

國立臺灣大學管理學院資訊管理研究所

碩士論文

Department of Information Management

College of Management

National Taiwan University

Master Thesis



以 DAE 向量雜訊移除為基礎之新進使用者冷啟動推薦

A Cold Start Recommendation Method for New Users

Based on DAE Vector Noise Removal

吳承翰

Cheng-Han Wu

指導教授：陳建錦 博士

Advisor: Chien-Chin Chen, Ph.D.

中華民國 109 年 6 月

June 2020

國立臺灣大學碩士學位論文
口試委員會審定書

以 DAE 向量雜訊移除為基礎之新進使用者冷啟動推薦

A Cold Start Recommendation Method for New Users
Based on DAE Vector Noise Removal

本論文係吳承翰君（學號 R07725029）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 109 年 6 月 29 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

陳建川


陳月亭

王長弘

所 長：

魏志平

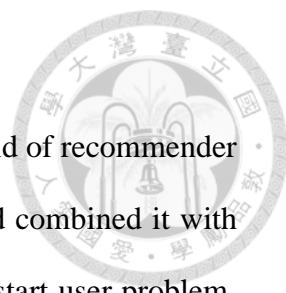
摘要



本篇論文主要專注於解決推薦系統中常見的冷啟動使用者問題，我們提出了一種稱為「使用者返老還童」的機制，並且結合深度學習模型 Denoising Autoencoder，以此來解決冷啟動使用者問題。首先，「使用者返老還童」機制會為不同使用者族群選出代表該群使用者的商品，我們稱這些選出的商品為「具代表性商品」，接著透過將使用者對於所有商品的評分向量隨機覆蓋大部分的不具代表性商品維度為 0 分，以及覆蓋小部分的具代表性商品維度為 0 分後，以此來模擬使用者冷啟動的狀態。當我們得到受「使用者返老還童」機制還原的冷啟動使用者後，接著我們可以為每一群使用者訓練一個深度學習模型 Denoising Autoencoder。Denoising Autoencoder 具備將受到雜訊干擾的輸入向量還原回未受雜訊干擾的向量，此種特性會有助於將冷啟動狀態的使用者向量還原回一般狀態的使用者。因此，當 Denoising Autoencoder 模型訓練完成後，模型便有能力把輸入的冷啟動使用者順利還原成富有評分資訊的使用者狀態，並且透過預測出的使用者評分向量來進行推薦。

關鍵字：深度學習、推薦系統、使用者冷啟動問題、自動去噪編碼器、具代表性商品選取

ABSTRACT



In our thesis, we focus on the cold start user problem in the field of recommender systems. We propose a mechanism called “User Rejuvenation” and combined it with the deep learning model Denoising Autoencoder to solve the cold start user problem. The “User Rejuvenation” first choose representative items for each group of users, after that we randomly set the dimensions corresponding to representative items in user vector to zero score with lower probability, and we randomly set the dimensions corresponding to non-representative items in user vector to zero score with higher probability. The main purpose for “User Rejuvenation” is to turn the non-cold start user vectors back to cold start user vectors for each group of users. After getting the group-specific cold start user vectors generated from “User Rejuvenation” mechanism, we can use them to train a Denoising Autoencoder model for the user group. When the training process is complete, the model will have capacity for restoring the cold start user vectors to non-cold start user vectors, and the recommendation is made through the predicted non-cold start user vectors.

Keywords : Deep Learning 、 Recommendation System 、 User Cold Start Problem 、 Denoising Autoencoder 、 Representative items selection

目錄



中文摘要/英文摘要	i、ii
目錄	iii
圖目錄	iv
表目錄	v
第一章 研究動機	1
1.1 推薦系統的重要性	1
1.2 冷啟動問題的難度與重要性	2
1.3 運用深度學習模型解決冷啟動問題	3
1.4 我們與現行深度學習方法的不同之處	5
第二章 文獻回顧	6
2.1 以深度學習為基礎之推薦系統	7
2.2 運用深度學習之冷啟動推薦	14
2.3 代表性商品探勘	17
第三章 論文方法	20
3.1 DAE 基本架構	21
3.2 採用 DAE 架構原因與初步方法	23
3.3 使用者返老還童階段	24
3.4 冷啟動使用者推薦	29
第四章 論文方法實驗與分析	29
4.1 實驗資料集與評估指標	29
4.2 訓練流程與實驗參數設定	31
4.3 所有模型效能比較	35
第五章 結論	37
第六章 參考文獻整理	38

圖目錄



圖一、文獻[3]的神經網路架構圖.....	8
圖二、文獻[6]的神經網路架構圖.....	9
圖三、文獻[2]的神經網路架構圖.....	10
圖四、文獻[2]的神經網路架構圖.....	11
圖五、文獻[17]的神經網路架構圖.....	13
圖六、文獻[19]的神經網路架構圖.....	14
圖七、文獻[15]的神經網路架構圖.....	15
圖八、文獻[13]的神經網路架構圖.....	15
圖九、文獻[13]的神經網路架構圖.....	16
圖十、文獻[8]的矩陣分解法.....	17
圖十一、文獻[12]的方法架構圖.....	18
圖十二、文獻[12]的矩陣分解圖.....	19
圖十三、Denoising Autoencoder 模型架構圖.....	22
圖十四、Denoising Autoencoder 訓練流程圖.....	24
圖十五、Denoising Autoencoder 測試流程圖.....	24
圖十六、「選出具代表性商品」流程圖.....	25
圖十七、DPC Decision Diagram.....	27
圖十八、masked RMSE 示意圖.....	31
圖十九、訓練誤差與訓練次數圖.....	32
圖二十、覆蓋 20%測試使用者喜歡的評分示意圖.....	33
圖二十一、Autoencoder 模型中間層實驗圖.....	34
圖二十二、DPC Decision Diagram 實驗圖.....	35
圖二十三、基準模型與我們方法的分群實驗圖.....	35

表目錄

表一、MovieLens 1M 統計資料表.....	29
表二、訓練與測試使用者統計表.	31
表三、所有模型效能比較.....	36



1 研究動機：

1.1 推薦系統的重要性

隨著網際網路中的資訊量呈現爆炸性的增長，網路上各式服務平台中令人目不暇的商品或服務選擇，早已令使用者處在無從應付的窘境當中，而這時推薦系統儼然已經成為能有效替使用者快速且正確地過濾掉大量不必要的資訊，並幫助使用者聚焦於他們所關注資訊的一種不可或缺存在。推薦系統除了能讓使用者更快速且正確的獲取符合他們需求的商品，也同時能通過精準的推薦來提高使用者的滿意度與增加使用者對於平台的黏著度，進而為服務平台供應商帶來可觀的效益。因此，將推薦系統應用在實務上，已經是現今許多電子商務平台（Amazon）或影音串流平台（Netflix），幫助企業提升使用者滿意度，以及增加平台收益的強大幕後推手。以 Netflix 為例，他們的首席產品官 Hunt 表示，Netflix 中有 80% 以上的電影觀看都是通過推薦生成的。並且如同 Netflix 的 Gomez-Urbe 和 Hunt 在[5]中所提到的，『如果能藉由提升 Netflix 推薦系統的推薦品質，以此增加用戶在平台上的黏著度，那 Netflix 每年可以來減少因客戶流失造成的 10 億美元損失』。透過維持客忠誠度來減少損失的觀點與 Forbes 專欄作家 Larry Myler 所提出的觀點不謀而合，『對於零售商來說，保留既有的客戶所要花費的成本往往會比開發新客戶的成本還要低上許多，而且擁有穩定的客群能使零售商獲取更高的顧客終生價值，以及使企業的營收更容易預測』。同樣的情況下，對於 Netflix 平台而言，減少客戶的流失會比去開發新客戶容易，是因為新客戶往往會在瀏覽少於 20 部影片推薦的情況之下，便決定是否加入平台，如此一來 Netflix 只能在極少的資訊與時間中，預測新客戶的喜好。相較於從既有客戶過往的評分資訊中，推薦符合他們期望的影片來說，提升推薦系統的品質便顯得實際許多。從 Netflix 的案例中，我們便可體會，一個設計優良的推薦系統，能同時為使用者與服務平台供應商帶來效益。

1.2 冷啟動問題的難度與重要性

一般而言，在建構推薦系統時，有一類經常被使用的方法是「協同式過濾推薦」(collaborative filtering)，協同式過濾推薦的概念為：利用與欲推薦的目標使用者擁有類似商品評分經驗之群體的偏好，來預測該目標使用者也可能會偏好的商品。舉個簡單的例子：在 Amazon 網路書店，當顧客選購了一本書籍後，網站的下方會顯示「購買此書的顧客，同時也購買了什麼商品」，這便是一個協同式過濾推薦的經典案例。

不過協同式過濾推薦存在一個問題為，當要對新進使用者推薦商品時，由於只有些許新進使用者過去對於商品的評分資訊，因此我們便很難找出與該新進使用者擁有類似經驗的群體，並以此為參考進行推薦。這樣的問題被稱作「新進使用者冷啟動問題」(new user cold start problem)。在過去有各式各樣的方法被提出來解決此問題，其中一種方法為「基於內容推薦法」(content-based method) [16]，透過結合使用者的額外資訊，如使用者簡歷、年齡或性別等，以這些額外的資料來建構新進使用者的向量表示法(vector representation)，替代原本無評分資訊的使用者評分向量，再藉由向量相似度計算(如 cosine similarity)來找出與新進使用者向量高度相似的使用者所喜歡的商品來完成推薦。不過基於內容推薦法必需建立在能夠獲取使用者額外資料的前提之下，在實務上我們並不一定能夠獲取到這些資訊。因此，就有了另外一種解決新進使用者冷啟動問題的方法，稱為「基於代表商品推薦法」(representative-based method) [8]、[4]、[12]，該方法的核心概念是去找出推薦系統中“具有代表性的商品”(representative items)，並且若能取得新進使用者對於部分具代表性商品的評分，則我們便可以透過具代表性的商品與一般商品之間的關係，間接地推論出新進使用者與其他一般商品之間的關係，然後進行推薦。

若我們可以提出一種能有效地解決新進使用者冷啟動問題的架構，則便可以推薦符合新進使用者需求的商品，並提高他們在平台上的黏著度與跟平台互動的次數，如此一來能使新進使用者脫離冷啟動的狀態，而得到更精確的商品推薦。最後，形成使用者與平台之間正向的互動循環。

1.3 運用深度學習模型解決冷啟動問題

近年來，隨著深度學習 (deep learning) 在文字、圖像、語音等各個領域中的應用逐漸成熟，並且在許多的任務上有著非常亮眼，甚至是超越人類的表現。也因此，不論是在業界或是在學術界，都有越來越多人開始研究該如何將深度學習的技術應用在推薦系統上，我們把這種應用稱之為「以深度學習為基礎之推薦方法」(deep learning based recommendation method)。在業界，Covington 等人提出使用深度學習的模型，來建構 YouTube 影片的推薦系統 [2]。該方法先運用一組具有三層隱藏層(hidden layers) 的 Deep Neuron Network (DNN) 模型，從輸入的用戶瀏覽歷史、搜索歷史、人口統計學訊息等資訊中，找出用戶可能會喜歡的候選影片集合，接著再將使用者與候選影片之間相關的資訊，例如：用戶瀏覽該頻道的次數、用戶最近一次瀏覽該頻道距離現在的時間等。以另外一個具備三層隱藏層的 DNN 模型來學習出用戶對這些候選影片的喜好排序。任職於 Google 的 Cheng 等人也使用深度學習的方法來建構 Google Play 的 APP 推薦 [1]。該方法將 Google Play 中使用者的連續型資料(例如：使用者年齡、使用者已安裝的 APP 數量等)和使用者的離散型資料(例如：使用者使用的裝置種類、使用者已經安裝的 APP 等)，當作 DNN 模型的輸入，產生使用者的特徵向量後，再同時考慮使用者特徵向量與 APP 的相關資訊來決定是否推薦某 APP 給某個使用者。在學術界，有鑑於使用深度學習方法結合推薦系統的研究文章大量且相繼的被提出與刊登，ACM RecSys 研討會便於 2016 年起將以深度學習為基礎的推薦系統

納入會議主題之一，以此促進深度學習技術在推薦系統上的發展與交流。見到如此多的業界技術人員或學者，趨之若鶩的投入這項新興研究議題，就能得知發展建構在深度學習基礎之上的推薦系統，是一個無法避免的趨勢。因此，我們的研究便是希望能立足於這些先輩們的研究基礎之上，建構一套有效的深度學習模型來有效解決推薦系統中新進使用者冷啟動問題。

