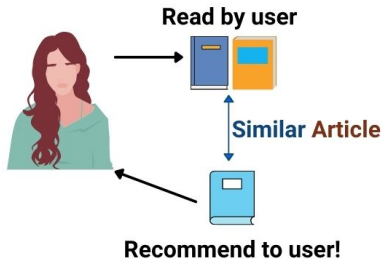


Коллаборативная фильтрация

Виктор Китов

victorkitov.github.io

Курс поддержан
фондом
'Интеллект'



Победитель
конкурса VK среди
курсов по IT



Содержание

- 1 Матричные разложения в рекомендациях
 - Использование матричных разложений
 - Учёт неявных откликов
- 2 Оценка и бизнес-особенности рекомендаций

- 1 Матричные разложения в рекомендациях
 - Использование матричных разложений
 - Учёт неявных откликов

Метод матричных разложений

- Будем строить эмбединг для каждого u и i в \mathbb{R}^d , $d \sim 100$:

$u \rightarrow \mathbf{p}_u \in \mathbb{R}^d$ интерес пользователя к категориям

$i \rightarrow \mathbf{q}_i \in \mathbb{R}^d$ представленность категорий в товаре

- Прогноз - соответствие интересующих и представленных категорий

$$\hat{r}_{ui} = \langle \mathbf{p}_u, \mathbf{q}_i \rangle$$

- В матричной записи ($\mathbf{p}_u, \mathbf{q}_i$ - строки P, Q):

$$\hat{R} = PQ^T$$

- Прогнозирование свелось к задаче низкорангового разложения R .
 - можем использовать SVD, pLSA и др.

Метод pure SVD

- Метод pure SVD использует trunkated SVD:

$$R \approx \hat{R} = \underbrace{(U\Sigma)}_P \underbrace{V^T}_Q$$

- Заполнение пропусков r_u :

$$\hat{\mathbf{r}}_u = V \left(\underbrace{V^T \mathbf{r}_u}_{\mathbf{p}_u} \right)$$

⊖ : SVD не допускает пропуски

- можно заполнить: 0, средним, оценкой другой моделью
- но это вызывает смещение оценок

Метод матричных разложений

- С поправкой на средние оценки пользователя и товара

$$\hat{r}_{ui} = w_u + w_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle$$

- Функция потерь (невыпуклая):

$$\sum_{(u,i) \in R} (w_u + w_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle - r_{ui})^2 + \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2$$

$$\rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i}$$

- Оценивание - SGD

- повторять до сходимости
 - сэмплируем случайную пару $(u, i) \in R$, обновляем¹:

$$\mathbf{p}_u := \mathbf{p}_u - \varepsilon \frac{\partial \mathcal{L}(u, i)}{\partial \mathbf{p}_u}; \quad \mathbf{q}_i := \mathbf{q}_i - \varepsilon \frac{\partial \mathcal{L}(u, i)}{\partial \mathbf{q}_i}; \quad \dots$$

¹Посчитайте аналитически.

Оценивание методов ALS

$$\sum_{(u,i) \in R} (w_u + w_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle - r_{ui})^2 + \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2$$
$$\rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i}$$

Метод ALS (alternating least squares):

- повторять по сходимости:
 - При фикс. P, W_u - найдем $w_i, q_i, i \in I$
 - При фикс. Q, W_i - найдем $w_u, p_u, u \in U$

Каждый раз - гребневая регрессия на $(u, i) \in R$

- аналит. решение, глобальный минимум

Метод матричных разложений

- Рассмотрим: при фикс. Q, W_i - нахождение $w_u, p_u, u \in U$
 - решаем для каждого u независимо
 - параллелизация
 - не нужно хранить все P
 - удобно для нового u
 - легко учитывать всю новую информацию о u
- p_u, q_i можно использовать:
 - для прогноза
 - как признаки в content-based модели
- q_i можно найти независимо как эмбединги i
 - удобно для новых товаров (без статистики) - строим их эмбединги по контенту (описание, музыка, текст, видео)

