

機器學習於材料資訊的應用

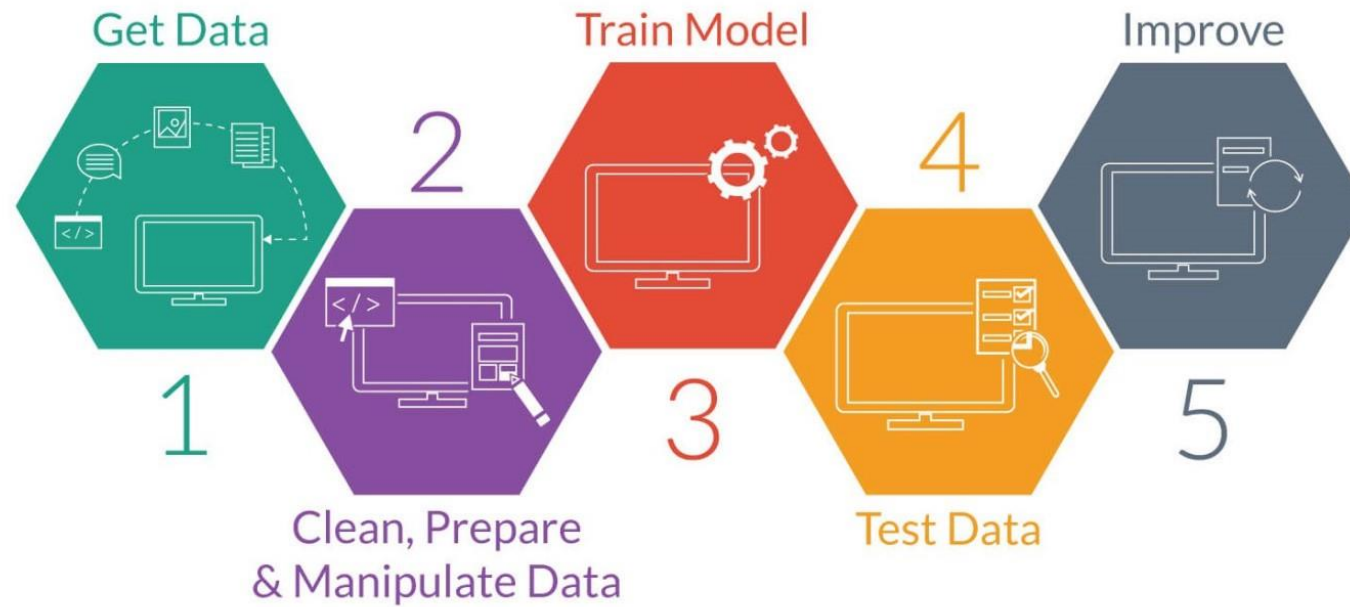
Machine Learning on Material Informatics

陳南佑(NAN-YOW CHEN)

nanyow@narlabs.org.tw

楊安正(AN-CHENG YANG)

acyang@narlabs.org.tw



使用軟體產生資料

檔案處理

建立網路

用測試資料
檢驗演算法

調整萃取特徵
方法

特徵萃取

分群演算法

LAMMPS



ASE

建立網路

- **Modularity**是基於網路分析的方法，所以比起其他分群演算法，需要多一個建立資料點的網路關係。
- 網路式建立於資料點的空間，不是原本問題的卡式座標。
- 距離的定義有許多種，歐式距離、曼哈頓距離、Dijkstra distances ...
- 網路連通的定義也要選擇。

Construct Networks ϵ -ball method

□ Steps:

1. 算出資料點兩兩距離。
2. 將所有距離排序，並依序設為 ϵ -ball的半徑。
3. 定義每個點與 ϵ -ball內的點都有連接，以此建立graph。
4. 檢查graph是否為連通圖(Connected graph)，若為否，則挑選更大的距離作為 ϵ -ball的半徑，重複3,4步驟。
5. 得到連通圖後再計算modularity。

Construct Networks

construct_network.ipynb

#載入資料點

```
file = 'features.csv'
```

```
features_space = pd.read_csv(file)
```

```
features_space.shape
```

