Question 1a

Advantages of multiple sensor fusion

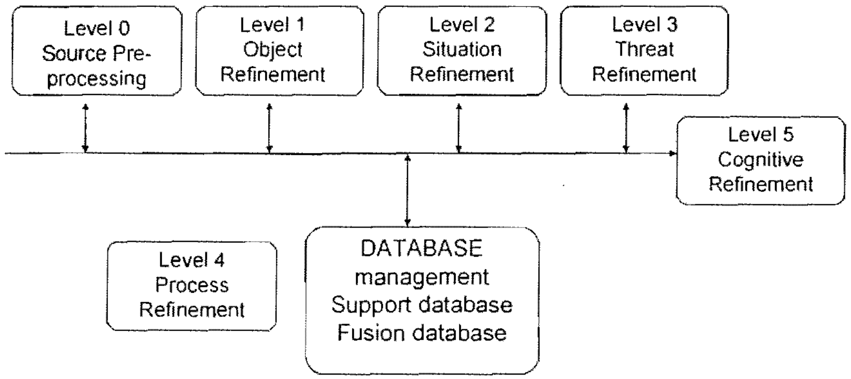
- Use of several identical sensors
  - e.g. radar tracking a moving object provides improved estimates of position and velocity
  - Improvement can be a factor of $\sqrt{N}$ where N is the number of sensors
- Relative placement to improve observation
  - e.g. two angular sensors on the bow and stern of a ship can be combined to give accurate position and velocity through triangulation.
- Use of different sensors to provide a combined integrated ‘picture’
  - e.g. Tracking a moving object such as an aircraft.
  - Radar can provide Range information and infrared imaging can provide Angular Direction
- Sensor reliability
  - e.g. Determine if a sensor in a multi-sensor system is faulty – conflicting information

Question 1b

Choose an example application for each level.

JDL Data Fusion Model

The results of data fusion and the decisions made as a response to the 'output' can span many hierarchical levels. In order to communicate central ideas in an integrated manner, the Joint Directors of Laboratories (JDL) data fusion model was conceived.



Level 0

Data from sensors or signals are often pre-processed, prior to fusion;

e.g. Filtered
- Aligned in time (or space)
- Signal processing
- Normalisation or unit conversions

Processing is normally specific to the particular sensor characteristics being used and the required data – types. (we wont be concentrating on this level in the course)

Level 1 (object refinement)

Combines data to provide a more cohesive 'picture' of an objects state (e.g. position and velocity) or identify (e.g. identify a 'friendly aircraft' from an 'enemy aircraft') Often uses data fusion techniques based on statistical evaluation or probability analysis, or dynamic system models.

An example of level 1 fusion is identifying and monitoring the trajectories of multiple aircraft using multiple sensors. An example is illustrated in the figure below:

(We shall look at techniques for this type of fusion (level))

Level 2 (situation refinement)

Uses level 1 data to try and develop a description of relationships among identified objects to give a meaning or interpretation of the situation or environment. This includes determining where squadrons of aircraft may be heading, their number and classifications, e.g. lots of bombers or lots of fighters?

Alternatively, from an industrial perspective, excessive heat, vibration and noise data may well indicate an impending machine fault.

Level 2 fusion techniques often employ automated reasoning or artificial intelligent, e.g. neural network, patter recognition, expert systems, decision trees, fuzzy logic.

Level 3 (threat refinement or consequence prediction)

Taking data from Levels 1 and 3 and making an assessment about future events and consequences.

e.g. predicting the time to complete failure of a machine, or trying to assess what an opponent might do and the consequences.

Typical level 3 tasks include:
- Estimate force capabilities
- Predicting the intentions of enemy forces based on environment, armaments and their distribution.
- Identifying threat opportunities – possible targets
- Estimating implications – future actions/projection of events after an immediate threat.
- Multi-perspective assessment – assess implications from 'more than one side'
  - e.g. from a neutral perspective
  - from a friendly perspective
  - from an aggressors perspective

Level 3 constitutes on of the most complex tasks of data fusion applications and usually involves the use of intelligent networks, neural networks, case based reasoning, logical templates, expert systems

Level 4 (process refinement)

Seeks to optimise the on-going fusion processes to deliver improved fusion products, e.g. improved estimates of position, velocity of an object by a different emphasis on using the outputs of different sensors – for example, based on weather conditions, e.g. use satellite radar, sensor? To this extent level 4 is considered a meta process.