在經過對於以深度學習為基礎之推薦方法全面的調查以後，我們發現 Denoising Autoencoder (DAE)[14]這種深度學習模型的運作概念，非常適合用於模擬推薦系統中的冷啟動問題。DAE 的運作方式若以圖像為例：會先把圖片中的部份像素隨機設定為 0，以此為原始的圖片產生雜訊 (noise)，之後將產生雜訊的圖片輸入 encoder-decoder 的深度學習模型架構中。在 encoder 階段，會將輸入的圖片向量經過一層以上的隱藏層來降低圖片的維度；而在 decoder 階段，則是通過一層以上的隱藏層來將圖片還原回原始輸入的圖片維度。DAE 模型的最終訓練目標便是希望能將受雜訊干擾的圖片在通過 encoder-decoder 架構後，能還原成起初未經雜訊干擾的圖片。也因此 DAE 在影像處理的領域中，常常會被用來去除圖像資料的雜訊，或是將圖像資料做降維。

而 DAE 與我們想要解決的冷啟動問題之間的關聯在於，我們可以將原始富有資訊的使用者輸入向量想像為評分過很多商品的一般使用者，而產生雜訊的過程，即是在模擬將一般使用者逐漸倒退回剛開始進入平台的新進使用者。但是，DAE 產生雜訊的方法通常都是應用在圖像或是語音這方面的資料上，並不一定直接適用於推薦系統的應用情境。因此，我們提出了一種為推薦系統的輸入向量產生雜訊的方法。以下我們便著重在於如何推薦商品給冷啟動的使用者這一類的冷啟動問題，以及簡單的介紹一下我們所提出的模型架構，該架構主要分為兩大部分：

1. User Rejuvenation Stage

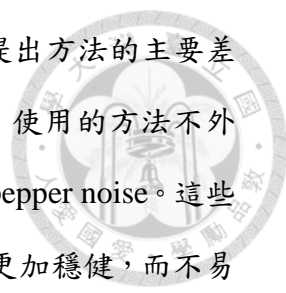
在此階段我們參照 DAE 的運作概念，將輸入模型的使用者向量，經過我們產生雜訊的方法後，把輸入的使用者向量模擬回剛進入平台的使用者新進狀態。其中為了更貼近使用者冷啟動問題，我們改稱 DAE 產生雜訊的過程為「使用者返老還童階段」(user rejuvenation stage)。在使用者返老還童階段中，會先將使用者進行分組，再分別挑選出代表不同組別使用者的 k 個具代表性的商品，而挑選出的具代表性商品必須具備「被大多數的使用者所評分過」(popularity) 與「使用者們對商品的評分分布需夠分散」(rating entropy) 這兩種特性。選出具代表性的商品後，我們會傾向保留使用者向量中具代表性商品的資訊，而其餘一般商品的資訊則給予較大的機率被丟棄。

2. Encode Decode Stage :

一旦能取得使用者返老還童階段用來模擬冷啟動狀態的使用者向量，接下來將使用者向量通過 encoder-decoder 的架構，將受復原為新進使用者狀態的輸入向量還原成原始輸入，便是類似於我們訓練出一個能將冷啟動使用者還原成一般使用者狀態的模型。

1.4 我們與現行深度學習方法的不同處

目前也有其他的研究將 DAE 的模型應用在推薦系統中，像是[17]提出了 Collaborative Denoising Autoencoder (CDAE) 的模型，把使用者對各個商品的偏好以 0、1 值來表示(若為 1：代表使用者喜歡此商品；若為 0：代表使用者尚未看過此商品)。如此一來，每一個商品會對應到使用者向量 y_u 中的一個維度。接著 CDAE 接收 y_u 和要預測的目標使用者的 ID 當作模型輸入，中間通過 encoder-decoder 架構，在最後 CDAE 模型的輸出向量 \hat{y}_u 的維度會與輸入向量 y_u 的維度相同，而 \hat{y}_u 的每一個維度值就是代表其所對應到的商品的預測推薦分數。而以下我們



說明現行使用 DAE 架構的推薦系統論文，與我們所提出方法的主要差別：現行的方法 [17]，在對輸入向量做雜訊干擾時，使用的方法不外乎是加入 Gaussian noise、masking noise 或是 salt-and-pepper noise。這些雜訊干擾方式雖然也能使訓練出來的模型，在預測上更加穩健，而不易受到輸入雜訊的干擾。不過對於在推薦系統實務上的可解釋性就沒有那麼的直觀，因為這些做法是將商品的評分資訊隨機的覆蓋為 0 分。而我們的方法能透過選出”具代表性的商品”來保留輸入向量中最為重要的商品評分資訊，然後將產生雜訊後的輸入，模擬成冷啟動的狀態。因此，我們的方法在實務上的可解釋性更為直觀。

2 文獻回顧：

近幾年以深度學習為基礎的推薦方法之所以大量的興起與被研究，主要是由於以深度學習為基礎的推薦系統有以下兩個優點[18]：(i) 可以模擬更複雜的非線性關係。以深度學習為基礎的方法能透過各種不同的非線性啟動函式 (Activation Function，如 Sigmoid、Tanh 與 ReLU)，來模擬推薦系統中使用者與商品之間複雜且非線性的關係。相較於非以深度學習為基礎的方法，像是 Matrix Factorization 就只能模擬使用者與商品之間簡單的線性組合關係。因此，以深度學習為基礎的方法顯然會更貼近於現實的情況。(ii) 可有效擷取資料特徵。以深度學習為基礎的方法能非常有效的擷取任何輸入型態的資訊，像是從使用者或商品的敘述性文字資料，亦或者是從商品圖片、商品廣告影片中，擷取出額外附帶的資訊。擁有這項特性便能節省以往使用非以深度學習為基礎的方法，需要手動去做特徵工程 (Feature Engineering) 的大量時間成本，因為以深度學習為基礎的方法能以監督或非監督式學習的方法，自動從原始輸入中提取重要的資訊。這項特性也能使我們在做推薦時，結合各種不同型態的內容資訊，使推薦結果更加精準。

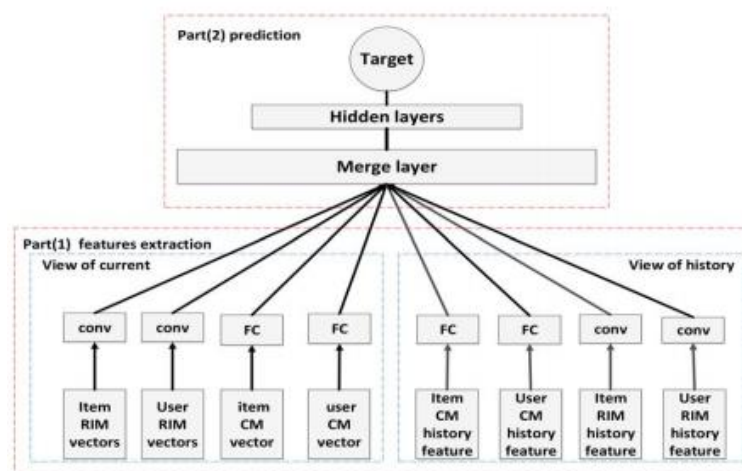
有鑑於以上的兩大優點，我們便針對現行以深度學習為基礎的推薦方法做調查與整理。另外，如何找出代表性商品也是我們的研究重點，所以我們也會在本章末整理一些找出代表性商品相關的論文。最後將所有相關的論文依照以下形式分類：

2.1 以深度學習為基礎之推薦系統文獻探討

在此分類下的深度學習推薦方法主要分為兩個階段，在第一階段必須決定要以怎麼樣的型式來表示輸入的使用者向量和商品向量。在第二階段則必須設計適合的類神經網路架構，能使最終預測的評分值與真實評分之間的誤差越小越好。就像是在[3]中，提出了兩個互補的方法來產生使用者和商品的 embedding vector，一個稱作 Constraint Model (CM)，其做法是利用找出對商品 A 和商品 B 皆評分過且評相同分數的使用者集合大小，來描述商品 A、B 之間的關聯，接著便可以求出所有商品之間的關聯矩陣 R，把商品之間的關聯矩陣 R 做矩陣分解後得到兩個子矩陣 \tilde{E} 與 \hat{E} ，分別取出子矩陣 \tilde{E} 的第 i 列向量 \tilde{e}_i 與子矩陣 \hat{E} 的第 i 行向量 \hat{e}_i ，將 \tilde{e}_i 和 \hat{e}_i 串接在一起形成商品 i 的 embedding vector。而另一個產生使用者和商品的 embedding vector 的做法為 Rating Independent Model (RIM)，其做法則是利用找出對商品 A 和商品 B 皆評過分但是評不同分數的使用者集合大小，來描述商品 A、B 之間的關聯，接著便可以求出所有商品之間的關聯矩陣 R'，而後續求出某個商品的 embedding 做法則和 CM 相同，此做法則是會考慮到，商品在不同評分時會有不同的 embedding vector。[3]最後結合了以上兩個方法，分別產生出 item RIM vector、user RIM vector、item CM vector 和 user CM vector，這四個向量被稱為「View of current」。另外，還會考慮四個稱作「View of history」的向量。在「View of history」中，會用某個使用者過去所評分過的商品集合中的所有商品向量做平均，來當作該使用者的向量(其中的 item 向量可以用 CM 或 RIM 來表示)；而某個商品的 history vector 會用過去曾評分過該商品的使

用者集合中的所有使用者向量做平均，來當作該商品的向量(其中的 user 向量可以用 CM 或 RIM 來表示)。在得到上述的八個向量之後，會分別將 CM embedding 和 RIM embedding 通過 DNN 和 CNN，再將所有特徵向量合併，輸入最終的深度網路預測某個使用者對某個商品的評分。訓練完模型後，假設想預測使用者 A 對商品 B 的評分，通過 CM、RIM 可以得出 View of current 中的四個輸入向量。然後將使用者 A 用過去使用者 A 所評分過的所有商品向量的平均來表示(item 向量用 CM 或 RIM 來表示)；將商品 B 用過去曾評分過商品 B 的所有使用者向量的平均來表示(user 向量用 CM 或 RIM 來表示)。因此，我們可以得到 View of history 的四個向量。

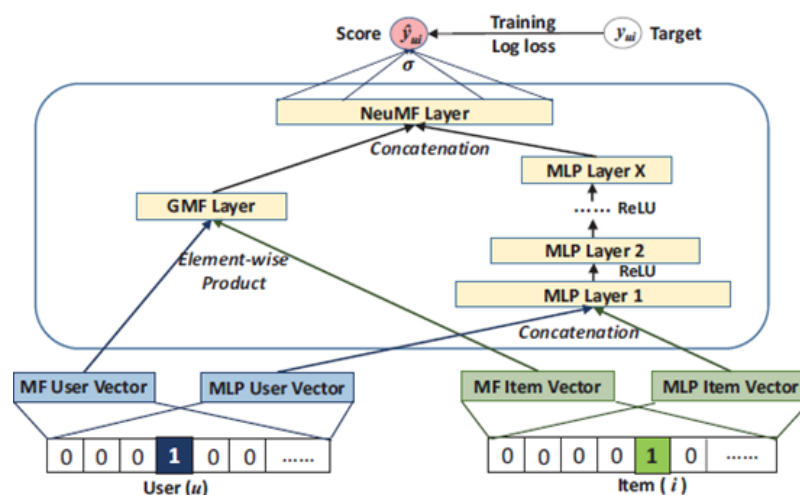
最後，把總共的八個向量輸入訓練後的模型，輸出的分數即代表，使用者 A 對商品 B 的預測分數。我們將分數高於某個門檻值的商品推薦給使用者。以下為[3]的神經網路架構圖：



圖一、文獻[3]的神經網路架構圖

再另一篇論文[6]中，推薦系統要預測的最終結果就不是使用者評分，而是使用者對某商品是否感興趣(若預測結果為 1：代表使用者感興趣，0：則表示使用者尚未與該商品互動過)。[6]在第一階段產生 user 和 item embedding 時，會將每一個 user 和 item 的 one-hot encoding 表示法，分別通過 4 組一層的 Embedding Layer，產生 MF User Vector、MLP User Vector、MF Item Vector

