本文件的參數標示為綠色，副程式標示為藍色。

要運作整個模型需先準備原始資料集檔名預設data\_XXX.csv，接著開啟runPaper.m即可開始運作，其中參數及函示解釋如下:

1. trial: 為想要跑實驗的次數
2. ExperimentationName: 為原始資料集檔名中的XXX
3. NumberOfTrainPoint: 為此資料集中所要當成訓練資料的筆數
4. ParameterOfPreSC: 此參數用來當作subclust() 函數中的參數，用以決定各個維度模糊集的個數
5. ParameterOfConsSC: 此參數用來當作subclust() 函數中的參數，用以決定後鑑部的個數
6. MODEL\_PSO(): 為整個模型主要內容

MODEL\_PSO()的整體程式碼大致流程如下:

1. 資料前處理
2. 結構學習
3. 模型學習
4. 測試部分
5. 繪圖與儲存
6. **資料前處理**
7. 此部分程式碼主要用 DataSplit.m 將資料讀取進來，並輸出
8. 接著用 FeatureSelection.m 做多目標的特徵挑選，運作內容如下:
   1. 對目標做一次差分，並從漲跌值中劃分成30個特徵
   2. 將差分後的資料和目標合併成矩陣 (DataMatrix)
   3. 將差分後的資料用CalculateIIM.m算出對每個目標的影響矩陣，第i個目標的影響矩陣儲存成IIM(i).value
   4. 計算特徵對每個目標的selection gain。第i個已選特徵池中，特徵索引儲存成SP(i).value，該特徵的selection gain儲存成SP(i).gain
   5. 定義Omega，Omega為每個已選特徵池中的特徵集合(重複記一次就好)，即為特徵的索引
   6. 計算NOL，NOL為特徵出現在各個特徵池的次數，第i個特徵儲存成NOL(i)
   7. 計算w，w為覆蓋率，為NOL除以總目標個數的值，第i個特徵的覆蓋率儲存成w(i)
   8. 計算gSum，為特徵在每個已選特徵池中的selection gain總合，第i個特徵的儲存成gSum(i)
   9. 計算p，p為有效貢獻率，以第i個特徵為例，p(i)為w(i)乘以gSum(i)的值
   10. 排序所有的有效貢獻率p，並透過上界 (Upper)和下界 (Lower)決定最後選取的個數
   11. 輸出為多目標挑選的特徵索引值 (FeatureIndex)
9. **結構學習**
10. 將各個特徵當作輸入維度的資料並分群，第i個輸入維度的資料存儲成h(i).value，第i個輸入維度的中心存儲成h(i).center，第i個輸入維度的標準差存儲成h(i).std
11. 將目標從實數型態組合成複數型態，第i個複數目標值儲存成y(i).value
12. 將各個維度的資料結合用subclust()函式決定後鑑部個數，其中函式參數使用ParameterOfPreSC，得到群數後用fcm()函式將每個目標做分群，第N個目標中，第Q個後鑑部中心儲存成C(N).q(Q)，第Q個後鑑部標準差儲存成S(N).q(Q)
13. 接著做前鑑部挑選，步驟如下:
    1. 創造一個矩陣 (OriginalFormationMatrix)紀錄每個前艦部在各個維度中所包含的模糊集
    2. 計算資料在每個前鑑部的啟動強度，第i個前鑑部啟動強度儲存成OriginalCubeBetaSum
    3. 用門檻值挑選前鑑部，並排序，最後透過上界 (upper)和下界 (down)篩選需要的前鑑部個數
    4. FormationMatrix為挑選過後即將給模型使用的前鑑部矩陣
    5. 透過FormationMatrix，將沒用到的模糊集刪除並重新編碼
14. **模型學習**
15. 設定PSO參數，解釋如下:
16. PSO.w: 為粒子的慣性權重
17. PSO.c1: PSO中個人最好位置的影響權重
18. PSO.c2: PSO中群體最好位置的影響權重
19. PSO.s1: PSO 中個人最好位置的隨機參數
20. PSO.s2: PSO 中群體最好位置的隨機參數
21. PSO.swarm\_size: 整個PSO的粒子總數目
22. PSO.iterations: PSO總共的迭代數
23. 將群體最好位置和每個粒子的速度、位置、個人最好位置初始化，PSO 的參數名稱如下:
24. 第i個粒子的位置: swarm(i).position
25. 第i個粒子的速度: swarm(i).velocity
26. 第i個粒子的最好位置: swarm(i).pBestPosition
27. 第i個粒子的最好位置誤差: swarm(i).pBestDistance
28. 群體中最好的位置: PSOgBest.Position
29. 群體最好位置的誤差: PSOgBest.Distance
30. 設定RLSE 參數
31. 用迴圈跑所有迭代，每次迴圈中每個粒子的流程如下:
    1. 將粒子位置轉換成模糊集，第M個輸入維度的第i個模糊集儲存成termSet.INPUT(M).fuzzyset(i).value
    2. 透過模糊集計算每個前鑑部的啟動強度
    3. 正規化啟動強度
    4. 轉換箭靶中心，並計算設在箭靶上的值，第K個前鑑部設到第Q個箭靶的第N層儲存成lambda(N).k(K).q(Q).value
    5. 將矩陣B、L、H相乘，最後得出Q個輸出P，B儲存成BOfLambda，L儲存成LOfLambda，H儲存成HOfLambda
    6. 透過RLSE優化後鑑部參數 swarm(i).RLSE.iteration(k).theata，其中k代表RLSE的迭代
    7. 將每個P乘上後鑑部參數並加總後即為模型輸出，第i個粒子對第N個目標所產生的輸出，儲存成swarm(i).yHead(:,N)
    8. 計算RMSE更新個人最好位置 (swarm(i).pBestPosition)及群體最好位置 (PSOgBest.Position)
    9. 更新該粒子的速度 (swarm(i).velocity)及位置 (swarm(i).position)
32. **測試部分**
33. 用testPart\_PSO.m來跑測試部分
34. 第M個輸入維度的測試資料定義為testh(M).value，目標定義為testy(M).value
35. 將最好的粒子位置轉換成模糊集
36. 計算每個前鑑部的啟動強度
37. 正規化啟動強度
38. 轉換箭靶中心，並計算設在箭靶上的值，第K個前鑑部設到第Q個箭靶的第N層儲存成testlambda(N).k(K).q(Q).value
39. 將矩陣B、L、H相乘，最後得出Q個輸出P，B儲存成BOfLambda，L儲存成LOfLambda，H儲存成HOfLambda
40. 將Q個P乘上優化過的後鑑部參數，並加總得出模型輸出，第N個目標的輸出儲存成testyHead(:,N)
41. **繪圖與儲存**
42. 畫出每個目標的圖
43. 將輸出每個輸出的實數型態和複數型態拆解，並畫到對應的目標圖上
44. 儲存每個目標訓練和測試的RMSE
45. 將workspace儲存成 Result\_XXX\_trial 的格式，其中XXX為實驗名稱 (ExperimentationName)