



ТЕХНОСФЕРА

Лекция 7 Recomendation Systems 2

Владимир Гулин

31 марта 2018 г.

План лекции

Напоминание

RBM for collaborative filtering

Netflix

Item embeddings & 2vec models

Deep Collaborative Filtering

Постановка задачи

Дано:

- ▶ Users, $u \in U$
- ▶ Items, $i \in I$
- ▶ Marks, $r_{ui} \in R$
- ▶ Events, $(u, i, r_{ui}, \dots) \in \mathcal{K}$

Задача:

- ▶ Выделить похожие объекты

$$i \rightarrow \{i_1, \dots, i_{I_S}\}$$

- ▶ Найти похожих пользователей

























$$u \rightarrow \{u_1, \dots, u_{U_S}\}$$

- ▶ Предсказать

$$\hat{r}_{ui} = h(u, i, \dots) \approx r_{ui}$$

Collaborative Filtering

Восстанавливаем пропущенные значения матрицы

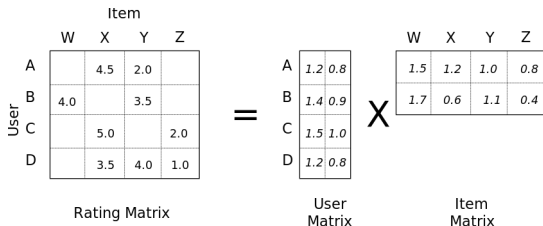
				
				
				
				
				
				

Model-based CF

Идея:

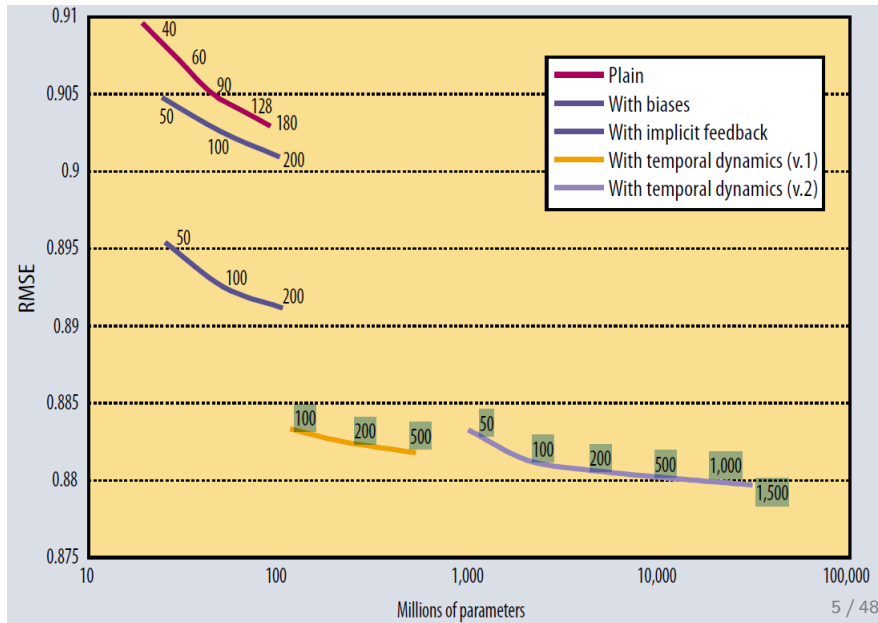
Будем искать решение в виде

$$\hat{r}_{ui} = \mathbf{p}_u^T \mathbf{q}_i$$

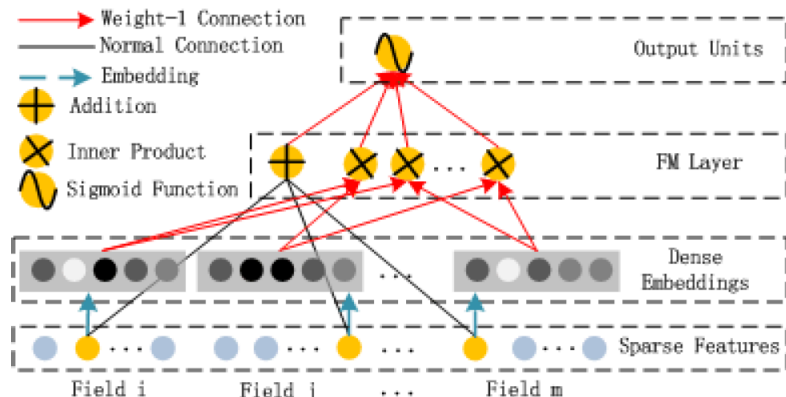


$$\sum_{(u,i) \in \mathcal{K}} (r_{ui} - \mathbf{p}_u^T \mathbf{q}_i)^2 \rightarrow \min_{\mathbf{p}, \mathbf{q}}$$