SVD++

- SVD++: для учёта не только явных оценок $r_{u,i}$, но и неявных (напр. просмотры).
- Вводится отдельный эмбединг для просмотренных товаров j : \mathbf{y}_j .
- Прогноз:

$$\sum_{(u,i) \in R} \left(w_u + w_i + \left\langle \mathbf{p}_u + \frac{1}{\sqrt{|\text{view}(u)|}} \sum_{j \in \text{view}(u)} \mathbf{y}_j, \mathbf{q}_i \right\rangle - r_{ui} \right)^2$$

$$+ \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2 + \gamma \sum_{i \in I} \|\mathbf{y}_i\|^2$$

$$\rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, \mathbf{y}_j, w_u, w_i}$$

- Нормировка почему-то на $\sqrt{|\text{view}(u)|}$.

Учёт социальных связей²

- Люди часто формируют рейтинги под влиянием друзей.
- Метод Social Regularization позволяет точнее оценивать профиль u , используя информацию о его круге друзей $\mathcal{F}(u)$.
- Предлагается метод матричных разложений с социальным регуляризатором:

$$R_1 = \sum_{u \in U} \left\| p_u - \frac{1}{|\mathcal{F}(u)|} \sum_{f \in \mathcal{F}(u)} p_f \right\|^2$$

$$R_2 = \sum_{u \in U} \sum_{f \in \mathcal{F}(u)} \|p_u - p_f\|^2$$

- В чём недостаток?

²Recommender Systems with Social Regularization

Учёт социальных связей

- Поскольку друзья могут различаться по вкусам, то правильнее их учитывать с учётом похожести:

$$R_1 = \sum_{u \in U} \left\| p_u - \frac{1}{|\mathcal{F}(u)|} \frac{\sum_{f \in \mathcal{F}(u)} \text{sim}(u, f) p_f}{\sum_{f \in \mathcal{F}(u)} \text{sim}(u, f)} \right\|^2$$

$$R_2 = \sum_{u \in U} \sum_{f \in \mathcal{F}(u)} \text{sim}(u, f) \|p_u - p_f\|^2$$

- Даёт прирост качества, лучше всего работала с R_2 и похожестью=корреляции.

Факторизационные машины^{3,4}

- Факторизационные машины (FM, factorization machines) применимы для регрессии, бинарной классификации, ранжирования и рек. систем.
 - для рек. систем - часто занимала топовые места на соревнованиях.
- Для рекомендаций - вектор признаков x_i в виде (вариант):

Feature vector x																	Target y					
$x^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(3)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$
A B C ... User				TI NH SW ST ... Movie				TI NH SW ST ... Other Movies rated				Time	TI NH SW ST ... Last Movie rated									

$$\hat{y}(x) = \tilde{w}_0 + \sum_{i=1}^D \tilde{w}_i x^i + \sum_{i=1}^N \sum_{j=i+1}^N \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

³<https://www.csie.ntu.edu.tw/~b97053/paper/Rendle2010FM.pdf>

⁴Реализация: <http://www.libfm.org>

Билинейная модель

- Обычная билинейная модель
 - напр. SVM+poly-kernel[d=2]

$$\hat{y}(x) = \tilde{w}_0 + \sum_{i=1}^D \tilde{w}_i x^i + \sum_{i=1}^D \sum_{j=i}^D w_{ij} x_i x_j$$

- Много параметров за счёт $W \in \mathbb{R}^{D \times D}$, переобучается
- Требует неразреженных данных, иначе не обучится.
- $O((\alpha D)^2)$ - прогноз и обучение
 - α - доля ненулевых признаков.
- Любая $W = W^T$ допускает спектральное разложение:

$$W = P \Sigma P^T = P \Sigma P^T = P \Sigma^{1/2} \left(\Sigma^{1/2} \right)^T P^T = S S^T$$

