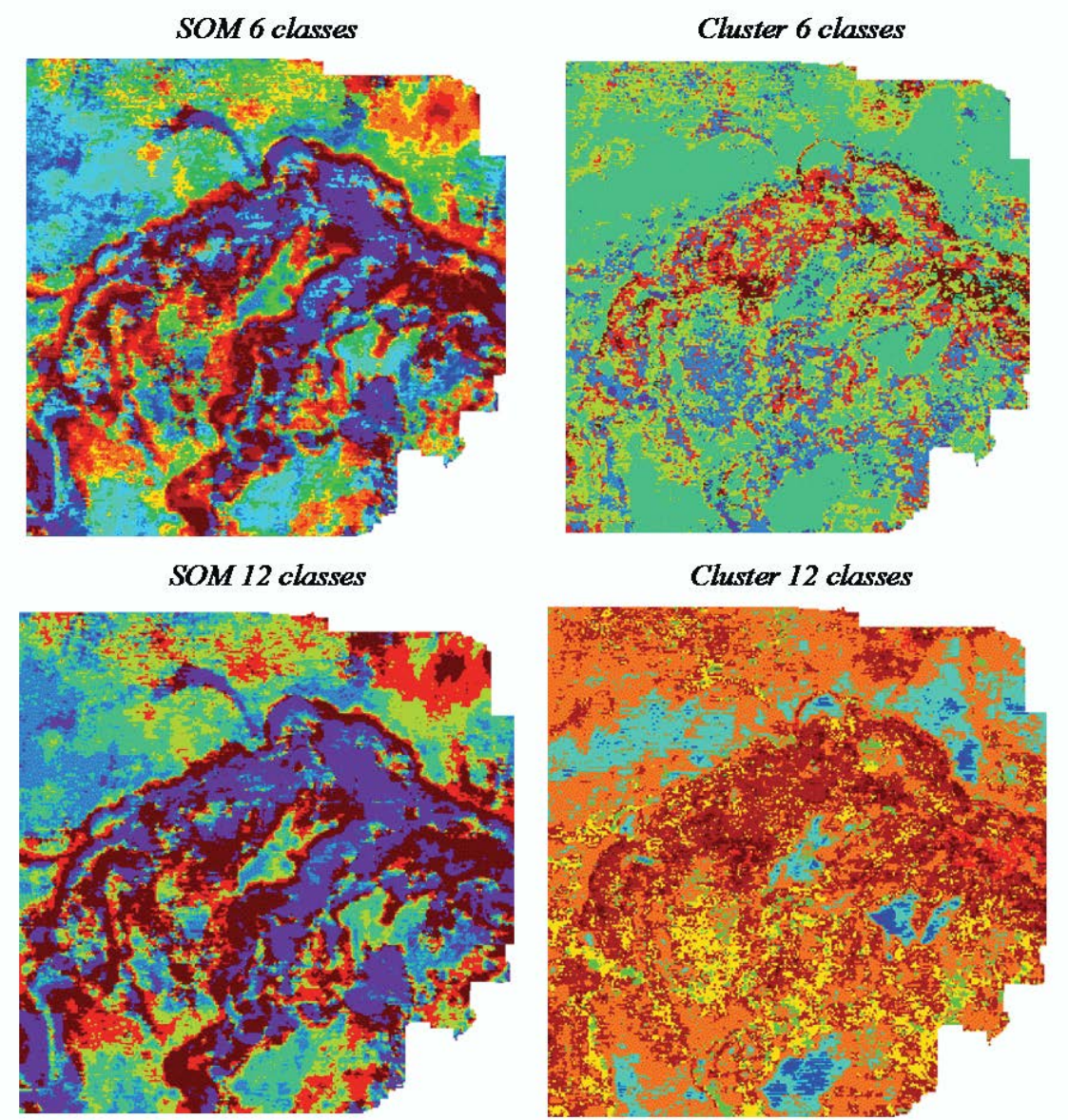
Unsupervised seismic facies classification using *K*-means and Gaussian Mixture Modeling

Brian Russell

NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience

Introduction

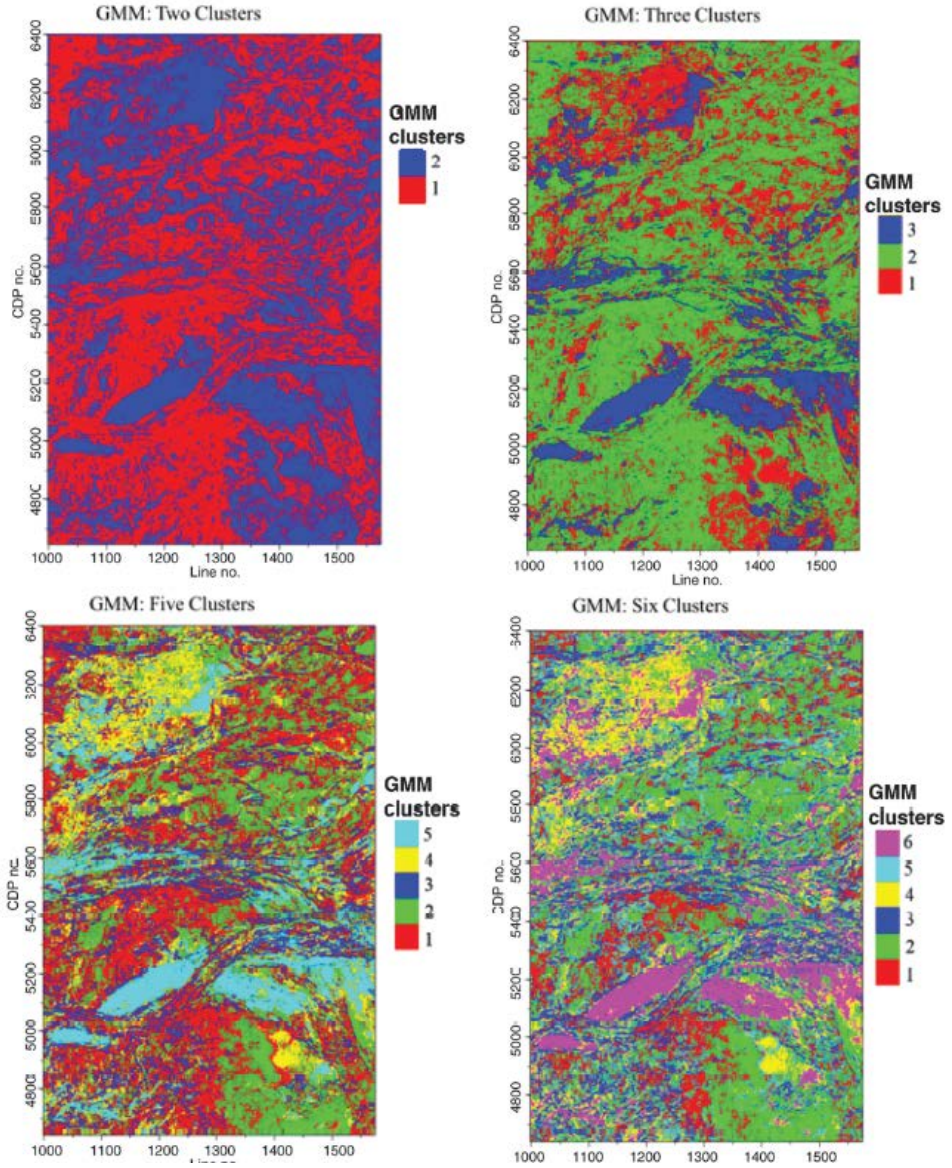
- Coléou et al. (2003), in a TLE article, were the first to apply unsupervised clustering techniques to seismic facies classification.
- The left two panels show the Self Organizing Map (SOM) technique with 6 classes (top) and 12 classes (bottom).
- The right two panels show the *K*-means technique with 6 classes (top) and 12 classes (bottom).
- I will not discuss the SOM technique today but will focus on the *K*-means technique as well as Gaussian Mixture Modelling (GMM), a newer clustering method.



Coléou et al, 2003

Gaussian Mixture Model Example

- Wallet and Hardisty (2019), in an article in Interpretation, applied the Gaussian Mixture Modeling (GMM) technique to seismic facies classification.
- The four panels on the right show the application of GMM to a set of amplitude slices through a seismic volume.
- The upper left shows two clusters, the upper right shows three clusters, the lower left shows five clusters, and the lower right shows six clusters.
- Let's now look at the theory of both K-means and GMM clustering.



Wallet and Hardisty (2019),

The *K*-means algorithm

- If we start with N M-dimensional data points (e.g., M attributes) the *K*-means algorithm will divide these points into *K* clusters.
- The *K*-means algorithm is implemented as follows:
  - Pick the number of clusters, *K*, and divide the input data points randomly into these *K* clusters.
  - Compute the *M*-dimensional means of the clusters.
  - Compute the distances between each point and each cluster and assign each point to the cluster for which this distance is a minimum.
  - Re-compute the means based on the new cluster assignments.
  - Iterate through the above three steps until convergence.
- The key assumption is that we know how many clusters are present in the data, so in a typical application you may want to try different values of *K*.

