# Recsys Advanced

Малышев Сергей
Одноклассники

# Plan

- DL4Rec
  - NN as MF, DSSM, DLRM, DCNv2
  - Sequential Recs (SASRec)
- Подходы в ранжировании
  - Pointwise
  - Pairwise
  - Listwise
- Двухуровневые модели рекомендаций
- Нерешенные проблемы в рекомендациях
  - Offline-online
  - Feedback-loop
- Summary

# DL4Rec

## NN as MF

$$\widehat{r_{ui}} = p_u^T \cdot q_i$$

- Идею мат. факт. Можно перекинуть на сетки и раскладывать матрицу взаимодействий с помощью них
- Как токен уже используем id юзера и id айтема
- Из лоссов можно выбирать регрессию (MSE), классификацию (cross-entropy)
- Нужны негативные примеры

# DL4Rec

## NN as MF

## Сэплинг негативных примеров

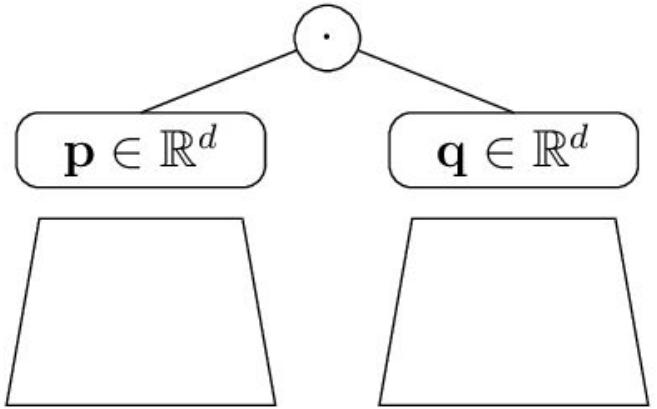
- Uniform - берем айтемы из равномерного распределения всего набора айтемов
    - Просто, но есть проблема easy negative-ов
- Popularity based - берем айтемы из частоного распределения всего набора айтемов
    - Уже учитываем популярность айтемов и примеры становятся качественней
- Hard-negatives - используем оценки от предыдущих моделей
    - Так как нужно скорить примеры во время обучения, время обучения увеличивается, но при этом и негативные примеры становятся еще лучше
- In-batch - сэпмлируем айтемы прямо в батче (uniform, popularity-based)

DL4Rec

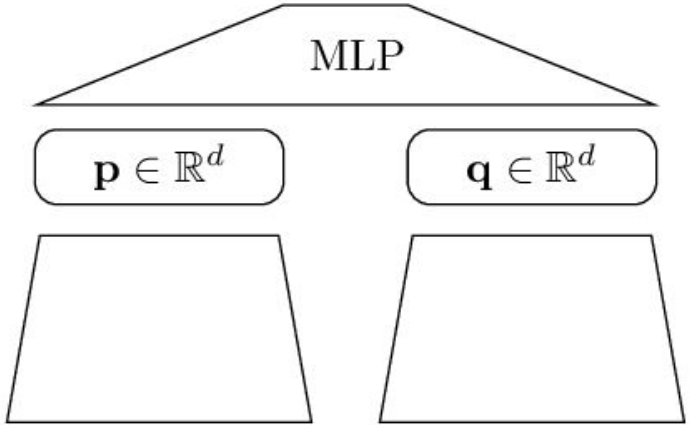
NN as MF

Dot-product можно усложнить - stack more layers

$$\phi^{\text{dot}}(\mathbf{p}, \mathbf{q}) = \langle \mathbf{p}, \mathbf{q} \rangle$$




$$\phi^{\text{MLP}}(\mathbf{p}, \mathbf{q}) = \mathbf{f}_{W_l, \mathbf{b}_l}(\ldots \mathbf{f}_{W_1, \mathbf{b}_1}([\mathbf{p}, \mathbf{q}]) \ldots)$$




- Оказывается, что линейные слои работают хуже, чем обычный dot-product, поэтому не стоит слишком усложнять модель

