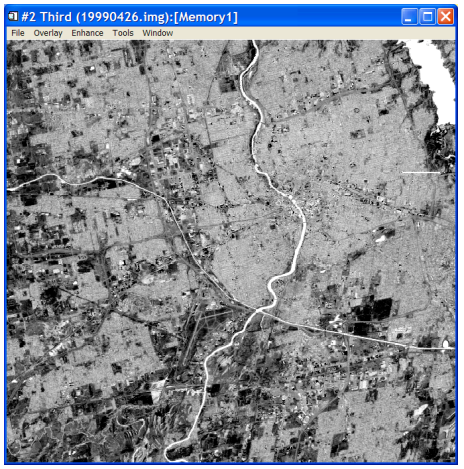
Feature Reduction

Tasseled Cap Transform (Kauth-Thomas Transform)

Landsat Thematic Mapper Transformation Coefficients

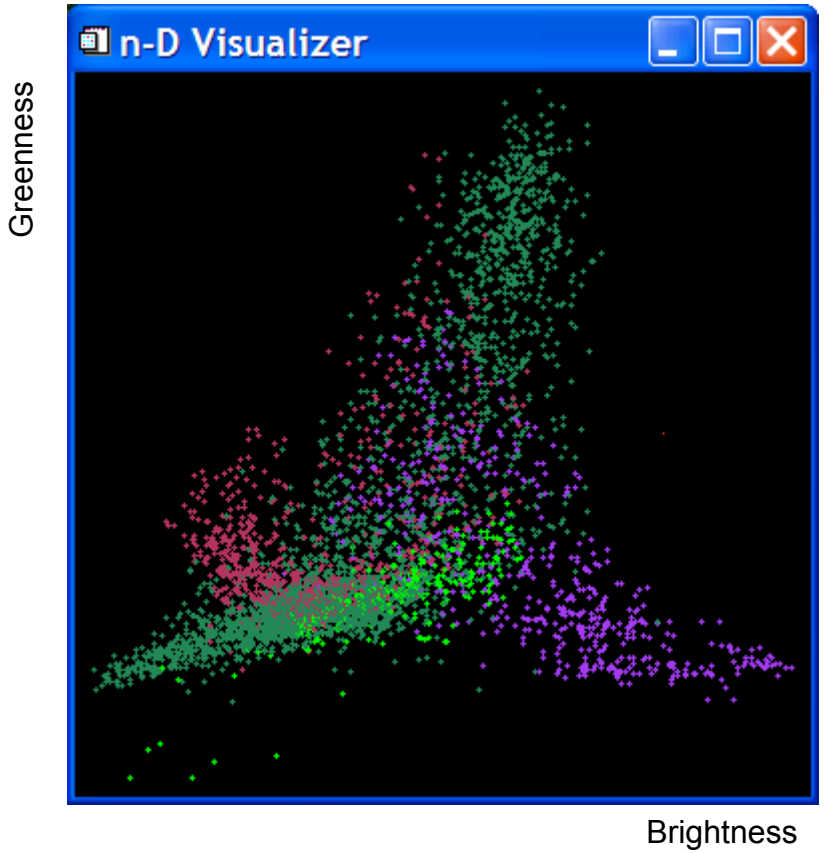
Feature	TM Band 1	TM Band2	TM Band 3	TM Band 4	TM Band 5	TM Band 7
Brightness	0.33183	0.33121	0.55177	0.42514	0.48087	0.25252
Greenness	-0.24717	-0.16263	-0.40639	0.85468	0.05493	-0.11749
Wetness	0.13929	0.22490	0.40359	0.25178	-0.70133	-0.45732
Fourth	-0.83104	0.07447	0.42144	-0.07579	0.23819	-0.25247
Fifth	-0.32530	0.05361	0.11485	0.11140	-0.46571	0.80549
Sixth	0.11381	-0.89714	0.42038	0.06686	-0.01629	0.02706

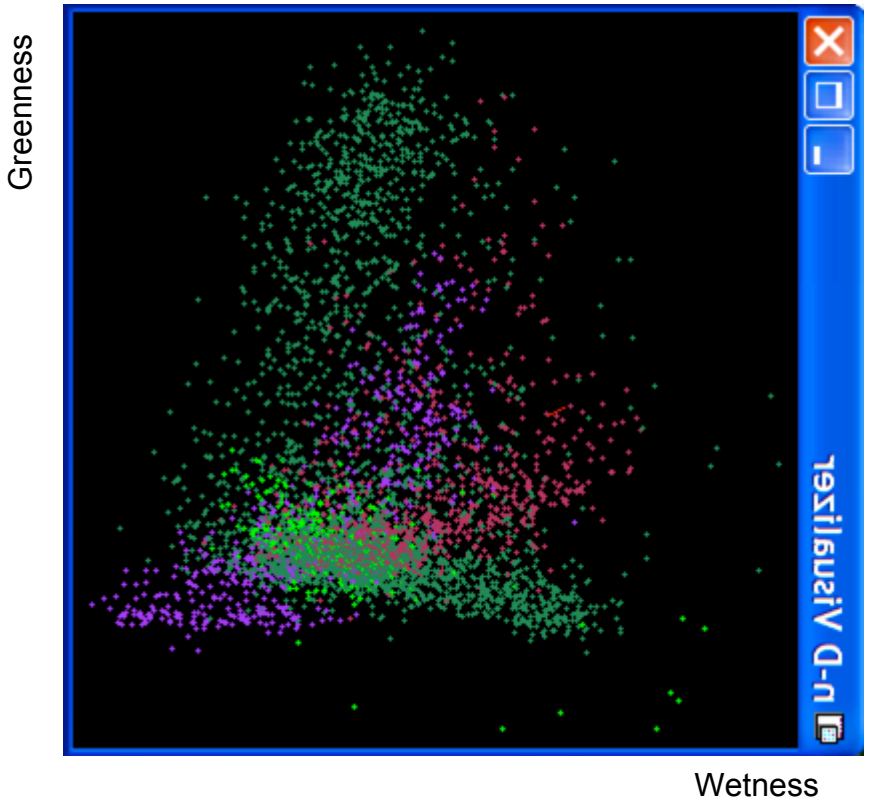




Feature Reduction

Tasseled Cap Transform (Kauth-Thomas Transform)