The *K*-means algorithm in two dimensions

- The general *K*-means algorithm can be applied in any number of dimensions *M*.
- But the algorithm is much easier to visualize in *M* = 2 dimensions, where the points, means and distances are written as follows:

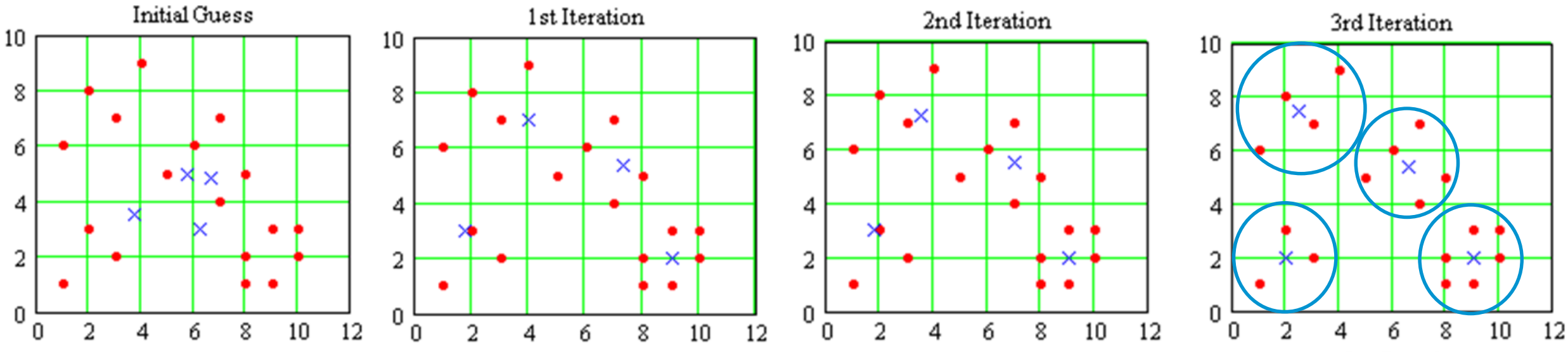
$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}, \mu_k = \begin{bmatrix} \mu_{k1} \\ \mu_{k2} \end{bmatrix} \Rightarrow d_{ik} = \sqrt{(x_{i1} - \mu_{k1})^2 + (x_{i2} - \mu_{k2})^2}, \text{ where}$$

$$i = 1, \ldots, N = \text{ no. of input points, } k = 1, \ldots, K = \text{ no. of clusters.}$$

- This is the standard *K*-means algorithm, in which distance is interpreted as the Euclidean distance (we will look at statistical distance later).
- I will first illustrate the *K*-means method with several “toy” datasets and then use a set of clusters derived from the seismic inversion of a Gulf Coast dataset.

A simple example of *K*-means

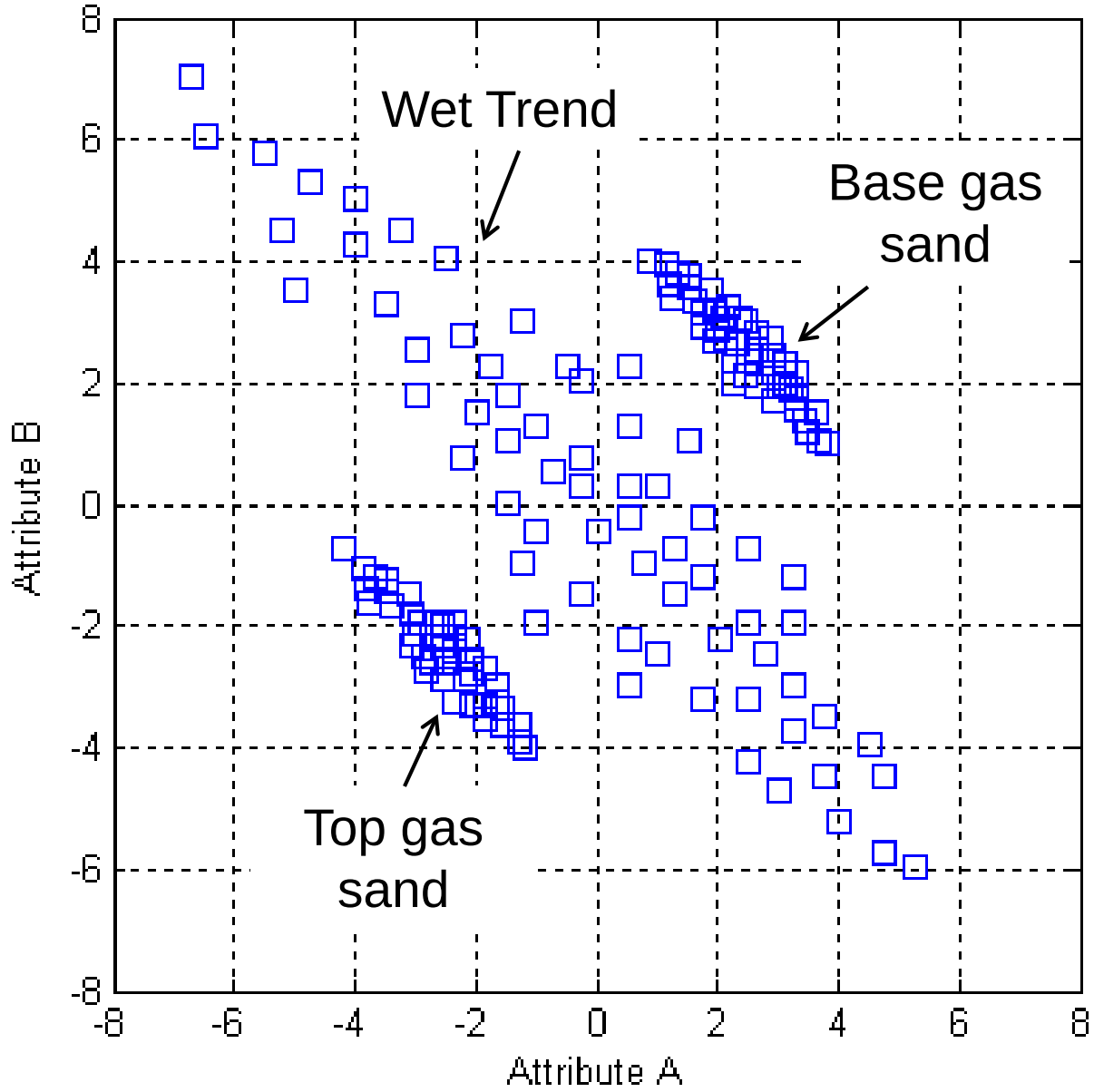
- This example has four clusters, where the cross-plot on the left shows the points as red dots (input in random order) and the initial means (incorrect) as blue crosses.



- The next three cross-plots shows the updates after the 1st, 2nd and 3rd iterations.
- After the 3rd iteration, the means have “locked in” to the cluster centres, and we can clearly identify the four clusters by drawing circles around the means.

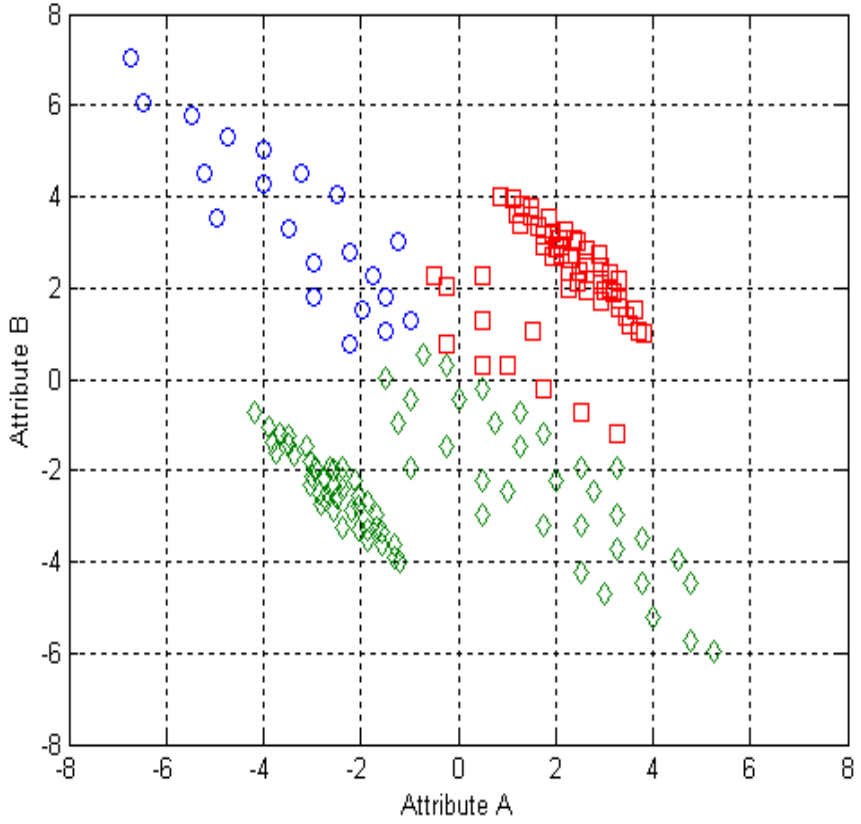
A second example

- Next, consider this second example, where the clusters have elliptical shapes.
- Although this is a synthetic example, it is based on an AVO Class 3 intercept (A) versus gradient (B) cross-plot.
- The three clusters represent a wet trend, the base of gas sand and top of gas sand.

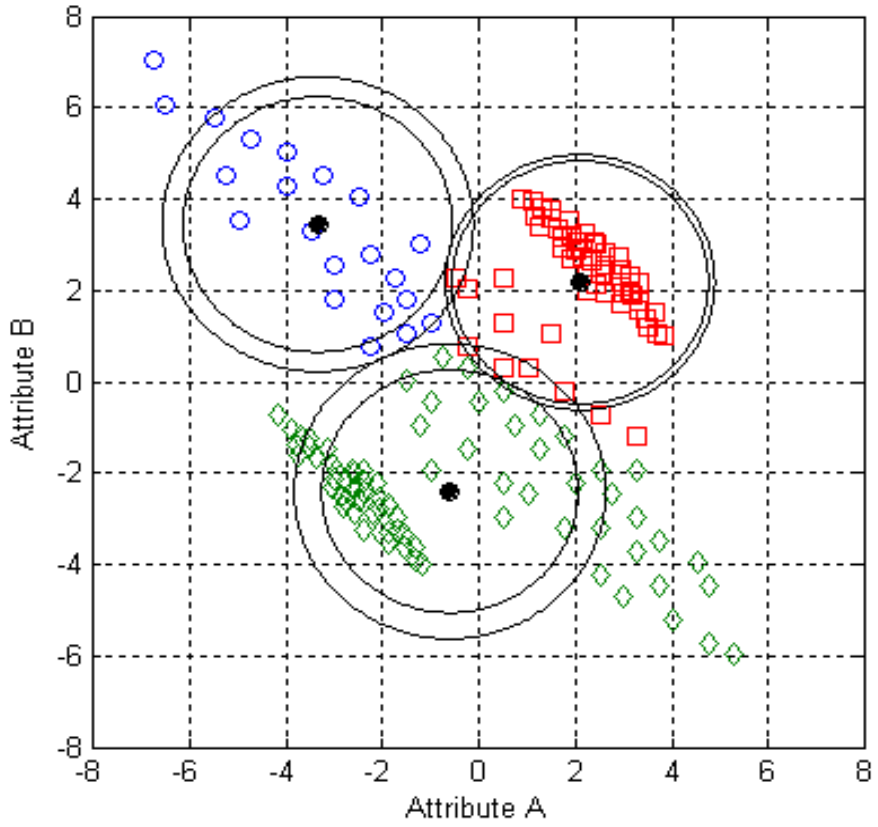


The *K*-means result

- Applying *K*-means with *K* = 3 leads to the following incorrect result.



