

Neural Collaborative Filtering

He et. al, 2017

Revisión de artículo

Introducción

Contexto y problemática

Aprendiendo de datos implícitos

Factorización matricial

Introducción

Problema

- Esfuerzo solo en mejorar MFs.
- Limitada solo a producto punto.
- El producto punto quizá no es suficiente para capturar la interacción de los datos.

Propuesta

- Usar DNN para aprender la función de interacción de los datos.
- Enfoque en implicit feedback.
- ANN capaces de aproximar cualquier función continua.
- DNNs han demostrado buenos resultados en RecSys.

Contexto

Learning from Implicit Data

Matriz de interacción Usuario-Item $\mathbf{Y} \in \mathbb{R}^{M \times N}$

$$y_{ui} = \begin{cases} 1, & \text{if interaction (user } u, \text{ item } i) \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

Se desea estimar las puntuaciones de los valores no observados de \mathbf{Y}

$$\hat{y}_{ui} = f(u, i | \Theta)$$

Existen 2 tipos de funciones objetivo:

- Pointwise loss: minimiza la pérdida cuadrada entre \hat{y}_{ui} y y_{ui}
- Pairwise loss: maximiza el margen entre \hat{y}_{ui} observada y una \hat{y}_{ui} no observada

Contexto

Matrix Factorization

MF asocia cada item y usuario a un valor de un espacio latente.

$$\hat{y}_{ui} = f(u, i | \mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^T \mathbf{q}_i = \sum_{k=1}^K p_{uk} q_{ik}$$

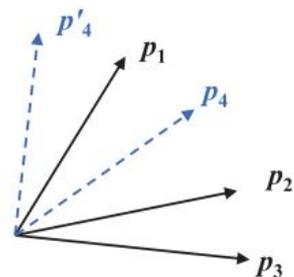
Se estima una interacción y_{ui} como producto punto de \mathbf{p} y \mathbf{q} .

- Es limitada en expresividad.
- Limitación de estimación de interacciones.
- Ejemplo: incurre en pérdida de ranking.

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

↑ users
← items

(a) user-item matrix



(b) user latent space

Solución

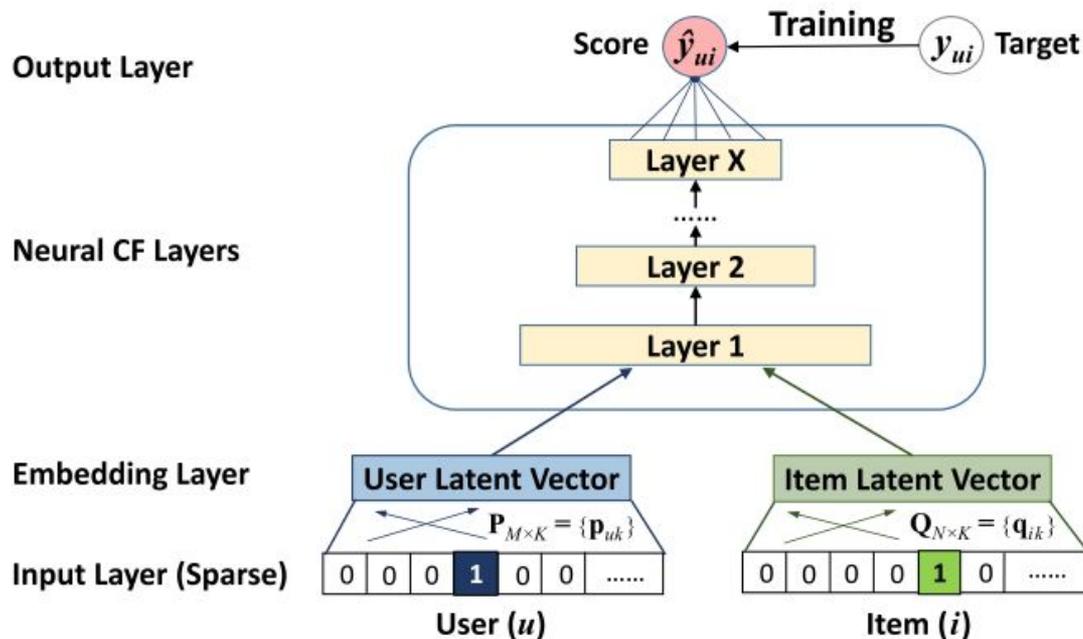
Neural Collaborative Filtering
(NCF)

