Matrix Factorization Model

Matrix Factorization Model 又稱為矩陣分解法，為推薦系統中協同過濾策略的經典算法之一，其主要想法是基於使用者和商品的互動評分向量化使用者和商品的潛在因素，越高的評分代表該商品應該被推薦。模型可以由矩陣作為呈現，矩陣的丨列代表一個使用者，矩陣的丨行代表一個商品，矩陣的元素代表該使用者和商品的互動評分。模型的資料來源有兩種，一種為品質較好的明確回饋（explicit feedback），包含使用者明確給出針對商品的評分，像是 YouTube 的按讚機制、Google Map 商家的評分機制等，此種資料雖然明確卻不容易取得，數量也會遠小於使用者和商品的數量，因此容易形成稀疏矩陣（sparse matrix），另一種資料為隱性回饋（implicit feedback），為間接可以代表使用者喜好的使用者行為包含使用者購買紀錄、瀏覽紀錄、搜尋紀錄等，通常會以這些行為存在與否作為紀錄，容易形成密集矩陣（dense matrix）。

Matrix Factorization Model 會將使用者和商品映射到一個共同的向量空間並有著相同的維度，並將向量間的內積用於模擬使用者和商品間互動評分。其定義每一個商品 i 對應著向量 $q_i \in \mathbb{R}^f$，向量每個維度意味著商品所擁有的潛在因素（Latent Factor）是多或少（在向量空間中是正或負），每一個使用者 u 對應著向量 $p_u \in \mathbb{R}^f$，其中的每個維度意味著使用者基於每個潛在因素對於商品的喜好程度高或低，而使用者 u 和商品 i 的互動評分 r_{ui} 即 q_i 和 p_u 的內積 $q_i^T p_u$，意味著使用者對於商品各項因素的總體喜好程度。模型的最大挑戰在於求得使用者和商品的向量，計算出所有的向量，即可以推測出使用者和商品的互動評分基於此公式 $\hat{r_{ui}} = q_i^T p_u$。

此種將矩陣拆解為使用者矩陣和商品矩陣的作法類似 奇異值分解（singular value decomposition），但由於 SVD 並不能有效針對稀疏矩陣進行分解，而替代的代入法（Imputation）如代入平均值等會導致增加相當大量的資料，並且不容易確保可信度。因此 Matrix Factorization 的做法是直接針對互動評分進行建模，並且透過正則化（regularization）來避免過擬合。最終的模型公式如下：

$$\mathcal{L} = min_{q^*,p^*} \sum_{i,u \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

其中 κ 是使用者和商品的互動評分資料集，λ 是正則化參數，q_i 和 p_u 是使用者和商品的向量，$q_i^T p_u$ 是使用者和商品的互動評分。學習時會以過去的互動評分為目標，因此需要透過正則化對向量中過大的值進行扣分來避免過擬合。

學習演算法

隨機梯度下降（Stochastic Gradient Decent）

Simon Funk 讓透過隨機梯度下降（Stochastic Gradient Decent, SGD）求得模型的 q_i, p_u 成為目前熱門的最佳化演算法，該做法透過取樣（sample）大量的訓練資料的互動評分（r_{ui}）計算該數值和模型預測的分數誤差：

$$e_{ui} = r_{ui} - q_i^T p_u$$

透過誤差修改參數往模型公式的梯度相反方向更新，並透過 γ 調整更新的幅度：

$$q_i \leftarrow q_i + \gamma(e_{ui} p_u - \lambda q_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui} q_i - \lambda p_u)$$

經過多次的更新，誤差會達到收斂，即代表訓練完成。

模型變形

加入偏差（bias）

在觀測的互動評分中，除了潛在因素會影響分數以外，也會受到資料品質和數量不平衡造成偏差，導致模型的預測會有偏差，這些偏差對於模型學習的參數應該是獨立的，因此應該另外加入偏差參數，以便模型可以接受這些偏差。偏差的建立會來自於使用者或是商品，衡量一位使用者對於商品的評分會是源於所有商品的平均評分加上商品自身相較平均的評分差距及使用者本身的標準不同所造成的評分差距，例如所有的電影評分是 3.5 分而復仇者聯盟較平均高過 1 分並且 Joe 這位使用者相較嚴格多扣了 1.3 分則造成的評分即為 3.5 + 1 - 1.3 。因此實際互動評分的偏差會來自總體平均的評分 + 商品與平均評分的差距 + 使用者對於商品的評分與平均評分的差距：

$$b_{ui} = \mu + b_i + b_u$$

而實際的評分為：

$$\hat{r_{ui}} = \mu + b_i + b_u + q_i^T p_u$$

因此加入偏差的模型公式為會修改如下：

$$\mathcal{L} = min_{q^*,p^*} \sum_{i,u \in \kappa} (r_{ui} - \mu - b_i - b_u - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_i^2 + b_u^2)$$

加入額外資訊

針對明確回饋資料不多的冷啟動問題（Cold Start Problem），Matrix Factorization 模型會需要加入隱性回饋資料，如使用者的購買紀錄、網頁瀏覽紀錄等作為推薦依據，這些資料為了簡化複雜會以布林型態作為判斷存在或不存在該行為，針對一個使用者 u 的有所有隱性回饋的商品形成一個集合 $N(u)$，模型透過這些隱性回饋建立使用者檔案（user profile），因此針對這些有隱性回饋的商品，模型需要加入一些商品因素（item factors）$x_i \in \mathbb{R}^f$ 以表示這些商品與使用者隱性回饋的關係，一個使用者對於在 $N(u)$ 集合中的商品的偏好便會被表示為向量：

$$\sum_{i \in N(u)} x_i$$

正則化的公式如下：

$$\|N(u)\|^{-0.5} \sum_{i \in N(u)} x_i$$

除此之外，使用者的屬性也是一項可以列入當作依據的資訊，同樣考慮一個使用者屬性存在與否會以布林來表示，一個使用者 u 所擁有的屬性集合為 $A(u)$，如性別、年齡、職業等，而每個屬性有著屬性因素（attribute factors）可以以向量表示 $y_a \in \mathbb{R}^f$，因此針對一位使用者的屬性可以表示如下：

$$\sum_{a \in A(u)} y_a$$

而預測的互動評分可以修正如下：

$$\hat{r_{ui}} = \mu + b_i + b_u + q_i^T [p_u + \sum_{a \in A(u)} y_a + \|N(u)\|^{-0.5} \sum_{i \in N(u)} x_i]$$

除了使用者以外，對於商品端缺少的資訊也可以透過這樣的方法處理。

https://note.josix.tw/data-science/recsys/matrix%20factorization%20model/