Тензоры и рекомендательные системы

Аспирант: Халкечев Р. В.

Научный руководитель: Воронцов К.В.

Аспирантская школа по компьютерным наукам

Национальный исследовательский университет "Высшая школа экономики"

9 июня 2016 г.

Содержание

1. Введение
2. Постановка задачи
3. User/Item based подходы
4. SVD подход
5. Context-aware recommender systems
6. Тензоры
7. Текущая работа и направления исследований

Введение

Рекомендательные системы – программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, веб-сайты) будут интересны пользователю, имея определенную информацию о его профиле.

Введение

Виды рекомендательных систем

- **Content-based**: Рекомендуются объекты, похожие (по контенту) на те, с которыми пользователь уже успешно взаимодействовал.
- **Collaborative Filtering**: Рекомендации на основе истории оценок как самого пользователя, так и других пользователей.

Постановка задачи

Дано

- $u \in U$ – множество пользователей.
- $i \in I$ – множество объектов.
- $(r_{ui}, u, i, ...) \in D$ – множество событий.

Постановка задачи

Найти

- $\hat{r}_{ui} = \text{Predict}(u, i, ...) \approx r_{ui}$ – предсказать результат взаимодействия.
- $u \to (i_1, ..., i_K) = \text{Recommend}_K(u, ...)$ – персональные рекомендации.
- $i \to (i_1, ..., i_M) = \text{Similar}_M(i)$ – похожие объекты.

Постановка задачи

Иллюстрация:

2			4	5	
5		4			1
		5		2	
	1		5		4
		4			2
4	5		1		

User based

Выберем некоторую меру схожести пользователей по их истории оценок $sim(u, v)$. Тогда предсказание вычисляется по формуле:

$$\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in U_i} sim(u, v)(r_{vi} - \overline{r}_v)}{\sum_{v \in U_i} sim(u, v)}$$

Item based

Симметричный подход:

$$\hat{r}_{ui} = \overline{r}_i + \frac{\sum_{j \in I_u} sim(i,j)(r_{uj} - \overline{r}_j)}{\sum_{j \in I_u} sim(i,j)}$$

User/Item based

Недостатки таких подходов:

- Холодный старт.
- Плохие предсказания для нетипичных пользователей.
- Ресурсоёмкость вычислений.
- Часто предсказания получаются тривиальными.

SVD модель

Сингулярное разложение матрицы:

$$\underset{m\times n}{A} = \underset{m\times m}{U} \times \underset{m\times n}{\Sigma} \times \underset{n\times n}{V^T}$$

Матрицы U и V ортогональные, а Σ – диагональная.

$$UU^T = I_m, \quad VV^T = I_n$$

$$\Sigma = \text{diag}(\lambda_1, ..., \lambda_{min(m,n)}), \quad \lambda_1 \geq ... \geq \lambda_{min(m,n)} \geq 0$$

SVD модель

Из лямбд оставляем только первые d чисел, остальные полагаем равными нулю:

$$\lambda_{d+1}, ..., \lambda_{min(m,n)} := 0$$

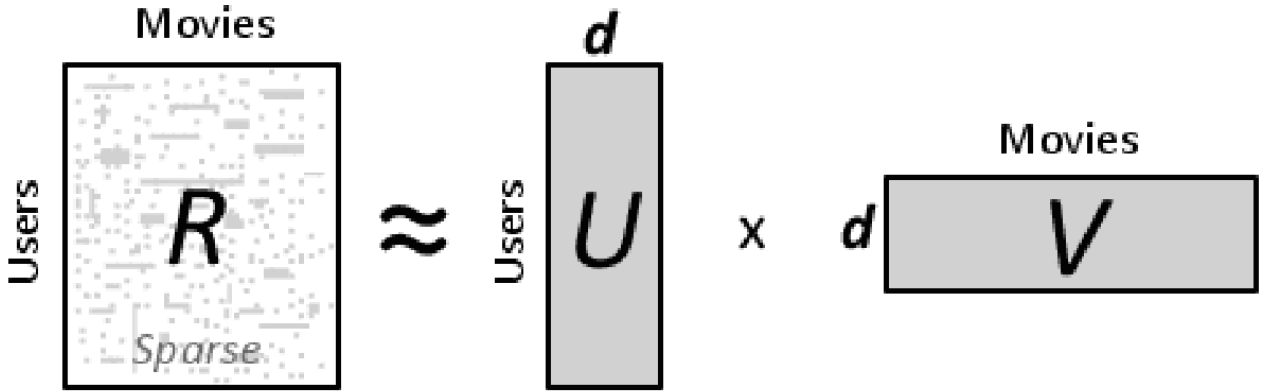
Получаем **усеченное** сингулярное разложение матрицы:

$$\underset{m \times n}{A'} = \underset{m \times d}{U'} \times \underset{d \times d}{\Sigma'} \times \underset{d \times n}{V'^T}$$

По теореме Эккарта-Янга A' – наилучшее низкоранговое приближение A с точки зрения средне-квадратичного отклонения.