Factorización matricial
generalizada

Perceptrón multicapa

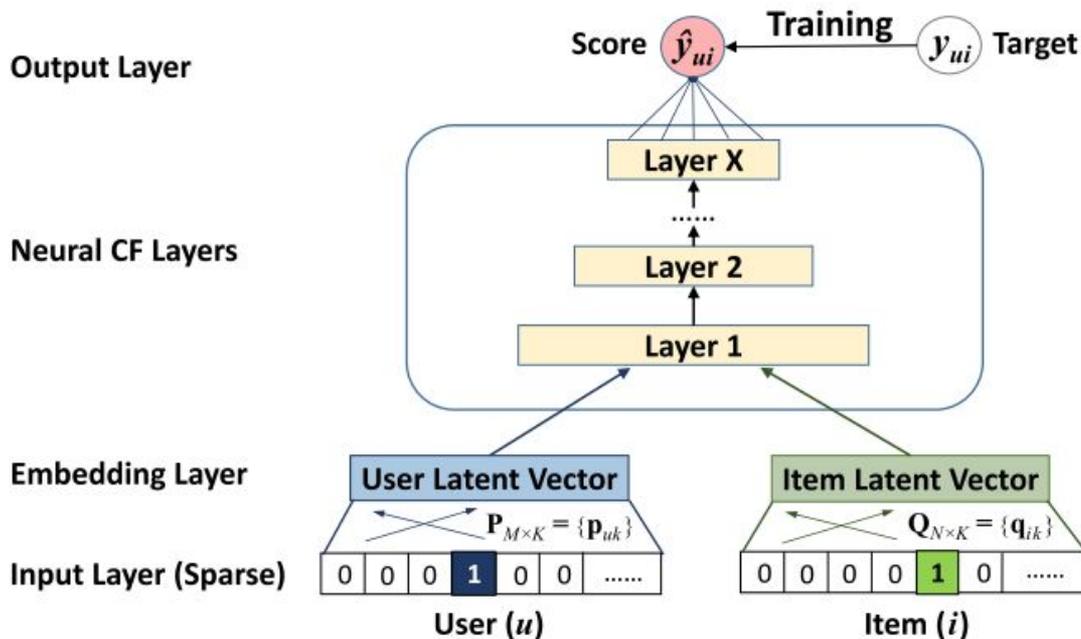
Sistema híbrido embebido

Framework General



- Representación multi-capa para modelar interacción usuario-item y_{ui}
- Identidad de un usuario y un item como entrada (one-hot)

Framework General



- Capa de embedding, genera vectores densos.
- NCF mapea vectores latentes.
- Cada capa puede ser personalizada.
- Capa final de predicción de score.

Framework General

Entrenamiento mediante pointwise learning y optimización con SCG.

Función objetivo (binary cross entropy loss, log loss)

$$\begin{aligned} L &= - \sum_{(u,i) \in \mathcal{Y}} \log \hat{y}_{ui} - \sum_{(u,j) \in \mathcal{Y}^-} \log(1 - \hat{y}_{uj}) \\ &= - \sum_{(u,i) \in \mathcal{Y} \cup \mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \end{aligned}$$

Recomendación con implicit feedback como clasificación binaria.

Factorización matricial generalizada (GMF)

Los autores definen una función de mapeo como cuasi producto interno:

$$\phi_1(\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u \odot \mathbf{q}_i$$

Con la que generan una predicción

$$\hat{y}_{ui} = a_{out}(\mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_i))$$

Factorización matricial generalizada (GMF)

MF es un caso especial de GMF con

- \odot , producto punto
- h , vector unitario
- función de activación igual a la identidad

GMF puede trabajar con entornos no lineales al especificar diferentes funciones de activación.