Feature Reduction

Minimum Noise Fraction (MNF)

Principal components transform do not always produce images that show steadily decreasing image quality with increasing component number

Principal components produces new components to maximize variance

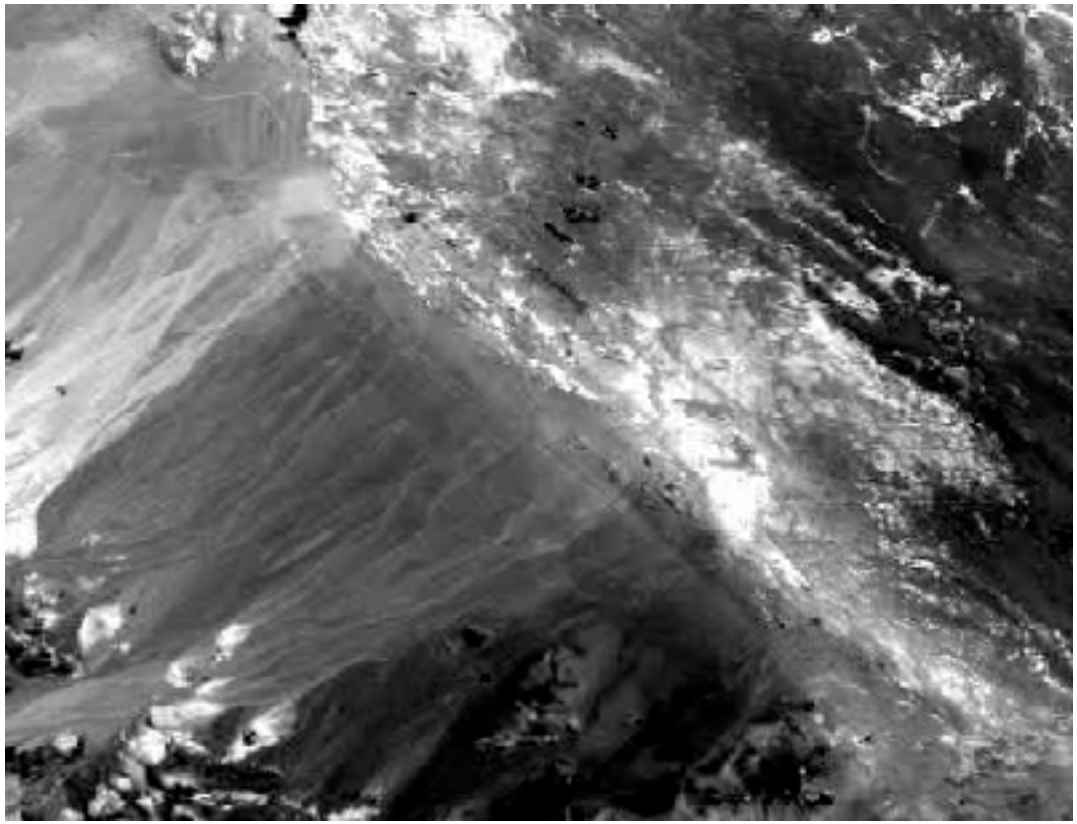
Maximum noise fraction produces new components to maximize the signal-to-noise ratio

This type of transform should always produce components that show decreasing image quality with increasing component number

Feature Reduction

Minimum Noise Fraction (MNF)

PCA Component Images



(click image to play movie)

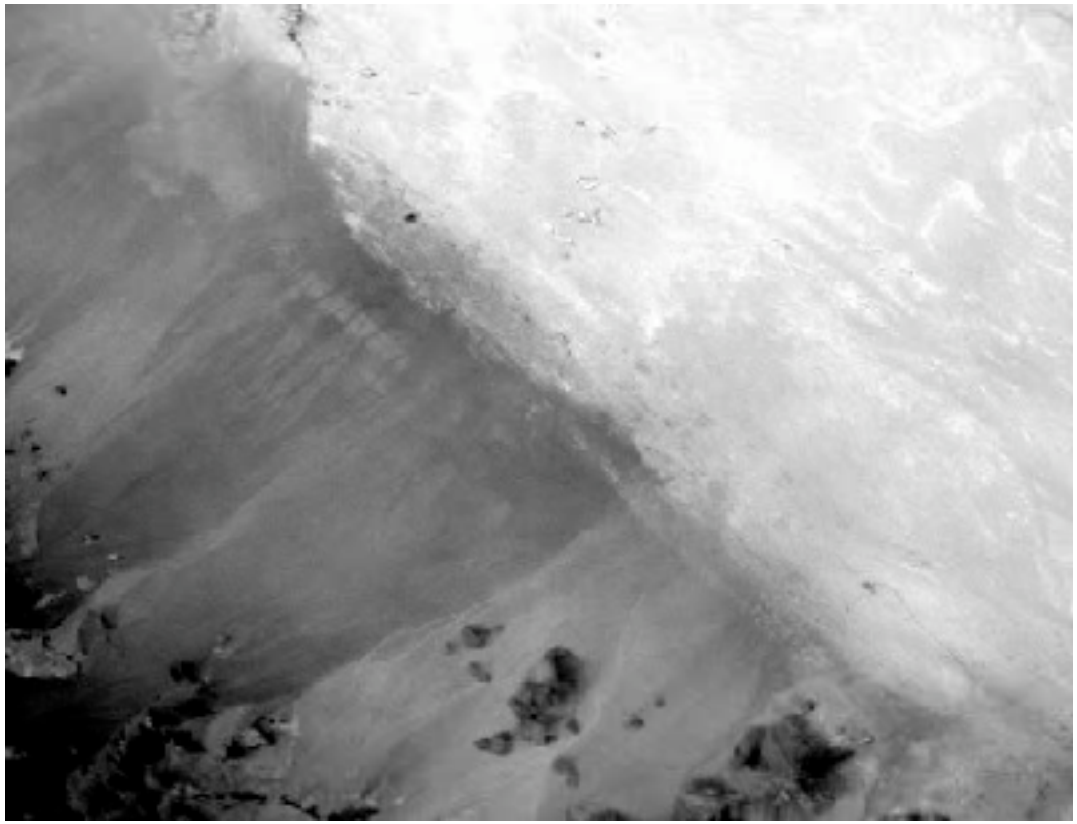
AVIRIS (Lunar Lake)