# DL4Rec

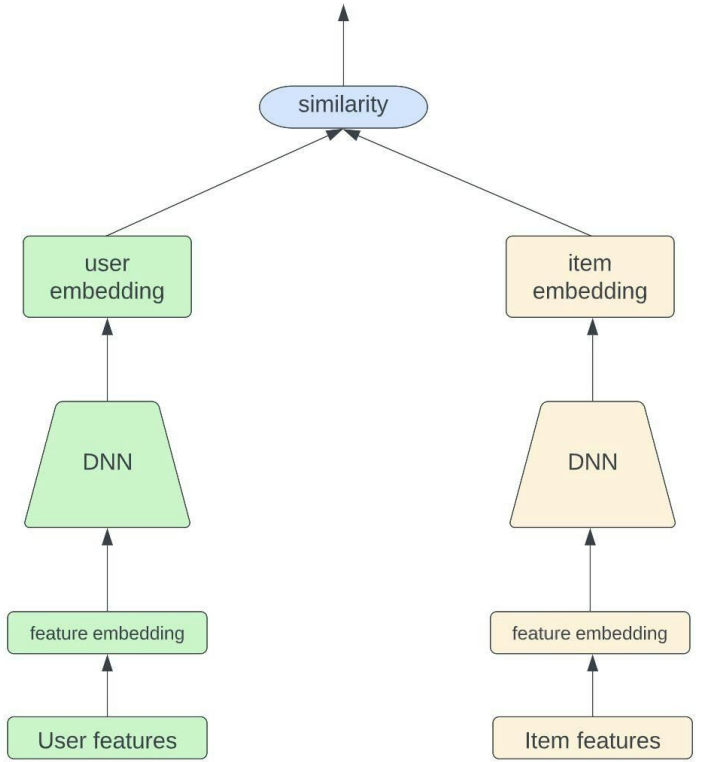
## NN as MF

- Pros:
  - Распространенность фреймворков
  - Можно легко дотюнить
- Cons:
  - Туда не запихнуть свойства объектов и пользователя

# DL4Rec

## DSSM (Two tower-model)

- Строим две башни - юзера и айтема
- В башне юзера передаем (фичи юзера, user_id)
  => передаем в MLP
  => получаем финальный эмбеддинг юзера
- То же самое с башней для айтемов
- Берем dot-product юзер и айтем емебедингов



# DL4Rec

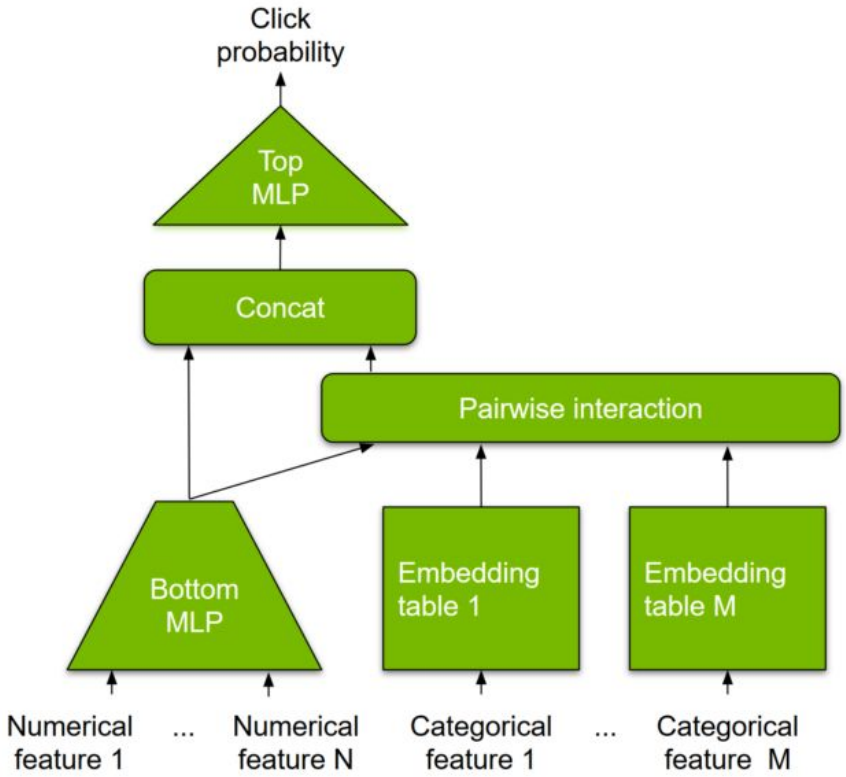
## DSSM (Two tower-model)

