

推薦系統 API 雛形 — 必要之參數物理意義簡介(Ver. 1.0)

作者： 陳韋辛 Frank Chen

前言： 由於此程式目的是為了讓使用者不用瞭解內部數學原理即可使用，此外此程式內部也包含不少數學模型、計算與分析方法，詳見文件附錄之關鍵字，所以此份文件只會描述 Demo.py 中 Train 方程式與輸入有關的重要數學意義，如神經元數量與計算迴圈數對預測結果的影響，若對該數學模型和演算法感興趣可以查閱此文件之參考資料，也歡迎隨時與我討論。

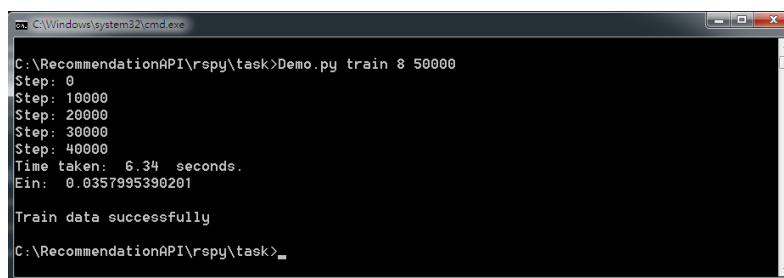
數學模型： Neural network with linear regression (Matrix Factorization Learning)

輸入： 多位顧客對多項商品的已評評價

輸出： 預測某位顧客對未知商品的評價

Train 方程式使用方法：

第一次使用時，必須計算所有用戶之評分資料，以產生矩陣 V 和矩陣 W 以便預測全體用戶之行為，因此使用者必須呼叫 Demo.py 中的 train 方程式，並輸入參數「神經元數量」以及「計算迴圈數」，若無輸入參數，則預設神經元數量為 10，計算迴圈數為 10000000。



```
C:\Windows\system32\cmd.exe
C:\RecommendationAPI\rsipy\task>Demo.py train 8 50000
Step: 0
Step: 10000
Step: 20000
Step: 30000
Step: 40000
Time taken: 6.34 seconds.
Ein: 0.0357995390201
Train data successfully
C:\RecommendationAPI\rsipy\task>
```

神經元數量的潛藏意義：

神經元數量的意義如同將顧客與商品視為由「多少種成分」所組成，一個神經元代表一種成分。舉例來說，若神經數量為 3，則電腦可能會判斷每個顧客是由「喜歡喜劇片之程度」、「喜歡動作片之程度」、「喜歡文藝片之程度」所組成，而每部電影則是由「喜劇片之比例」、「動作片之比例」、「文藝之比例」所組成，因此只要藉由兩者向量之乘積即可計算該顧客對該電影之評價。當然實際上每個神經元所代表的物理意義完全是由電腦所決定，因此該成分真正在實際生活中所代表的物理意義通常不會像「喜劇成分」那樣如此直覺，而此套件輸出之 V 矩陣與 W 矩陣分別記錄每位顧客與商品所含的成分比例。

神經元數量對預測結果的影響：

$$\text{預測結果誤差} \cong \text{模型複雜度} + \text{測試資料誤差}$$

當神經元過少時，雖然模型複雜度(model complexity)會較低，但測試資料誤差(in-sample error)會較高，因此預測結果誤差(out-of-sample error)仍會很高，如 Fig. 1。當神經元過多時，雖然測資誤差較低，但模型複雜度較高，因此預測結果誤差仍會很高，此外神經元數量過高也會大幅提升運算時間。

由此可見，較低或較高的神經元數量皆會使預測結果變差，因此必須選擇較適當的神經元數量，可以使用 Cross Validation 等技巧找尋此適當值。

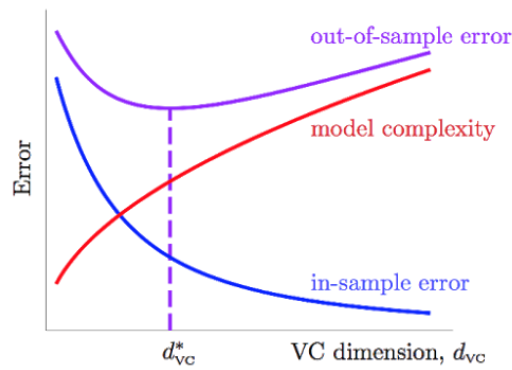


Fig. 1 複雜度與預測誤差之關係^[1]

計算迴圈數對預測結果的影響：

如同神經元數量與預測結果之關係。當迴圈數過少時，數學模型尚未截取到測資中的 pattern 便停止，因此測資誤差 E_{in} 與預測誤差 E_{test} 都較高，如附圖。當迴圈數過多時，雖然測資誤差較低，但模型太過追求於降低測資誤差，導致預測誤差隨之提升，此外迴圈數過高也會大幅提升運算時間。由於過少或過多的計算迴圈數都會導致預測誤差變大，因此使用者也必須使用 Cross Validation 等技巧選取適當之迴圈數值。

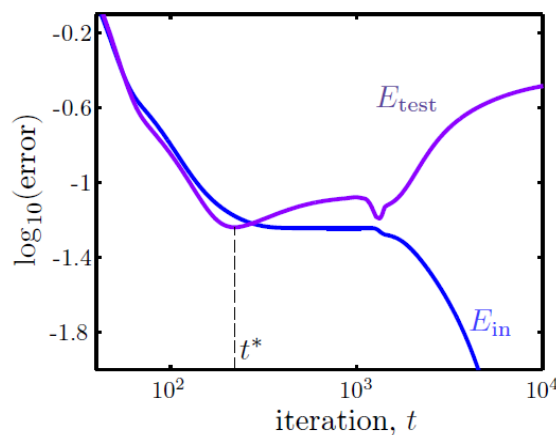


Fig. 2 迴圈數與預測誤差之關係^[1]

關鍵字：

VC Dimension, Binary Vector Encoding, Neural Network, Regression and Stochastic Gradient Descent

參考資料：

- [1] Handouts from Machine Learning Techniques by Hsuan-Tien Lin, National Taiwan University.
- [2] Machine Learning Foundations in Coursera, National Taiwan University
<https://www.coursera.org/course/ntumlone>
- [3] Machine Learning Techniques in Coursera, National Taiwan University.
<https://www.coursera.org/course/ntumltwo>
- [4] Machine Learning in Coursera, Stanford University.
<https://www.coursera.org/course/ml>