Image quality does not necessarily decrease with increasing component number for principal component transformations

Feature Reduction

Minimum Noise Fraction (MNF)

MNF Component Images



(click image to play movie)

AVIRIS (Lunar Lake)

Image quality shows a steady decrease with increasing component number for MNF transformations

AVIRIS (Lunar Lake)

Feature Reduction

Minimum Noise Fraction (MNF)

Consider an N-dimensional multivariate data set

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

where each observation is made up of an uncorrelated signal and noise component as

$$\mathbf{x} = \begin{bmatrix} S(x_1) + N(x_1) \\ S(x_2) + N(x_2) \\ \vdots \\ S(x_N) + N(x_N) \end{bmatrix}$$

The covariance of such a data set is given by

$$\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{S}} + \Sigma_{\mathbf{N}}$$

the sum of the individual signal and noise covariance matrices

Feature Reduction

Minimum Noise Fraction (MNF)

The noise fraction for the i^{th} band is defined as

$$NF(x_i) = \frac{\sigma^2_{N(x_i)}}{\sigma^2_{x_i}}$$

the ratio of the noise variance to the total variance for that band.

The maximum noise fraction transformation chooses a linear transformation

$$\mathbf{y} = \mathbf{G}\mathbf{x}$$

such that the noise fraction for y_i is maximum among all linear transformations orthogonal to y_j for all j greater than i.

The individual transforms that make up the matrix $\mathbf{G}$ are defined as the eigenvectors of $\Sigma_{\mathbf{N}}\Sigma_{\mathbf{x}}^{-1}$ and the corresponding eigenvalues are the individual noise fractions $NF(x_i)$. The noise fractions (and the corresponding noise fraction transforms) are ordered such that

$$NF(x_1) \geq NF(x_2) \geq \ldots \geq NF(x_N)$$

Feature Reduction

Minimum Noise Fraction (MNF)

Since the signal and noise are assumed uncorrelated, the orthogonalization of $\mathbf{x}$ also orthogonalizes the signal, $S(\mathbf{x})$, and noise, $N(\mathbf{x})$, terms.

Noise removal occurs by spatial filtering the noisiest components of y and then back transforming the data set to the original domain, namely

$$\mathbf{x}' = \mathbf{G}^{-1}\mathbf{y}_{filtered}$$

This data set, $\mathbf{x}'$, with its noise removed, can then be taken through the a principal component transformation which will maximize the variance along each component axis, where this variance is now exclusively in the signal of concern (the noise component has been removed).

Feature Reduction

Minimum Noise Fraction (MNF)