- Pros:
  - Распространенность фреймворков
  - Можно легко дотюнить
  - Можно добавить признаки
- Cons:
  - Не учесть попарное взаимодействие признаков и временную динамику

# DL4Rec

## DLRM

- Строим одну MLP под все признаки
- Строим эмбеды для пользователей и айтемов чисто под коллаборативку
- Берем попарные dot-product-ы эмбедов всех фичей, эмбедов юзеров и айтемов
- Concat-им эмбеды признаков и попарных dot-product-ов
- Скармливаем финальной MLP



DL4Rec

DCNv2

- Левый блок:
  - Перемножаем вектор фичей на инкрементальный вектор с добавлением матрицы весов и байеса
  - Делаем так несколько раз
  - Передаем в concat-слой
- Правый блок:
  - Скармливаем фичи MLP
  - Передаем в concat-слой
- Домножением на финальную матрицу весов + баес получаем финальный скор

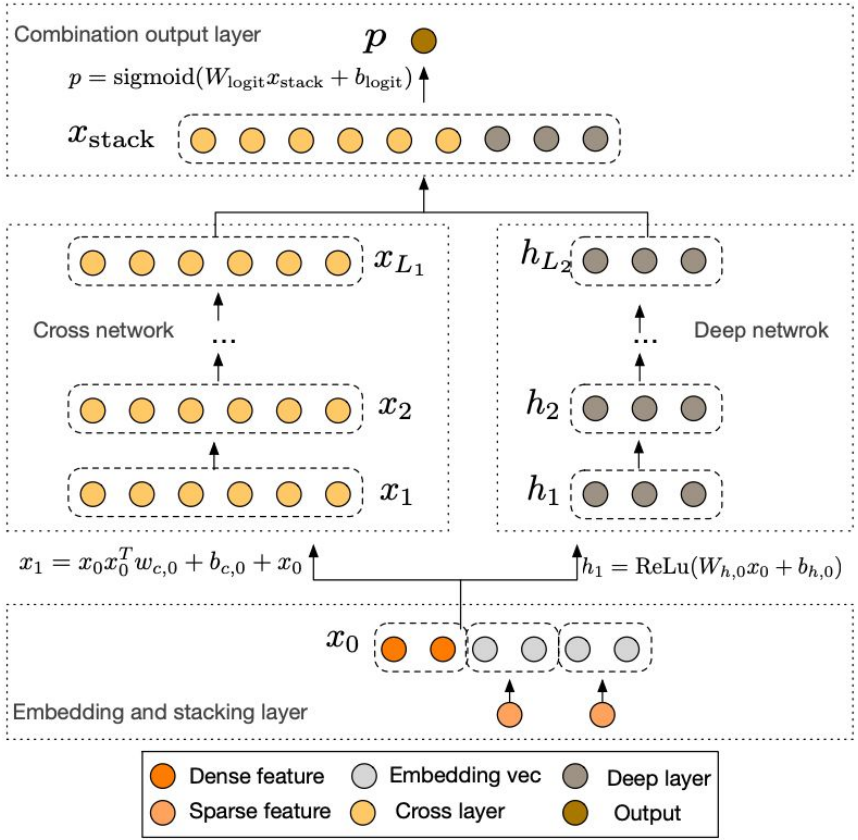



**Figure 1: The Deep & Cross Network**

# DL4Rec

# DCNv2

- Pros:
  - Распространенность фреймворков
  - Можно легко дотюнить
  - Можно добавить признаки
  - Учитываем попарное взаимодействие признаков
- Cons:
  - Не учесть временную динамику