(61, 730)

61顆原子 ← → $1+9*9*9=730$ voxels

Construct Networks

construct_network.ipynb

```
natom = features_space.shape[0]
ndim = features_space.shape[1]
#建立連線關係表作為networkx產生圖的輸入
outfile = 'network.csv'
with open(outfile, 'w') as output:
    writer = csv.writer(output, delimiter=' ')
```

Construct Networks

construct_network.ipynb

```
for i in list(range(natom)):
    features_i = features_space.iloc[i, :].values
    for j in list(range(i+1,natom)):
        features_j = features_space.iloc[j, :].values
        dist=0.0
        for k in list(range(1,ndim)):
            dist += (features_i[k] - features_j[k])**2
        print( int(features_i[0]), int(features_j[0]), dist)
        writer.writerow( [int(features_i[0]), int(features_j[0]), dist] )
```

start point

end point

distance

Construct Networks

construct_network.ipynb

```
import community
import networkx as nx
import matplotlib.pyplot as plt

#讀入先前建立的連線關係表
Edges = pd.read_csv('Edges.csv', sep=' ', header=None )

E = np.array(Edges.iloc[:, 0:3].values)
```


Construct Networks

construct_network.ipynb

```
#Use pre-defined linkage (Edges.csv) to construct whole network
```

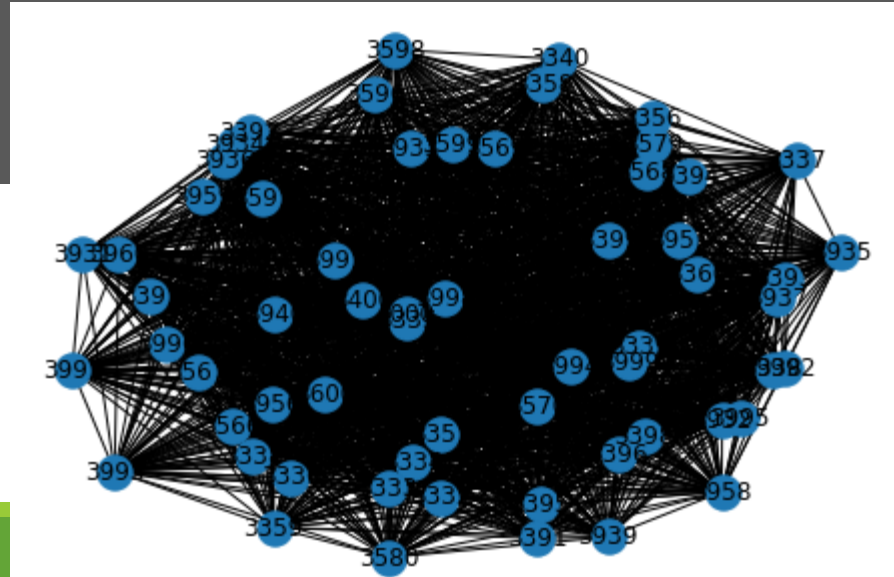
```
G = nx.Graph()
```

```
for i in range(0, len(E)):
```

```
    e = ( str(int(E[i,0])), str(int(E[i,1])), E[i,2] )
```

```
    G.add_weighted_edges_from([(e)])
```

```
nx.draw(G, with_labels=True)
```