Сравнительный анализ на данных Netflix Prize



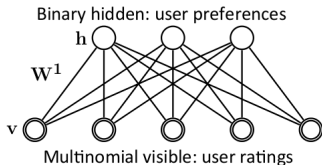
Factorization Machines As Neural Network



$$\hat{y}(x) = w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \mathbf{v}_j^T \mathbf{v}_{j'}$$

RBM for collaborative filtering

$$P_{\theta}(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp \left(\sum_{ijk} W_{ij}^k v_i^k h_j + \sum_{ik} b_i^k v_i^k + \sum_j a_j h_j \right)$$



Netflix dataset:

480,189 users

17,770 movies

Over 100 million ratings



Learned features: ``genre''

Fahrenheit 9/11
Bowling for Columbine
The People vs. Larry Flynt
Canadian Bacon
La Dolce Vita

Friday the 13th
The Texas Chainsaw Massacre
Children of the Corn
Child's Play
The Return of Michael Myers

Independence Day
The Day After Tomorrow
Con Air
Men in Black II
Men in Black

Scary Movie
Naked Gun
Hot Shots!
American Pie
Police Academy

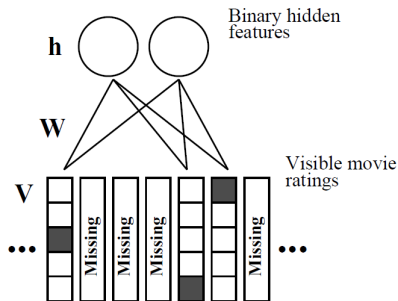
State-of-the-art performance
on the Netflix dataset.

(Salakhutdinov, Mnih, Hinton, ICML 2007)

RBM for collaborative filtering

Дано:

- ▶ Users, $u \in N$
- ▶ Movies, $i \in M$
- ▶ Marks, $1, 2, 3, 4, K = 5$
- ▶ Hidden vars, $h_j, j = 1, \dots, F$
- ▶ W_{ij}^k - interaction between feature j and rating k of film i
- ▶ b_i^k - bias of rating k for movie i
- ▶ b_j - bias of feature j



$$p(v_i^k = 1|\mathbf{h}) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)}, \quad p(h_j = 1|\mathbf{V}) = \sigma(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k)$$

RBM for collaborative filtering

Marginal distribution over visible ratings \mathbf{V}

$$p(\mathbf{V}) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{V}, \mathbf{h}))}{\sum_{\mathbf{V}', \mathbf{h}'} \exp(-E(\mathbf{V}', \mathbf{h}'))}$$

Energy

$$E(\mathbf{V}, \mathbf{h}) = - \sum_{i=1}^m \sum_{j=1}^F \sum_{k=1}^K W_{ij}^k h_j v_i^k + \sum_{i=1}^m \log Z_i - \sum_{i=1}^m \sum_{k=1}^K v_i^k b_i^k - \sum_{j=1}^F h_j b_j$$

$$Z_i = \sum_{l=1}^K \exp(b_i^l + \sum_j h_j W_{ij}^l)$$

RBM for collaborative filtering

Learning

$$\delta W_{ij}^k = \epsilon \frac{\partial \log p(\mathbf{V})}{\partial W_{ij}^k} = \epsilon (\langle v_i^k h_j \rangle_{data} - \langle v_i^k h_j \rangle_{model})$$

$\langle v_i^k h_j \rangle_{data}$ - частота, с которой фильм i с рейтингом k и фичой j вместе встречаются в обучающем датасете

Вопрос:

- ▶ Что такое $\langle v_i^k h_j \rangle_{model}$ и как его посчитать?