Отличие факторизационных машин

- В FM W приближается с помощью:

$$W \approx VV^T, \quad V \in \mathbb{R}^{D \times K}, \quad K \ll D$$

Факторизационные машины (FM):

$$\hat{y}(x) = \tilde{w}_0 + \sum_{i=1}^D \tilde{w}_i x^i + \underbrace{\sum_{i=1}^N \sum_{j=i+1}^N \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j}_{\text{почти } \frac{1}{2} x^T V V^T x = \frac{1}{2} \langle V^T x, V^T x \rangle}$$

- Сложность прогноза и сложность обновления всех весов через SGD: $O(\alpha DK)$
 - α - доля ненулевых признаков.
- FM лучше учатся на разреженных данных за счёт факторизации (с малым K)

Метод матричных разложений

- $\hat{r}_{ui} = w_u + w_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle = w_u + w_i + \|\mathbf{p}_u\| \|\mathbf{q}_i\| \cos \phi$
 - $\|\mathbf{q}_i\| \gg 1 \Rightarrow$ часто $|\hat{r}_{ui}| \gg 1$ (популярные) можно хранить только их
 - чтобы не зацикливаться на популярных, можно подмешивать товары по cos-sim $(\mathbf{p}_u, \mathbf{q}_i)$
- Neural collaborative filtering: нейросетевое обобщение $\langle \mathbf{p}_u, \mathbf{q}_i \rangle \rightarrow F_\theta(\mathbf{p}_u, \mathbf{q}_i)$

$$\sum_{(u,i) \in R} (w_u + w_i + F_\theta(\mathbf{p}_u, \mathbf{q}_i) - r_{ui})^2 + \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2$$

$$\rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i, \theta}$$

- но нейросеть вычисляется долго и не так хорошо способна выучить $\langle \mathbf{p}_u, \mathbf{q}_i \rangle^5$.

⁵<https://arxiv.org/pdf/2005.09683.pdf>

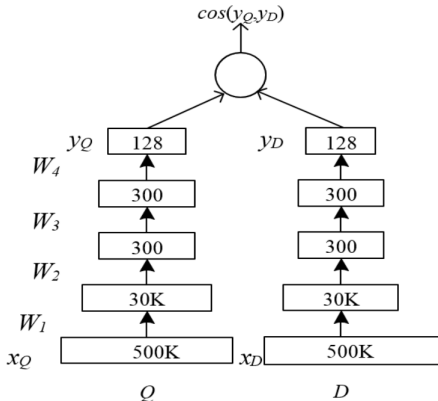
Модель DSSM⁶

- Модель DSSM (Deep Structured Semantic Model)
вычисляют соответствие
 - пользователя и товара (рек. система)
 - документа и запроса (ранжирование)
 - p_u, q_i можно брать content-based
 - вариант: $p_u = \text{mean}_{r_{ui} > 0} \{q_i\}$
 - либо подставлять p_u и q_i из др. модели

⁶ A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems.

Модель DSSM

- Используется сиамская сеть, отображающее все в общем семантическое пространство.
- $\cos\text{-sim}$ используется для оценки похожести



- 1 Матричные разложения в рекомендациях
 - Использование матричных разложений
 - Учёт неявных откликов

Модель Implicit CF

- Большинство рек. систем работают с явными оценками товаров (explicit feedback).
- Модель Implicit CF⁷ ориентирована на неявные оценки.
 - время просмотра, количество покупок, потраченные ср-ва (чем больше, тем достовернее связь)
- Прогноз - интерпретируемый в виде item-based рекомендации с автонастраиваемой похожестью товаров
 - похожесть своя для каждого u

⁷Collaborative Filtering for Implicit Feedback Datasets.

