

Foursquare

組員：謝綺珊、劉育彤





Foursquare - Location Matching

- 目標：將匹配 POI(points-of-interest)，使用包含超過一百萬個地方目的地數據集，這些資料經過大量修改包含雜訊、重複、無關或不正確的信息，將生成一種算法來預測哪些 **地方目的地** 代表相同的興趣點，成功的提交確定最準確的匹配項，透過有效且成功地匹配 POI，可以更輕鬆地確定新商店或企業在哪些地方對人們最有利。

資料

Data Explorer

346.93 MB

-  pairs.csv
-  sample_submission.csv
-  test.csv
-  train.csv

- 配對地方標的物
- 文件格式
- 訓練及測試資料(名稱、經緯度、地址、城市、州名稱、國家)

計算 Haversine 距離(經緯度)

Haversine:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

With:

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

and:

$$d = R \cdot c$$

Where:

ϕ = latitude

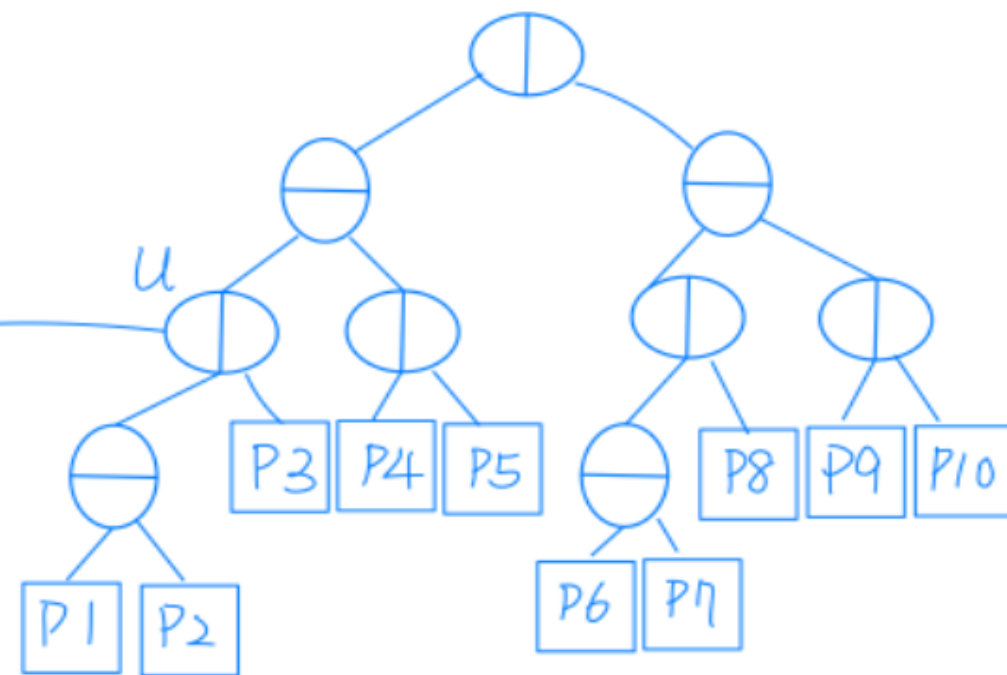
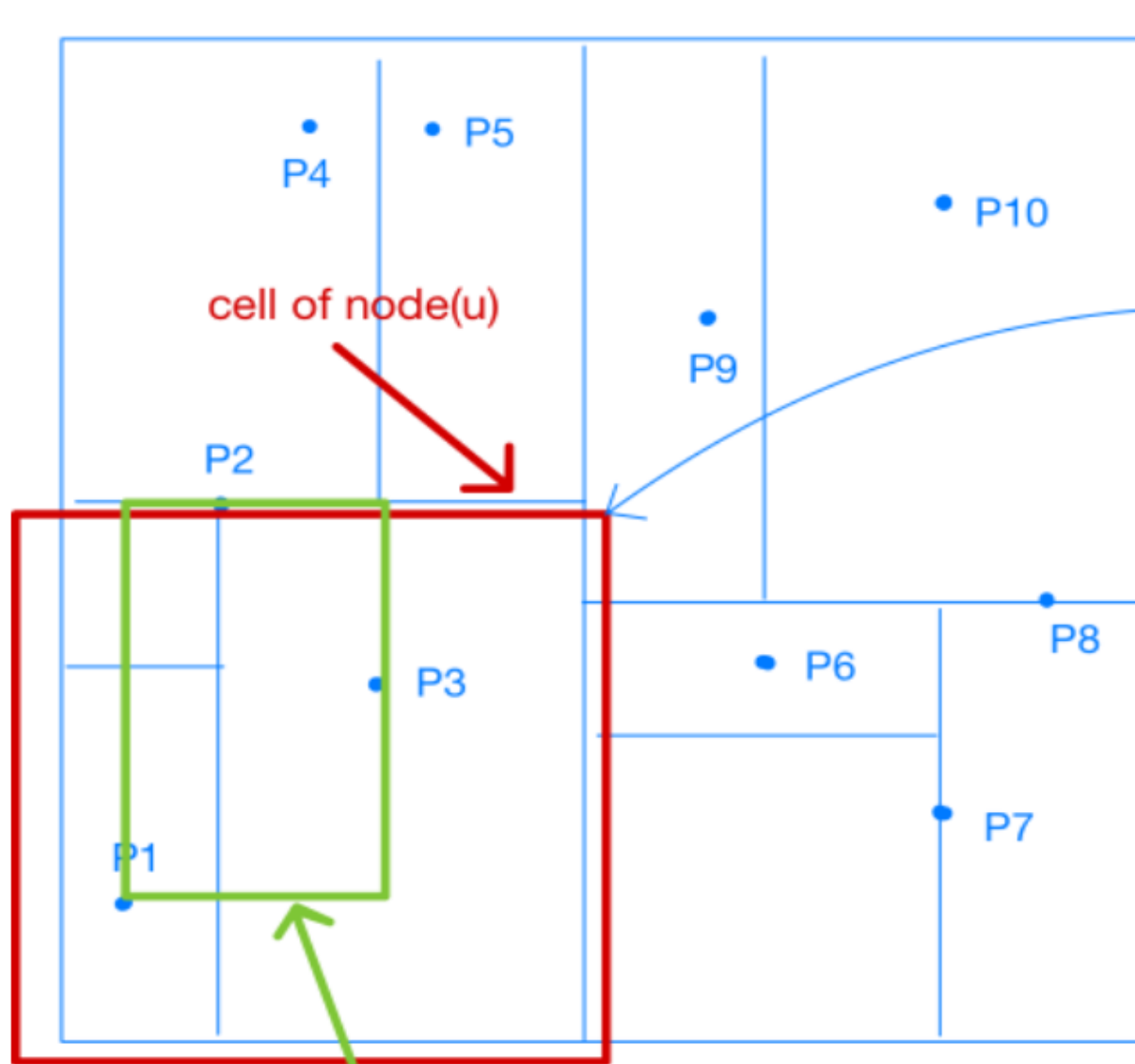
λ = longitude