Calculate Modularity

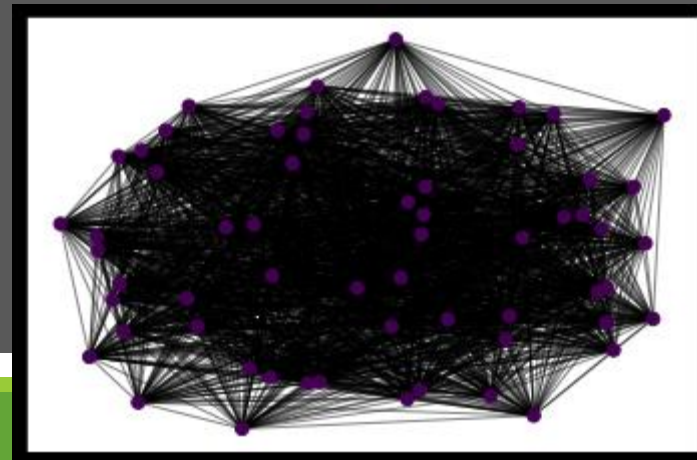
construct_network.ipynb

```
# https://python-louvain.readthedocs.io/en/latest/
partition = community.best_partition(G)
# community.best_partition回傳的物件是dict，利用.values取出各群的内容，。
# 利用set建立無序的資料集合，再用len取得這個集合的長度，這就是分群後的結果。
size = float(len(set(partition.values())))
print("community:", size)
# community.modularity就直接計算modularity，graph太大的話，就要考慮自己實做平行版本的modularity
mod = community.modularity(partition, G)
print("modularity:", mod)
```

Calculate Modularity

construct_network.ipynb

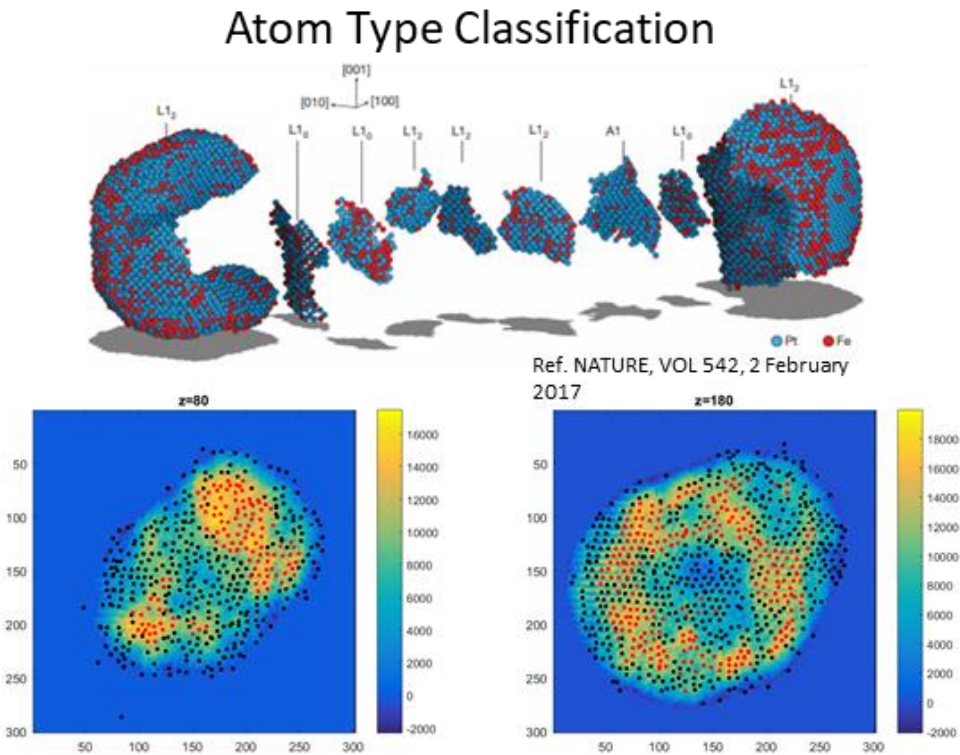
```
# draw the graph based on the result of modularity
pos = nx.spring_layout(G)
# color the nodes according to their partition
cmap = matplotlib.cm.get_cmap('viridis', max(partition.values()) + 1)
nx.draw_networkx_nodes(G, pos, partition.keys(), node_size=40,
                       cmap=cmap, node_color=list(partition.values()))
nx.draw_networkx_edges(G, pos, alpha=0.5)
plt.show()
```