和 MLP Item Vector，然後再分別經過 Generalized Matrix Factorization Layer (GMF)和 Multi-Layer Perceptron (MLP) Layer，來學習 user 跟 item 之間線性和非線性關係的 embedding。其中在 GMF Layer 裡進行的運算是將 user 和 item 向量中對應的維度相乘(element-wise product)，目的希望能模擬 Matrix Factorization 在捕捉向量之間線性關係的運作，而在 MLP Layer 裡進行的運算則是，將 user 和 item 向量合併後經過一連串的非線性轉換，以此捕捉它們之間非線性的關係。最後再同時考慮 GMF 與 MLP 的 embedding 來預測使用者的喜好。訓練完模型後，若要知道 user A 是否對 item B 感興趣，只要將 user A 和 item B 的 ID 轉換成 one hot encoding 後輸入模型，我們便可以透過模型輸出的數值得知 user A 對 item B 的喜好分數。以下為[6]的網路架構圖：



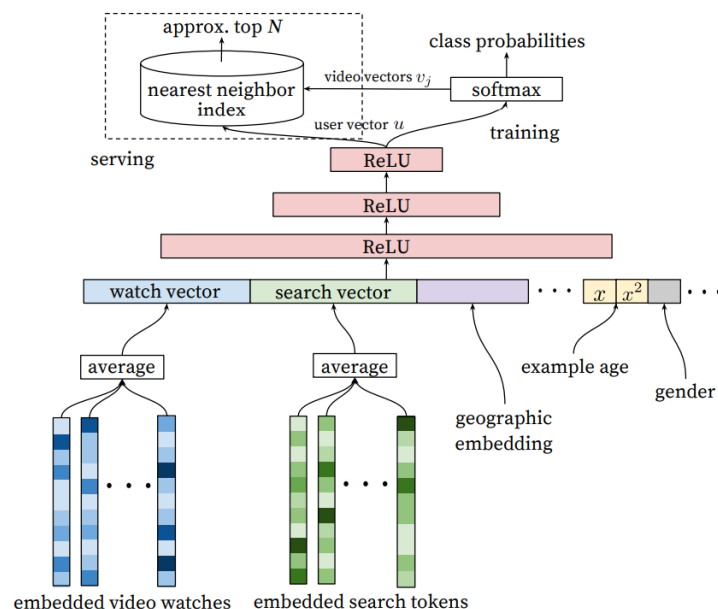
圖二、文獻[6]的神經網路架構圖

[2]主要分成兩個階段：candidate generation 階段和 ranking 階段，此兩個階段各自使用一個類神經網路模型。

在 candidate generation 階段中，類神經網路的主要輸入特徵有「用戶觀看影片的歷史紀錄」、「用戶的歷史搜尋紀錄」、「人口統計信息」和「其他上下文信息」。對於「用戶觀看影片的歷史紀錄」這組特徵，論文中使用類似於 word2vec 的做法，將每個影片映射到固定維度的向量中，我們可以通過對使用者曾觀看過的所有影片向量，做加權平均得到一個固定維度的用戶歷史觀

看影片向量，來當作「用戶觀看影片的歷史紀錄」特徵向量 (watch vector)。對於「用戶的歷史搜尋紀錄」這組特徵，會將使用者歷史搜尋的關鍵字分詞後，將每個詞的嵌入向量 (embedding) 進行加權平均，來產生能反映用戶歷史搜尋狀態的特徵向量 (search vector)。對於「人口統計信息」這組特徵，會使用性別、年齡、地域等特徵來形成「人口統計信息」特徵向量。對於「其他上下文信息」這組特徵，則是由使用者使用的設備或登入狀態等特徵，來產生「其他上下文信息」特徵向量。將上述四種特徵向量串接，然後輸入三層全連接層的 DNN 模型，最後模型的輸出即為該使用者的向量 u (user vector)。

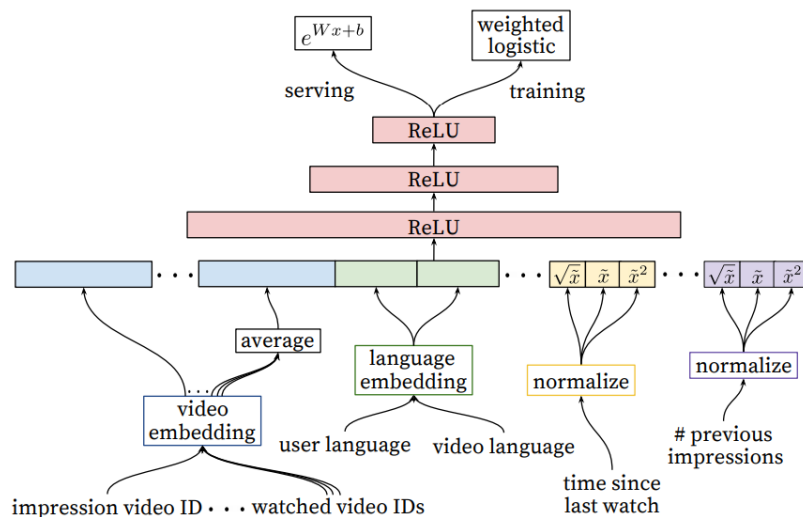
使用者向量 u 在模型訓練時與模型使用時分別有不同的用途。在模型訓練時，模型的訓練目標是：希望學習出的使用者向量能在預測該使用者是否會觀看某影片的預測任務上，誤差越小越好；而在模型使用時，使用者向量 u 被用來找出與之相似的其他使用者 (nearest neighbor)，並且取出這些相似使用者喜歡的前 N 個影片當作候選的影片 (candidate)。以下為 candidate generation 階段的模型架構圖：



圖三、文獻[2] 的神經網路架構圖

在 ranking 階段中，類神經網路模型的輸入除了考慮使用者過去的活動紀錄與使用者上下文資訊，還會考慮第一階段所選出的候選影片的影片特徵。

在第二階考慮到候選影片的特徵有：「候選影片 ID」(impression video ID)、「候選影片語言」(video language)、「候選影片最後一次的觀看時間」(time since last watch)和「候選影片曝光次數」(# previous impression)等。對於「候選影片 ID」這個特徵，「候選影片 ID」會共用第一階段產生影片向量的嵌入矩陣(embedding matrix)來產生候選影片的向量，並同時考慮目前這個候選影片和使用者過去曾看過的影片。對於「候選影片語言」這個特徵，會同時考慮使用者的語言和候選影片的語言，在將此兩個特徵共用一個語言嵌入矩陣(language embedding matrix)，形成使用者語言向量和影片語言向量。對於「候選影片最後一次的觀看時間」這個特徵，此特徵計算影片最後一次觀看到目前所經過時間。對於「候選影片曝光次數」這個特徵，此特徵計算使用者觀看候選影片之前該影片被系統顯示了多少次。類似於 candidate generation 的架構，我們會將上述的特徵全部串接起來輸入三層全連接層的 DNN 模型，最終模型的輸出會使用者對於候選影片的預測觀看時間，我們可以依觀看時間由長至短排序後推薦給使用者。以下為 ranking 階段的模型架構圖：



圖四、文獻[2]的神經網路架構圖

除了上述方法，另一派深度學習推薦系統是基於 Auto-Encoder (AE) 架構，這類方法可以分為兩大類，(i)使用 AE 來擷取使用者或商品附帶的內容資訊的低維度特徵向量[16] [14] [9] (ii)使用 AE 來預測使用者與商品互動行

為[17]。

在第 1 類 Autoencoder 的應用中，[16] 提出了一個大架構適合用於預測 CCS (complete cold start) item 和 ICS (incomplete cold start) item，這兩種冷啟動商品的評分。這個大架構是由 DL 模型和 timeSVD++所組成的。其中使用的 DL 模型為 Stacked Denoising Autoencoder (SDAE)，SDAE 模型接收冷啟動商品的文字描述當作輸入，並擷取 SDAE 的中間層當作商品向量。之後再將商品向量輸入，預測評分的 timeSVD++ 模型中，同時考慮商品流行度和使用者評分模式這些隨時間變化的因素，以及考慮要預測的目標使用者過去評過分的所有商品集合的平均分數等因素，預測最終冷啟動商品的評分。針對 CCS item，模型預測評分的公式如下：

$$\hat{r}_{ui}(t) = b_u(t) + \theta_i^T \left[p_u(t) + |N(u)|^{\frac{1}{2}} \sum_{j \in R(u)} y_j \right] + \frac{\sum_{j \in S^M(u,i)} r_{uj} s_{ij}}{\sum_{j \in S^M(u,i)} s_{ij}}.$$

$\hat{r}_{ui}(t)$ ：表示在時間點 t，使用者 u 對商品 i 的評分

$b_u(t)$ ：表示使用者 u 在時間點 t，給與評分的偏差（使用者評分的模式）

θ_i^T ：表示商品 i 的商品描述資訊經過 SDAE 後，隱藏層所產生的商品向量

$|N(u)|^{\frac{1}{2}} \sum_{j \in R(u)} y_j$ ：表示使用者 u 過去所評過分的商品集合的平均向量

$\frac{\sum_{j \in S^M(u,i)} r_{uj} s_{ij}}{\sum_{j \in S^M(u,i)} s_{ij}}$ ：表示 CCS 商品 j 與跟它前 M 個最相似的非冷啟動商品 i，依使

用者 u 給商品 i 的評分對商品 i 和商品 j 的相似度(s_{ij})作加權。意義為：使用與 CCS 商品 j 相似的非冷啟動商品估計 CCS 商品 j 可能會得到的分數。

而針對 ICS item，模型預測評分的公式如下：

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T \left[p_u(t) + |N(u)|^{\frac{1}{2}} \sum_{j \in N(u)} y_j \right]$$

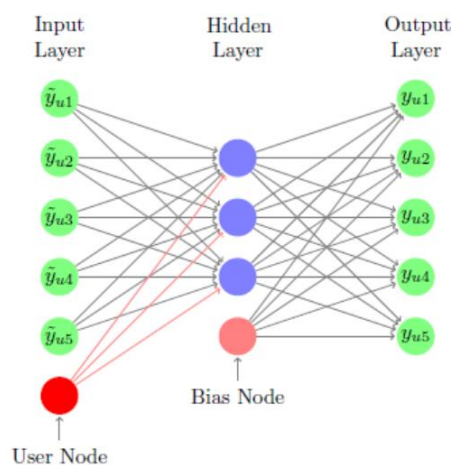
μ ：所有商品的平均評分

$b_i(t)$ ：表示商品 i 在時間點 t，收到評分的偏差（商品流行度）

$q_i(t)$ ：因為 ICS 商品仍有一些評分資訊存在，所以可以取得該商品的向量

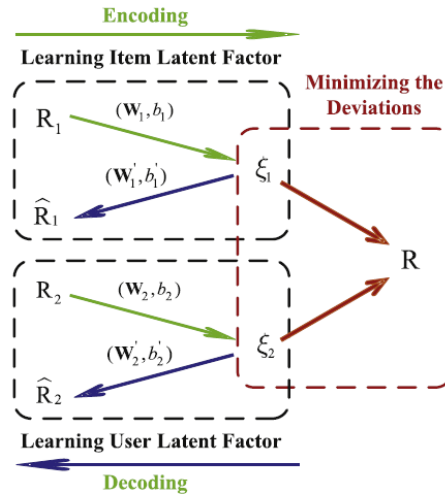
在第 2 類 Autoencoder 的應用中，[17] 提出了 Collaborative Denoising Autoencoder (CDAE) 的模型，CDAE 接收使用者對各個商品的喜好值(若為 1：代表使用者喜歡此商品；若為 0：代表使用者尚未看過此商品)和該使用

者的 ID 當作模型輸入，然後透過隨機將某些輸入值設為 0，再經過一層維度遠比輸入向量小的隱藏層，最後將輸出層設定為輸入向量的維度。透過 Autoencoder 的訓練，來還原被雜訊干擾的輸入向量，而還原出的輸出向量即為使用者對於商品喜好值。模型訓練完後，若今天有一個使用者向量 y_u ，向量中的每一個維度對應到一個商品的喜好(0 或 1)，可以想像使用者向量 y_u 會是一個稀疏的向量(因為使用者可能有很多商品都沒看過，商品所對應到的維度值 = 0)。推薦的方法是將 y_u 輸入 CDAE，得到輸出 \hat{y}_u ，推薦 \hat{y}_u 中維度值越接近 1 所對應的商品給該使用者。以下為 CDAE 的模型圖：