RBM for collaborative filtering

Learning

$$\delta W_{ij}^k = \epsilon \frac{\partial \log p(\mathbf{V})}{\partial W_{ij}^k} = \epsilon (\langle v_i^k h_j \rangle_{data} - \langle v_i^k h_j \rangle_T)$$

$\langle v_i^k h_j \rangle_T$ - ожидание, полученное Contrastive Divergence (семплирование T раз по Гиббсу из распределения)

Predictions

$$\begin{aligned} p(v_q^k = 1 | V) &= \sum_{\mathbf{h}} \exp(-E(v_q^k, \mathbf{V}, \mathbf{h})) = \\ &= \exp(v_q^k b_q^k) \prod_{j=1}^F (1 + \exp(\sum_{il} v_i^l W_{ij}^l + v_q^k W_{qj}^k + b_j)) \end{aligned}$$

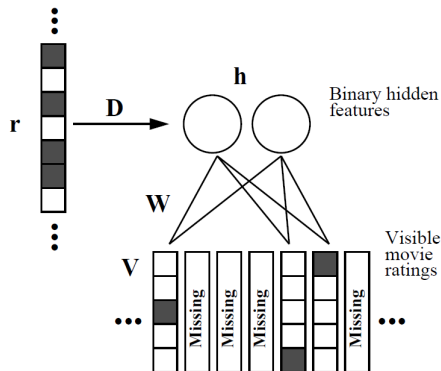
- Что брать в качестве предсказания модели?

Conditional RBM for collaborative filtering

Идея:

До этого момента мы рассматривали случаи, когда оценка пользователя для фильма заданна, однако в тестовых данных могут встречаться фильмы, для которых нет ни одной оценки в тренировочных данных. Давайте использовать такую информацию.

- $\mathbf{r} \in \{0, 1\}^M$ - инфа о фильмах с рейтингами



$$p(v_i^k = 1 | \mathbf{h}) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)}, \quad p(h_j = 1 | \mathbf{v}, \mathbf{r}) = \sigma(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k + \sum_i r_i D_{ij})$$

Conditional Factored RBM

Проблема:

Одним из недостатков рассмотренной модели является количество свободных параметров для $W \in R^{M \times K \times F}$. К примеру, при $M = 17770$, $K = 5$ и $F = 100$ (число латентных переменных) количество параметров около 9 миллионов, что является потенциальным источником переобучения.

Решение:

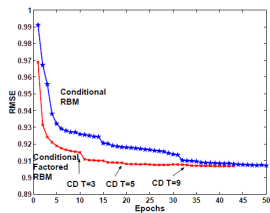
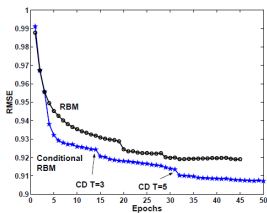
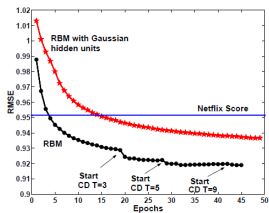
Факторизуем матрицу параметров W в произведения двух матриц низкого ранка A и B

$$W_{ij}^k = \sum_{c=1}^C A_{ic}^k B_{cj}$$

где $C \ll M$ и $C \ll F$.

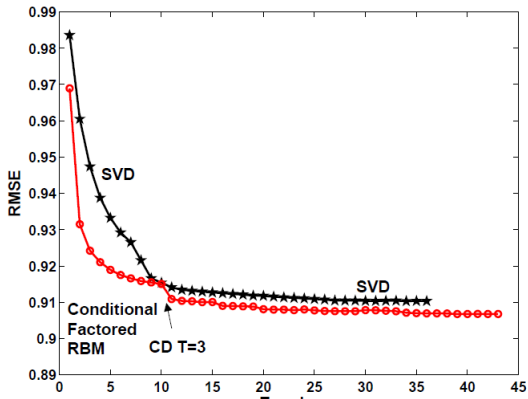
RBM for collaborative filtering

Результаты:



RBM for collaborative filtering

Сравнение с SVD:



Netflix Prize

- ▶ 480189 users
- ▶ 17770 movies
- ▶ 100480507 оценок $\{1, 2, 3, 4, 5\}$
- ▶ 2.10.2006 - 21.09.2009
- ▶ Призовой фонд 1 000 000\$
- ▶ Задача улучшить RMSE на 10% (с 0.9514 до 0.8563)

Netflix Prize

Основные модели №1

$$b_{ui} = \mu + b_u + \alpha_u dev_u(t_{ui}) + b_{u,t_{ui}} + (b_i + b_{i,Bin(t_{ui})}) \cdot c_u(t_{ui}) + b_{i,f_{ui}}$$

μ - глобальный bias

$b_i + b_{i,Bin(t_{ui})}$ - bias для фильма во времени

$b_u(t) = b_u + \alpha_u dev_u(t_{ui}) + b_{ut}$ - динамический bias user-a,

$dev_u(t) = sign(t - t_u) \cdot |t - t_u|^\beta$

$c_u(t_{ui})$ - шкалирующая поправка на оценку user-a во времени

$b_{i,f_{ui}}$ - bias фильма, в зависимости от того сколько пользователь поставил рейтингов в день t_{ui} .

Результат:

► $RMSE = 0.9278$

Основные модели №2

$$r_{ui} = b_{ui} + q_i^T(p_u(t_{ui}) + |N(u)|^{-0.5} \sum_{j \in N(u)} y_j)$$

$N(u)$ - множество всех фильмов, для которых пользователь поставил рейтинг (этот рейтинг неизвестен, если пример взят из тестового набора)

Результат:

- ▶ $RMSE = 0.8787(f = 200, iterations = 40)$

Netflix Prize

Основные модели №3

$$\begin{aligned} r_{ui} = & b_{ui} + |N(u)|^{-0.5} \sum_{j \in N(u)} e^{-\beta_u |t_{ui} - t_{uj}|} c_{ij} + \\ & + |R(u)|^{-0.5} \sum_{j \in R(u)} e^{-\beta_u |t_{ui} - t_{uj}|} (r_{uj} - \hat{b}_{uj}) w_{ij} + \\ & + \sum_{j \in R(u)} e^{-\gamma_u |t_{ui} - t_{uj}|} (r_{uj} - \hat{b}_{uj}) d_{ij} \end{aligned}$$

$R(u)$ - множество всех фильмов, для которых пользователь поставил рейтинг и он известен

Результат:

► $RMSE = 0.8870$

Netflix Prize

Основные модели №4

$$p(v_i^k = 1 | h, u, t, f) = \frac{\exp(b_{if}^k + b_{uf}^k + b_u^k + b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^5 \exp(b_{if}^l + b_{uf}^l + b_u^l + b_i^l + \sum_{j=1}^F h_j W_{ij}^l)}$$

Результат:

- ▶ $RMSE = 0.8950$

Основные модели №5

$$p(h_j = 1|V, r_t) = \sigma(b_j + \sum_{i=1}^M \sum_{k=1}^5 r_i^t v_i^k W_{ij}^k + \sum_{i=1}^M r_i^t D_{ij})$$

r_t - бинарный вектор, характеризующий набор фильмов, которые получили рейтинг в конкретный день t

Результат:

► $RMSE = 0.8920$

Netflix Prize

Blending

- ▶ GBDT
- ▶ trees = 200
- ▶ tree-size = 20
- ▶ shrinkage = 0.18
- ▶ sampling rate = 0.9

Результат:

- ▶ $RMSE = 0.8603(454predictors)$

Item embeddings & 2vec models

Prod2Vec (Yahoo 2016)

Идея:

Учим эмбединги на наборах товаров. Контекстами являются другие товары

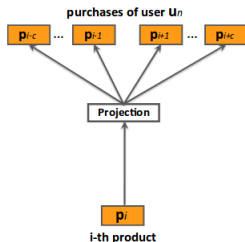


Figure 2: prod2vec skip-gram model

Bagged Prod2Vec

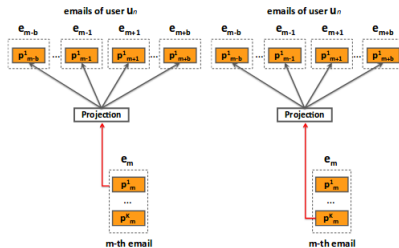


Figure 3: bagged-prod2vec model updates

User Prod2Vec

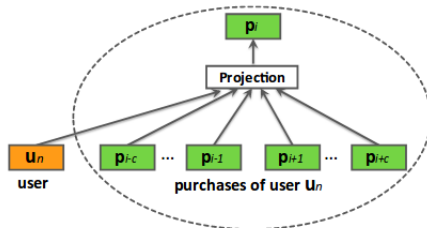


Figure 4: User embeddings for user to product predictions

Item2Vec (Microsoft 2016)