Typical functions include:

- -System performance management
- -System control (feedback control systems) to optimise an objective function

As well as military applications, this type of process is now being used in e-commerce and e-business.

Actions are often initiated on the results of multi-objective optimisations, based on measures of effectiveness (MOE) and measures of performance (MOP).

For example we may want to maximise the probability of detection of an aircraft and its range (conflicting requirements) based on constraints of;

- Computer performance
- Sensor type and location
- Observation conditions (weather etc)

Level 5 (cognitive refinement)

(recently, 2000, introduced additional level)

•Centred on Human-Computer-Interaction (HCI) tasks.

- Interaction of computer displays and menus
- Displaying data in a cohesive manner
- Gesture recognition
- Help, on-line, efficient (Microsoft; paperclip character) "fill-in the blanks"
- Search engines
- Aids to eliminate cognitive bias-humans often put too much dependency on positive information and ignore contradictory or negative evidence.
- Balancing the need to focus on particular areas, objects or topics while maintaining general surveillance, e.g. air traffic control
- Needs special aids or displays with appropriate visual clues:- impending collision in red!

-Data fusion algorithm performance

In general this is a multi-objective optimisation problem of the form

weight

maximise $$MOE = \sum_{i=1}^{N} \omega_i \quad MOPi$$

Measure of performance of i'th component

Subject to

$MOPi:<C_i$

Typical algorithms used include:
Linear programming
Quadratic programming
Goal programming
Minimum co-variance error analysis
Fuzzy membership functions

Question 1 C

- There are 37 readings from each sensor, therefore from the table use $\nu = \bar{n} - 1 = 36$
- There are 8 sensors, therefore $K = 7$.
- Using the confidence table for $\alpha = 0.05$ (given in question), it is clear that

$$C_{36}(0.05) = 0.2278$$

To determine the possibility of outliers, use:

$$C = \frac{\max \sigma_{si}^2}{\sum_{i=1}^{7} \sigma_{si}^2} = \frac{0.42}{0.925} = 0.4541$$

Since $C > C_{36}(0.05)$ then sensor S_5 is suspect

If S_5 is taken out of the system, then there are $K = 6$ sensors $\rightarrow$ $\nu = \bar{n} - 1 = 36$ as before

$C_{36}(0.05)$ now equals $= 0.2612$.

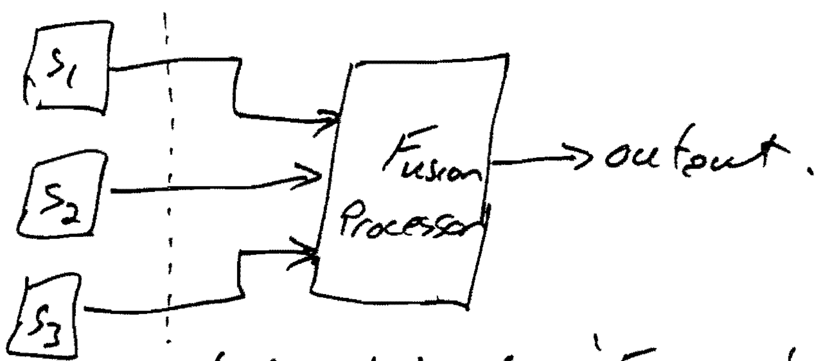
$\therefore C = 0.3960$ i.e. $\left(\frac{0.2}{0.505}\right)$

The system is still suspect $\rightarrow$ $\underbrace{S_4}_{\text{rogue sensor.}}$?