- This is because the basic *K*-means algorithm finds circles, not ellipses.



The 2-D Gaussian (or Normal) Distribution

- The elliptical shapes of those clusters can be modelled by the 2D Gaussian distribution:

$$f(\boldsymbol{x}) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right], \text{ where}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}, \text{ in which } \sigma_1^2,\ \sigma_2^2 = \text{ the variances, } \sigma_{12} = \text{ the covariance.}$$

- A simpler way to write the Gaussian distribution is as follows:

$$f(\boldsymbol{x}) = S \exp\left[-\frac{\Delta^2}{2}\right], \text{ where } S = \frac{1}{2\pi|\Sigma|^{1/2}} \text{ is a scaling term,}$$

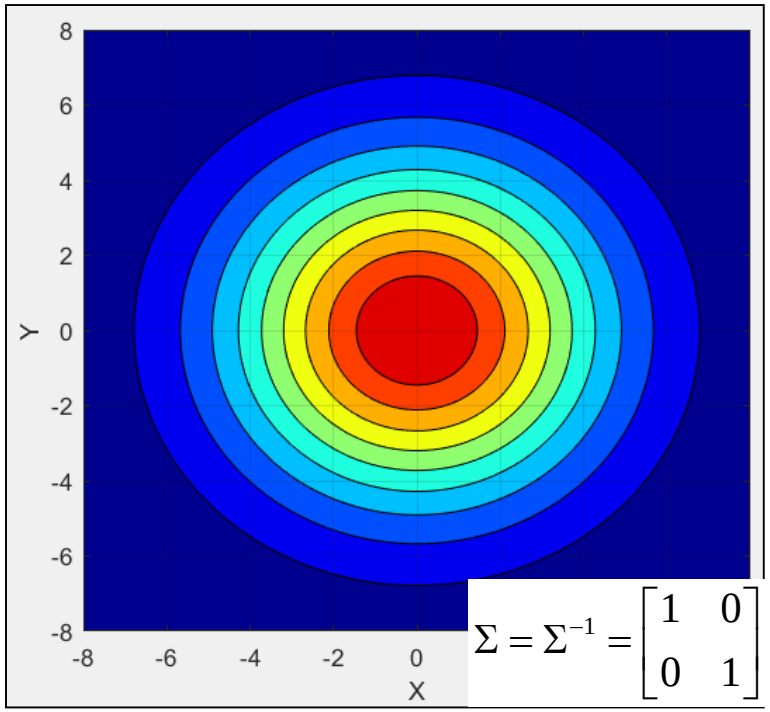
$$\text{and } \Delta = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \text{ is the Mahalanobis, or statistical, distance.}$$

Three special cases

- Using zero means we can write the Mahalanobis distance as follows:

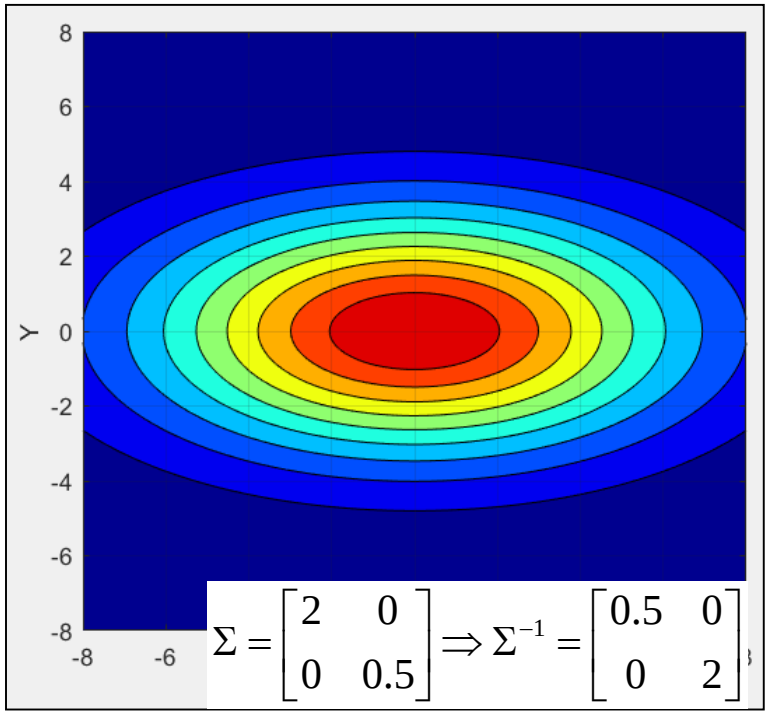
$$\Delta^2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a & -b \\ -b & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = ax_1^2 - 2bx_1x_2 + cx_2^2$$

If $b = 0$ and $a = c$, the Mahalanobis distance is the Euclidean distance:



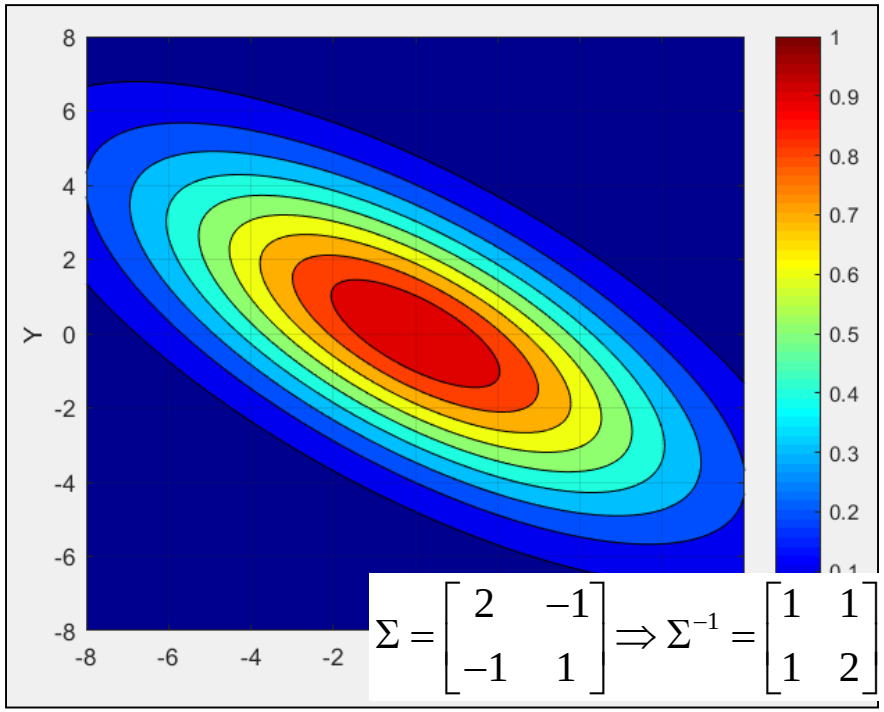
$$\Sigma = \Sigma^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

If $b = 0$ and $a \neq c$, we get vertical ($a < c$) or horizontal ($a > c$) elliptical curves:



$$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix} \Rightarrow \Sigma^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}$$

If $b \neq 0$, we get tilted ellipses (negative slope if $b < 0$ and vice versa).



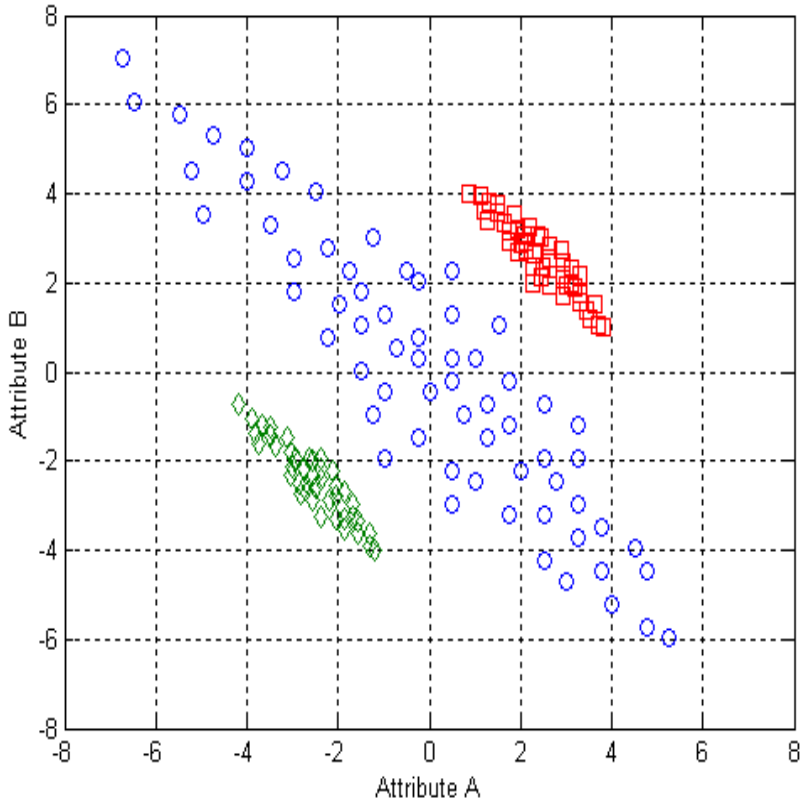
$$\Sigma = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \Rightarrow \Sigma^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