Perceptrón multicapa (MLP)

La primera capa concatena las entradas de los vectores latentes.

$$\mathbf{z}_1 = \phi_1(\mathbf{p}_u, \mathbf{q}_i) = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}$$

Las siguientes capas utilizan la arquitectura de predicción en cada neurona

$$\begin{aligned} \phi_2(\mathbf{z}_1) &= a_2(\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_2), \\ &\dots\dots \\ \phi_L(\mathbf{z}_{L-1}) &= a_L(\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L), \end{aligned}$$

Perceptrón multicapa (MLP)

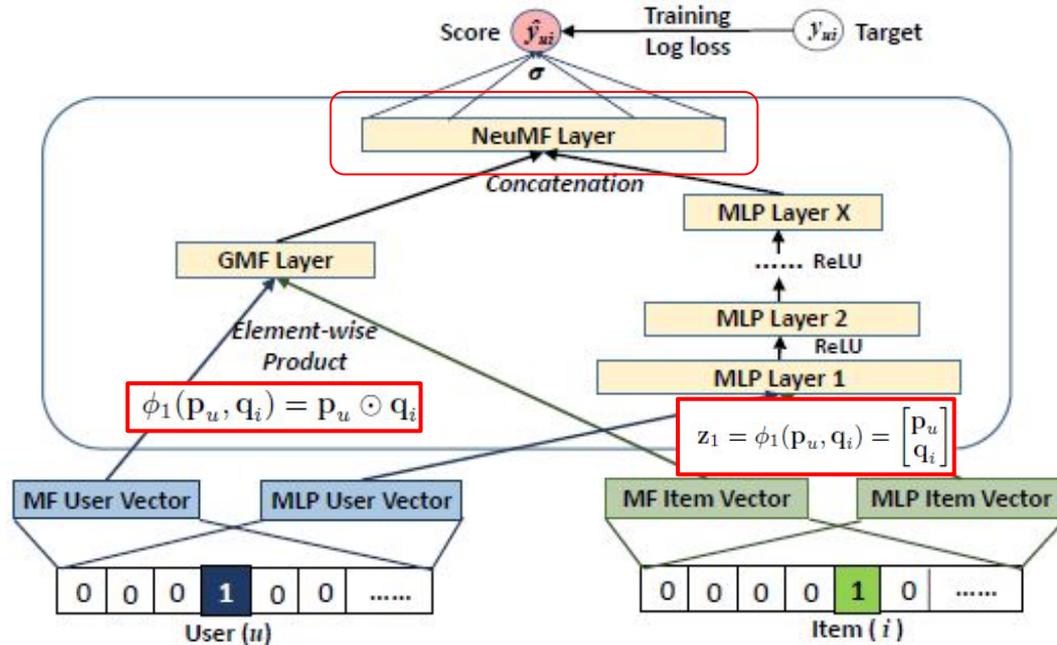
Esta arquitectura elimina el producto elemento a elemento de GMF.

La función de activación comúnmente elegida es ReLU para las capas ocultas.

La capa final utiliza una sigmoide y da la predicción.

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \phi_L(\mathbf{z}_{L-1}))$$

Sistema híbrido embebido NCF



Para fusionar estas dos aproximaciones cada técnica puede compartir la primera capa embebida y luego sus salidas.

Sistema híbrido embebido NCF

Para darle flexibilidad al modelo se le permite a cada técnica aprender de manera separada y luego combinar sus salidas.

$$\phi^{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G,$$

$$\phi^{MLP} = a_L(\mathbf{W}_L^T(a_{L-1}(\dots a_2(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2)\dots)) + \mathbf{b}_L),$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}),$$

Sistema híbrido embebido NCF

La no convexidad del sistema puede llevar a soluciones locales.

El autor sugiere pre-entrenar GMF y MLP con varios puntos iniciales aleatorios.

Utilizó estimación adaptativa de momentos (Adam) para adaptar la razón de aprendizaje de la red.

Experimentos

[https://github.com/hexiangnan/
neural_collaborative_filtering](https://github.com/hexiangnan/neural_collaborative_filtering)

Preguntas científicas

Definición de experimentos

Comparación de desempeño

Logaritmo de la pérdida para
muestreo negativo

¿Ayuda el aprendizaje profundo?

Preguntas científicas

RQ1: ¿Puede NeuMF sobrepasar al desempeño de filtros colaborativos del estado del arte con información implícita?

RQ2: ¿Cómo funciona este marco de optimización para tareas de recomendación?

RQ3: ¿Son útiles las capas ocultas para aprender de interacciones usuario-item?