Centralised or Decentralised Data Fusion

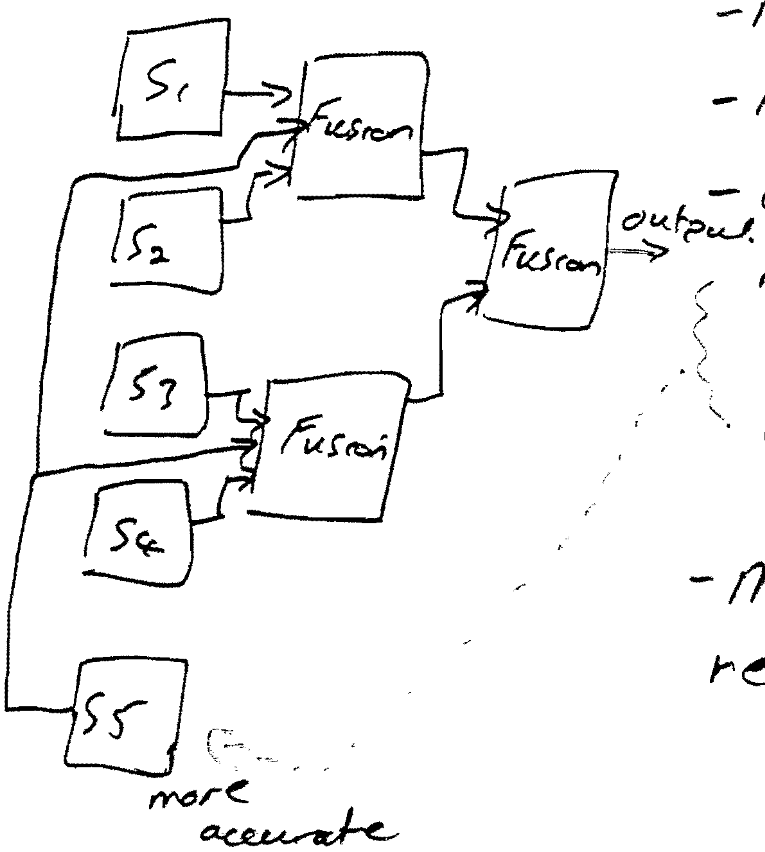
Question 2 (a)

Centralised (eg measurementfusion)



- High computational load on 'Fusion' processor.
- Can cause communication bottleneck
- Difficult to expand.
- Vulnerable to failure in fusion processor.
- Theoretically has the best performance

Decentralised (eg. state fusion. many topologies)



- More easily scalable
- More robust to failures
- Often uses, or passes redundant data or information, therefore performance can suffer.
- Modular and more readily scaleable.

Relative merits:

STATE FUSION

- Reduces computational load on central processor since KF's can be calculated locally and just combined centrally.

- Generally shown to be a sub-optimal solution.

Measurement Fusion

- Reduced covariance of resulting estimates ie generally more accurate than state fusion.

- Requires more centralised processing.

Both techniques can be generalised to more sensors or KFs with obvious increases in computational effort.

Question 2 c

Maximum likelihood decision making.
i)Combine the sensor decision matrices for the 2 sensors as follows:

Fighter

Bomber

$$\begin{array}{cc} & \begin{array}{cccc} m_{11}\,m_{21} & m_{11}\,m_{22} & m_{12}\,m_{21} & m_{12}\,m_{22} \end{array} \\ \begin{array}{r} \cancel{Enemy} \\ \cancel{Friend} \end{array} & \begin{bmatrix} 0.9\times 0.9=0.81 & 0.9\times 0.1=0.09 & 0.1\times 0.9=0.09 & 0.1\times 0.1=0.01 \\ 0.3\times 0.1=0.03 & 0.3\times 0.9=0.27 & 0.7\times 0.1=0.07 & 0.7\times 0.9=0.63 \end{bmatrix} \end{array}$$

ii) For each row, add up the elements that we regard as contributing to a positive identification:

Fighter

Bomber

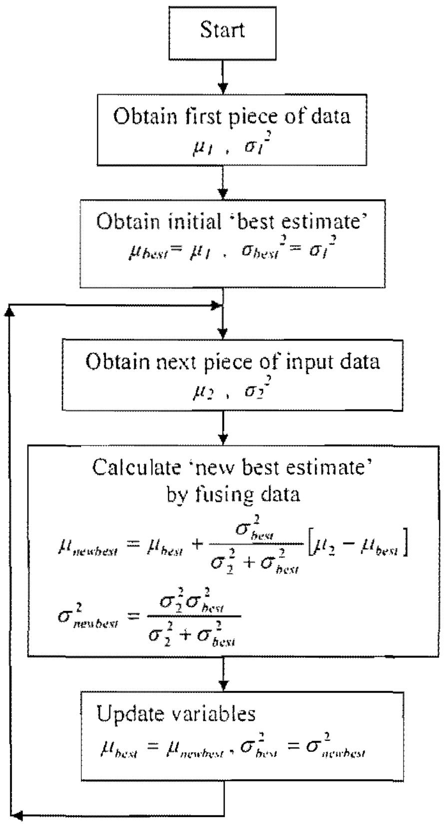
$$\begin{array}{r} \cancel{Enemy} \\ \cancel{Friend} \end{array} \begin{bmatrix} 0.81+0.09+0.09=0.99 & 0.01 \\ 0.1 & 0.27+0.63=0.9 \end{bmatrix}$$

Fighter

This means that the network will statistically identify an Enemy correctly 99% of the time, and will identify a ~~Friend~~ correctly 90% of the time.

Bomber

Question 3a *Flowchart of KF*



Question 3b

mu_best_and_sigma_square_best_vec =

4.1893	0.3000
3.6363	0.1200
3.8343	0.0857
3.7237	0.0600
3.7458	0.0500
3.5559	0.0400
3.6120	0.0353
3.5273	0.0300
3.5659	0.0273

3 c

P(k), Q, R require initialisation at the start of the estimator

R

R is chosen (often) to have the variance of the measurement noise sources on the main diagonals:

$$R = \begin{bmatrix} \sigma_{v1}^2 & 0 & \cdots \\ 0 & \sigma_{v2}^2 & \ddots \\ 0 & & \end{bmatrix}$$

Q

Q is chosen (often) to have the variance of the system noise on the main diagonals:

$$Q = \begin{bmatrix} \sigma_{w1}^2 & 0 & \cdots \\ 0 & \sigma_{w2}^2 & \ddots \\ 0 & & \\ \vdots & & \end{bmatrix}$$

P(k=0)

P(0) allows us to give an initial estimate of our confidence that we know what the initial value ~~of~~ should be. If we know exactly what x is to begin with then

$$\underline{\hat{x}}(0) = x(0) \qquad \text{and} \qquad P = \begin{bmatrix} 0\,0 \cdots & \\ 0 \ddots & \\ 0 & \end{bmatrix} = \underline{0}$$