The Mahalanobis *K*-means algorithm

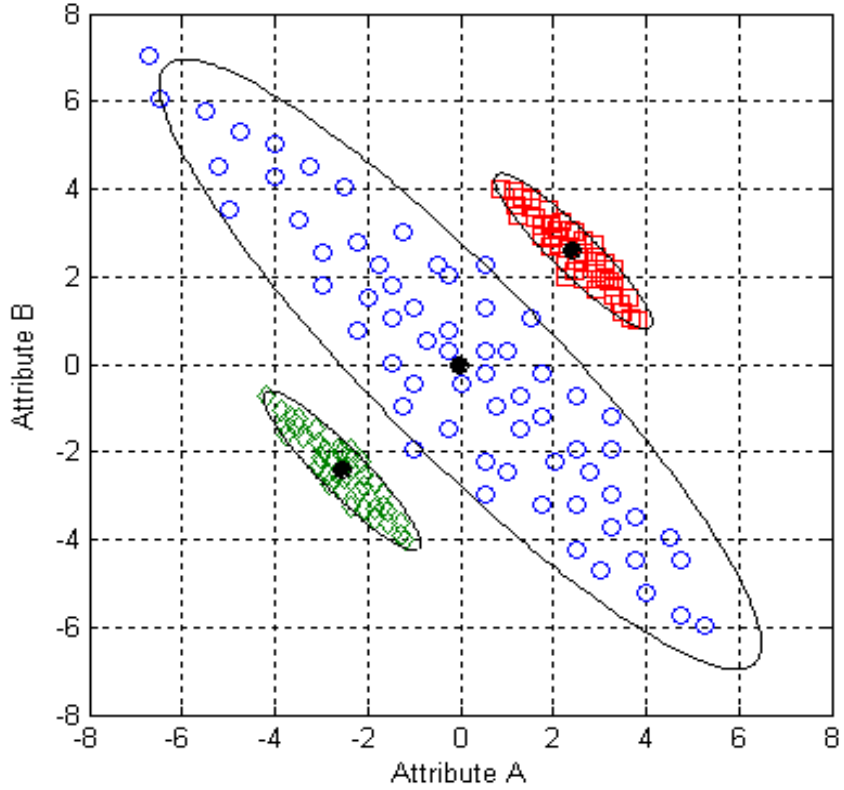
- This suggests a modification of the *K*-means algorithm as follows:
  - Pick the number of clusters, *K*, and divide the input data points randomly into these *K* clusters.
  - Compute the *M*-dimensional means, μ_k, of the clusters, as well as the covariance matrices Σ_k within each cluster, where $k = 1, 2, \ldots, K$.
  - Compute the Mahalanobis distances between each point and cluster and assign each point to the cluster for which this distance is a minimum.
  - Re-compute the means and covariances based on the new cluster assignments.
  - Iterate through the above three steps until convergence.
- Now, let's see how well this algorithm works on our second dataset.

The Mahalanobis *K*-means result

- Applying modified *K*-means leads to the following correct result:



- This plot annotates the cluster means and lines of constant variance:



- I presented this algorithm at the 2003 CREWES meeting (Russell and Lines, 2003), not realizing that the algorithm had already been invented by Sung and Poggio (1994), who called it the elliptical *K*-means method

The Gaussian Mixture Model (GMM)

- A more recent implementation of this method is called the Gaussian Mixture Model (GMM), which is a mixture pdf of N M-dimensional feature vectors $\boldsymbol{x}_n$, which are grouped into K classes C_K.
- Each feature vector has a conditional probability given by:

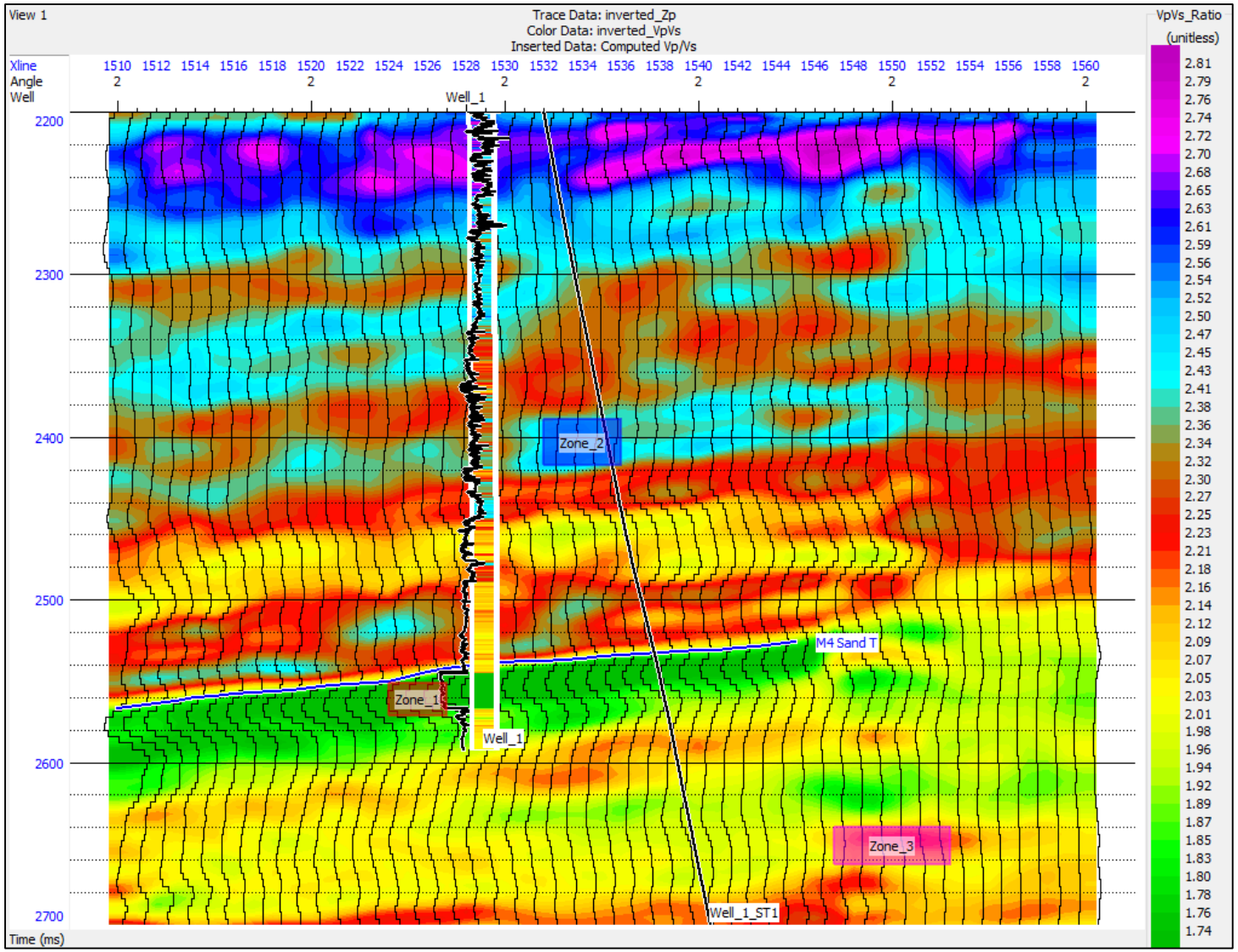
$$p(\boldsymbol{x}_n \mid C_k) = \frac{1}{(2\pi)^{M/2}\left|\Sigma_k\right|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right], \text{ where}$$

$$k = 1,\ldots,K,\ n = 1,\ldots,N,\ \boldsymbol{x}_n = \begin{bmatrix} x_{1n} \\ \vdots \\ x_{Mn} \end{bmatrix},\ \boldsymbol{\mu}_k = \begin{bmatrix} \mu_{1k} \\ \vdots \\ \mu_{Mk} \end{bmatrix} \text{ and } \Sigma_k = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1M} \\ \vdots & \ddots & \vdots \\ \sigma_{M1} & \cdots & \sigma_{MM} \end{bmatrix}.$$

- GMM starts with an initial guess of the means and covariance matrices of each class, and determines the correct values by iterating to a solution.
- Unlike *K*-means, the data is never physically re-ordered during the process.

