

A Food Recommendation System Based on Graph Neural Networks

張碩文
NCKU CSIE

p76134692@gs.ncku.edu.tw

蘇祐蓁
NCKU AIM

ne6131021@gs.ncku.edu.tw

陳子輝
NCKU CSIE

p76135062@gs.ncku.edu.tw

許漢權
NCKU IMI

q56135019@gs.ncku.edu.tw

Abstract

隨著近年線上外送平台如 *Foodpanda*、*UberEat* 逐漸成為主流，餐點推薦系統的重要性日益增加。有效的推薦系統不僅能夠精確地向使用者推薦合適的餐點，還可以幫助平台最大化收益並提升用戶體驗。本研究將提出一種基於圖神經網路 (Graph Neural Networks; GNN) 的推薦系統模型，通過分析平台上用戶與餐點之間的關聯性，從中萃取出有價值的資訊，進而提供更精準的餐點推薦。此模型將針對使用者的偏好與歷史行為進行學習，結合圖結構的數據特徵，實現更個性化且高效的推薦效果。最後，我們將對該模型進行實驗與評估，並與現有的推薦系統進行比較。

1. Introduction

因線上外送平台如 *Foodpanda* 和 *UberEat* 的迅速發展，餐飲產業的數位化轉型日益顯著。這些平台不僅簡化了顧客與餐廳之間的聯繫，還透過大規模數據分析來改進用戶體驗及提高平台的經濟效益。餐廳推薦系統在這一過程中扮演了至關重要的角色，能夠根據使用者的歷史消費行為、偏好和時下趨勢，及時推薦合適的餐廳，從而提升平台服務的精確性和客戶滿意度。

傳統的推薦系統方法，如基於內容的過濾 (content-based filtering) 和協同過濾 (collaborative filtering)，雖然在一定程度上能夠提供有效的推薦，但往往忽視了用戶與餐廳之間更深層次的聯繫結構。隨著圖神經網路 (Graph Neural Networks; GNN) 的發展，發現可以將平台上用戶與餐廳之間的關聯性表示為圖結構，從而在更高層次上捕捉到它們之間的複雜互動，其中這個圖結構能以二分圖 (bipartite graph) 表示。如圖 1 所示，能將所有的節點分成，左邊的使用者節點，右邊的推薦餐廳節點，因左右兩邊的節點類型不同，該圖同時為異質圖 (attributed heterogeneous graph)。本研究的架構圖如圖 2 所示，從 *Foodpanda*、*GoogleMap* 上使

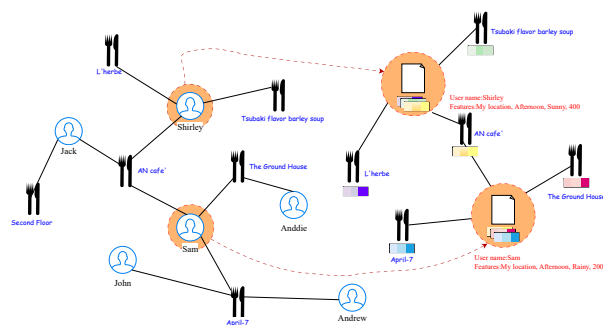


Figure 1. 二分圖於推薦系統 [1]

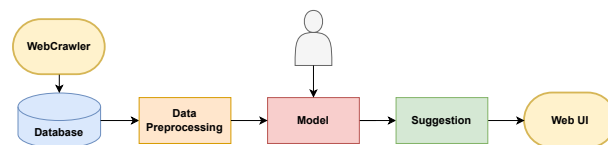


Figure 2. 推薦系統架構圖

用爬蟲將每個店家的資訊儲存到資料庫，並使用基本的資料前處理，對於使用者節點，其節點內容如同表格 1，會將這些資訊同時輸入模型，並得出最後的推薦結果，再以 Web UI 的方式呈現。本研究將使用

節點類型	節點內容
使用者	當下位置、使用時間、天氣、預算
推薦餐廳	餐廳名稱、價位、評分、評論

Table 1. 使用者與推薦餐廳節資訊

Zhang 等人所提出的圖卷機網路 [2] 應用在店家推薦的資料集，並使用歸一化折扣累積增益 (Normalized Discounted Cumulative Gain; NDCG)、召回率 (Recall) 去衡量推薦系統之表現。