SPECIAL CASE

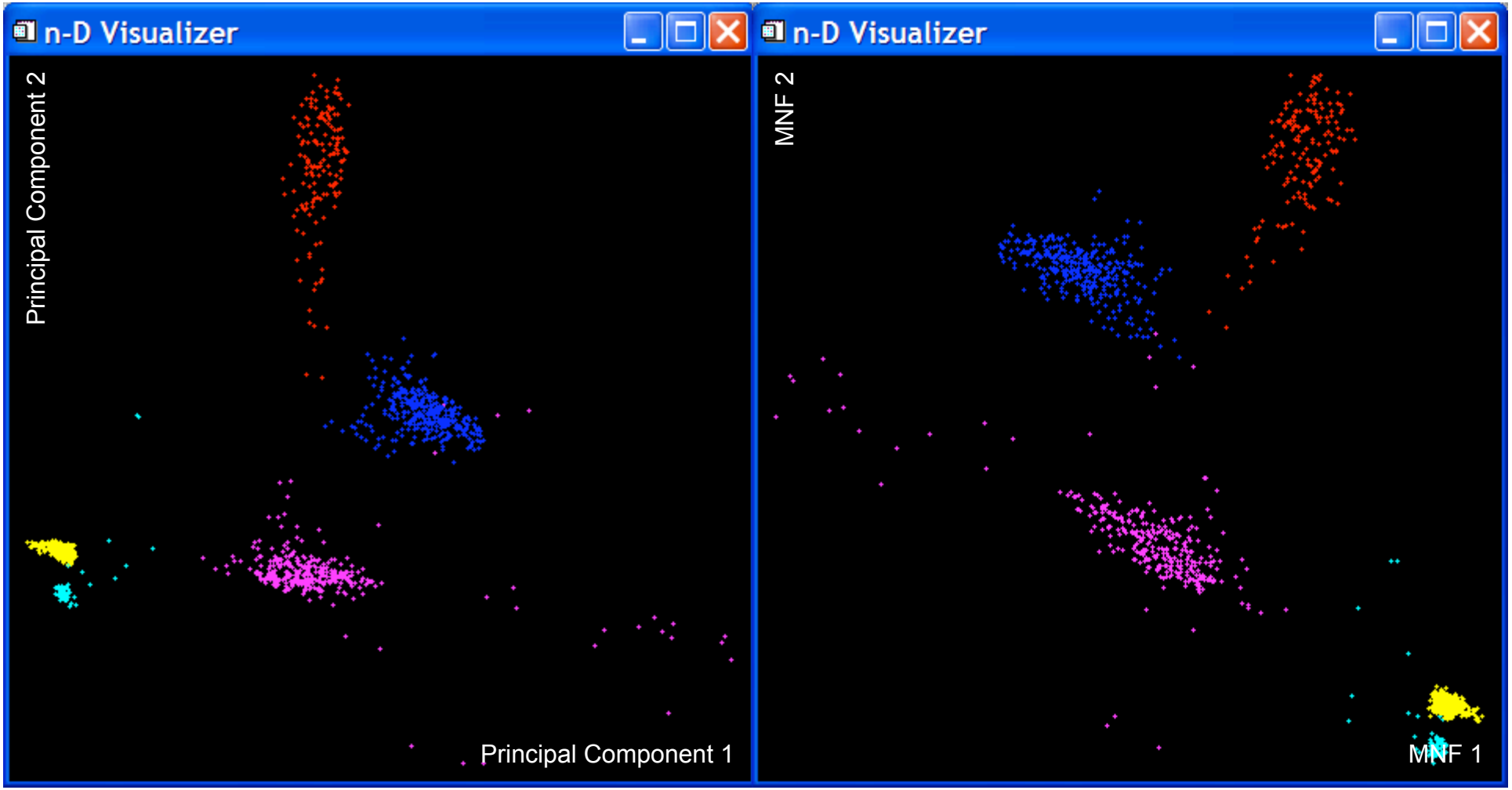
If the noise in every band of the data set is the same, the noise component of the variance is spherically distributed. The removal of this component will have no effect on the subsequent principal component transformation, the direction of the computed eigenvectors will be the same.

Therefore, when this is the case, the principal component and minimum noise fraction transforms are identical.

This is why the principal component transform works so well on low-dimensional data sets like Landsat TM where the signal-to-noise ratio is the same in all bands but not on higher dimensional data sets like AVIRIS where the signal-to-noise ratio is variable across the bands.

Feature Reduction

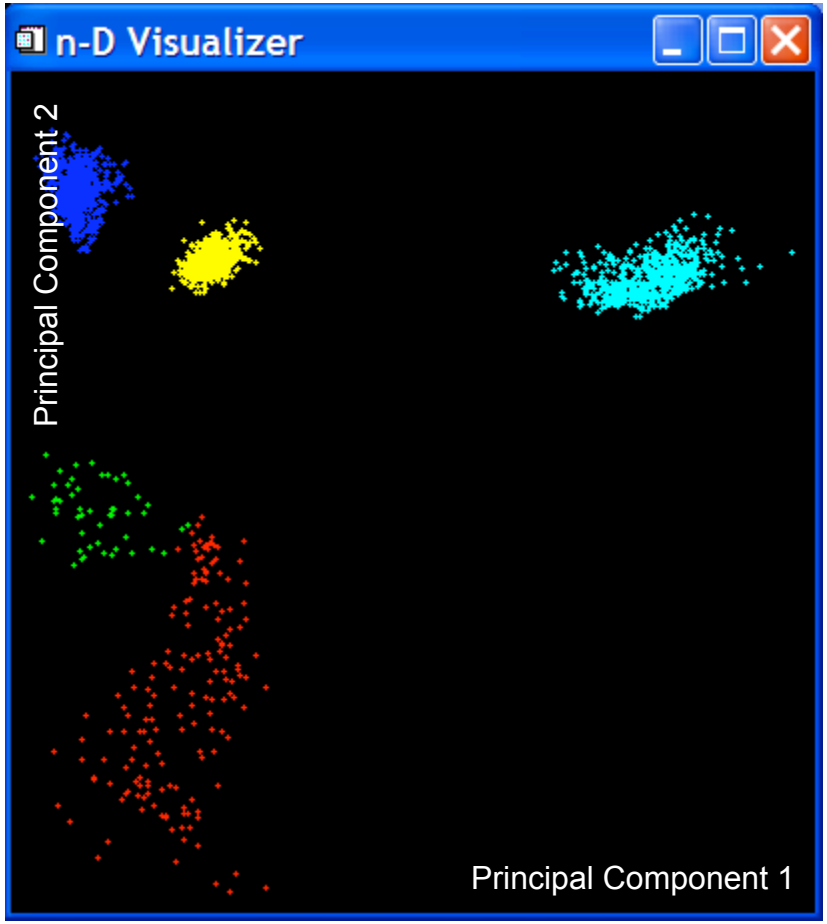
Minimum Noise Fraction (MNF)