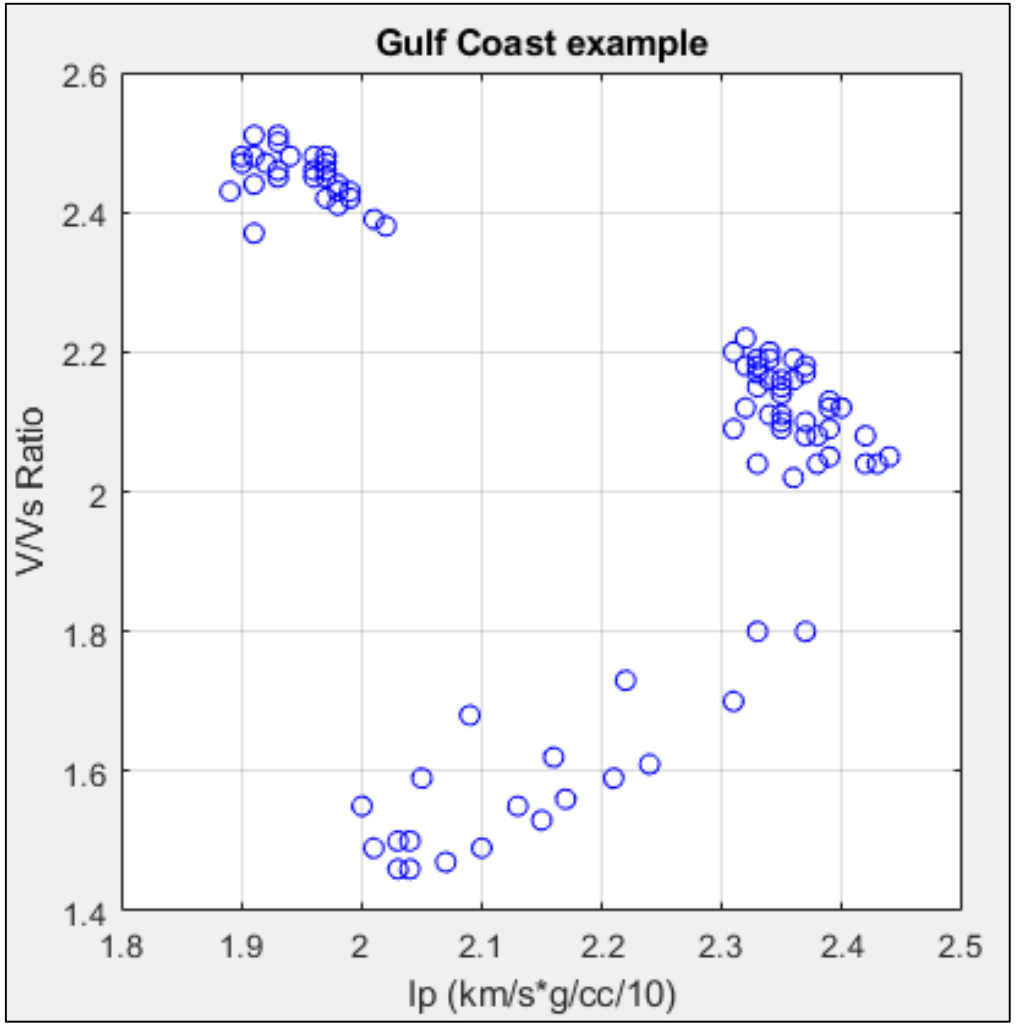
Gulf Coast example

- Here is a plot of one inline from the pre-stack inversion of a Gulf Coast dataset, which is intersected by a well that found gas at a time of 2550 ms.
- The traces represent acoustic impedance (I_P) and the colour represents V_P/V_S ratio (the green is low V_P/V_S at the gas).
- Three zones have been picked on the line, from the gas sand, shallow sands and shales and deeper calcite cemented sands.



The Gaussian Mixture Model (GMM)

- On this cross-plot, the upper set of points represent the shale, the lower points the gas sand, and the rightmost points the carbonate.
- These are the values that were extracted from the pre-stack inversion of a Gulf Coast dataset shown previously.
- Notice that we are cross-plotting inverted Vp/Vs ratio against inverted P-impedance (Ip).
- The points are presented to the algorithm in random order.

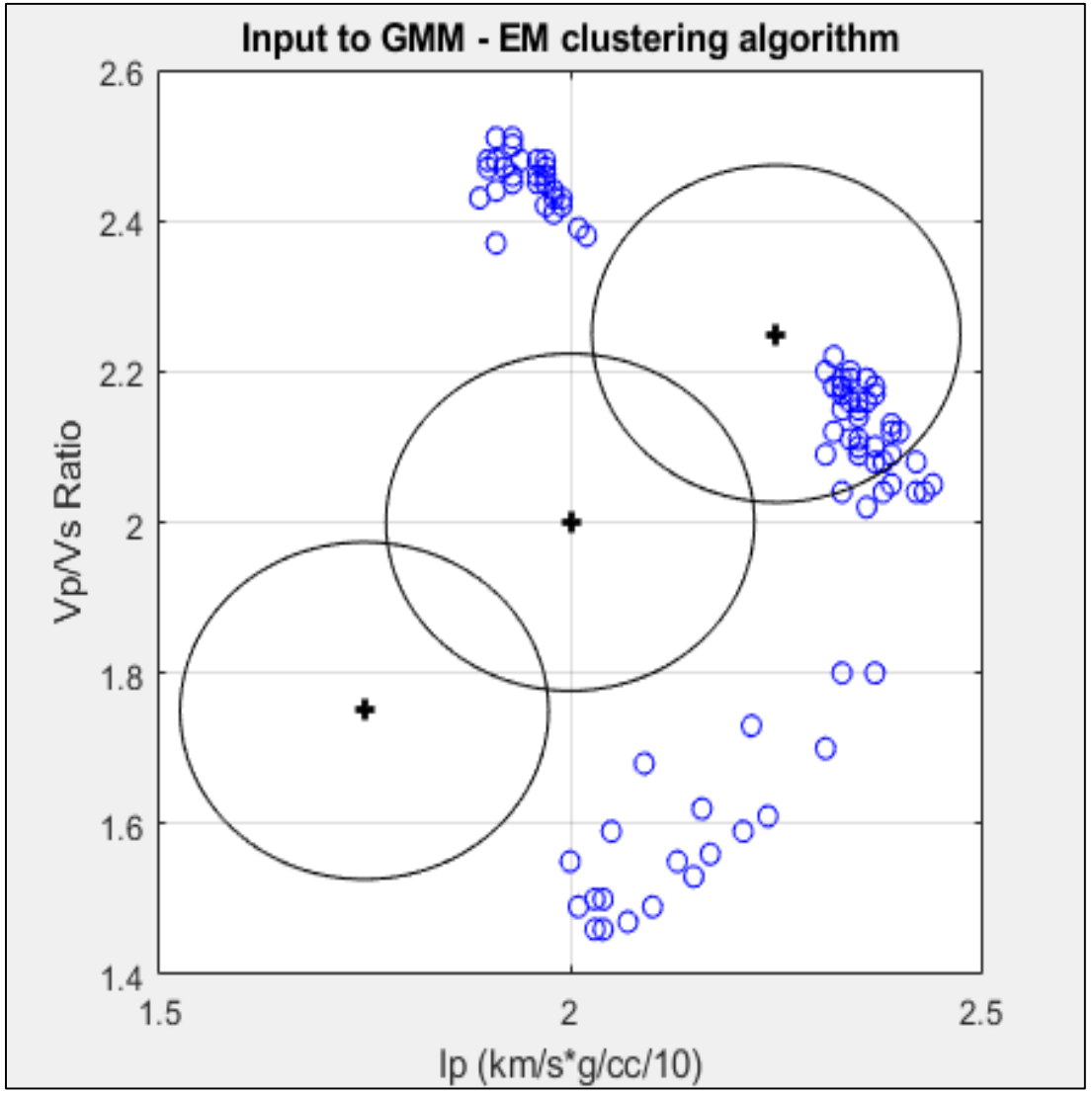


Training the GMM via Expectation-Maximization (*EM*)

- The GMM is then trained using the Expectation-Maximization, or *EM*, algorithm.
- The Gaussian functions can have full, diagonal or spherical covariance matrices.
- I will initialize the GMM with three means and covariances given by:

$$\mu_1 = \begin{bmatrix} 2.25 \\ 2.25 \end{bmatrix}, \mu_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mu_2 = \begin{bmatrix} 1.75 \\ 1.75 \end{bmatrix},$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \frac{1}{4}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$



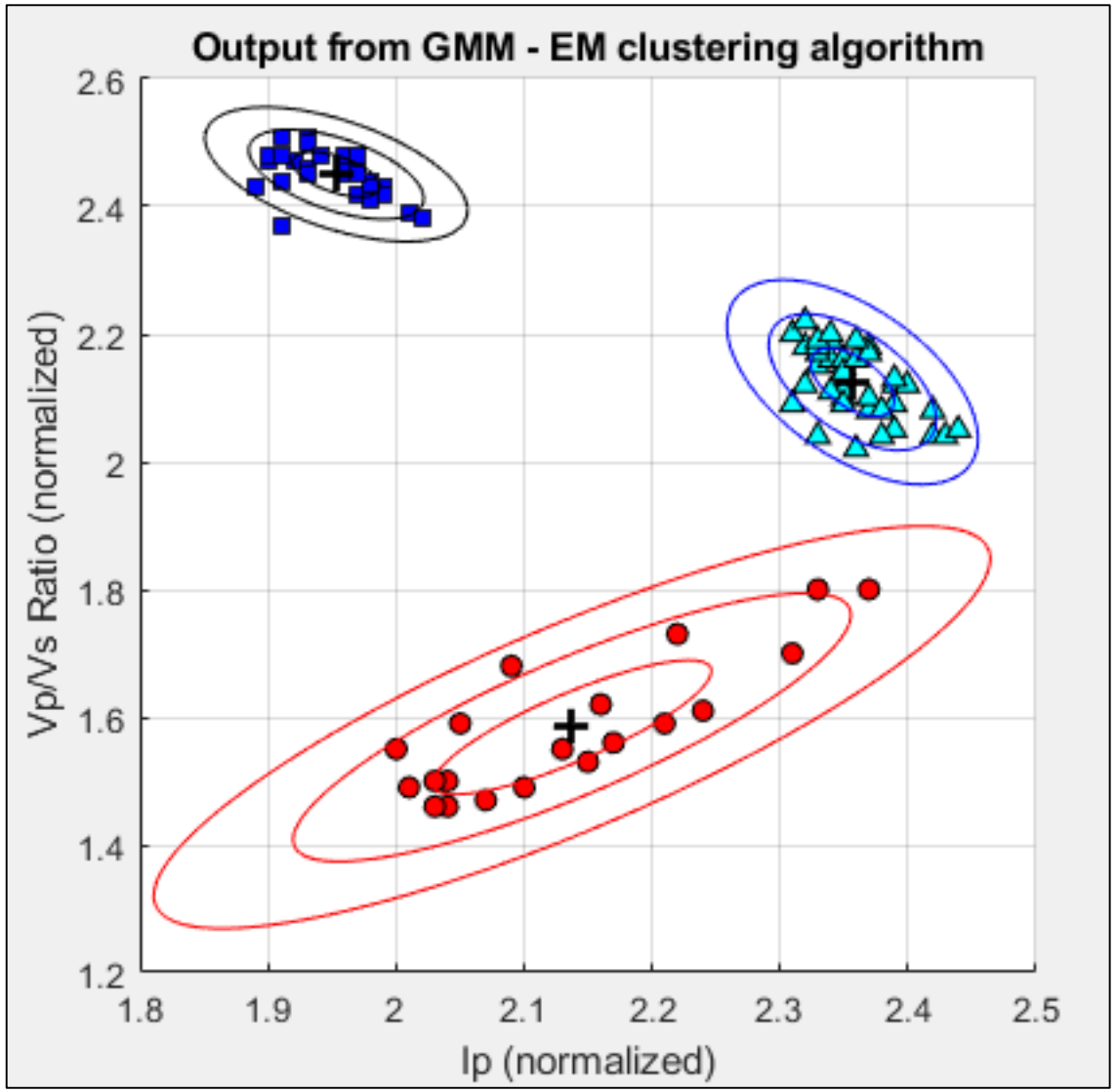
Clustered result

- This figure shows our clustered result, with their Gaussian contours shown.
- The colours can be thought of as “labels”, which will tell us how to classify the points.
- The final statistics are:

$$\mu_1 = \begin{bmatrix} 2.1375 \\ 1.5840 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 0.0119 & 0.0096 \\ 0.0096 & 0.0110 \end{bmatrix}.$$

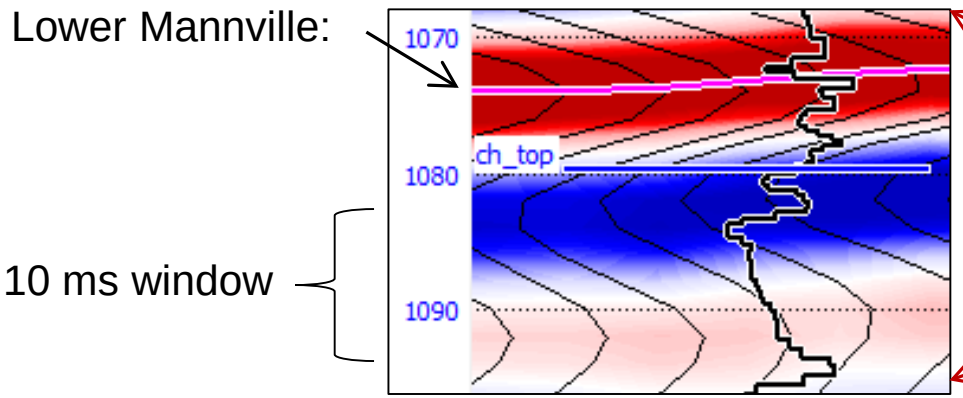
$$\mu_2 = \begin{bmatrix} 1.9529 \\ 2.4503 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.0012 & -0.0006 \\ -0.0006 & 0.0012 \end{bmatrix}.$$

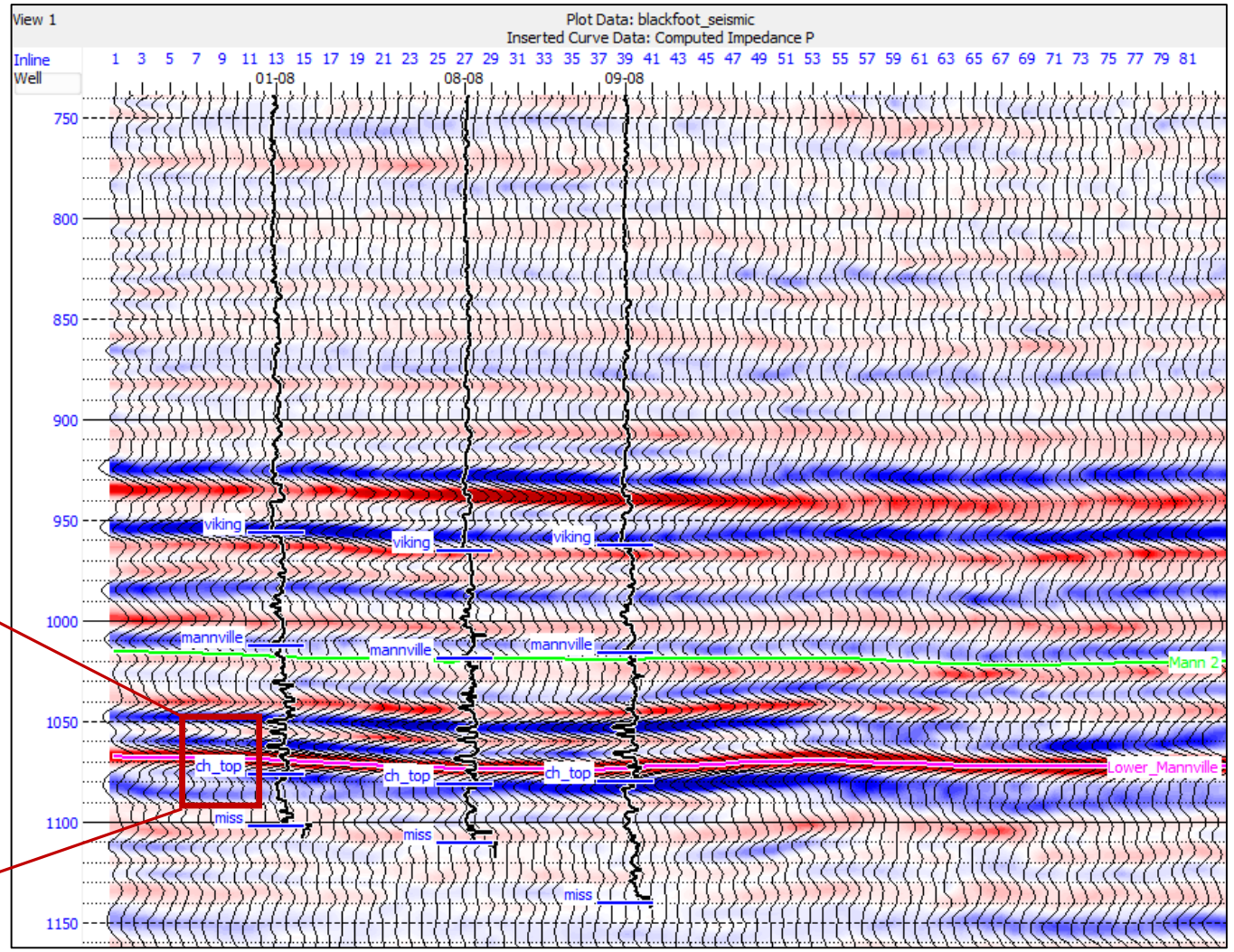
$$\mu_3 = \begin{bmatrix} 2.3571 \\ 2.1248 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.0011 & -0.001 \\ -0.001 & 0.0029 \end{bmatrix}.$$



Facies clustering in the Blackfoot dataset

- Finally, I will apply *K*-means and GMM to the Blackfoot dataset (Dufour et al. 2002), shown on the right, to perform facies clustering.
- The algorithm will be applied to a 10 ms window starting 10 ms below the Lower Mannville event.



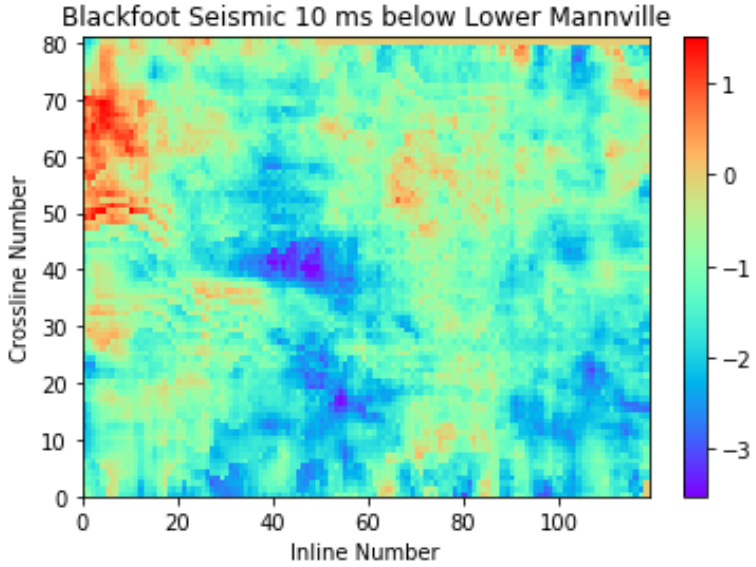


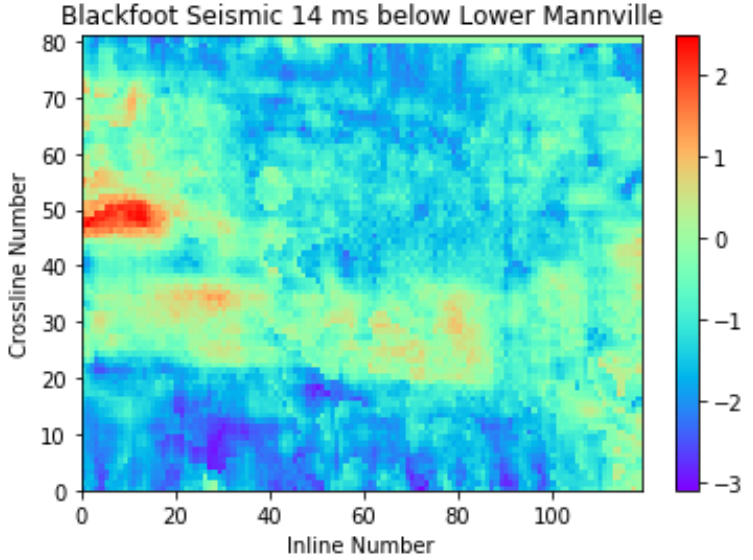
- This means that we are now doing the clustering in 5-dimensional space.