Идея:

Ищем эмбединги как в word2vec (SGNS), только убираем пространственные данные

People also like

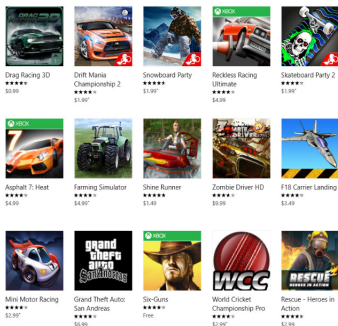


Fig. 1. Recommendations in Windows 10 Store based on similar items to Need For Speed.

$$\frac{1}{K} \sum_{i=1}^K \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{i+j} | w_i) \rightarrow \frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \log p(w_j | w_i)$$

Item2Vec

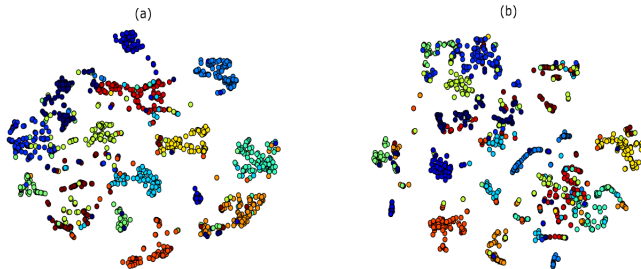
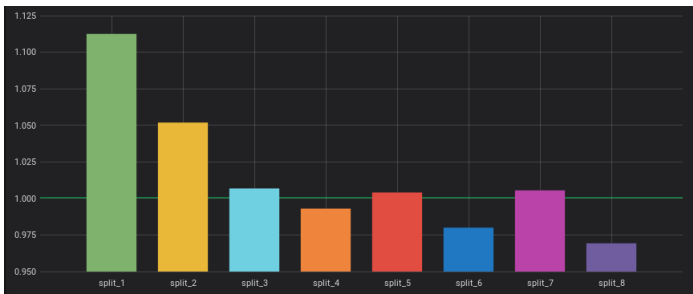


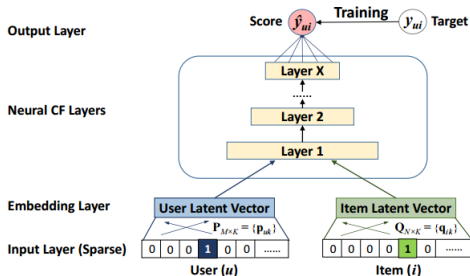
Fig.2: t-SNE embedding for the item vectors produced by item2vec (a) and SVD (b). The items are colored according to a web retrieved genre metadata.

Задача максимизации среднего чека

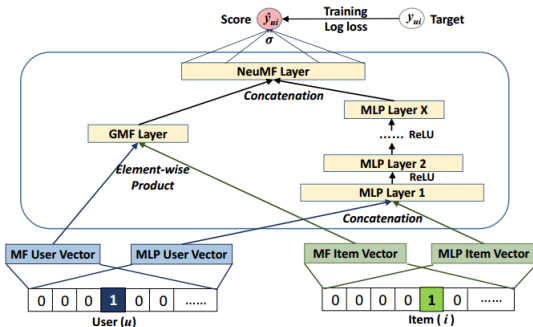


Deep Collaborative Filtering

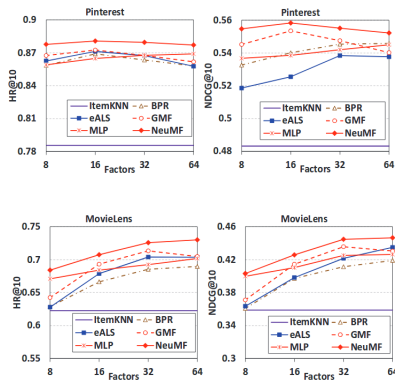
Neural Collaborative Filtering (2017)