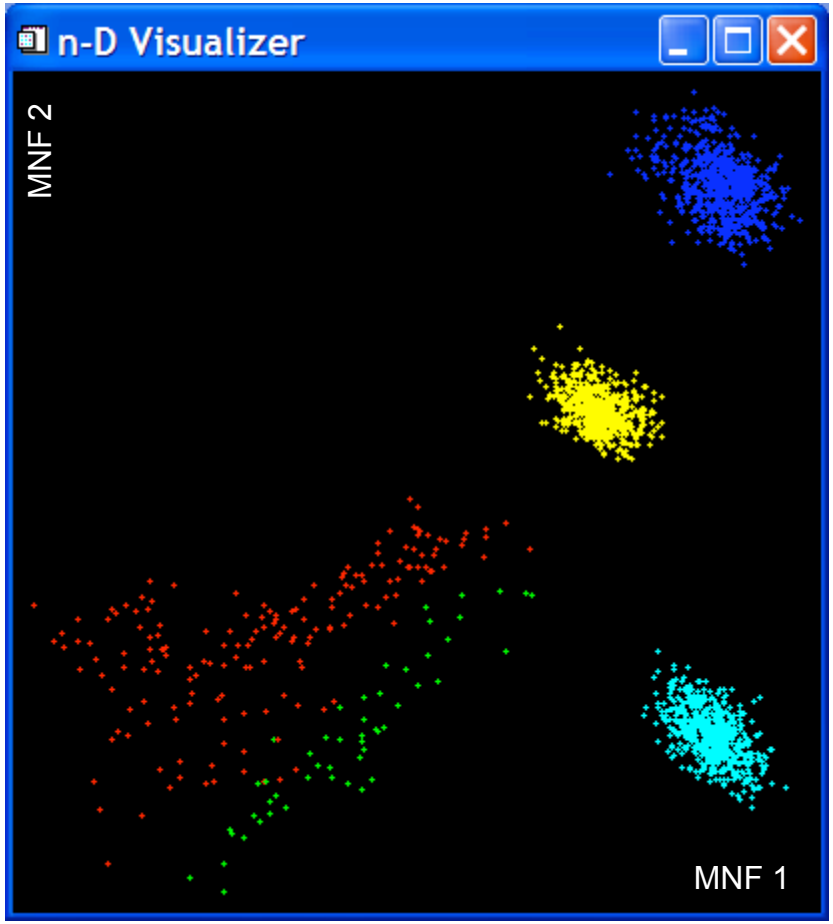


Landsat TM (Rochester)

Feature Reduction

Minimum Noise Fraction (MNF)





AVIRIS (Lunar Lake)

Feature Reduction

Minimum Noise Fraction (MNF)

Computationally, you need to determine the noise and overall covariance matrices for the original data set. The overall covariance matrix is simply the unbiased estimate of the sample covariance matrix for this data set, just as we have computed it in the past

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \xi\left\{\left(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}\right)\left(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}\right)^{t}\right\}$$

The noise covariance is not as easily arrived at.

Switzer and Green, 1984, have presented a method known as minimum/maximum autocorrelation factors (MAF) to arrive at an estimate of this noise covariance matrix. Assumptions of this technique are

1) the signal at any point in the image is highly correlated to the signal at neighboring pixels (*i.e.* high spatial correlation)
2) the noise at any point in the image shows only weak spatial correlation

Both assumptions are valid with most imaging systems.

Feature Reduction

Minimum Noise Fraction (MNF)

The noise covariance matrix is estimated as

$$\boldsymbol{\Sigma}_{\mathbf{N}} \cong \boldsymbol{\Sigma}_{\Delta} = \boldsymbol{\Sigma}_{\frac{1}{2}(\mathbf{x}-\mathbf{x}_{\Delta})}$$

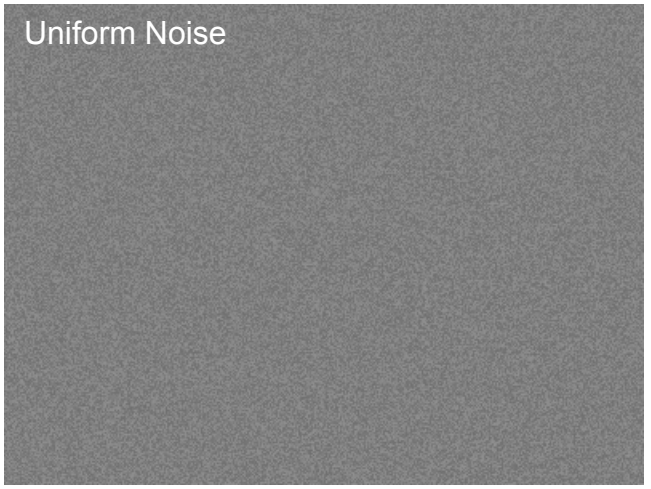
where $\mathbf{x}$-$\mathbf{x}_{\Delta}$ is an image formed by subtracting a slightly offset version of an image from itself. If the noise component of the signal shows little spatial correlation while the signal component exhibits high spatial correlation within a band, then this difference will in effect remove the underlying signal while leaving the statistics of the noise unaffected.

Feature Reduction

Minimum Noise Fraction (MNF)