Definición de experimentos

Datasets

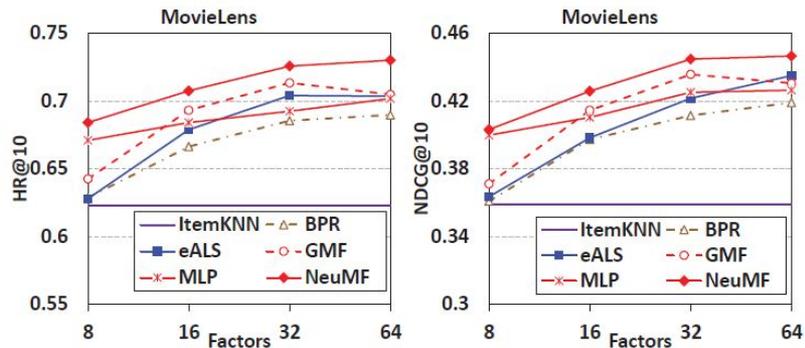
- MovieLens: Más de un millón de ratings con por lo menos 20 ratings por usuario.
- Pinterest: Más de 55 mil usuarios con sobre el millón y medio de interacciones. Datos altamente escasos.

Definición de experimentos

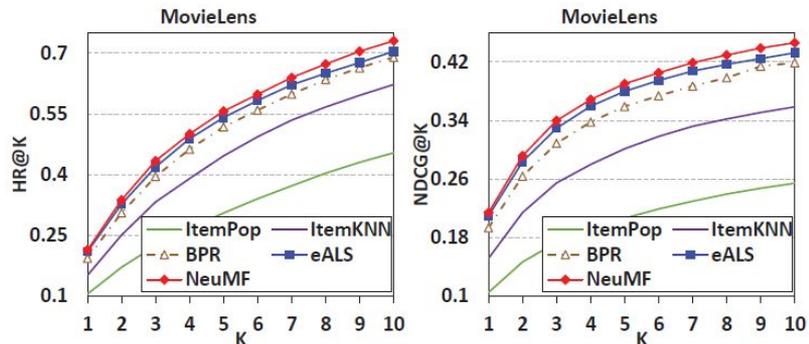
Métricas

- Hit ratio (HR@N): Truncada para medir muestras de los ítems e indicar de manera intuitiva si un ítem de test está presente en un top 10 de recomendación.
- Normalized Discount Cumulative Gain (NDCG@N): Va a observar la posición del ítem relevante y asigna un valor a ésta.

Comparación de desempeño (RQ1)



(a) MovieLens — HR@10 (b) MovieLens — NDCG@10

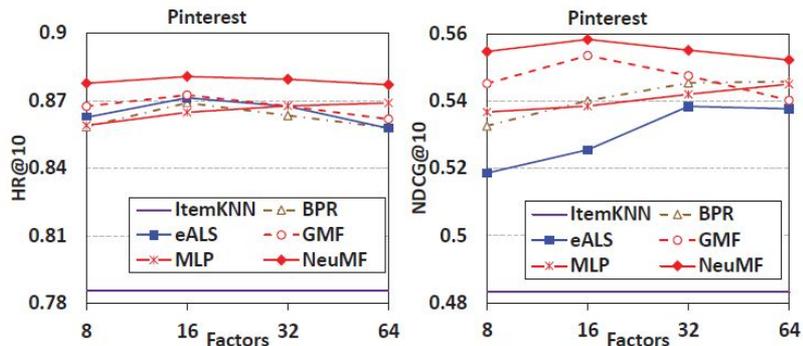


(a) MovieLens — HR@K (b) MovieLens — NDCG@K

Se utilizan cuatro modelos para comparar:

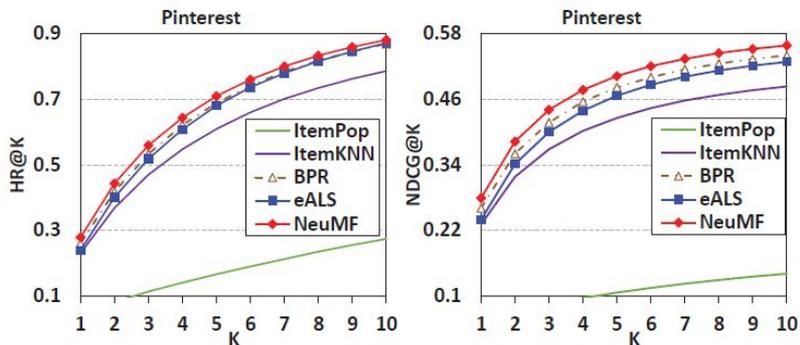
- Item most popular
- Knn para item
- BPR
- eALS