圖五、文獻[17]的神經網路架構圖

另外，[19] 提出了一種以 Autoencoder 來進行推薦的新架構，稱之為 Dual Autoencoder。Dual Autoencoder 架構會同時訓練使用者的向量表示法 u 與商品的向量表示法 v ，並且將向量 u 與向量 v 做內積來預測使用者 u 對商品 v 的評分。其中，訓練使用者向量表示法 u 與商品向量表示法 v 的過程是透過 Autoencoder 來訓練。最後訓練完成的 Dual Autoencoder 在進行推薦時，若今天輸入一個使用者向量 y_u ， y_u 向量中的每一個維度對應到使用者對一個商品的評分，則 Dual Autoencoder 會輸出該使用者對於所有商品的評分，我們可以將模型的輸出向量當作預測值，並將預測分數高的商品推薦給該使用者。以下為 Dual Autoencoder 的模型圖：

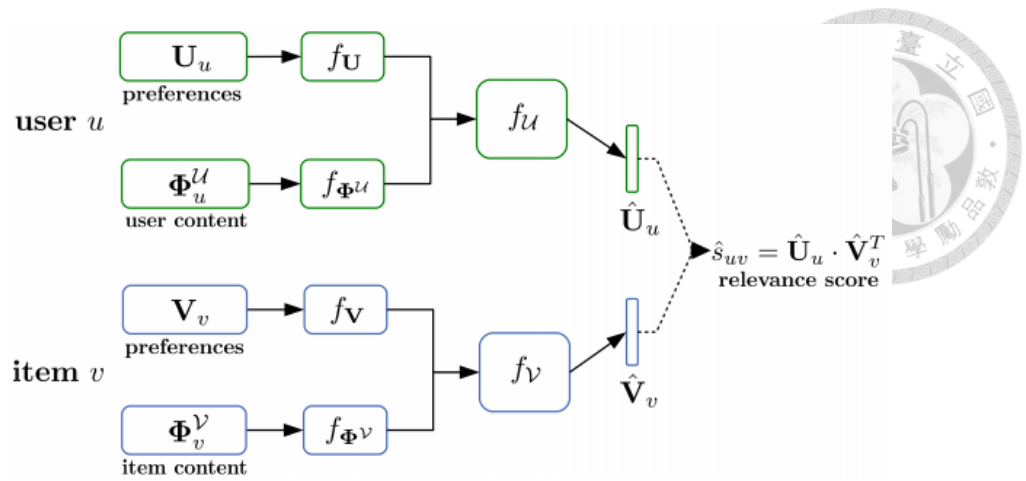


圖六、文獻[19]的神經網路架構圖

2.2 運用深度學習之冷啟動推薦文獻探討

雖然以深度學習為基礎的推薦方法比非以深度學習為基礎的方法，在做協同式推薦時有更強的能力，不過面對新進的使用者或新進商品只擁有即少數評分資訊時，仍然會面臨冷啟動的問題。因此，也有許多的研究設法使用深度學習的方法解決冷啟動問題。

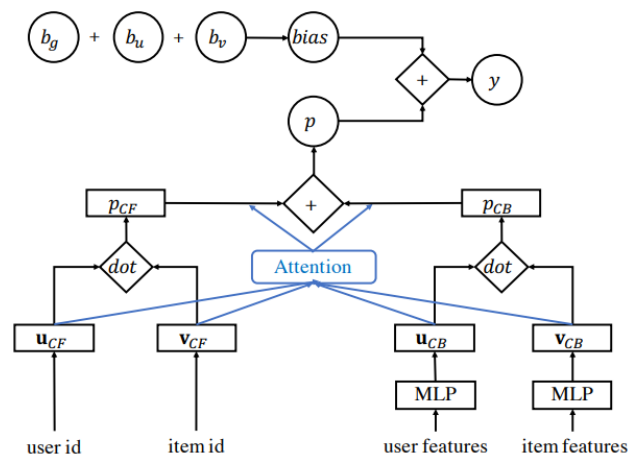
像是[15] 提出的深度學習模型，結合了 preferences (rating)和 content 的資訊當作模型輸入，以及使用 input dropout 的機制。在訓練模型時，當我們遇到某個冷啟動商品或冷啟動使用者時，因為我們並沒有該商品或使用者的評分資訊，所以我們將對應的 item preferences 或 user preferences 輸入設為 0 向量。如此的設計稱為「dropout」，其想法是源自於 DAE 產生雜訊的設計，並希望最終模型能還原出該冷啟動使用者的 latent representation。以下為[15] 中模型的示意圖：



圖七、文獻[15]的神經網路架構圖

另外，[13] 提出以深度學習中的 Attention 機制為基礎的模型稱為 Attentional Content & Collaborate Model (ACCM)，透過 Attention 機制動態的給予內容資訊(content information)與歷史活動資訊 (historical feedback)相應的權重。其核心想法為：對於新進的使用者或商品輸入，由於它們的歷史活動資訊比較少，所以模型會傾向於使用內容資訊。而對於非新進的使用者或商品輸入，由於它們的歷史活動資訊比較多，則模型會傾向於使用歷史活動資訊。下圖為[13] 中提出的兩個模型架構，分別被稱為 Result Level Attention 和 Vector Level Attention。

Result Level Attention：

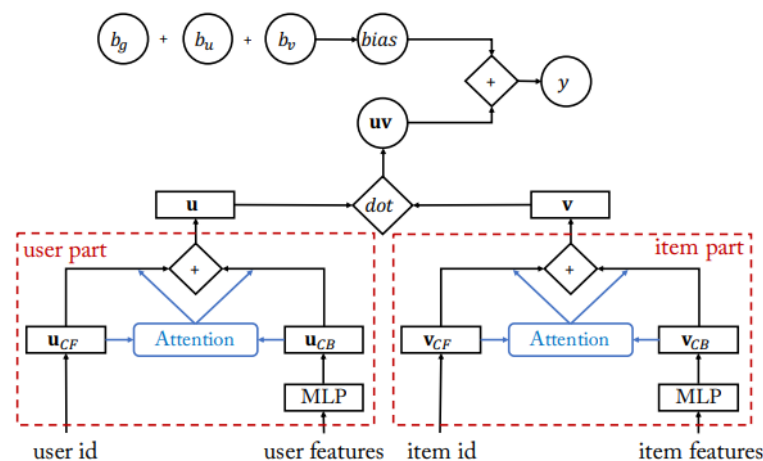


圖八、文獻[13]的神經網路架構圖

在 Result Level Attention 的模型架構中，輸入分別為 user id 的 one-hot encoding、item id 的 one-hot encoding 與 user features (例如：年齡、職業...)、

item features (例如：物品描述...)。其中 user id 和 item id 的 one-hot encoding 會分別通過一層 embedding layer 得到向量 u_{CF} 、 v_{CF} ，這些 embedding layer 中的權重會根據訓練時預測值與真實值的誤差來更新。而 user features 和 item features 則是事先透過其他方法訓練出來的向量，並非與 Result Level Attention 模型一起訓練的，將它們再分別通過一層 MLP 後可以得到向量 u_{CB} 、 v_{CB} 。接著將向量 u_{CF} 、 v_{CF} 做內積就類似於使用傳統 CF 方法來預測使用者對商品的評分 p_{CF} ，同樣也可以對向量 u_{CB} 、 v_{CB} 做內積得到評分 p_{CB} 。當有了評分 p_{CF} 與 p_{CB} 後，就會使用 Attention 機制動態的學習出 p_{CF} 與 p_{CB} 各自佔的權重，再依權重對兩個評分加權得到評分 p 。在最後模型輸出預測評分時，除了預測的評分 p 以外，還會考慮 user bias b_u 、item bias b_v 和 global bias b_{global} 。

Vector Level Attention：

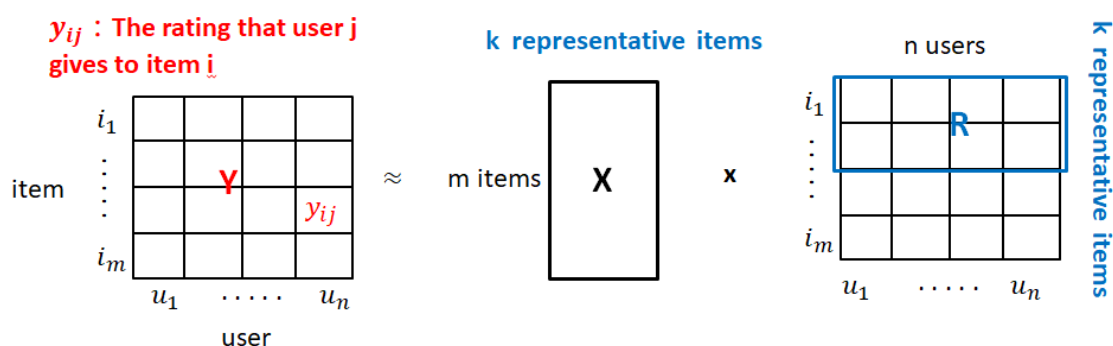


圖九、文獻[13]的神經網路架構圖

在 Vector Level Attention 的模型架構中，模型的輸入與 Result Level Attention 的模型輸入完全相同。不同的地方在於 Attention 的機制被使用到產生使用者向量 u 與產生商品向量 v ，而不是對預測出的評分做加權。在得到使用者向量 u 與商品向量 v 後，我們便將這兩個向量做內積來當作預測的評分。在最後模型輸出預測評分時，同樣的也會考慮 user bias b_u 、item bias b_v 和 global bias b_{global} 。

2.3 代表性商品探勘之文獻探討

在上一個小節中用來解決冷啟動問題的深度學習方法，都是使用基於內容的推薦這種方法。而還有另一種解決冷啟動問題的方法是基於代表性商品的推薦。不同於基於內容的推薦需要蒐集額外關於使用者或商品相關的資訊，再使用深度學習模型來提取附帶資訊的特徵。若以解決冷啟動使用者問題為範例，基於代表性商品的推薦透過選出推薦系統中的具代表性商品，並以這些具代表性商品和其他一般商品之間的關係，來預測冷啟動使用者對其他一般商品的評分。文獻 [8] 提出 RBMF (representative-based matrix factorization)，將 user 對 item 的 rating matrix $Y \in R^{m \times n}$ ，拆解為兩個子矩陣。若是在要找出 k 個具代表性商品的情境之下，則兩個被拆解出的子矩陣分別為 $X \in R^{m \times k}$ 、 $R \in R^{k \times n}$ ，即($Y \approx XR$)，其中的子矩陣 R 代表：所有 n 個 users 與 k 個 representative items 之間的關係；子矩陣 X 則代表： k 個 representative items 與所有 m 個 items 之間的關係。至於如何選定 k 個 representative items，是透過 maximal volume algorithm 所求得，此演算法被用在將拆解後的子矩陣，拿來逼近原始矩陣的任務中。maximal volume algorithm 能從原始矩陣中，選出最能還原原始矩陣 Y 的 k 個 columns 來形成子矩陣 X 。因此我們認為這 k 個 columns 所對應的 k 個 items 就是 representative items。下圖為 RBMF 在選出 representative items 的示意圖：

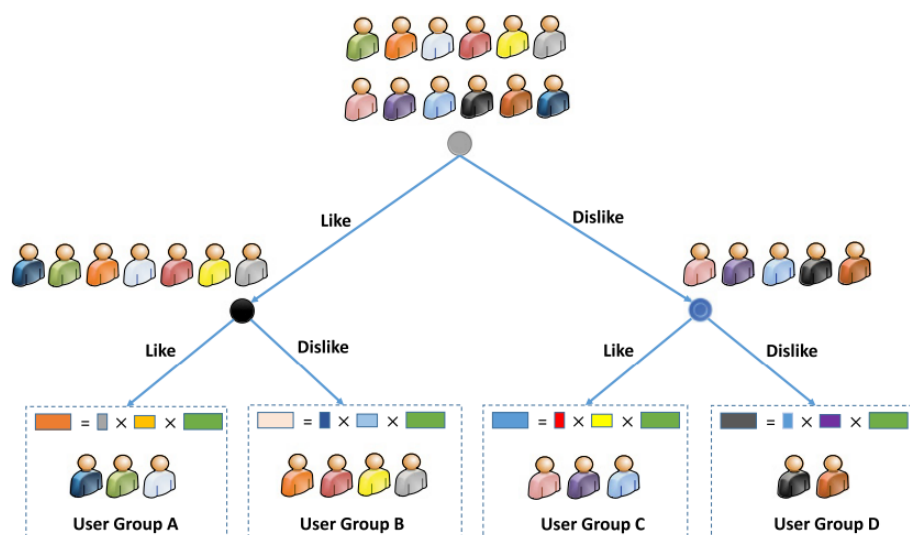


圖十、文獻[8]的矩陣分解法

當我們透過 maximal volume algorithm 找出 k 個具代表性的商品後，

所有的模型參數都存在於 representative items 與所有 items 之間的關聯矩陣 $X \in R^{m \times k}$ 中。當我們要進行推薦時，只要能取得新使用者對於這 k 個具代表性商品的評分，即可得到該 user 的向量 $u \in R^{1 \times k}$ ，最後將向量 u 與矩陣 X 做內積，便可以得到此 user 對所有商品的評分。