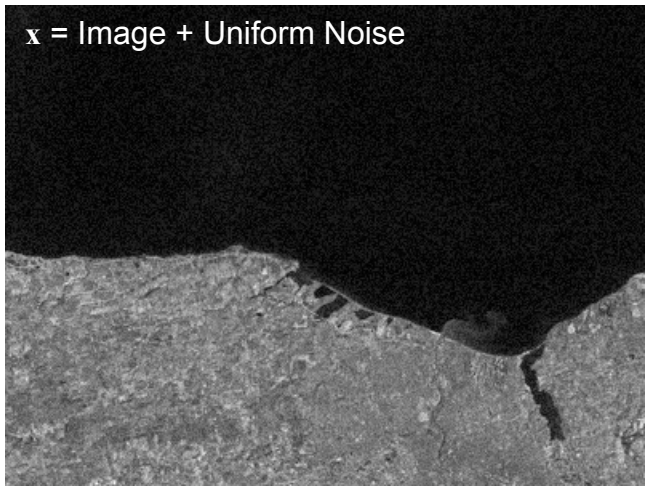
$\mu = 48.3$

$\sigma^2 = 2276.9$



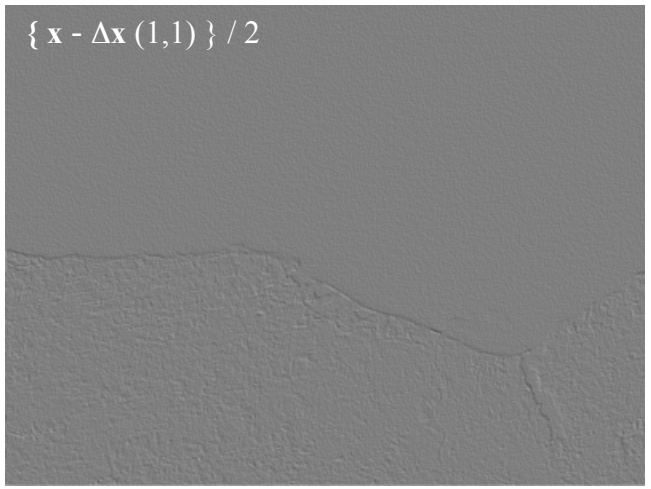
$\mu = 0.02$

$\sigma^2 = 85.4$



$\mu = 48.3$

$\sigma^2 = 2361.1$



$\mu = 0.00$

$\sigma^2 = 89.3$

Δx (1,1) signifies a shift of the original image one pixel in both the horizontal and vertical directions

Feature Reduction

Minimum Noise Fraction (MNF)

IDL Script for previous example

```
READ_JPEG, "c:\tmp\19990328.jpg", image
WINDOW, 0, XSIZE=400, YSIZE=300
TV, image

noise = FLTARR(400,300)
FOR row=0,299 DO FOR col=0,399 DO noise[col,row] = RANDOMU(seed)*32-16
WINDOW, 1, XSIZE=400, YSIZE=300
TV, noise+128

imageN = image + noise
WINDOW, 2, XSIZE=400, YSIZE=300
TVSCL, imageN

deltaImage = ( imageN - SHIFT( imageN, 1, 1 ) ) / 2
WINDOW, 3, XSIZE=400, YSIZE=300
TV, deltaImage+128

PRINT, MEAN(image), VARIANCE(image)
PRINT, MEAN(noise), VARIANCE(noise)
PRINT, MEAN(imageN), VARIANCE(imageN)
PRINT, MEAN(deltaImage), VARIANCE(deltaImage)
```

Image-to-Image Registration

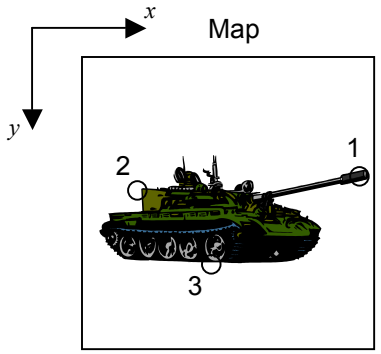
Mapping Polynomials

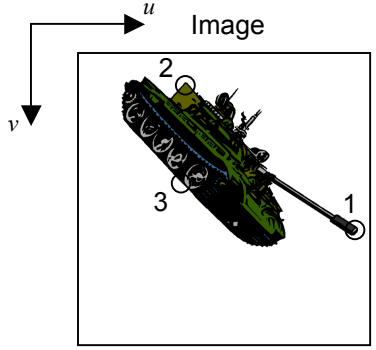
Let's assume we have a map (or reference image) and an image we would like to have overlay this map. We describe the points in the map as (x,y) and the points in the image as (u,v), two unique coordinate systems.

Since the explicit mathematical form of a mapping function between the two systems is rarely known explicitly, we rely on simple mapping polynomials to define the relationship between the two coordinate systems. For example,

$$u = f(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

$$v = g(x,y) = b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2$$





$$(x_1, y_1) \rightarrow (u_1, v_1)$$

$$(x_2, y_2) \rightarrow (u_2, v_2)$$

$$(x_3, y_3) \rightarrow (u_3, v_3)$$

Image-to-Image Registration

Mapping Polynomials

$$u = f(x,y) = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2$$

$$v = g(x,y) = b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2$$

$$\begin{aligned}
(x_1,y_1) &\rightarrow (u_1,v_1)\\
(x_2,y_2) &\rightarrow (u_2,v_2)\\
(x_3,y_3) &\rightarrow (u_3,v_3)\\
(x_4,y_4) &\rightarrow (u_4,v_4)\\
(x_5,y_5) &\rightarrow (u_5,v_5)\\
(x_6,y_6) &\rightarrow (u_6,v_6)\\
(x_7,y_7) &\rightarrow (u_7,v_7)
\end{aligned}$$

$$\mathbf{X} = \begin{bmatrix}
1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2\\
1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2\\
1 & x_3 & y_3 & x_3y_3 & x_3^2 & y_3^2\\
1 & x_4 & y_4 & x_4y_4 & x_4^2 & y_4^2\\
1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2\\
1 & x_6 & y_6 & x_6y_6 & x_6^2 & y_6^2\\
1 & x_7 & y_7 & x_7y_7 & x_7^2 & y_7^2
\end{bmatrix} \qquad \mathbf{Y}_u = \begin{bmatrix} u_1\\ u_2\\ u_3\\ u_4\\ u_5\\ u_6\\ u_7 \end{bmatrix} \qquad \mathbf{Y}_v = \begin{bmatrix} v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6\\ v_7 \end{bmatrix}$$