Comparación de desempeño (RQ1)



(c) Pinterest — HR@10

(d) Pinterest — NDCG@10



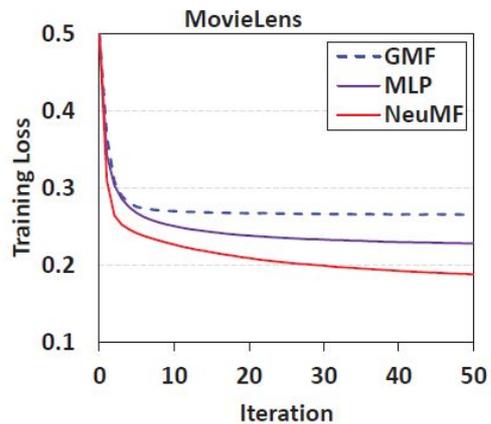
(c) Pinterest — HR@K

(d) Pinterest — NDCG@K

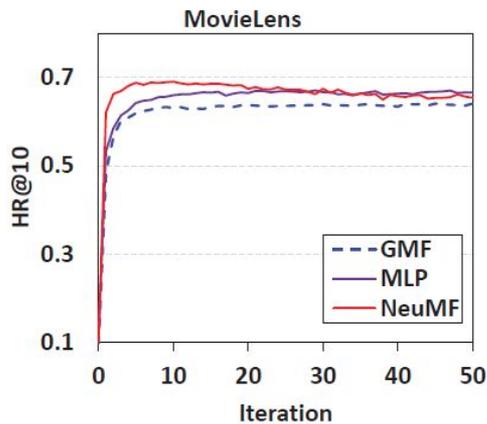
NeuMF logra superar a todos los métodos con ambas métricas.

El artículo no informa de otras métricas conocidas, o de cómo generaliza la red.

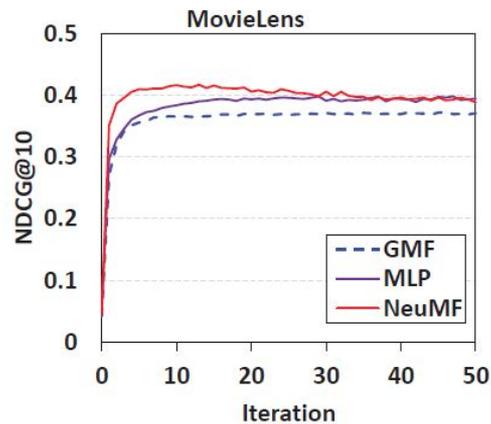
Logaritmo de la pérdida (RQ2)



(a) Training Loss



(b) HR@10



(c) NDCG@10

Existe una cantidad de iteraciones óptima para entrenamiento.

Más allá se observa overfitting en la red. Peor recomendación.

¿Es útil el aprendizaje profundo? (RQ3)

Adherir capas (profundidad) a la DNN ayuda al sistema a mejorar su desempeño.

Table 3: HR@10 of MLP with different layers.

Factors	MLP-0	MLP-1	MLP-2	MLP-3	MLP-4
MovieLens					
8	0.452	0.628	0.655	0.671	0.678
16	0.454	0.663	0.674	0.684	0.690
32	0.453	0.682	0.687	0.692	0.699
64	0.453	0.687	0.696	0.702	0.707
Pinterest					
8	0.275	0.848	0.855	0.859	0.862
16	0.274	0.855	0.861	0.865	0.867
32	0.273	0.861	0.863	0.868	0.867
64	0.274	0.864	0.867	0.869	0.873

Conclusiones

- Se presenta un framework con 3 instancias, para aprender interacciones de 2 entidades.
- No se limita a los modelos presentados.
- Abre las posibilidades para nuevas investigaciones.

Trabajo Futuro

- Pairwise Learners
- Extensiones con información adicional como reviews, bases de conocimiento, señales temporales.

Gracias