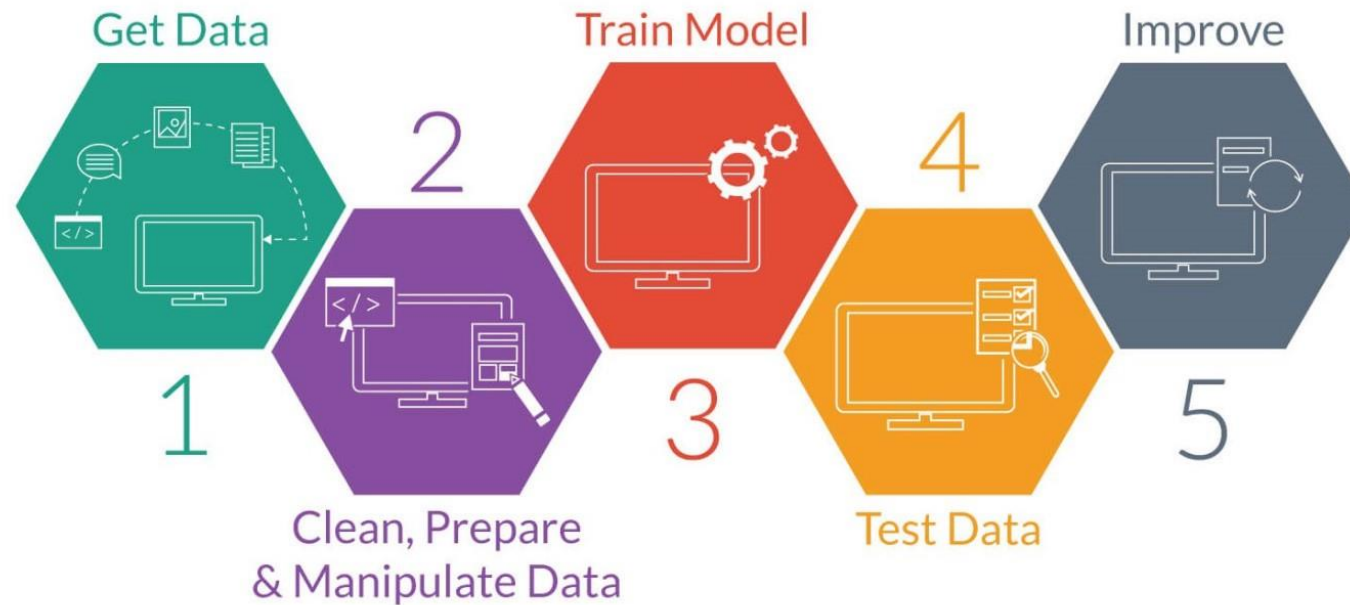
奈米粒子影像重建

種類與缺陷型態深度辨識機器

合作計畫	國研院創新計畫
合作團隊	清大工科陳健群實驗室



案例	三維斷層造影之原子種類與缺陷型態辨識
客戶目標	透過深度學習技術，定位原子座標、辨識原子種類、分析缺陷型態
問題困難描述	需要人工判斷原子種類與缺陷邊界，會因人為偏見造成不一致性的誤判。
訓練資料來源	原子級三維斷層造影顯微技術之實驗資料
機器學習引擎	Convolutional Neural Network, Ensemble Learning, Active Learning
結果	成功完成原子座標定位、種類辨識、缺陷邊界型態與相似度分析。降低人為誤判機會。
Status	通過實驗組驗證 論文撰寫中(Nature Materials)



3D X-ray 影像重建得到影像資料
(Volume Data)

檔案處理

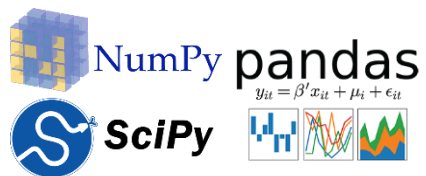
特徵萃取

建立3D CNN
模型

分類問題

用測試資料
檢驗演算法

調整萃取特徵
方法



資料說明

	類型	Feature	Label Data
TrainData.mat TestData.mat	實驗資料	原子座標	K-mean 結果(論文發表用)
dAtomIntensity_TrainData_New.mat dAtomIntensity_TestData_New.mat	後處理資料	ROI區域內強度分佈 (7*7*7的格點)	
iAtomType_New.mat			人工挑選可信度高的分類結果進行標記