Neural Collaborative Filtering



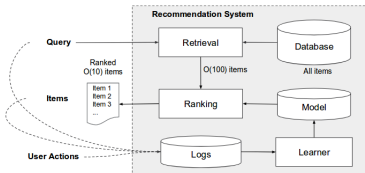
Neural Collaborative Filtering



Wide and Deep (Google 2016)

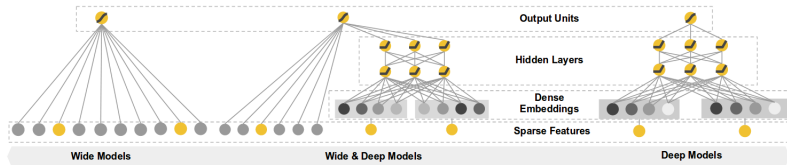
Задача:

Нужно увеличивать количество установок приложений в стор

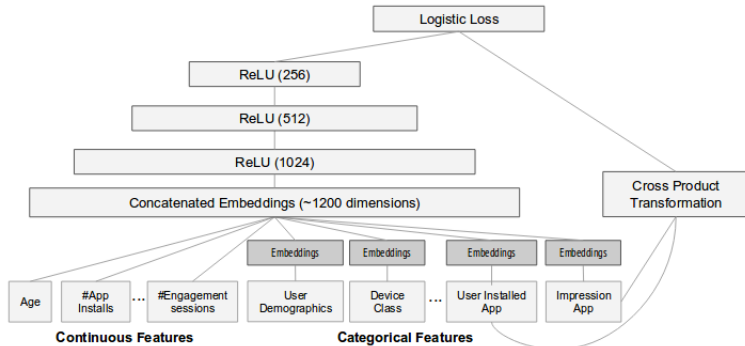


Архитектура рекомендательной системы

Wide and Deep (Google 2016)



Wide and Deep (Google 2016)



Wide and Deep (Google 2016)

Table 1: Offline & online metrics of different models. Online Acquisition Gain is relative to the control.

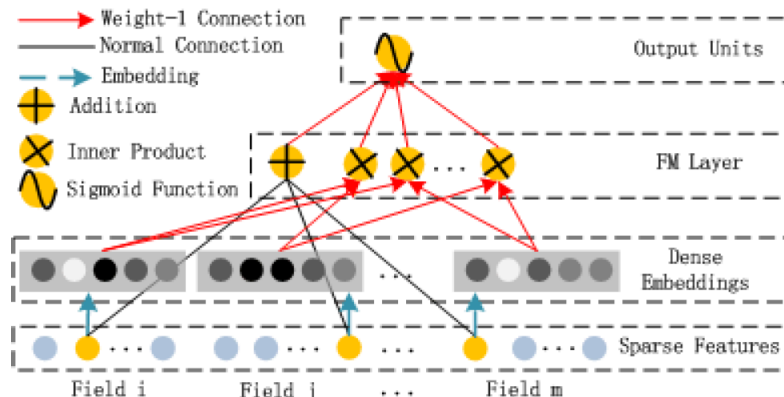
Model	Offline AUC	Online Acquisition Gain
Wide (control)	0.726	0%
Deep	0.722	+2.9%
Wide & Deep	0.728	+3.9%

Factorization Machines

$$\hat{y}(x) = w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \mathbf{v}_j^T \mathbf{v}_{j'}$$

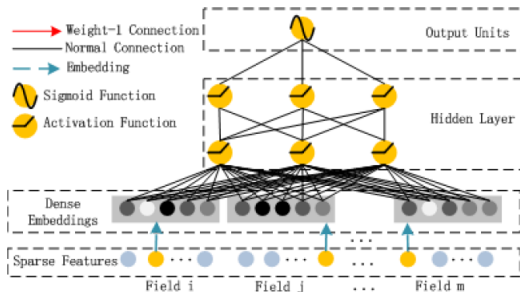
Feature vector \mathbf{x}																	Target y					
\mathbf{x}_1	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	y_1
\mathbf{x}_2	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	y_2
\mathbf{x}_3	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	y_3
\mathbf{x}_4	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	y_4
\mathbf{x}_5	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	y_5
\mathbf{x}_6	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	y_6
\mathbf{x}_7	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	y_7
A B C ... User				TI NH SW ST ... Movie					TI NH SW ST ... Other Movies rated					Time	TI NH SW ST ... Last Movie rated							