Least Squares Regression $\rightarrow$

$$\mathbf{a} = \left(\mathbf{X}^t\mathbf{X}\right)^{-1}\mathbf{X}^t\mathbf{Y}_u$$

$$\mathbf{b} = \left(\mathbf{X}^t\mathbf{X}\right)^{-1}\mathbf{X}^t\mathbf{Y}_v$$

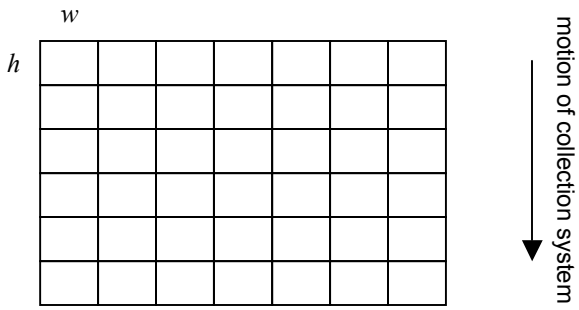
Image-to-Image Registration

Direct Transformations

Aspect Ratio Correction

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$a = \frac{h}{w}$$



Tangent Correction

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \theta\cot\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

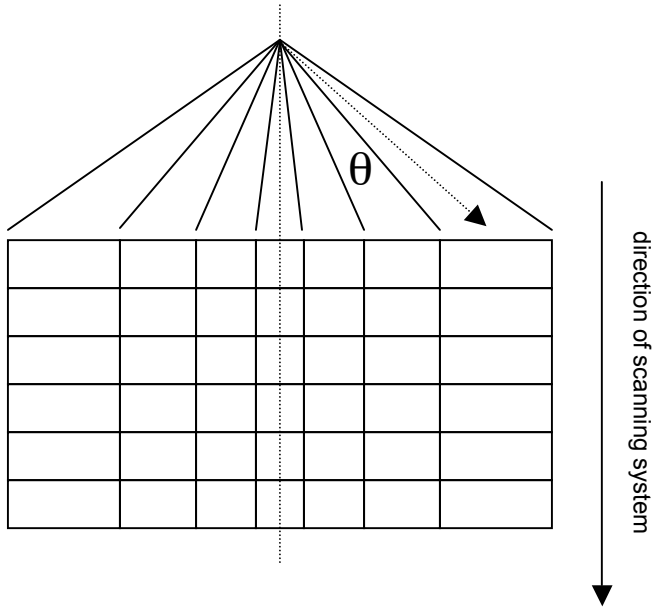


Image-to-Image Registration

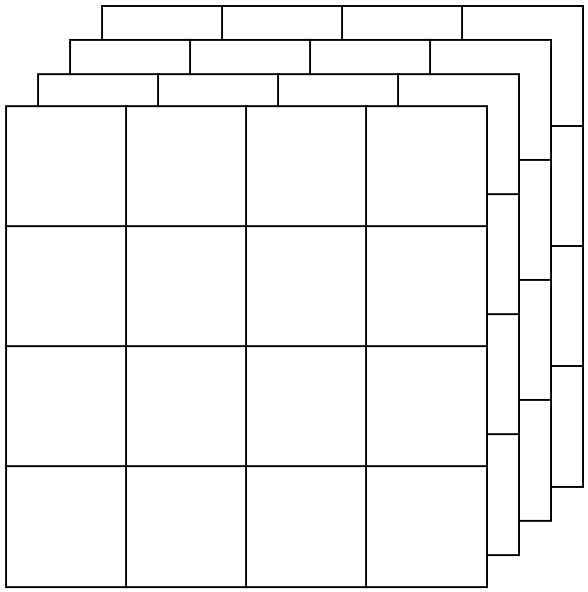
Direct Transformations

Rotation

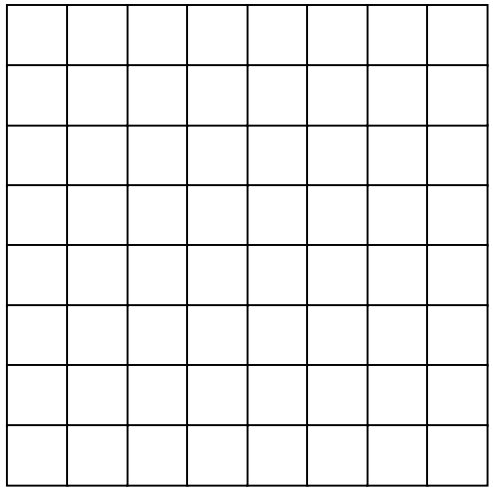
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

α is the CCW rotation angle

Image Fusion



Low Resolution Multiband Images



High Resolution Broadband Image

GOAL: To create an image with the spectral resolution of the low resolution multiband image and the spatial resolution of the high resolution broadband image

Image Fusion

Methodology Classes

- Merger by separate manipulation of spatial and spectral data components
  - IHS
  - PCA
  - High-pass filter
- Merger to maintain radiometric integrity

Image Fusion

Separate Manipulation of Spatial/Spectral Data Content

Any spectral transformation (IHS, PCA) that produces a transformed component that represents the average brightness/intensity of the individual spectral components, can be used. The basic premise is to replace this average brightness component in the transformed space before back transforming these components.