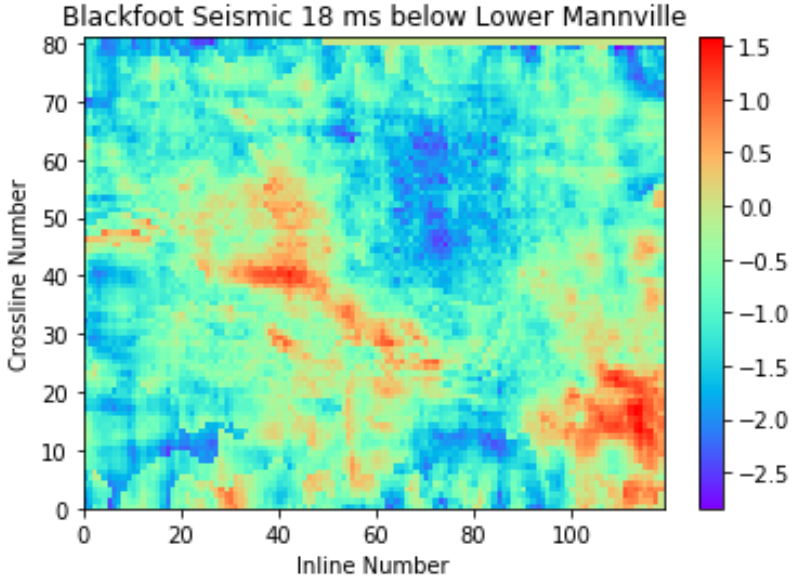
The input data slices

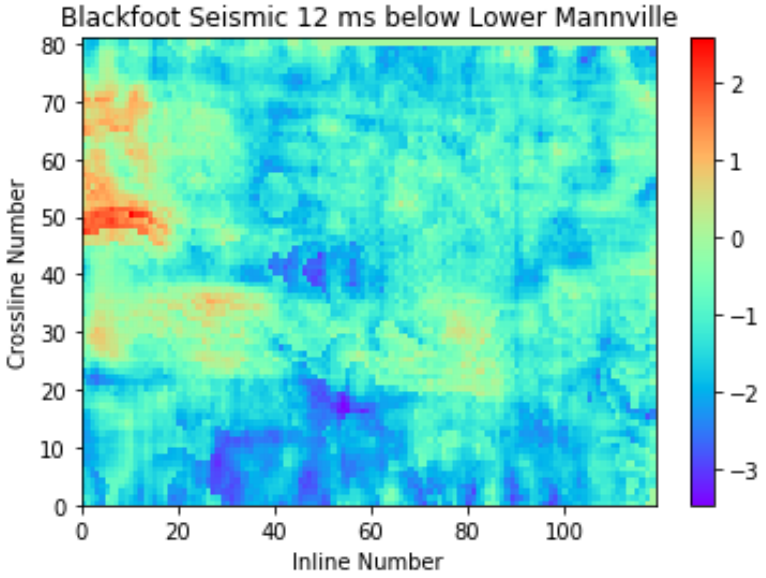
- The Blackfoot dataset consists of 5 data slices from the seismic volume between times of 10 ms and 18 ms below the Lower Manville.
- I extracted the data slices the HampsonRussell GeoView program using our Python Ecosystem.
- I then saved it as a five column ASCII file which was input to a Python program for a recent CREWES Learning Lab that I presented.
- In the Python implementation, the *K*-means algorithm uses only the Euclidean option.
- Therefore, it was necessary to use the GMM algorithm to implement the Mahalanobis distance.
- The data and results are shown in the final slides.

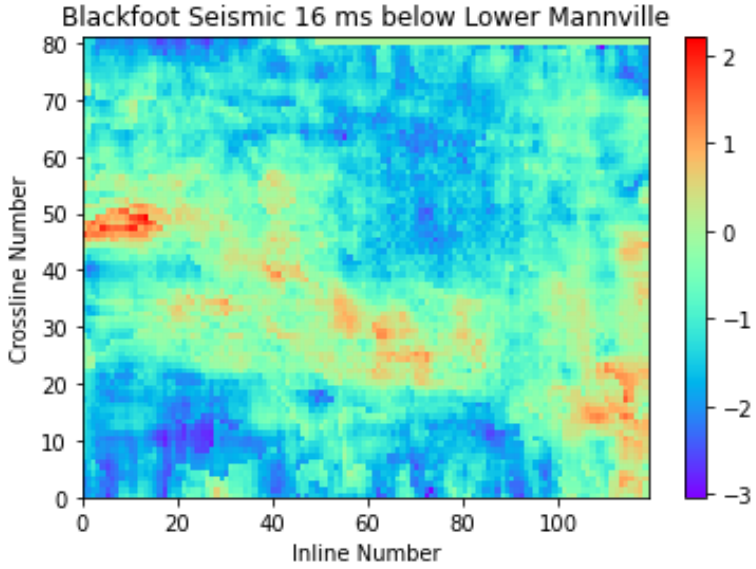
The input data slices





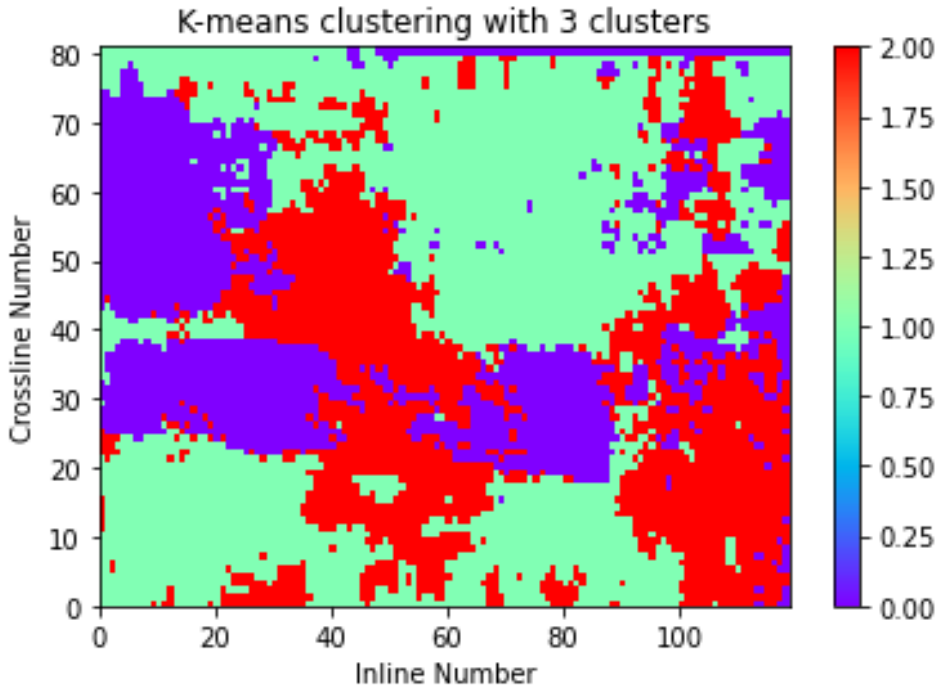


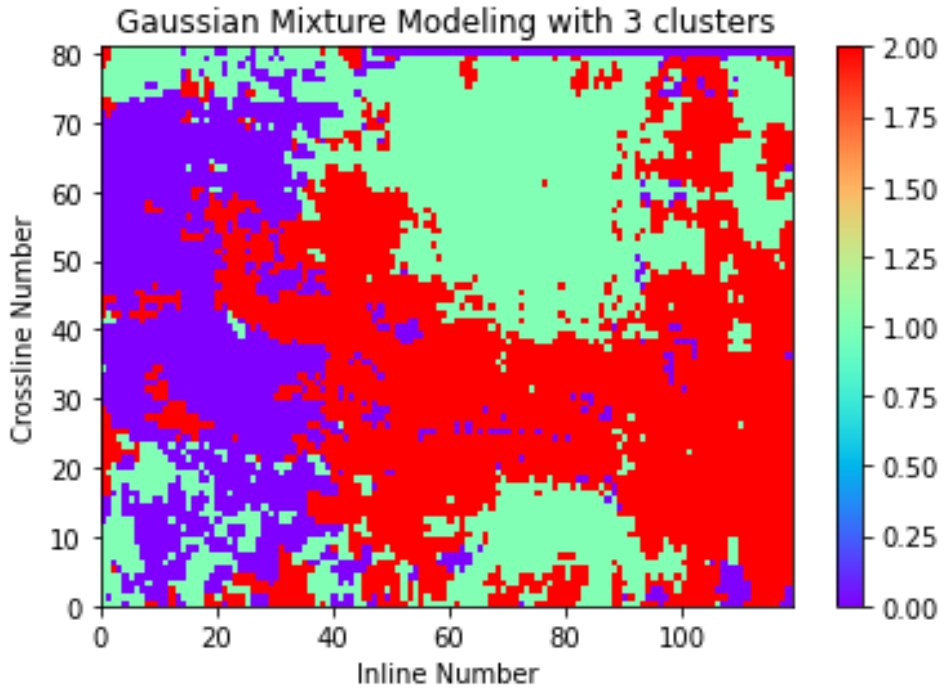




- Here is a display of the 5 data slices from the Blackfoot dataset.
- Note the change in the stratigraphic features from slice to slice.