Модель Implicit CF

- Пусть r_{ui} - неявная вещественная оценка (напр. время просмотра видео)
- Определим

$$s_{ui} = \begin{cases} 1, & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

- Силу взаимодействия закодируем через вес

$$c_{ui} = 1 + \alpha r_{ui}, \quad \alpha \sim 40.$$

- Предлагается решать⁸

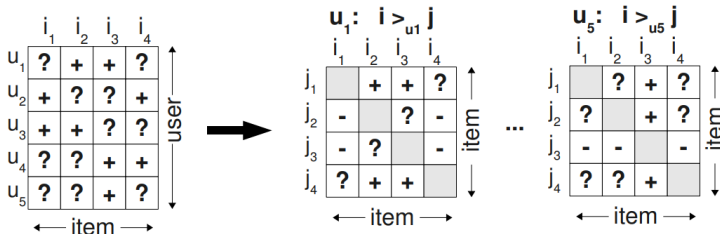
$$\sum_{(u,i) \in R} c_{ui} (w_u + w_i + \langle \mathbf{p}_u, \mathbf{q}_i \rangle - s_{ui})^2 + \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2$$

$$\rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i}$$

⁸В оригинальной статье - без w_u, w_i и $\alpha = \beta$.

Рекомендации с помощью ранжирования

- Модель BPR (Bayesian Personalized Ranking)⁹ настраивается на $r_{u,i} \in \{0, 1\}$ (implicit feedback)
- Используются потери из ранжирования.
- Если $r_{ui} = 1$ и $r_{uj} = 0$, то предполагаем, что для u $i \succ_u j$.



⁹<https://arxiv.org/pdf/1205.2618.pdf>

Рекомендации с помощью ранжирования

$$\sum_{\{u,i,j: i \succ_u j\}} \underbrace{\ln \sigma(\langle \mathbf{p}_u, \mathbf{q}_i \rangle - \langle \mathbf{p}_u, \mathbf{q}_j \rangle)}_{P(i \succ_u j)} - \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 - \beta \sum_{i \in I} \|\mathbf{q}_i\|^2$$

$$\rightarrow \max_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i}$$

Похожий метод¹⁰ (ранжирование через hinge):

$$\sum_{\{u,i,j: i \succ_u j\}} \max \{0; 1 + \langle \mathbf{p}_u, \mathbf{q}_j \rangle - \langle \mathbf{p}_u, \mathbf{q}_i \rangle\}$$

$$+ \alpha \sum_{u \in U} \|\mathbf{p}_u\|^2 + \beta \sum_{i \in I} \|\mathbf{q}_i\|^2 \rightarrow \min_{\mathbf{p}_u, \mathbf{q}_i, w_u, w_i}$$

¹⁰ Improving Maximum Margin Matrix Factorization

Содержание

- 1 Матричные разложения в рекомендациях
- 2 Оценка и бизнес-особенности рекомендаций

Рекомендательные системы на практике

Рекомендательные системы на практике состоят из этапов:

- 1 Отбор кандидатов ~ 3000
- 2 Препроцессинг кандидатов
- 3 Переранжирование кандидатов сложной рек. моделью
- 4 Постпроцессинг ранжирования из бизнес требований.

Рекомендательные системы на практике

Рекомендательные системы на практике состоят из этапов:

- ❶ Отбор кандидатов ~ 3000
 - по пользовательским предпочтениям (подпискам, просмотрам, интересам в анкете, поисковому запросу)
 - дополнения к предыдущим покупкам
 - по близости content-based эмбедингов пользователей и товаров
 - по облегченной рекомендательной системе
 - по товарам с высоким $\|\mathbf{q}_i\|$ (потенциально популярные)
 - близкие к \mathbf{p}_u (существуют м-ды быстрого поиска похожих)
- ❷ Препроцессинг кандидатов
 - удаляем сомнительные
 - удаляем слишком похожие
 - например, кластеризуем и берём по одному из каждого кластера
- ❸ Переранжирование кандидатов сложной рек. моделью
- ❹ Постпроцессинг ранжирования из бизнес требований.