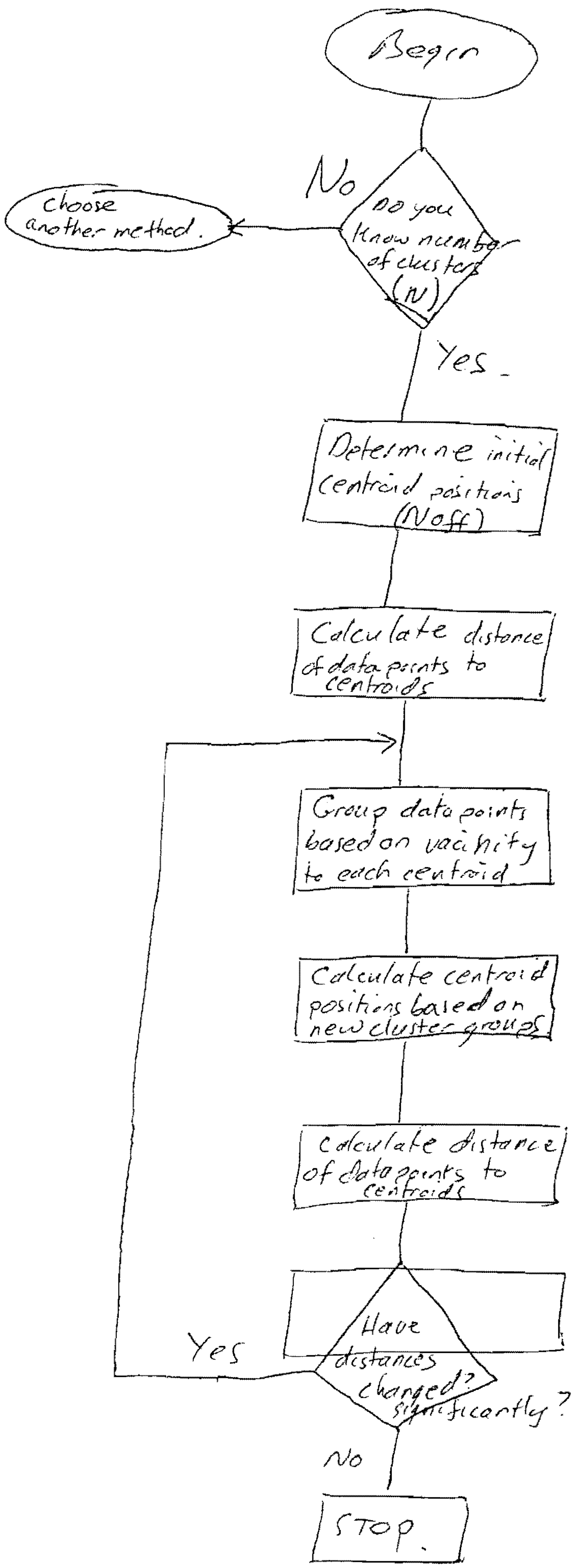
However, if we don't know the exact position then we choose P(0) to reflect our uncertainty—alpha's large

$$\underline{x}(0) = \begin{bmatrix} ? \\ ? \\ \vdots \end{bmatrix} \quad P = \begin{bmatrix} \alpha_1\, 0\, 0 & \\ 0\, \alpha_2\, 0 \ddots & \\ 0\, 0 & \end{bmatrix}$$

All we now have to do is know how to calculate K(k)! It is this calculation that differentiates between state estimation techniques. In some systems $\underline{K}(k) = \underline{K}$ i.e. is chosen to be constant (and assign Eigenvalues to the 'error dynamics'). However, in the KF, $\underline{K}(k)$ is calculated on each iteration to minimise the error variance by taking into consideration the statistical properties of our measurement and system noise.

Question 4 a

i)



Question 4

ii)

Data =

1.5000	2.0000	3.0000	2.0000	3.0000
1.0000	1.0000	3.0000	2.5000	4.0000

Initialise Centroids:

c =

2.5000	1.0000
2.0000	2.0000

ITERATION 0

D_dist =

1.4142	1.1180	1.1180	0.7071	2.0616
1.1180	1.4142	2.2361	1.1180	2.8284

Grouping

0	1	1	1	1	Cluster Group 1
1	0	0	0	0	Cluster Group 2

ITERATION 1

c =

2.5000	1.5000
2.6250	1.0000

D_dist =

1.9080	1.7002	0.6250	0.5154	1.4631
0	0.5000	2.5000	1.5811	3.3541

Grouping

0	0	1	1	1
1	1	0	0	0

ITERATION 2

c =

2.6667	1.7500
3.1667	1.0000

D_dist =

2.4608	2.2669	0.3727	0.9428	0.8975
0.2500	0.2500	2.3585	1.5207	3.2500

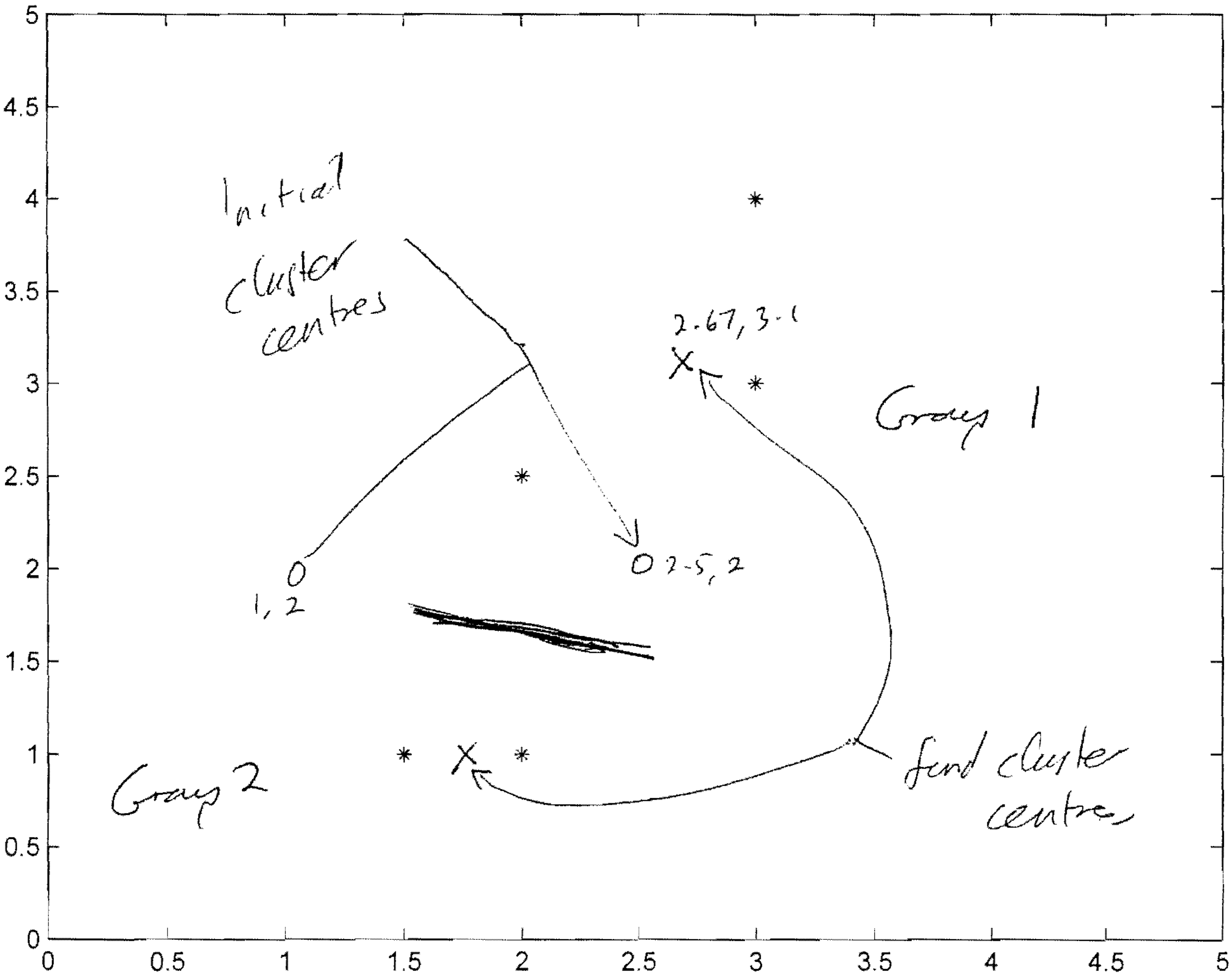
Grouping

0	0	1	1	1
1	1	0	0	0

No Change, therefore A,B are in cluster Group 2, and C,D,E are in cluster Group 1

Question 4

ii) Check.



G = 0 0 1 1 1
1 1 0 0 0

Question 4~~b~~ c

If you weren't aware that there were only 2 vehicles in the vicinity, you may choose to re-run the K-means method with different number of clusters, N, and interpret the data based on how 'tight' the datapoints are within each cluster grouping. Fuzzy clustering methods may also be useful for this purpose.

An alternative method would be to use Hierarchical clustering (either Agglomerate (bottom-up) or Divisive (top-down). Based on the production of a similarity matrix, clusters close to one another are successively combined. The resulting 'hierarchical tree' can be used decide how many clusters you consider to be present (a significant advantage), and their relative vicinity.

The above techniques are based on 'distance' principles. More generally, Probabilistic clustering can be used with great effect based on statistical knowledge of the data being presented.

Importantly, no method is perfect.