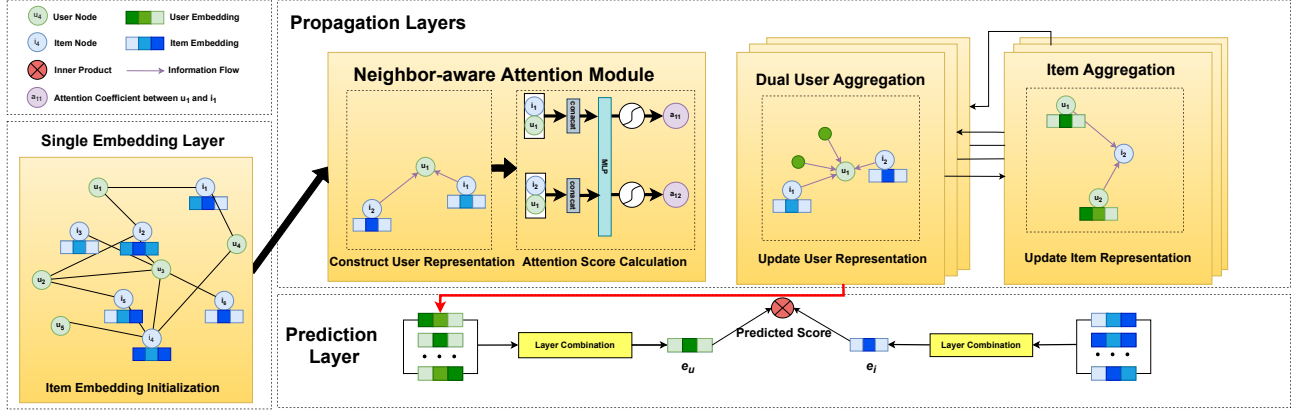


Figure 3. 模型架構圖

2. Proposed

2.1. Single Embedding Layer

本研究所採用的NIE-GCN[2]模型可拆解成幾個主要步驟，首先是對餐廳節點進行向量嵌入。第一步會針對每個餐廳節點 i 生成一個嵌入向量 e_i ，此向量位於 d 維空間中，即 $e_i \in \mathbb{R}^d$ 。在嵌入過程中，所有餐廳節點的嵌入向量會被收集成一組矩陣 $E_I^{(0)}$ ，此矩陣表達了所有餐廳節點的初始嵌入狀態，如下式所示：

$$E_I^{(0)} = [e_{i_1}^{(0)}, e_{i_2}^{(0)}, \dots, e_{i_N}^{(0)}] \in \mathbb{R}^{N \times d}, \quad (1)$$

其中， N 代表餐廳節點的總數，而 d 則是嵌入向量的維度；上標0表示第0層傳播的初始狀態。藉由得到每個餐廳節點的嵌入向量後，模型便可利用這些向量作為基礎，來進一步推斷每位使用者的偏好。因為每個使用者節點 u 的鄰居必定為餐廳節點，故模型會透過與該使用者互動過的餐廳節點來推敲其個人喜好。

在完成餐廳節點的嵌入後，第二步則是建構每位使用者節點的初始嵌入向量 $e_u^{(0)}$ 。此過程需要依賴與使用者節點 u 相鄰的餐廳節點 i 的嵌入向量 $e_i^{(0)}$ 。第0層的嵌入構造方式如下：

$$e_u^{(0)} = \sigma \left(\sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} e_i^{(0)} \right), \quad (2)$$

其中 N_u 表示使用者節點 u 的鄰居集合，而 N_i 則代表餐廳節點 i 的鄰居集合； $\sigma(\cdot)$ 是激活函數，選擇使用雙曲正切函數(tanh)。透過這一層的加權平均，可以有效地融合使用者與其鄰近餐廳節點的特徵信息，以更準確地反映該使用者的行為特徵。

