

## Collaborative Filtering

**user-based** 考慮的是 user 和 user 之間的相似程度

Insight: Personal preferences are correlated • If Jack loves A and B, and Jill loves A, B, and C, then Jack is more likely to love C

給定一個用戶 A

計算用戶 A 跟其他所有用戶的相似度

找出最相似的 m 個用戶

再找出這些用戶有評分但是用戶 A 沒有評分的物品（也可以額外限制至少要幾個用戶有評分過）

以「相似用戶的相似度」和「該用戶對該物品的評分」來加權算出用戶 A 對這些未評分物品的評分

最後推薦給 A 評分最高的 n 個物品

**Item-based** 考慮的是 item 和 item 之間的相似程度

如果物品數比用戶數還少得多的話

可以事先計算好所有物品之間的相似度

給定一個用戶 A

找出用戶 A 的所有未評分物品

以「用戶 A 的已評分物品對該未評分物品的相似度」和「用戶 A 對已評分物品的評分」來加權算出用戶 A 對這些未評分物品的評分

最後推薦給用戶 A 評分最高的 n 個物品

CF 的缺點：

- 如果沒有用戶的歷史數據就沒辦法做任何推薦
- 以及無論 user-based 或 item-based 都需要消耗大量的運算資源
- 大部分用戶有評分紀錄的資料都只佔所有資料中的很小一部分，matrix 相當稀疏，很難找到相似的資料
- 越熱門的物品越容易被推薦，所以通常都會降低熱門物品的權重

## Content Based Filtering

比較商品的屬性，找出最相似的商品。

- Each item is described by a set of features
- Measure similarity between items
- Recommend items that are similar to the items the User liked