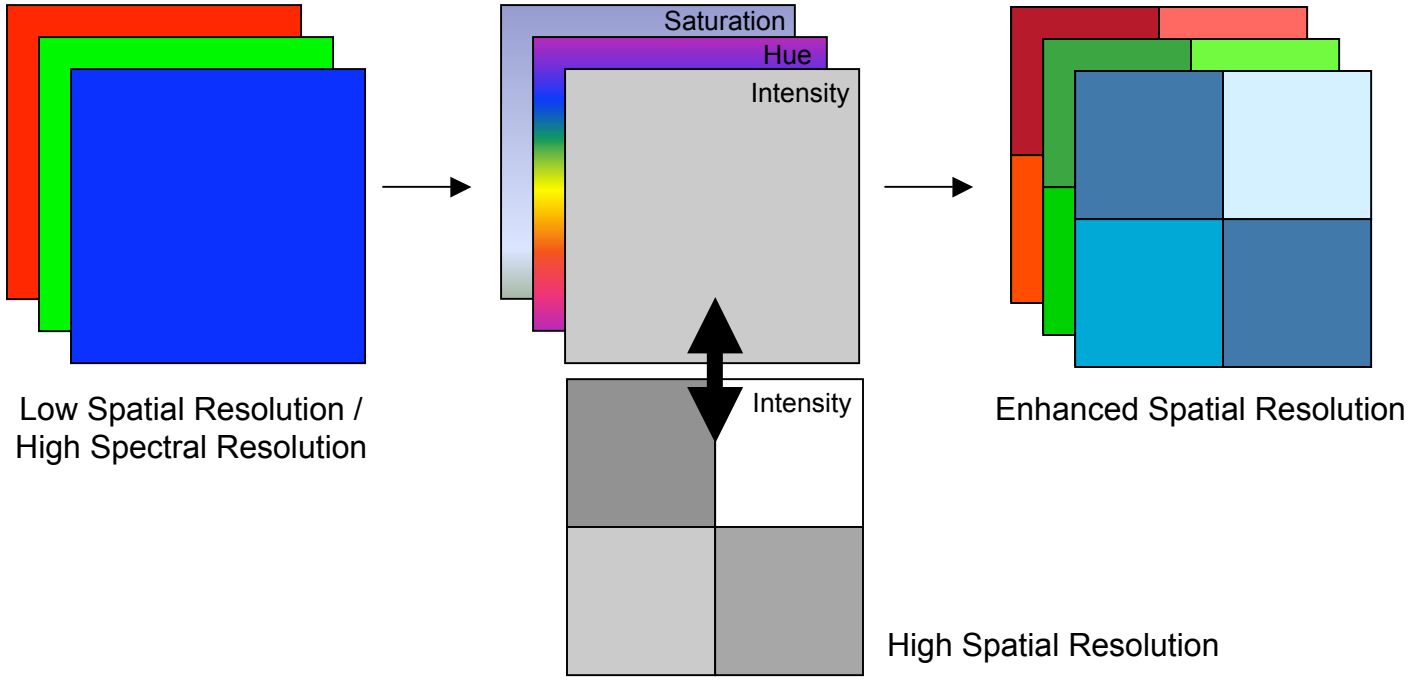
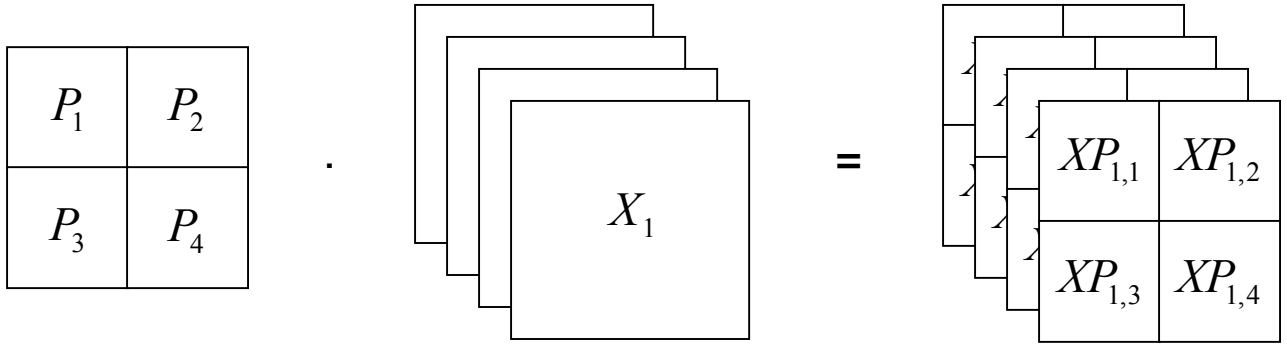


Image Fusion

Maintaining Radiometric Integrity

$$XP_{i,j} = X_i \cdot \frac{P_j}{P_1 + P_2 + P_3 + P_4} \qquad \text{for } i = 1,\ 2,\ \ldots,\ N \text{ and } j = 1,\ 2,\ 3,\ 4$$

Pradines, 1986



The individual bands of the multiband image MUST exhibit high correlation with the high resolution panchromatic band (typically the spectral response functions must overlap)

Image Fusion

Maintaining Radiometric Integrity

$$XS_i = a_i P_{average} + b_i \qquad \text{for } i = 1,\ 2,\ \ldots,\ N$$

$$XS_i' = a_i P_{high\ resolution} + b_i$$

Price, 1987

$$Hybrid_i = \frac{XS_i \cdot XS_i'}{XS_{average,i}'}$$

XS_i	is the digital count of a superpixel in the i^{th} spectral band,
$P_{average}$	is the average of the high resolution pan pixels in a superpixel,
a_i, b_i	least-squares regression coefficients,
XS_i'	high resolution estimate of i^{th} spectral band,
$XS_{average,i}'$	average digital count in the XS_i' image in the superpixel,
$Hybrid_i$	the digital count in the i^{th} band of the high resolution hybrid multispectral band