# DL4Rec

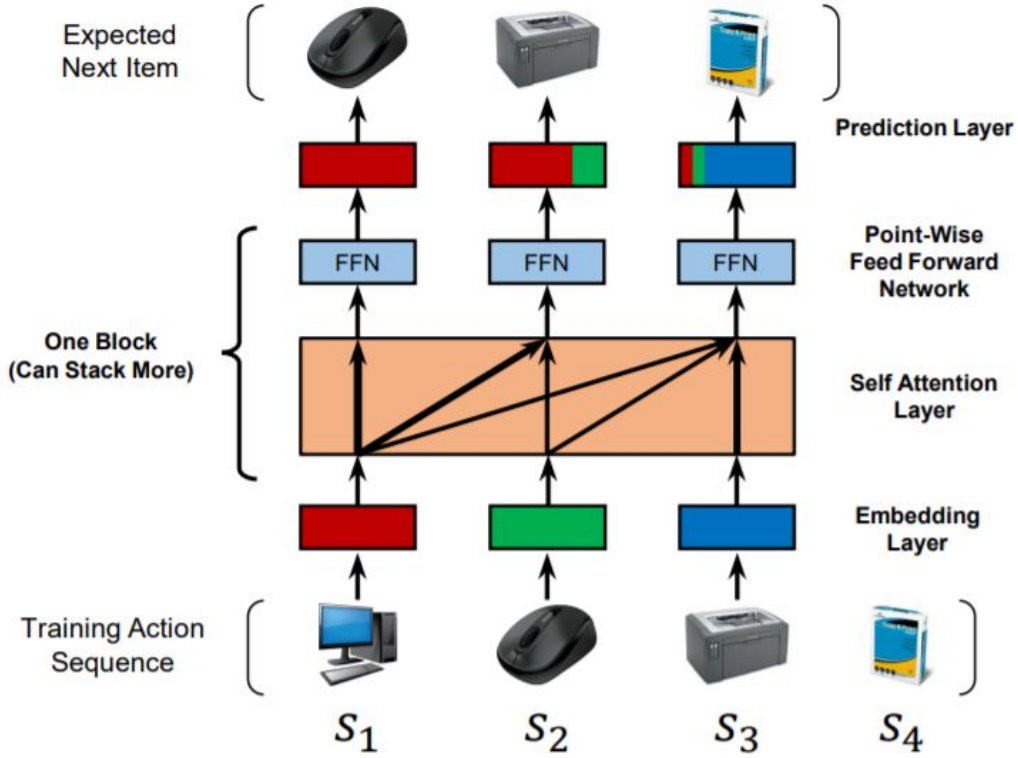
## Sequential recommendations

- Токены - айтемы
- Последовательности - юзеры
- Предсказываем либо следующий токен по левому контексту, либо айтем, зная левый и правый контекс

# DL4Rec

## SASRec (Self-Attentive Sequential Rec)

- Используя attention-маску стараемся предсказать следующий айтем
- От трансформера используется только decoder



# DL4Rec

## SASRec (Self-Attentive Sequential Rec)

- Pros:
    - Очень сильный бейзлайн, использующий временную динамику
- Cons:
    - Не засунуть признаки

Подходы в рекомендациях / ранжировании

Pointwise

Идея: пытаемся приблизить оценки моделей к меткам объектов для каждого объекта отдельно

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(P\left(\hat{y}_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(1 - P\left(\hat{y}_i\right)\right)$$

Pros:

- Просто, быстро

Cons:

- Мы рассматриваем объекты сами в себе не используя контекст всех остальных объектов

Подходы в рекомендациях / ранжировании

Pairwise

Идея: хотим использовать уже пары и сравнивая их друг с другом строить оценки модели

$$f : x_i \to s_i$$

$$sign\left( l_i - l_j \right) = sign\left( s_i - s_j \right)$$

$$L_{ij} = -y_{ij} \cdot \log\left( P\left( \widehat{y_{ij}} \right) \right) - \left( 1 - y_{ij} \right) \cdot \log\left( 1 - P\left( \widehat{y_{ij}} \right) \right)$$

$$P\left( \widehat{y_{ij}} \right) = \sigma\left( s_i - s_j \right)$$

$$y_{ij} = \begin{cases} 1 \leftrightarrow l_i > l_j \\ 0 \leftrightarrow l_i < l_j \end{cases}$$