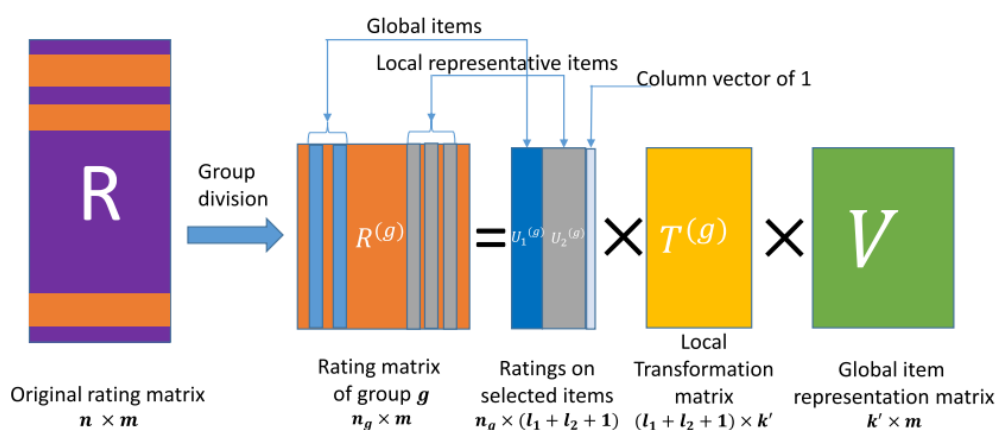
[12]想改良[8] 所提出的 RBMF 方法，進而捕捉更細微的新進使用者喜好，相較於 RBMF 只使用一組具代表性的商品，來當作描述所有使用者的特徵。[12] 提出的 LRBMF 方法，會先動態的將使用者以決策樹，依不同「全域具代表性商品」(global representative items)來分組。分組過後，再根據各組選取「區域具代表性商品」(local representative items)。最後，每一個使用者向量的維度是由全域與區域具代表性商品共同組成的。在選取「全域具代表性商品」時的依據是，希望各組所得到的評分矩陣與各組所預測的評分矩陣之間的誤差總和最小。而在選取「區域具代表性商品」時，則是透過 Maximal Volume algorithm 來選出前 k 個「區域具代表性商品」。以下為整個 LRBMF 模型兩個階段的示意圖：第一階段為將使用者分組



圖十一、文獻[12]的方法架構圖

第二階段的矩陣 $R^{(g)}$ ，每列(row)代表該組的使用者，每行(column)代表

一個商品，我們會從所有 m 個商品中，依照第一階段選出的 l_1 個全域具代表性商品，和第二階段利用 Maximal Volume algorithm 所選出的 l_2 個區域具代表性商品，來形成矩陣 U 。而其他兩個矩陣 $T^{(g)}$ 與 V 中所包含的內容便是我們需要訓練用來逼近 $R^{(g)}$ 的參數。當我們訓練完所有矩陣 $T^{(g)}$ 與 V 中的參數後，進行推薦的方法便會類似於之前所提到的 RBMF，只要能取得某使用者對於「全域具代表性商品」與「區域具代表性商品」的評分值，便可以得到該使用者的向量 u ，然後再將向量 u 與矩陣 $T^{(g)}$ 、 V 做內積後，就可以得到此使用者對所有商品的評分。



圖十二、文獻[12]的矩陣分解圖

[4] 將推薦系統中的所有使用者做 K-means 分群後，找出各群集中最具代表性的使用者，對各群中的其他使用者做推薦。在為每一個群集找出 representative user 時，使用的方法是找出該群集中的某個 user，若他與其他群集內的成員 similarity 總和最大，則此 user 便是該群集的 representative user。此方法的關鍵在於如何定義 user 之間的 similarity。本篇論文透過改良以往的 similarity metric (cosine similarity、Pearson correlation)，來修正過去的 similarity metric 會選出 highly active user 當作 representative user 的偏差。提出的修正方法是：在原本的 Pearson correlation 計算中，乘上 $\frac{N_I^{ij}}{\max(N_I^i, N_I^j)}$ 。 N_I^{ij} ：表示 user i 和 user j 共同評分過的 item 數量。 N_I^i ：表示 user i 評分過的 item

數量。 N_I^j ：表示 user j 評分過的 item 數量。我們可以發現，若兩個 user 共同評過分的 item 數量越多，則他們之間的相似度就會越高。並且若 user i 或 user j 有其中一個人是 highly active user 時，則分母 $\max(N_I^i, N_I^j)$ 會越大，造成兩個 users 之間的 similarity 變小。最後在進行推薦時，會找出目標使用者所屬群集的 representative user，然後推薦 representative user 所評分過的 item set 中，分數超過某個門檻值以上的 item 給目標使用者。

[10] 的目標是選出能代表新進使用者的具代表性商品集合，主要提出的方法為 HELF (Harmonic mean of Entropy and Logarithm of Frequency)。HELf 設計的概念為：希望具代表性的商品必須是被大多數的使用者所評分過的(popularity)，換句話說就是商品的總評分人數需要夠多。而具代表性商品的評分分布還必須夠分散(entropy)，意即所有使用者對商品的評分若越不一致，則該商品所包含的資訊量越多。最後 HELF 同時考慮 popularity 與 entropy 這兩個因素的調和平均，當作選取具代表性商品的指標，其公式如下：

$$HELF_{a_i} = \frac{2 * LF'_{a_i} * H'(a_i)}{LF'_{a_i} + H'(a_i)}$$

LF'_{a_i} ：商品 a_i 收到的評分次數，取對數再除以所有使用者人數(normalization)

$H'(a_i)$ ：商品 a_i 收到評分分布(entropy)，再除以 5 分(normalization)

最後 HELF 指標值越大的商品，我們就認為這些商品是具代表性的商品。

3 論文方法：

在本章節中，我們首先將介紹 DAE 的運作概念 (3-1)，再來說明 DAE 如何運用到新進使用者冷啟動推薦的問題上 (3-2)，接著就是本篇論文的核心方法 (3-3)：如何將使用者向量透過我們設計的「使用者返老還童」(user rejuvenation) 方法，來把使用者向量模擬(倒退)回剛進入平台時的新進使用

者向量。最後 (3-4)則說明 DAE 模型如何訓練以及我們怎麼使用訓練好的 DAE 產生推薦給使用者。

3.1 DAE 基本架構

在深入了解 DAE 之前，首先我們必須對傳統的 Auto-Encoder (AE) 有一定的認識，AE 是由 encoder 和 decoder 兩部分所組成的。在 encoder 中，會使用一層全連接層的類神經網路當作函式 f_θ ， f_θ 會將輸入向量 x 映射成較低維度的向量 y ，我們稱此函式 f_θ 為 encoder。函式 f_θ 如以下公式所表示：

$$f_\theta(x) = \sigma(Wx + b)$$

其中 f_θ 的函式參數為 $\theta = \{W, b\}$ ， W 是一個 $k \times n$ 的轉置矩陣 ($k \ll n$)，而 b 是一個維度等於 k 的偏差向量 (bias vector)， $\sigma(\cdot)$ 是 sigmoid 函式。

在 decoder 中，會使用一層全連接層的類神經網路當作函式 g_φ ， g_φ 會將低維度的向量 y 映射回與輸入向量 x 相同維度的向量 \hat{x} 。我們稱此函式 g_φ 為 decoder。函式 g_φ 如以下公式所表示：

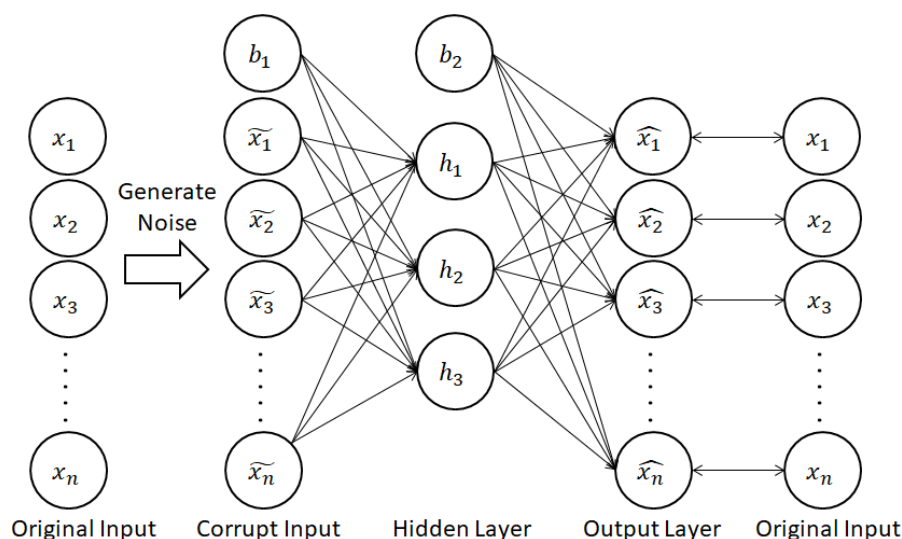
$$g_\varphi(x) = \sigma(W'x + b')$$

其中 g_φ 的函式參數為 $\varphi = \{W', b'\}$ ， W 是一個 $n \times k$ 的轉置矩陣 ($k \ll n$)，而 b' 是一個維度等於 n 的偏差向量 (bias vector)， $\sigma(\cdot)$ 是 sigmoid 函式。最終整個模型要優化的損失函式，便是輸入向量 x 與輸出向量 \hat{x} 之間的誤差總和，可以使用以下的公式來表示：

$$\operatorname{argmin}_{\theta, \varphi} \sum_{i=1}^N \|x^{(i)} - \hat{x}^{(i)}\|^2 = \|x^{(i)} - g_\varphi(f_\theta(\hat{x}^{(i)}))\|^2$$

其中 $\|\cdot\|^2$ 是計算兩個變數平方誤差的函式、 $x^{(i)}$ 代表第 i 筆輸入向量、 $\hat{x}^{(i)}$ 代表第 i 筆輸入向量經過 AE 後的輸出向量。

DAE 是 AE 的一種變形，該模型的訓練目標仍然是希望最小化輸入向量與輸出向量之間的誤差，不過 DAE 會對輸入向量中的一部分維度產生雜訊，才將帶有雜訊的向量輸入 AE 中進行輸入向量的還原。整個 DAE 的運作流程如下圖：



圖十三、Denoising Autoencoder 模型架構圖

而其中常見的雜訊產生方法有三種：第一種方法是 Gaussian noise，Gaussian noise 適用於當輸入向量的維度值是實數的情況，會依照平均數為 0，變異數為 1 的常態分布來產生雜訊值，並將原本輸入向量中的維度值加上產生的雜訊值。第二種方法是 Masking noise，Masking noise 會隨機將輸入向量中的一部份維度值設定為 0。第三種方法是 Salt-and-pepper noise，Salt-and-pepper noise 則是會隨機將輸入向量中的一部份維度值設定成該維度值可容許的最大值或最小值，例如對於彩色照片的每個像素(對應到向量中的維度值)，可容許的維度值便會介於 0 到 255 之間。通過加入雜訊的機制，可以讓後續的 AE 模型在訓練時，能擷取出輸入向量中更高階的特徵表示法，使模型不容易因為輸入向量受到些許雜訊的干擾，就改變模型的預測結果。