Factorization Machines As Neural Network

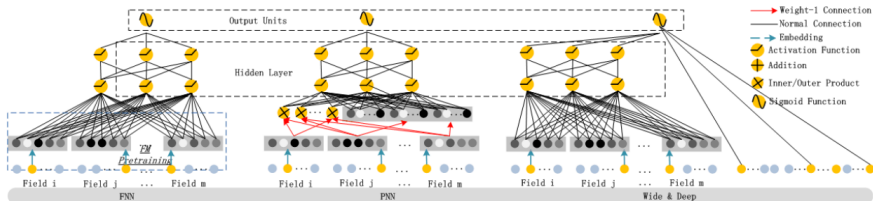


$$\hat{y}(x) = w_0 + \sum_{j=1}^p w_j x_j + \sum_{j=1}^p \sum_{j'=j+1}^p x_j x_{j'} \mathbf{v}_j^T \mathbf{v}_{j'}$$

Deep Neural Network



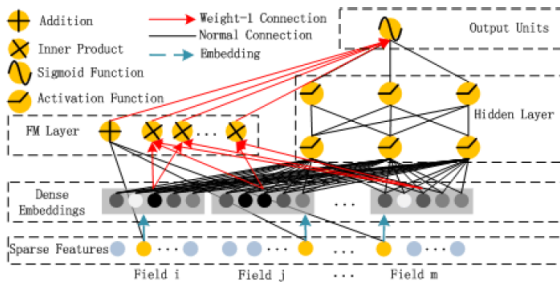
Нейросетевые архитектуры для CF



FNN - FM initialized feed forward neural network

PNN - Product based neural network

DeepFM (2017)



DeepFM

Результаты:

	Company*		Criteo	
	AUC	LogLoss	AUC	LogLoss
LR	0.8640	0.02648	0.7686	0.47762
FM	0.8678	0.02633	0.7892	0.46077
FNN	0.8683	0.02629	0.7963	0.45738
IPNN	0.8664	0.02637	0.7972	0.45323
OPNN	0.8658	0.02641	0.7982	0.45256
PNN*	0.8672	0.02636	0.7987	0.45214
LR & DNN	0.8673	0.02634	0.7981	0.46772
FM & DNN	0.8661	0.02640	0.7850	0.45382
DeepFM	0.8715	0.02618	0.8007	0.45083

Итоги

- ▶ Neighborhood methods
 - ▶ Проблема холодного старта
 - ▶ Матрица оценок используется непосредственно для получения предсказаний
 - ▶ У больших компаний с большим количеством данных чаще всего реализуется эвристический item-to-item подход
- ▶ Model-based methods
 - ▶ Основаны на оффлайнных расчетах
 - ▶ В продакшене используется уже обученные модели
 - ▶ Модели необходимо часто обновлять и перестраивать
 - ▶ Есть множество методов, которые показывают хорошие результаты
- ▶ Можно применять комбинированный подход
- ▶ Все это можно использовать в композициях

Открытые вопросы

- ▶ Как улучшать качество рекомендаций а не RMSE?
- ▶ Как обосновывать рекомендации?
- ▶ Как решать проблему холодного старта?

Открытые вопросы

- ▶ Как учитывать неявные предпочтения?
- ▶ Как учитывать контекст в рекомендациях?
- ▶ Как дополнительно учитывать признаки объектов?
- ▶ Как учитывать связи между пользователями?
- ▶ Как учитывать информацию из других источников (например из поиска)?

Технические вопросы

- ▶ Как быстро обновлять рекомендации?
- ▶ Как масштабировать расчет рекомендаций?
- ▶ Как быстро выбирать топ документов для предсказаний?
- ▶ Как измерять качества контекстных рекомендаций?

Вопросы