2.2. Propagation Layers

完成初始嵌入後，下一步是計算每個使用者節點 u 與其相鄰的餐廳節點 i 之間的注意力分數 $\rho(u, i)$ ，藉此衡量不同鄰居節點的重要性。該分數計算方式如下：

$$\rho(u, i) = Q^T \sigma(W(e_u^{(0)} || (e_i^{(0)} + b))), \quad (3)$$

其中 $W \in \mathbb{R}^{2d \times d}$ 、 $Q \in \mathbb{R}^{1 \times d}$ 、 $b \in \mathbb{R}^{1 \times d}$ ，這三個參數分別為注意力機制中的權重矩陣與偏置項。此處的 $\sigma(\cdot)$ 同樣為雙曲正切函數(tanh)，而 $||$ 則表示向量的串接操作。 $\rho(u, i)$ 的分數越高，代表使用者 u 與餐廳 i 之間的關聯性越強。因此，透過該分數可以量化每個餐廳節點對於預測使用者偏好的貢獻度。

為將注意力分數限制在 $[0, 1]$ 的範圍內，利用Softmax函數對這些分數進行範圍歸一化處理，得到最終的注意力值 $\alpha(u, i)$ 。公式如下：

$$\alpha(u, i) = \frac{\exp(\rho(u, i))}{\left(\sum_{j \in N_u} \exp(\rho(u, j)) \right)^\beta}. \quad (4)$$

其中 $\alpha(u, i)$ 代表了使用者節點 u 與餐廳節點 i 之間的最終注意力值。透過 $\alpha(u, i)$ ，每個使用者節點均能依據其鄰居的特徵，綜合考量注意力分數來完成向量更新。

在初始嵌入狀態完成後，為進行更深層的特徵傳播，使用者節點 u 的嵌入向量 $e_u^{(k)}$ 將在每一層根據其鄰居節點的資訊進行更新。使用者的嵌入向量 $e_u^{(k)}$ 的更新方式會依據先前計算出的注意力權重 $\alpha(u, i)$ ，並利用相鄰餐廳節點 i 在上一層的嵌入向量 $e_i^{(k-1)}$ 進行加法聚合(Add Aggregation)。此聚合方式可以有效融合鄰居的特徵信息，更新的計算公式如下：

$$e_u^{(k)} = \sigma \left(\sum_{i \in N_u} \alpha(u, i) e_i^{(k-1)} \right), \quad (5)$$

其中， $\sigma(\cdot)$ 為雙曲正切函數(tanh)， N_u 表示使用者節點 u 的鄰居集合。透過加權聚合，使用者節點可以自適應地調整對不同鄰居特徵的重視程度，進一步增強模型對於不同使用者偏好的捕捉能力。

同樣地，初始狀態後的餐廳節點的嵌入向量 $e_i^{(k)}$ 也會根據其相鄰使用者節點的嵌入進行更新。這個更新方式與公式(2)類似，餐廳節點 i 的嵌入向量會根據其相鄰的使用者節點 u 的嵌入向量 $e_u^{(k)}$ 進行加權平均，

計算方式如下：

$$e_i^{(k)} = \sigma \left(\sum_{u \in N_i} \frac{1}{\sqrt{|N_u||N_i|}} e_u^{(k)} \right), \quad (6)$$

其中， N_u 和 N_i 分別表示使用者節點 u 和餐廳節點 i 的鄰居集合，並透過加權平均來控制每個鄰居對於嵌入向量的貢獻。此公式 (6) 與公式 (2) 中的加權項 $\frac{1}{\sqrt{|N_u||N_i|}}$ 可以平衡不同鄰居數量對嵌入更新的影響，從而避免由於鄰居數量不均而引起的偏差。藉由此加權項，每個餐廳節點的嵌入向量都將根據與其相鄰使用者的特徵進行有效更新，從而更準確地反映餐廳與使用者間的潛在關係。

2.3. Prediction Layers

在完成每一層使用者節點的嵌入向量 $e_u^{(k)}$ 和餐廳節點的嵌入向量 $e_i^{(k)}$ 的計算後，接下來的步驟是將這些嵌入向量進行聚合，以獲得最終的嵌入表達，在本研究將所有層的使用者嵌入向量和餐廳嵌入向量分別相加，計算公式如下：

$$e_u^* = \sum_{k=1}^L e_u^{(k)}, \quad e_i^* = \sum_{k=1}^L e_i^{(k)}, \quad (7)$$