因此，以下我們總結 DAE 的訓練過程：

1. 首先，我們會將輸入向量 x 通過上述的三種雜訊產生方法或自行設計的機率分布，將輸入向量映射成受雜訊干擾的向量 \tilde{x} 。此過程可以表示為：

$$\tilde{x} \sim q_D(\tilde{x}|x)$$

其中的 $q_D(\cdot)$ 為產生雜訊的機率分布

2. 接著，將受雜訊干擾的向量 \tilde{x} 當作傳統 Autoencoder 的輸入，使用 Encoder 將 \tilde{x} 映射到低維度的向量 y 。此過程可以表示為：

$$f_\theta(\tilde{x}) = \sigma(W\tilde{x} + b)$$

3. 最後，使用 Decoder 將隱藏層的向量 y 映射回輸入向量的空間中，產生輸出向量 \hat{x} 。此過程可以表示為：

$$g_\phi(x) = \sigma(W'y + b')$$

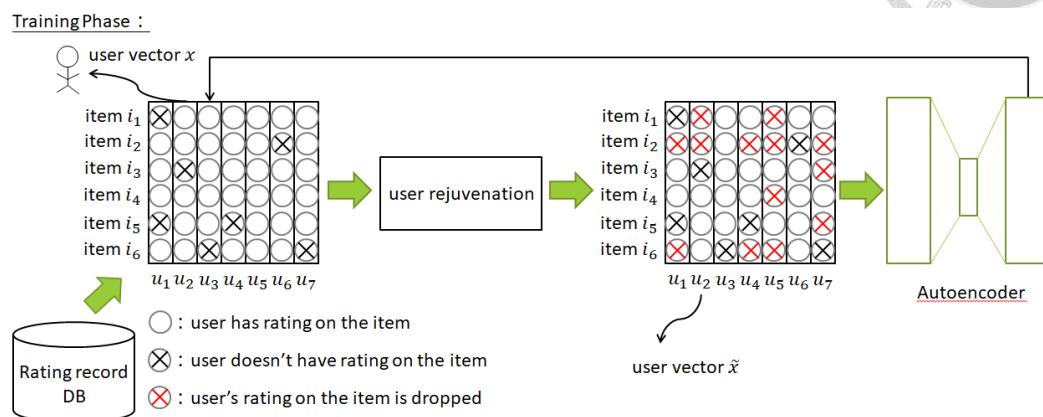
而整個 DAE 模型參數 θ 、 ϕ 的優化過程可以使用一般的「反向傳遞演算法」(backpropagation)，通過將輸入層與輸出層之間的差距 (loss) 逐步由模型輸出層往輸入層傳遞，以此來更新參數 θ 、 ϕ 。

3.2 採用 DAE 架構原因與初步方法

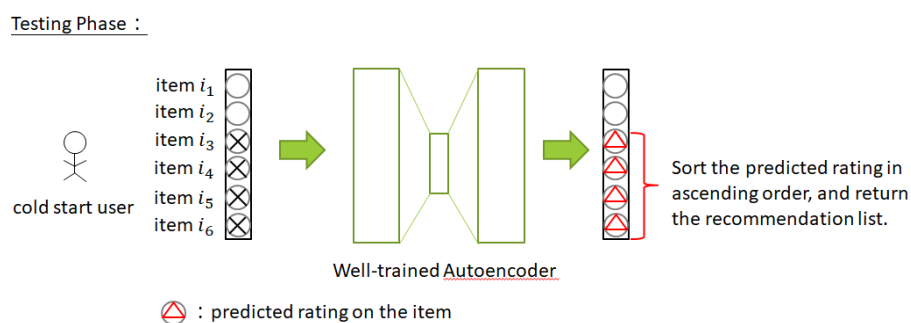
DAE 模型與冷啟動推薦的關聯性在於：DAE 在訓練時的其中一種雜訊產生方式 (Masking noise) 會將輸入向量的一部份維度值隨機設定為 0，這樣的雜訊產生做法對應到推薦系統的情境就是從使用者向量中，抹去使用者對於部分商品的評分。我們的目標便是提出適用於推薦系統情境的雜訊產生方式，藉此把使用者向量模擬(倒退)成剛進入平台的新進使用者向量。而後續 Autoencoder 模型訓練的目標就是把處於新進使用者狀態的向量成長(快轉)為一開始資料豐富的使用者向量。如此一來，訓練完畢的 encoder-decoder 模型可以被視為，能預測冷啟動使用者未來將會偏好哪些商品的預測模型。

以下我們將以圖例來說明 DAE 與冷啟動使用者推薦流程，其中為了更

貼近新進使用者冷啟動問題，我們改稱 DAE 訓練時的雜訊產生階段為「使用者返老還童階段」(user rejuvenation stage)，而「使用者返老還童階段」的方法細節將會在下一個小節中介紹。



圖十四、Denoising Autoencoder 訓練流程圖



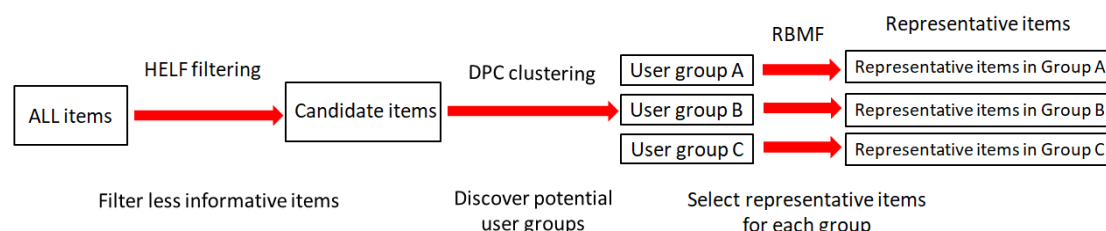
圖十五、Denoising Autoencoder 測試流程圖

3.3 使用者返老還童階段

有鑑於原始 DAE 隨機抹去使用者對商品的互動資訊，此種作法所產生的稀疏向量未必能確切反映新進使用者的冷啟動狀態。因此在「使用者返老還童階段」中，我們在將使用者還原成新進使用者狀態時，並不會隨機抹去使用者與商品互動的資訊，而是透過找出具代表性的商品，然後在還原使用者向量時盡可能的抹去非具代表性商品的互動資訊，讓最後的使用者稀疏向量中保留與具代表性商品的互動資訊。因為我們認為，新進使用者進入一個服務平台，往往都是由平台中具有代表性的商品開始接觸，之後隨著時間的推移才會探索那些非具代表性的商品。而我們希望選出的具代表性商品必須具備兩個特性：首先，具代表性的商品必須被大多數的使用者評分過

(popularity)，換句話說就是大多數的使用者必須對此商品足夠熟悉。以及，具代表性商品的評分分布還必須夠分散(rating entropy)，意即使用者們對商品的評分若越不一致，則我們才能夠透過該商品來區隔不同類型的使用者。

以下我們將以圖例展示「選出具代表性商品」的運作流程：



圖十六、「選出具代表性商品」流程圖

由上面的流程圖可得知，所有的商品會通過一條處理路徑來形成最終的具代表性商品集合。在以下的各個小節中將會說明各個區塊間的處理運作方式。

● Harmonic mean of Entropy and Logarithm of Frequency (HELf)

在選取候選具代表性商品時，我們使用[10]中所提出的 HELf 來當作初步篩選商品的指標，HELf 指標會考慮到兩個因素：其一是商品的熱門度(Popularity)，也就是某個商品曾被多少的使用者評分過；其二是使用者們對某個商品評分的熵(Rating entropy)，意即使用者們對於某個商品評分意見的分散程度，舉個例子來說：若所有使用者對某商品一致給予 5 分的評分，則該商品的熵為最小值 0；反之，若所有使用者的評分均勻的分布在 1~5 分中，則該商品可以得到最大的熵值。綜合上述的兩個因素的調和平均數便可以得到指標 HELf：

$$HELf_{a_i} = \frac{2 \times LF'_{a_i} \times H'(a_i)}{LF'_{a_i} + H'(a_i)}$$

LF'_{a_i} ：商品 a_i 收到的評分次數，取對數再除以所有使用者人數(normalization)

$H'(a_i)$ ：商品 a_i 收到評分分布(entropy)，再除以 5 分(normalization)

我們希望選出的候選具代表性商品需要有越大的 HELf 值，換句

話說就是該商品需要被大多數人所評分過，並且每個人對該商品的意見要越分歧。因此，我們會先過濾掉 HELF 值較小的商品來產生候選具代表性商品集合 I_{cand} 。先使用 HELF 指標濾掉極端冷門(unpopular)或沒有任何使用者鑑別力(low rating entropy)的商品，因為這些被濾掉的商品留下來當作使用者向量的一部份其實並沒有任何的資訊含量，而且會拖慢後續幫使用者分群的速度與效果。

● Density Peaks Clustering (DPC)

DPC [11] 是一種基於區域性密度的分群演算法，通過事先設定一個距離範圍 d_c (cutoff distance)，我們便可以計算出某個資料點 i 在 d_c 範圍內所包含的其他資料點數量，也就是資料點 i 的區域密度(local density) ρ_i 。對應到推薦系統的應用情境中，因為我們的目標是找出潛在的使用者分組，所以每個資料點 i 就是一個使用者向量，而使用者向量中的每個維度由 HELF 指標初步篩選出的候選具代表性商品來表示。對於使用者 i 的區域密度 ρ_i 計算公式如下：

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}$$

d_{ij} ：使用者 i 與使用者 j 的距離

d_c ：事先設定的距離範圍

$\chi(x)$ ：若使用者 i 與使用者 j 的距離在 d_c 內，則區域密度 ρ_i 值增加 1

其中使用者 i 與使用者 j 之間的距離 d_{ij} 被定義為：兩個使用者共同評分過的商品集合中，相同商品評分差的總和平均，計算公式如下。

$$d_{ij} = \frac{1}{|M|} \sum_{m \in M} \|s_{m_i} - s_{m_j}\|$$

M ：使用者 i 與使用者 j 共同評分過的商品集合

m ：商品集合 M 中的一個商品

s_{m_i} ：使用者 i 對商品 m 的評分

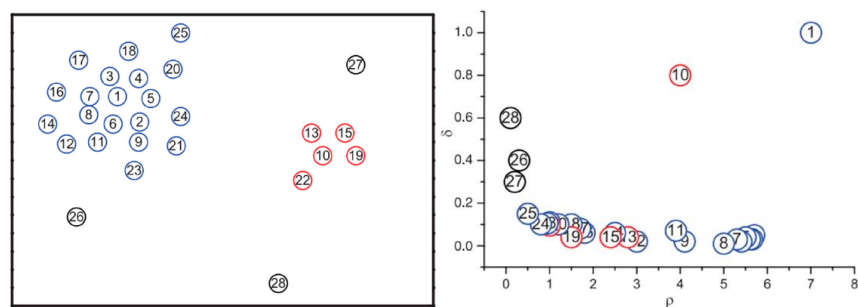
s_{m_j} ：使用者 j 對商品 m 的評分

得到區域密度 ρ_i 後，DPC 便可以依據 ρ_i 計算出群聚中心相對距離(relative distance) δ_i ，計算的方法是：當計算出所有使用者的區域密度 ρ_i 後，我們可以將區域密度由大到小排序，假設為 $\rho_i > \rho_j > \rho_k > \dots$ 。對於區域密度最大的使用者 i，它的群聚中心相對距離 δ_i 會等於與使用者 i 相距最遠的那個使用者的距離。而其他非區域密度最大的使用者，例如使用者 j 的群聚中心相對距離會等於，區域密度大於使用者 j 的使用者集合中，與使用者 j 距離最小的使用者的距離。 δ_i 的計算公式可以依據 ρ_i 是否為最大區域密度，表示成以下的兩個公式：

$\delta_i = \max_j(d_{ij})$ 若 ρ_i 為所有使用者中，區域密度最大的點

$\delta_i = \min_{j:\rho_j>\rho_i}(d_{ij})$ 若 ρ_i 不是區域密度最大的使用者

DPC 分群演算法會依據每個使用者的區域密度 ρ_i 和群聚中心相對距離 δ_i 分別當作二維平面的座標軸(x 軸： ρ_i 、y 軸： δ_i)來產生決策圖，如下圖：



圖十七、DPC Decision Diagram

左圖為 28 個使用者在空間中分布的情況，且每個使用者上面的數字是依照區域密度由大到小進行編號(編號 1 的區域密度最大、編號 28 的區域密度最小)。右圖則是將這 28 個使用者各自的 (ρ_i, δ_i) 繪製在 x 軸為區域密度、y 軸為群聚中心相對距離的座標平面之上。

從右圖中可以明顯的看出當 ρ_i 與 δ_i 皆大時，表示這些使用者(1、

10)不僅區域密度很大，且密度比這些使用者大的使用者集合(其他可能的群聚中心)，它們離使用者 1、10 的最小距離仍然很遠，因此可以將 1、10 視為群聚中心。當 ρ_i 很小但是 δ_i 大時，表示這些使用者(26、27、28)的區域密度很小，而且距離其他密度比它們大的使用者集合還很遠，因此可以將它們視為離群值的使用者。當 ρ_i 大但是 δ_i 很小時，表示這些使用者 (除了 1、10、26、27、28 的其他使用者)的區域密度雖然大，但是周圍密度比它們大的使用者集合也離它們很近，所以它們無法成為群聚中心。在我們可以使用決策圖來選出群聚中心 1 和 10 後，我們就可以將其他使用者依照距離指派給不同的群聚中心，完成使用者的分群任務。