R = 6371 (Earth's mean radius in km)

```
def haversine(lon1, lat1, lon2, lat2):  
    """  
    Calculate the great circle distance in kilometers between two points  
    on the earth (specified in decimal degrees)  
    """  
    # convert decimal degrees to radians  
    lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2])  
  
    # haversine formula  
    dlon = lon2 - lon1  
    dlat = lat2 - lat1  
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2  
    c = 2 * asin(sqrt(a))  
    r = 6371 # Radius of earth in kilometers. Determines return value units.  
    return c * r
```

KDTree

- KD 樹又稱 K 維樹 (K-dimensional tree)，是一種可以對 K 維資料進行劃分的資料結構，可以看成二元搜尋樹的一種延伸，不斷的對空間中的維度做劃分，利用搜尋樹剪枝的特性縮短時間複雜度，主要應用在多維空間搜尋，例如最近鄰居搜索。
- 1. 從資料中每次選擇其中一個維度進行劃分
- 2. 在某個維度上選擇節點進行劃分，得到子空間
- 重複2點過程直到每個子空間都不能再劃分為止



結果

..		precision	recall	f1-score	support
	False	0.35	0.34	0.35	35901
	True	0.71	0.72	0.71	79881
accuracy				0.60	115782
macro avg		0.53	0.53	0.53	115782
weighted avg		0.60	0.60	0.60	115782

SVC

```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(X), y, test_size=0.2, random_state=42)

svc = svm.SVC()
svc.fit(X_train, y_train)

print(classification_report(y_test, svc.predict(X_test).astype(int)))
```

	precision	recall	f1-score	support
False	0.00	0.00	0.00	35901
True	0.69	1.00	0.82	79881
accuracy			0.69	115782
macro avg	0.34	0.50	0.41	115782
weighted avg	0.48	0.69	0.56	115782

Softmax Regression

```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(X), y, test_size=0.2, random_state=42)
softmax_reg = LogisticRegression(multi_class='multinomial', C=1)
softmax_reg.fit(X_train, y_train)

print(classification_report(y_test, softmax_reg.predict(X_test).astype(int)))
```

	precision	recall	f1-score	support
False	0.00	0.00	0.00	35901
True	0.69	1.00	0.82	79881
accuracy			0.69	115782
macro avg	0.34	0.50	0.41	115782
weighted avg	0.48	0.69	0.56	115782

SGD classifier

```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(X), y, test_size=0.2, random_state=42)
sgd_clf = SGDClassifier(loss='log', penalty='l2', alpha=0.0001)
sgd_clf.fit(X_train, y_train)

print(classification_report(y_test, sgd_clf.predict(X_test).astype(int)))
```

	precision	recall	f1-score	support
False	0.00	0.00	0.00	35901
True	0.69	1.00	0.82	79881
accuracy			0.69	115782
macro avg	0.34	0.50	0.41	115782
weighted avg	0.48	0.69	0.56	115782

Decision Tree

```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(X), y, test_size=0.2, random_state=42)

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

print(classification_report(y_test, clf.predict(X_test).astype(int)))
```

	precision	recall	f1-score	support
False	0.35	0.37	0.36	35901
True	0.71	0.69	0.70	79881
accuracy			0.59	115782
macro avg	0.53	0.53	0.53	115782
weighted avg	0.60	0.59	0.59	115782

Native Bayes

```
X_train, X_test, y_train, y_test = train_test_split(pd.DataFrame(X), y, test_size=0.2, random_state=42)

BS = GaussianNB()
BS.fit(X_train, y_train)
BS.predict(X_test)

print(classification_report(y_test, BS.predict(X_test).astype(int)))
```

	precision	recall	f1-score	support
False	0.00	0.00	0.00	35901
True	0.69	1.00	0.82	79881
accuracy			0.69	115782
macro avg	0.34	0.50	0.41	115782
weighted avg	0.48	0.69	0.56	115782

結論

- 測試多種模型，顯示一開始使用的KD tree 較好的表現，而一般的決策樹也比他種模型好
- 有多組模型的結果幾乎是一樣

網頁連結

- <https://www.kaggle.com/competitions/foursquare-location-matching>