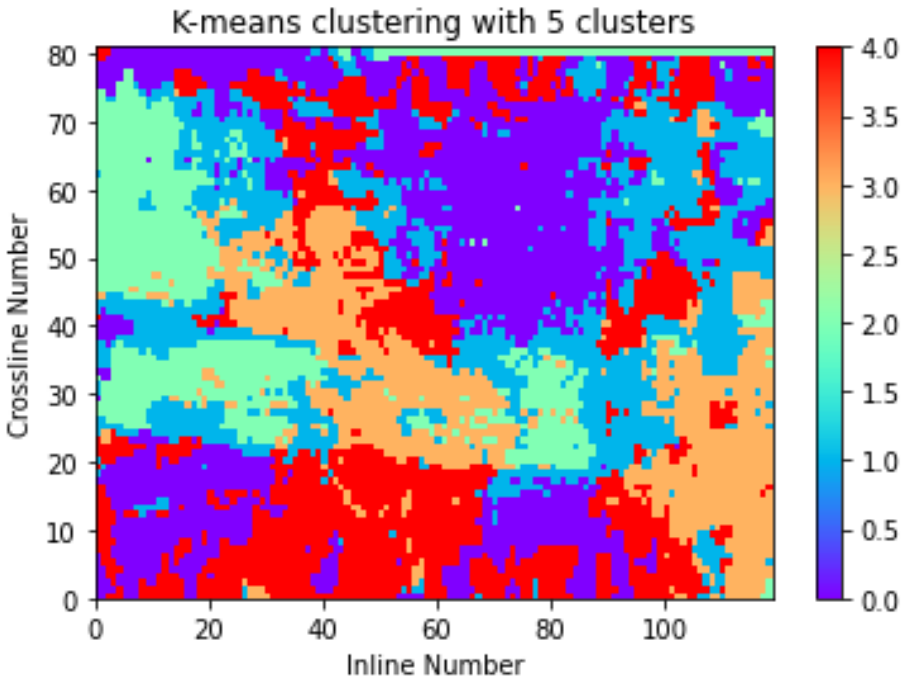
Results using three clusters

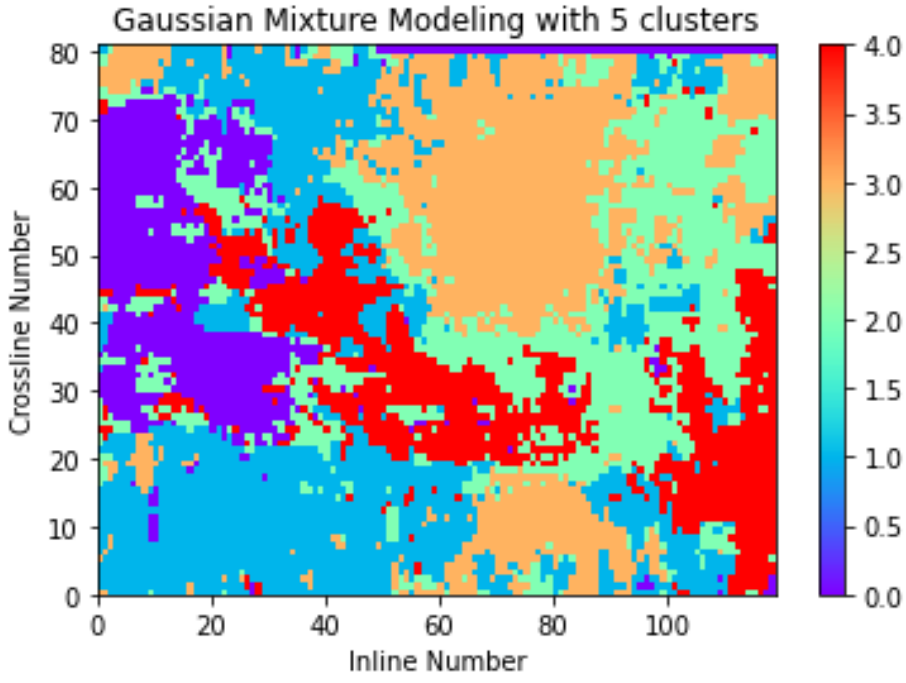




- With 3 clusters the *K*-means and GMM results are similar except that *K*-means sees the NW-SE feature as less extensive than GMM and shows a prominent event (in blue) on both sides of this feature.
- The facies on the NW part of the map is clearest on the *K*-means result.
- The circular facies in the top east side of the map is more obvious in the GMM result.

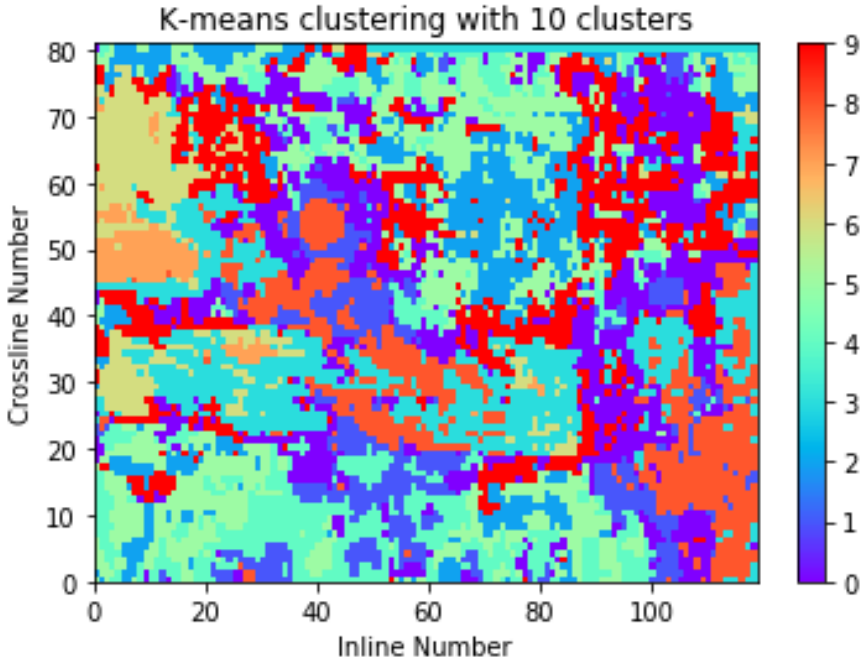
Results using 5 clusters

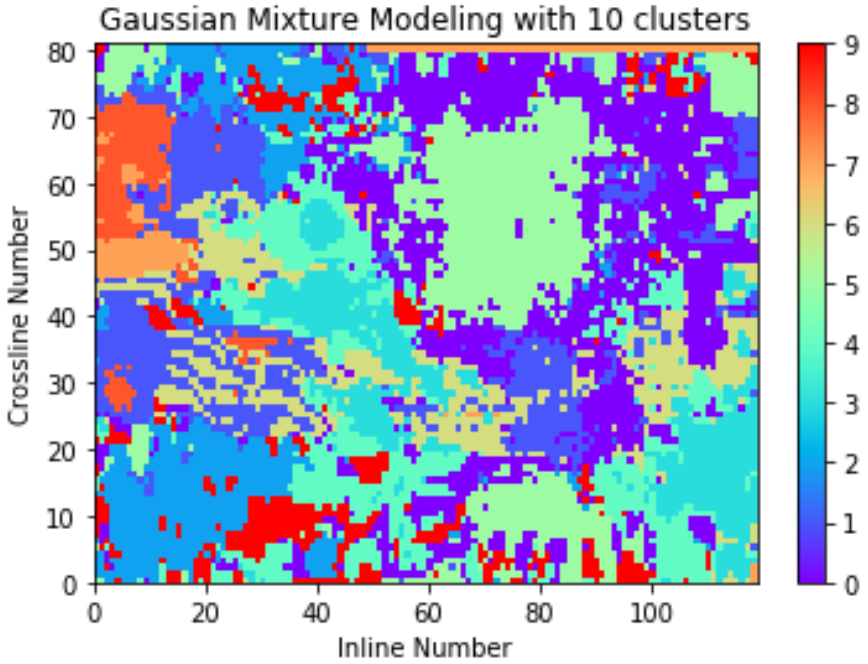




- With 5 clusters the *K*-means and GMM results are even more similar than with 3 clusters if you note the different colour schemes for the cluster order.
- Both the *K*-means and GMM algorithms show similar detail in the NW-SE feature and also on the event in the NW corner of the map.
- Also, the large circular facies in the top east part of the map is becoming more distinct in both methods.

Results using 10 clusters





- With 10 clusters the *K*-means result is much noisier than the GMM result.
- Also, the GMM result is more consistent with the previous examples using 3 and 5 clusters.
- Thus, the GMM method appears to be more robust than the K-means approach in this area.

Conclusions

- In this talk I discussed unsupervised clustering, specifically *K*-means clustering and Gaussian Mixture Modeling.
- The traditional *K*-means algorithm starts with random means and then updates the means by sorting the points using minimum Euclidean distance.
- We found that the *K*-means algorithm becomes much more powerful if we use statistical distance as a sorting criteria instead of Euclidean distance, which requires updating both the means and the covariance matrices.
- The Gaussian Mixture Model (GMM) is equivalent to *K*-means with statistical distance, except it starts with a random guess of the means and covariance matrices and finds the solution with the expectation-maximization (EM) algorithm.
- In my first three examples (two synthetic and one real data example) I performed clustering in 2D space.
- In my last example, for facies classification, the clustering was performed in 5D space and I varied the number of clusters.

Acknowledgements

I wish to thank the CREWES sponsors and my colleagues at Hampson-Russell, CGG, and CREWES.

References

Coléou, T., Poupon, M., and Azbel, K., 2003, Unsupervised seismic facies classification: A review and comparison of techniques and implementation: The Leading Edge, 942-953.

Dufour, J., Squires, J, Goodway, W.N., Edmunds, A., and Shook, I., 2002, Integrated geological and geophysical interpretation case study, and Lamé rock parameter extractions using AVO analysis on the Blackfoot 3C-3D seismic data, southern Alberta, Canada: Geophysics, 67, 27-37.

Russell, B.H and Lines, L.R., 2003, Mahalanobis clustering, with applications to AVO classification and seismic reservoir parameter estimation: CREWES Research Report **15**.

Sung, K.K. and T. Poggio, 1994, Example-based learning for view-based human face detection: AI Memo 1521, MIT.

Wallet, B.C. and Hardisty, R., 2019, Unsupervised seismic facies using Gaussian mixture models: Interpretation, August 2019, SE93 – SE111.