其中 L 表示嵌入層的總數。這樣計算得到的 e_u^* 和 e_i^* 分別代表了使用者節點和餐廳節點的最終嵌入向量，這兩個向量綜合了多層的信息，能夠更全面地捕捉使用者與餐廳之間的潛在關聯。為了進一步評估使用者 u 和餐廳 i 之間的關聯性，將這兩個最終嵌入向量進行內積操作，為了將內積值限制在 0 到 1 的範圍內，引入了 S 型函數 (sigmoid function)，計算公式如下：

$$y_{ui} = f(u, i) = \sigma(e_u^{T*} e_i^*), \quad (8)$$

其中， y_{ui} 表示模型對於使用者 u 和餐廳 i 之間關聯性的預測結果。S 型函數 $\sigma(\cdot)$ 將內積的結果映射到 $[0, 1]$ 的範圍，使得模型的輸出可以被解釋為一個概率值，表達使用者 u 對餐廳 i 的喜好程度。

2.4. Model Optimization

待完成

3. Experiment

3.1. Data Preprocessing

本研究的資料前處理架構如圖 4 所示，利用爬蟲技術從 Foodpanda 和 Google Maps 獲取餐廳資料。每間餐廳的資料涵蓋了餐廳名稱、餐廳類型、評分以及顧客的評論。由於每家店通常會累積大量的評論，這些評論內容對於描述店家的品質、服務及顧客的體驗具有高度相關性。若將所有評論合併後再丟入大規模語言模型 (LLM) 進行分析，可能會因評論主題多樣性導

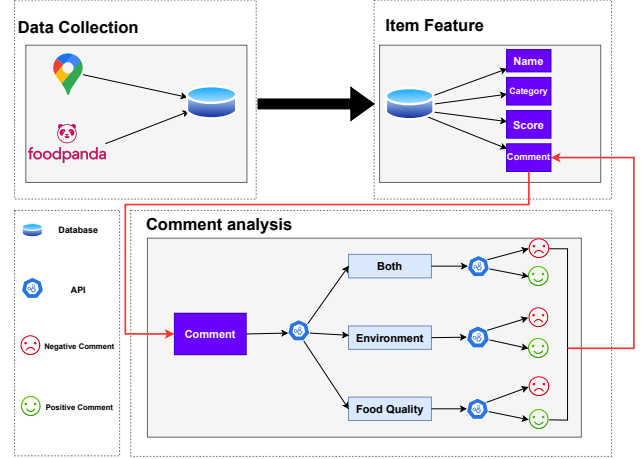


Figure 4. 資料前處理架構圖

致訊息混雜，進而影響模型的微調 (Fine Tuning) 效果。當不同主題的評論一起進行訓練時，模型可能難以識別每個特徵的意圖，進而影響結果的精準度。

為提升模型的針對性，本研究首先將評論依主題分為三類：餐點品質、店內環境氣氛，以及兩者皆有的綜合評價。針對這三類評論，我們分別對每一類進行獨立的 LLM 微調。這種獨立的微調方法有助於每個模型專注於該主題的特徵，避免不同主題之間的訊息干擾。對於餐點品質類評論，模型學習專注於食物的口感、質量及呈現；對於店內環境氣氛的評論，模型則聚焦在用餐氛圍、環境設計及舒適度等描述上；而綜合評價類別則幫助模型掌握顧客的整體體驗。通過分開微調，每個模型能精確捕捉該主題的特徵表現，使得每一類評論的處理結果更具針對性。

完成初步主題分類與獨立微調後，針對每一類主題中的評論再進行情感分析，以判斷其情感取向為正面或負面。這一分層且分主題的處理架構，能夠生成更為精確的特徵向量，使得每條評論的語意信息充分表達，進一步提升了推薦系統的精準度和可靠性。

References

- [1] Y. Xiao, C. Li, and V. Liu, "Dfm-gcn: A multi-task learning recommendation based on a deep graph neural network," *Mathematics*, vol. 10, no. 5, 2022. 1
- [2] Y. Zhang, Y. Zhang, D. Yan, Q. He, and Y. Yang, "Nie-gcn: Neighbor item embedding-aware graph convolutional network for recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 5, pp. 2810–2821, 2024. 1, 2