因此，依照以上的分群方法，事先設定一個好的 d_c (cutoff distance)，分別計算出每個使用者的 ρ_i 與 δ_i 後，產生出決策圖來手動選取 ρ_i 和 δ_i 值皆大的使用者當作群聚中心，最後將其他非群聚中心的使用者指派給距離最接近的群聚中心，以此找出潛在的使用者分組。

- Representative-based matrix factorization (RBMF)

假設在我們的推薦系統中使用 DPC 分群方法可以將所有使用者分為 A、B、C 三組，意即我們可以將包含所有使用者與候選具代表性商品的評分矩陣 $R_{all} \in R^{n \times m}$ 分割成三個評分子矩陣 $R_{group A} \in R^{n_A \times m}$ 、 $R_{group B} \in R^{n_B \times m}$ 、 $R_{group C} \in R^{n_C \times m}$ ，其中 n 代表系統中所有使用者的數量、 m 代表系統中經過 HELF 指標初步篩選過的候選商品數量， n_A 、 n_B 、 n_C 分別代表不同組的使用者數量。我們可以利用 Maximal Volume Algorithm 分別找出這三個評分子矩陣中的 k 個行(k 個具代表性商品)，使拆解後的子矩陣能最逼近原始的矩陣。而透過 RBMF 為每個使用者分組選出的商品便被稱為「具代表性商品」，我們

會在「使用者返老還童階段」中給予「具代表性商品」高於一般商品的機率被保留下來。



3.4 冷啟動使用者推薦

在進行推薦時，我們可以將訓練完成的 DAE 模型視為：能依據只有少數商品評分資訊的使用者向量，推演出該使用者往後評分模式的模型。意即，我們只需將冷啟動狀態的使用者向量輸入 DAE 模型，DAE 模型的輸出向量就是我們預測的該使用者對各個商品的評分。因此，我們可以依照各個商品所預測出的分數，由高至低的推薦商品給使用者。

4 論文方法實驗與分析：

4.1 實驗資料集與評估指標

● 實驗資料集

MovieLens 1M 資料集是一組從 20 世紀 90 年代末到 21 世紀初，由 MovieLens 使用者提供的電影評分資料。資料集中包含了使用者對電影的評分、電影類型、電影年代以及關於使用者的人口統計學資料(年齡、性別、職業)。但由於我們論文的方法著重在使用者對於電影的評分，因此只將資料集中，使用者對電影評分的資訊整理於下表。

	Num of ratings	Num of movies	Num of users
MovieLens 1M	1,000,209	3,900	6,040

表一、MovieLens 1M 統計資料表

● 評估指標介紹

在推薦系統效能評估階段，我們選擇常用來評估排序效能的指標 NDCG@k，和 Precision@k、Recall@k 來檢視我們所提出模型的表現，並於以下詳細介紹這些指標是如何計算的。

■ 評估指標 NDCG@k

NDCG [7]全名為 Normalized Discounted Cumulative Gain，它是一個衡量搜索引擎排序結果好壞的指標。NDCG 背後的核心概念為：若能將愈相關的結果排序在愈前面的話，便可以得到愈高的分數。因此，將 NDCG 的概念套用到推薦系統的情境中，我們會希望推薦系統能將使用者喜歡的电影排序在推薦清單中愈前面的位置愈好。以下便展示 NDCG@k 的計算方式，其中的 IDCG 是當推薦系統能完美的將推薦結果正確排序時，可以得到的最大 DCG 分數：

$$NDCG@k = \frac{DCG}{IDCG}$$

$$DCG@k = \frac{1}{k} \sum_{j=1}^k \frac{rel_j}{\log(1+j)}$$

k：只關注前 k 個排序的結果

rel_j ：推薦清單中第 j 個電影的分數

■ 評估指標 Precision@k、Recall@k

Precision@k 與 Recall@k 分別代表 Precision at k 和 Recall at k。在推薦系統的情境中，首先我們會依照使用者給定的評分來將電影分成喜歡與不喜歡兩類，若某部電影的評分高於使用者的平均評分，我們就將該電影視為使用者喜歡的電影；反之，若某部電影的評分低於使用者的平均評分，我們就將該電影視為使用者不喜歡的電影。這時 Precision@k 代表當我們推薦 k 個電影給某個使用者時，在這 k 部電影中有多少部電影是使用者喜歡的。而 Recall@k 則代表使用者所有喜歡的電影，有多少部出現在推薦清單中的前面 k 個位置中。以下便展示 Precision@k 與 Recall@k 的計算方式：

$$Precision@k = \frac{\text{num of recommended items at k that user likes}}{k}$$

$$\text{Recall@k} = \frac{\text{num of recommended items at k that user likes}}{\text{total num of items that user likes}}$$

4.2 訓練流程與實驗參數設定

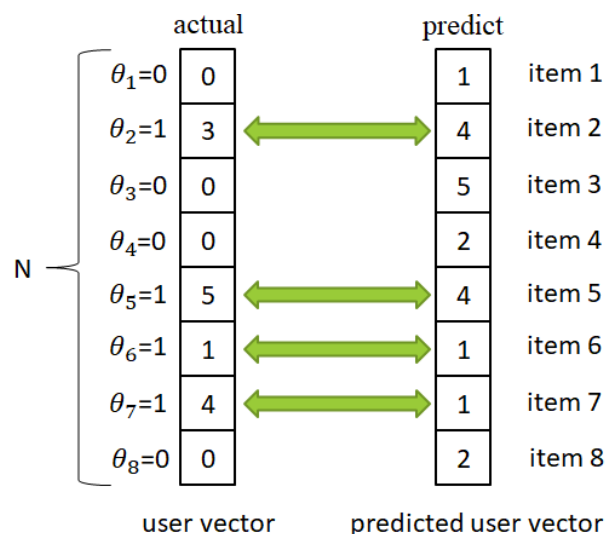
● 訓練流程

我們會將 MovieLen 1M 中，評分數量超過 25 個商品的使用者當作訓練使用者(training user)，而評分數量小於 25 個評分的使用者當作測試使用者(test user)，依此分割方法得到的分割結果如下表：可以看到訓練集中的使用者總共有 5549 個，而這 5549 個使用者總共對 3702 部電影有評過分。在測試集中，則有 491 個使用者，並且他們只有對 1995 部電影有評分。

	Training set		Test set	
Split methods	Num of users	Num of movies	Num of users	Num of movies
Cold Start split	5549	3702	491	1995

表二、訓練與測試使用者統計表

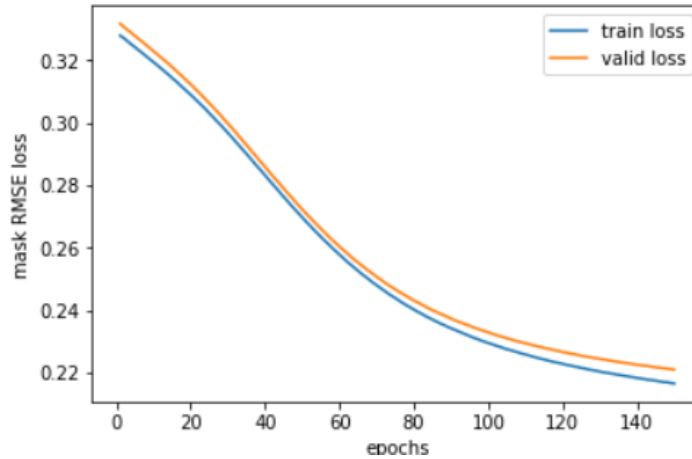
在訓練模型時，我們使用前述切割出的訓練集使用者來訓練模型。而訓練過程中，我們的 Denoising Autoencoder 模型要優化的損失函數是 masked RMSE，masked RMSE 的計算方法是針對使用者真實評分過的商品計算預測評分與真實評分之間的誤差，示意圖與公式如下：



圖十八、masked RMSE 示意圖

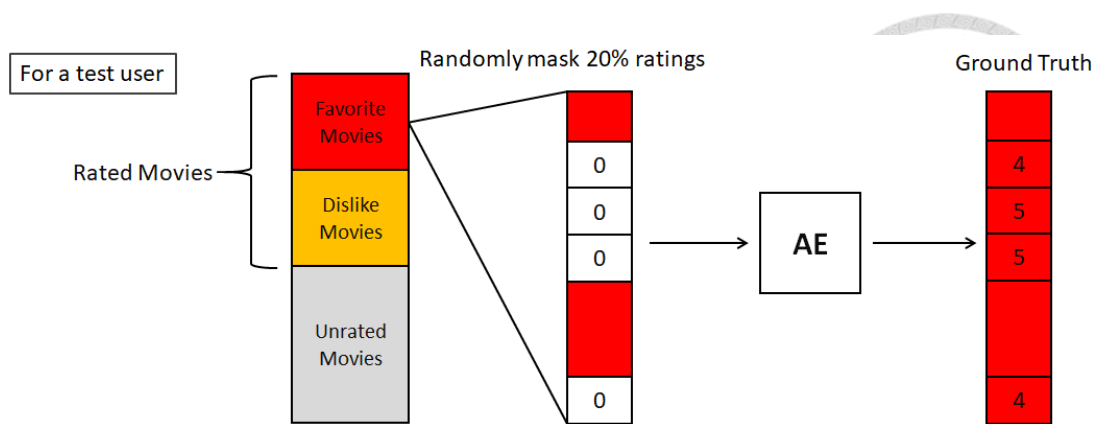
$$\sqrt{\frac{1}{N} \sum_{i=1}^N \theta_i * (\text{predict}_i - \text{actual}_i)^2}, \begin{cases} \theta_i = 1, \text{if } \text{actual}_i \neq 0 \\ \theta_i = 0, \text{if } \text{actual}_i = 0 \end{cases}$$

其中 N 代表所有的商品數量， actual_i 代表使用者對於商品 i 的真實評分； predict_i 代表我們 DAE 模型預測使用者對於商品 i 的預測評分。而 $\theta_i \in \{0, 1\}$ ，只有當 $\text{actual}_i \neq 0$ 時， $\theta_i = 1$ ，否則 $\theta_i = 0$ 。最後我們會透過訓練類神經網路常用的參數更新演算法 SGD (Stochastic Gradient Descent)，來更新模型的參數。下圖為訓練 DAE 時，訓練誤差隨著訓練次數(epochs)的圖，可以發現訓練次數大概在 150 次時訓練誤差達到收斂，因此我們在之後的實驗中設定訓練次數為 150 次。



圖十九、訓練誤差與訓練次數圖

在評估模型訓練的成效時，我們會將每個測試使用者喜歡的 20% 電影評分隨機覆蓋為 0，當作 NDCG@10、P@10、R@10 三個評估指標排序時要驗證的正確答案，最後將覆蓋後的測試使用者向量輸入訓練完成的 Autoencoder 得到預測結果，並依預測結果與原本的評分值計算 NDCG@10、P@10、R@10。示意圖如下：



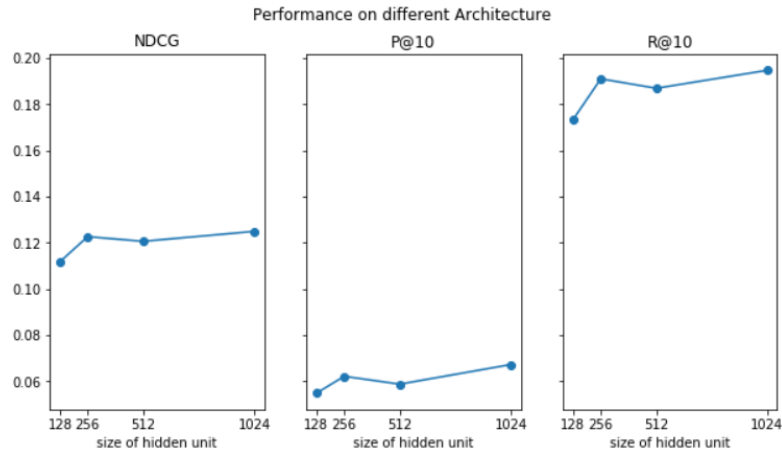
圖二十、覆蓋 20%測試使用者喜歡的評分示意圖

● 超參數設定

我們的模型總共有三部分的超參數需要決定：

第一部分是將使用者分群時，首先我們會濾掉資訊量不足的電影，意即電影 HELF 值過小的電影，我們設定該 HELF 門檻值為 0.65。接著我們會將過濾後的電影當作使用者向量的特徵，以此來計算使用者與使用者之間的距離進行後續的分群。在分群時，我們會依照 Decision diagram 來選擇適當的群數。