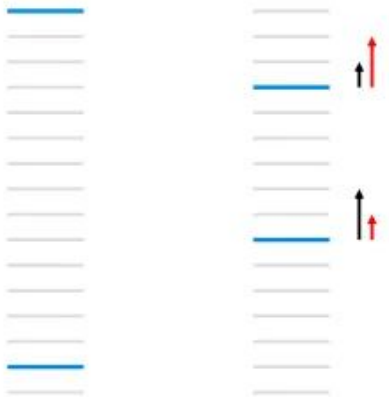
Подходы в рекомендациях / ранжировании

Pairwise

Как считать градиенты?

$$L_{ij} = -\left( y_{ij} \cdot \log\left( P\left( \widehat{y_{ij}} \right) \right) + \left( 1 - y_{ij} \right) \cdot \log\left( 1 - P\left( \widehat{y_{ij}} \right) \right) \right)$$

$$L_i = \sum_{j:\, l_i \neq l_j} L_{ij}$$



Интуиция: аккумулируем все тянущие силы айтема в одну равнодействующую

Подходы в рекомендациях / ранжировании

Pairwise

Ranknet: Как выглядят градиенты

$$\frac{\partial L_{ij}}{\partial w_k} = \frac{\partial L_{ij}}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_k} + \frac{\partial L_{ij}}{\partial sj} \cdot \frac{\partial s_j}{\partial w_k}$$

$$\frac{\partial L_{ij}}{\partial s_i} = y_{ij} - \frac{1}{1+e^{(s_i - s_j)}} = -\frac{\partial L_{ij}}{\partial s_j}$$

$$\lambda_{ij} = \frac{\partial L_{ij}}{\partial s_i}$$

$$\frac{\partial L}{\partial w_k} = \sum_{i,j}\left(\frac{\partial L_{ij}}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_k} + \frac{\partial L_{ij}}{\partial sj} \cdot \frac{\partial s_j}{\partial w_k}\right) = \sum_{i,j}\lambda_{ij} \cdot \left(\frac{\partial s_i}{\partial w_k} + \frac{\partial s_j}{\partial w_k}\right) =$$

$$= \sum_i \frac{\partial s_i}{\partial w_k} \cdot \left(\sum_{(i.j)}\lambda_{ij} - \sum_{(j.i)}\lambda_{ji}\right) = \sum_i \frac{\partial s_i}{\partial w_k} \cdot \lambda_j$$


Интуиция: таким образом косвенно оптимизируем метрику ROC-AUC

Подходы в рекомендациях / ранжировании

Pairwise

А если хотим оптимизировать другую метрику, например NDCG?

$$\frac{\partial L_{ij}}{\partial w_k} = \sum_{i,j} \lambda'_{ij} \left( \frac{\partial s_i}{\partial w_k} - \frac{\partial s_j}{\partial w_k} \right), \quad \lambda'_{ij} = \lambda_{ij} \cdot \left| \triangle NDCG_{i,j} \right| - LambaRank$$

Вместо NDCG можно добавить например MAP и уже оптимизировать MAP

## Подходы в рекомендациях / ранжировании

Pairwise

Pros:

- Решаем уже задачу, связанную с ранжированием
- Результаты получаются лучше чем у pointwise подхода

Cons:

- Оценки моделей становятся неинтерпретируемы

Подходы в рекомендациях / ранжировании

Listwise

Идея: хотим использовать уже весь список

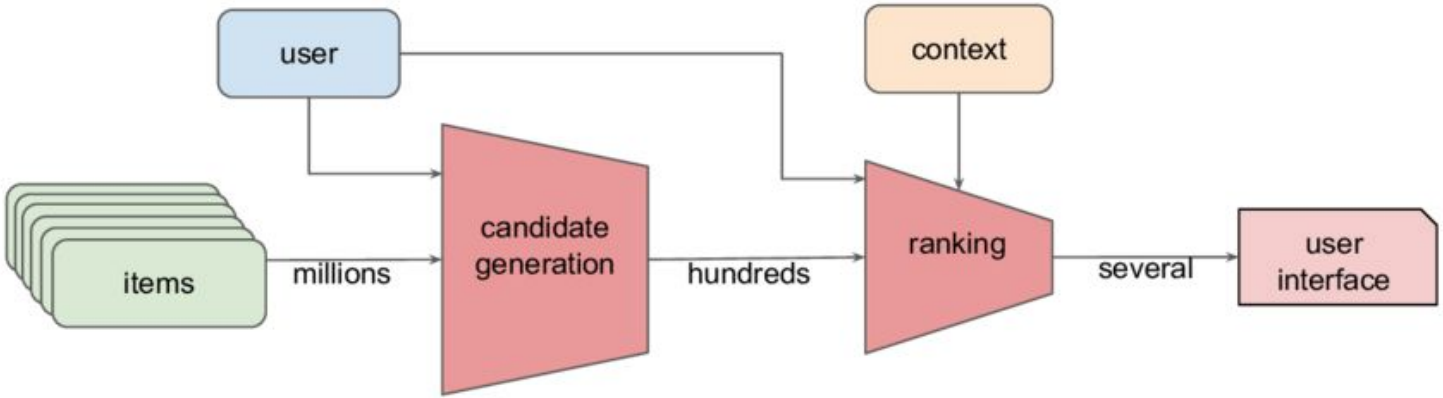
## Двухуровневые модели рекомендаций

### Мотивация

- Айтемов очень много (миллионы, миллиарды)
- Модели матричных разложений работают хорошо, но
  - Линейные модели не учитывают нелинейные зависимости
  - В целом у матричных разложений контекст, связан только со взаимодействиями пользователей и айтемов
- Хочется использовать не только коллаборативную инфу, но и использовать априорные и поведенческие свойства пользователей и объектов

# Двухуровневые модели рекомендаций

## Идея двухуровневой модели




- Первый уровень - достаточно простые и легковесные, которые могут прожевать огромный список айтемов и выдать грубую оценку по каждому айтему и отсечь самый треш

- Второй уровень - достаточно сложная и тяжелая модель, которая обрабатывает только самый топ айтемов и дает по каждому айтему более точную оценку и по этим оценкам уже строится финальное ранжирование

Нерешенные проблемы в рекомендациях

Offline-online evaluation

Проблема: увеличение метрик в оффлайне не всегда коррелирует с метриками в онлайне

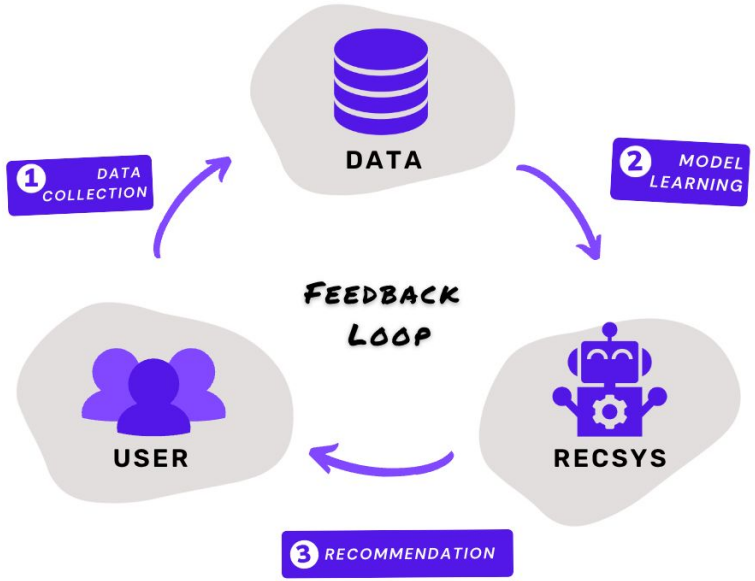
- Делаем предсказания вместо рекомендаций
    - Нужно встраивать все бизнесс-правила и постобработку в оффлайн-оценку
    - Использовать простые таргеты
    - Делать оффлайн-оценку в соответствии с поведением прода
- В онлайне напрямую влияем на поведение пользователя

Нерешенные проблемы в рекомендациях

Feedback loop

Проблема: обучаем на том, что рекомендуем => попадаем в пузырь

- Можно подмешивать в рекомендации:
  - Айтемы от других моделей
  - Случайные айтемы (epsilon-greedy)



# Summary

- DL4Rec
- Pairwise ранжирование
- Двухуровневые модели
- Нерешенные проблемы в рекомендациях

Спасибо за внимание!

тг: @malyshevsergey