Холодный старт

Холодный старт (cold start) - проблема построения прогнозов

- для нового пользователя:
- для нового товара:

Холодный старт

Холодный старт (cold start) - проблема построения прогнозов

- для нового пользователя:
 - рекомендовать популярное (summary-based)
 - вручную модерируемые подборки
 - опросить в начале, что пользователю интересно
- для нового товара:
 - рекомендовать по тематике
 - нужно гарантировать опр. кол-во просмотров
 - exploration vs. exploitation tradeoff

Контроль качества

- Как разбивать на train/test?

Контроль качества

- Как разбивать на train/test?
 - главное: разбить по времени
 - иначе модель будет знать заранее о трендах моды и вкусов
 - можно стратифицировать по категориям товаров
- Важно в онлайне следить за качеством модели
 - донастраивать при деградации
- Все нововведения должны проверяться через A/B тест

Метрики качества

- Кликовые метрики по top-K рекомендованным:
- Бизнес-метрики:

Метрики качества

- Кликовые метрики по top-K рекомендованным:
 - precision@K: доля кликнутых товаров
 - hitrate@K: 1, если кликнул на что-либо, 0 иначе
 - др. метрики из ранжирования (nDCG, pFound, ...)
- Бизнес-метрики:
 - время просмотра рекомендованных товаров
 - частота покупок рекомендаций
 - прибыль от рекомендаций
 - доля вернувшихся пользователей (retention)

Метрики качества

Что еще важно для качественных рекомендаций?

Метрики качества

Что еще важно для качественных рекомендаций?

- Фильтровать сомнительные рекомендации
 - используя ассессоров или отзывы пользователей
- Нетривиальность рекомендаций, "способность удивлять" (serendipity)
 - рекомендовать масло к хлебу - тривиально
- Обеспечивать разнообразие рекомендаций
 - повышает $\text{hitrate}@K$, почему?

Метрики качества

Что еще важно для качественных рекомендаций?

- Фильтровать сомнительные рекомендации
 - используя ассессоров или отзывы пользователей
- Нетривиальность рекомендаций, "способность удивлять" (serendipity)
 - рекомендовать масло к хлебу - тривиально
- Обеспечивать разнообразие рекомендаций
 - повышает hitrate@K , почему?
 - модель максимизирует для каждого i $p(\text{click on } i|u)$
 - а $\text{hitrate@K} = p(\text{click on any } i \in i_1, i_2, \dots, i_K|u)$

Комментарии

- Получается много метрик m_1, \dots, m_S
- Можно обучить веса наилучшей смеси, макс. коррелирующую с целевой бизнес-метрикой (прибыль, retention, ...)

$$\text{TargetMetric} \approx w_0 + w_1 m_1 + \dots + w_S m_S$$

- Особенность алгоритмов CF:
 - усиление моды (популярные i становятся еще популярнее)
 - усиление сегментации интересов
 - прививочники смотрят видео о пользе прививок
 - антипрививочники - об их возможном вреде

Комментарии

- Важно учитывать время:
 - многие товары имеют сезонный спрос
 - со временем меняется интерфейс системы, вкусы, мода.
 - для некоторых товаров (новости), важна временная свежесть.
- В данных много шума:
 - семейные аккаунты
 - возможны атаки на систему
 - положит. комментарии на "свои" товары
 - отрицат. комментарии на "чужие" товары

Заключение

- Рекомендации могут быть основаны:
 - на характеристиках пользователя и товара (content-based)
 $F : (\text{признаки пользователя, признаки товара}) \rightarrow \text{соответствие}$
 - на матрице рейтингов (collaborative filtering)
 - усредняющий алгоритм
 - memory-based алгоритмы (user/item based)
 - матричная факторизация (потери - поточенные или попарные)
- Итоговые рекомендации учитывают прогноз и др. факторы
 - разнообразие, нетривиальность, приличность и др.