第二部分需要決定的超參數是 Autoencoder 模型的架構、具代表性電影的丟棄比例與不具代表性的丟棄比例，經過實驗我們將 Autoencoder 模型的架構設定為 $[N, 1024, N]$ ， N 代表以資料集中的所有電影來當作使用者向量的維度，1024 代表使用者向量經過類神經網路壓縮成 1024 維的向量。而具代表性與不具代表性電影的丟棄比例我們分別設定為 0.4、0.8。以下為 Autoencoder 模型架構的實驗結果，可以發現當 Autoencoder 中間層的單元數為 1024 時表現最好。



圖二十一、Autoencoder 模型中間層實驗圖

第三部分的超參數便是決定訓練模型時所使用的最佳化演算法 (optimizer)、啟動函數(activation function)、以及模型參數初始方法。經過實驗後我們發現使用 SGD 最佳化演算法搭配 lecun normal 參數初始方法和 selu 啟動函數可以得到最佳的訓練效果。

● 基準模型

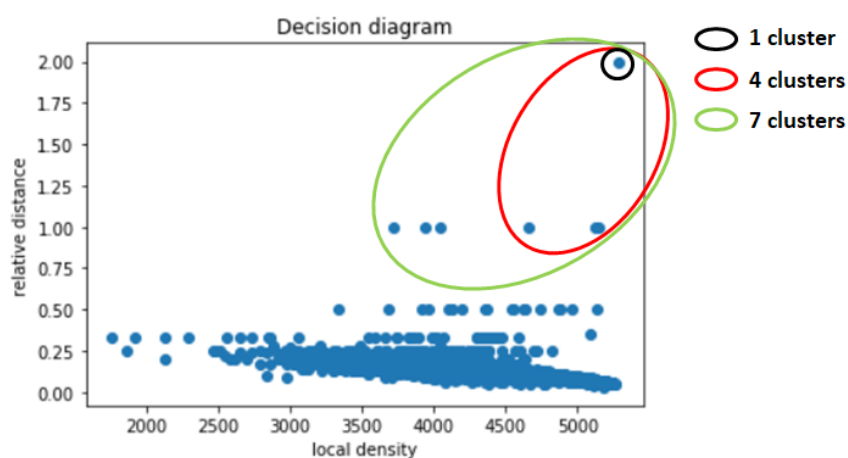
為了檢視我們所提出的「返老還童」機制是否有效，我們使用兩種基準模型(baseline model)和我們提出的模型來比較效能。

1. Autoencoder: 使用最原始的 Autoencoder 模型，並且神經網路架構為 $[N, 1024, N]$ 。
2. Random Autoencoder: 使用最原始的 Autoencoder，神經網路架構為 $[N, 1024, N]$ ，並且在訓練時隨機的覆蓋掉一些評分。

● 分群數量如何影響實驗效能

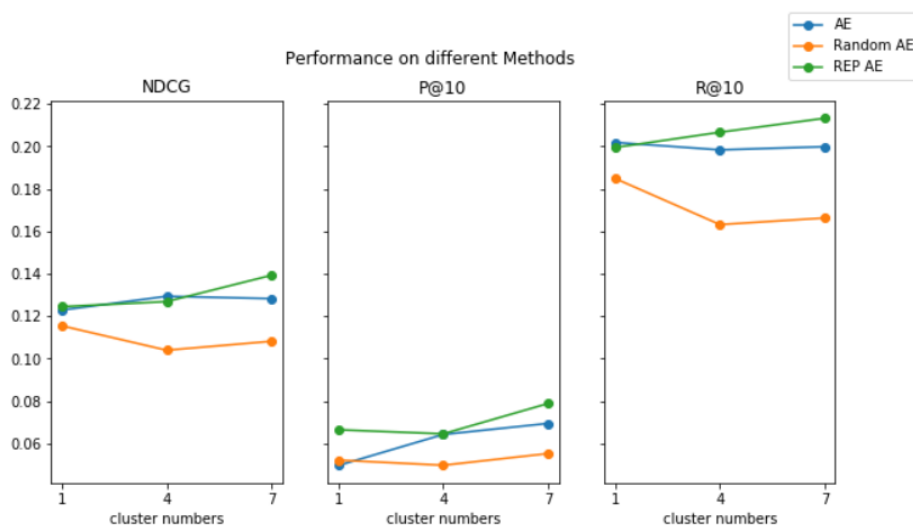
將 MovieLen 1M 資料集中的使用者分群時，我們會依照 Decision diagram 來選擇適當的群數。下圖為進行使用者分群時產生的 Decision diagram，圖上的每個點分別代表一個使用者，而每一個橢圓框代表將所有使用者分 1、4、7 群時，這些群數對應的群中心使用者們。在之後的分群實驗中，我們之所以選擇把所有使用者分成 1、4、7 群的原因是，這些群中心

的使用者符合 DPC 分群演算法群中心的特性：群中心的區域密度要夠大，並且群中心與群中心的相對距離也要夠大。



圖二十二、DPC Decision Diagram 實驗圖

我們實驗不同群數對我們的模型和其他基準模型，在三種效能評估指標上的差異，並整理於下圖中。可以發現隨著分群的數量增加我們方法的表現也逐漸上升。



圖二十三、基準模型與我們方法的分群實驗圖

4.3 所有模型效能比較

在此章節中，會比較我們的方法和前一小節提到的基準模型，以及現今其他模型[17] [19]在冷啟動資料測試集中的推薦效能，並且設定顯著水準為 0.05，然後再分別對於每個方法對於每個評估指標抽樣 40 次，計算他們的

t 檢定統計量，以此檢測我們的方法對於其他方法的效能是否在統計上有顯著的差異。以下我們會先介紹模型[17] [19]，最後將各個模型的推薦效能整理於以下表格中：

- CDAE [17]：如同在文獻回顧中所介紹的，CDAE 接收使用者對各個商品的喜好值(若為 1：代表使用者喜歡此商品；若為 0：代表使用者尚未看過此商品)和該使用者的 ID 當作模型輸入。在這裡我們將使用者對各個商品的喜好值改成使用者對於商品的評分。然後以 0.4 的機率隨機將某些輸入向量維度設為 0，再經過一層維度為 512 的隱藏層，最後將輸出層維度設定為輸入向量的維度。
- Dual Autoencoder [19]：如同在文獻回顧中所介紹的，Dual Autoencoder 會在訓練模型時，同時訓練使用者與商品的向量表示法，我們在此設定使用者與商品向量壓縮後的維度為 1024，並將壓縮後的使用者向量和商品向量內積來當作模型的預測輸出。

Models	NDCG@10	P@10	R@10
Autoencoder	0.1196**	0.0573	0.1934**
Random Autoencoder	0.1087**	0.0510**	0.1754**
CDAE	0.1120**	0.0493**	0.1865**
Dual Autoencoder	0.0875**	0.0395**	0.1457**
Our Method	0.1321	0.0689	0.2029

表三、所有模型效能比較表

表中的結果若有**，表示我們的方法在 95%的信心水準之下，有顯著的優於其他方法。此處的基準模型(Autoencoder、Random Autoencoder)與我們提出的模型皆為沒做分群的效能。

由上表可以看到我們提出的方法，效能在統計上顯著的優於其他方法，可能的原因在於，我們選出的具代表性商品所包含的資訊量是有助於將冷啟動狀態的使用者還原回正常狀態的使用者，並且這種現象會隨著分割使用者的群數愈多而有愈好的表現，因為分群的效果會將有相似特性的使用者分到

同一組，對於同一組的使用者來說，我們可以更精準的選出代表這一群使用的具代表性商品。至於 CDAE 模型的表現不如我們模型好，可能的原因在於 CDAE 模型是隨機的覆蓋使用者向量中的某些商品來產生雜訊，導致後續 Autoencoder 無法如我們的方法般有效進行訓練。而 Dual Autoencoder 模型表現最差的原因，可能在於雖然該架構能同時訓練使用者的低維表示向量和商品的低維表示向量，不過最終在模型輸出時的預測僅使用使用者低維向量與商品低維向量做內積，可能沒有辦法順利的捕捉使用者與商品之間複雜的非線性關係，以至於最終的成效不佳。

5 結論：

本篇論文主要專注於解決推薦系統中常見的冷啟動使用者問題，我們提出了一種稱為「使用者返老還童」的機制，該機制首先會為不同使用者族群選出該群使用者的具代表性商品，並且透過將富有評分資訊的使用者向量隨機覆蓋大部分的不具代表性商品維度為 0 分，以及覆蓋小部分的具代表性商品維度為 0 分後，以此來模擬使用者冷啟動的狀態。最後，我們使用被「使用者返老還童」機制還原的冷啟動使用者來為每一群使用者訓練一個深度學習模型 Denoising Autoencoder。當模型訓練完成後，模型便有能力把輸入的冷啟動使用者順利還原成富有資訊的使用者狀態，並且進行推薦。

總結我們的論文主要有兩大貢獻，第一、我們是第一個提出雜訊產生並結合 Denoising Autoencoder 的概念來解決推薦系統中的冷啟動問題。第二、我們所提出的「使用者返老還童」機制可以有效的幫助 Denoising Autoencoder 進行訓練。

而在未來我們除了可以透過商品資訊量的多寡來為每一群使用者們挑選出具代表性商品，我們可能還可以額外的考慮商品被評分的時間順序作為依據來進一步的模擬冷啟動狀態的使用者。以及，我們希望可以持續的改進模型的架構而不是使用只有一層隱藏層的 Denoising Autoencoder，嘗試將模型的架構變得更複

雜以此來增加模型學習的能力。最後，我們也應該在更多不同的資料集上進行實驗，以此證明我們方法的一般性。



I. 參考文獻整理：

- [1] H.-T. Cheng, et al., Wide & deep learning for recommender systems, in: Proceedings of the 1st workshop on deep learning for recommender systems, (2016), pp. 7-10.
- [2] P. Covington, et al., Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM conference on recommender systems, (2016), pp. 191-198.
- [3] M. Fu, et al., A novel deep learning-based collaborative filtering model for recommendation system, IEEE transactions on cybernetics, 49(3) (2018) 1084-1096.
- [4] O. Georgiou, N. Tsapatsoulis, The importance of similarity metrics for representative users identification in recommender systems, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, (Springer, 2010), pp. 12-21.
- [5] C.A. Gomez-Uribe, N.J.A.T.o.M.I.S. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Transactions on Management Information Systems, December 2015 Article No.: 13, 6(4) (2015) 1-19.
- [6] X. He, et al., Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, (2017), pp. 173-182.
- [7] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems, 20(4) (2002) 422-446.
- [8] N.N. Liu, et al., Wisdom of the better few: cold start recommendation via representative based rating elicitation, in: Proceedings of the fifth ACM conference on Recommender systems, (2011), pp. 37-44.
- [9] A. Majumdar, A. Jain, Cold-start, warm-start and everything in between: an autoencoder based approach to recommendation, in: 2017 International Joint Conference on Neural Networks (IJCNN), (IEEE, 2017), pp. 3656-3663.
- [10] A.M. Rashid, et al., Learning preferences of new users in recommender systems: an information theoretic approach, ACM Sigkdd Explorations Newsletter, 10(2) (2008) 90-100.
- [11] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science, 344(6191) (2014) 1492-1496.
- [12] L. Shi, et al., Local representative-based matrix factorization for cold-start recommendation, ACM Transactions on Information Systems, 36(2) (2017) 1-28.
- [13] S. Shi, et al., Attention-based adaptive model to unify warm and cold starts

recommendation, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, (2018), pp. 127-136.

[14] P. Vincent, et al., Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of machine learning research, 11(Dec) (2010) 3371-3408.

[15] M. Volkovs, et al., Dropoutnet: Addressing cold start in recommender systems, in: Advances in Neural Information Processing Systems, (2017), pp. 4957-4966.

[16] J. Wei, et al., Collaborative filtering and deep learning based recommendation system for cold start items, Expert Systems with Applications, 69(2017) 29-39.

[17] Y. Wu, et al., Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, (2016), pp. 153-162.

[18] S. Zhang, et al., Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys, 52(1) (2019) 1-38.

[19] F. Zhuang, et al., Representation learning via Dual-Autoencoder for recommendation, Neural Networks, 90(2017) 83-89.