SVD модель

Обозначим произведение первых двух матриц за одну матрицу U и получим для нашей задачи рекомендаций:



SVD модель

Чтобы предсказать оценку пользователя u для фильма i, мы берем некоторый вектор p_u (вектор пользователя) и вектор q_i (вектор объекта). Их скалярное произведение и будет предсказанием:

$$\hat{r}_{ui} =< p_u, q_i >$$

SVD модель

Проблемы:

- Матрица R нам полностью неизвестна и мы не можем просто взять её SVD разложение.
- SVD разложение не единственно.

Обучение SVD модели

Наша модель будет зависеть от векторов пользователей и объектов, а предсказание будем рассчитывать по скалярному произведению вектора пользователя на вектор объекта.

$$\hat{r}_{ui}(\Theta) =< p_u, q_i >$$

$$\Theta = \{p_u, q_i \mid u \in U, i \in I\}$$

Обучение SVD модели

Но векторов мы не знаем, их нужно обучить на оценках, которые нам известны:

$$E_{(u,i)}(\hat{r}_{ui}(\Theta) - r_{ui})^2 \to \min_{\Theta}$$

Обучение SVD модели

Для борьбы с переобучением осталось добавить регуляризатор:

$$\sum_{(u,i)\in D} (\hat{r}_{ui}(\Theta) - r_{ui})^2 + \lambda \sum_{\theta\in\Theta} \theta^2 \to \min_{\Theta}$$

Обучение SVD модели

Окончательно, получаем задачу численной оптимизации:

$$J(\Theta) = \sum_{(u,i)\in D} (p_u^T q_i - r_{ui})^2 + \lambda(\sum_u ||p_u||^2 + \sum_i ||q_i||^2) \to \min_{p,q}$$

Обучение SVD модели

Подходы к решению:

- Градиентный спуск
- Alternating Least Squares

Обучение SVD модели

Градиентный спуск:

$$\Theta_{t+1} = \Theta_t - \eta \nabla J(\Theta)$$

Обучение SVD модели

Alternating Least Squares:

$$p_u^*(\Theta) = \arg\min_{p_u} J(\Theta)$$

$$q_i^*(\Theta) = \arg\min_{q_i} J(\Theta)$$

Обучение SVD модели

Alternating Least Squares:

$$p_u^*(\Theta) = \arg\min_{p_u} J(\Theta) = (Q_u^T Q_u + \lambda I)^{-1} Q_u^T r_u$$

$$q_i^*(\Theta) = \arg\min_{q_i} J(\Theta) = (P_i^T P_i + \lambda I)^{-1} P_i^T r_i$$

Обучение SVD модели

Итеративный процес:

$$p_u^{2t+1} = p_u^*(\Theta_{2t}) \quad \forall u \in U$$

$$q_i^{2t+2} = q_i^*(\Theta_{2t+1}) \quad \forall i \in I$$

Учитываем контекст

Предпочтения пользователей не статичны, а зависят от многих факторов, например:

- Время
- Геолокация
- Предыдущий объект
- Тип девайса пользователя
- ...

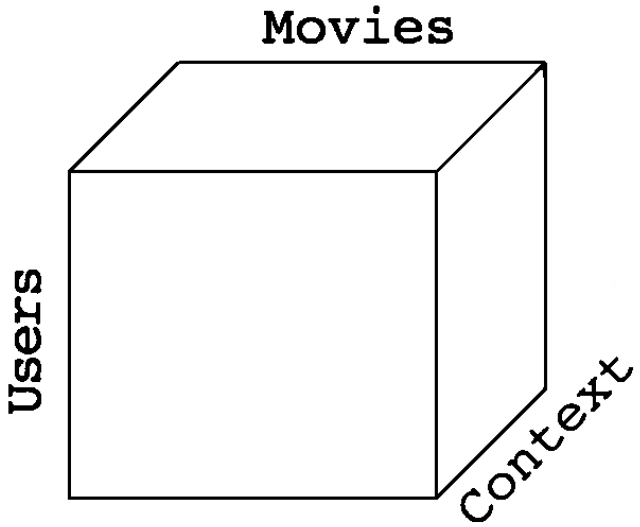
Учитываем контекст

В этом случае нам нужно аппроксимировать функцию от $N + 2$ переменных:

$$F : \text{User} \times \text{Item} \times \text{Context}_1 \times ... \times \text{Context}_N \to \text{Relevance}$$

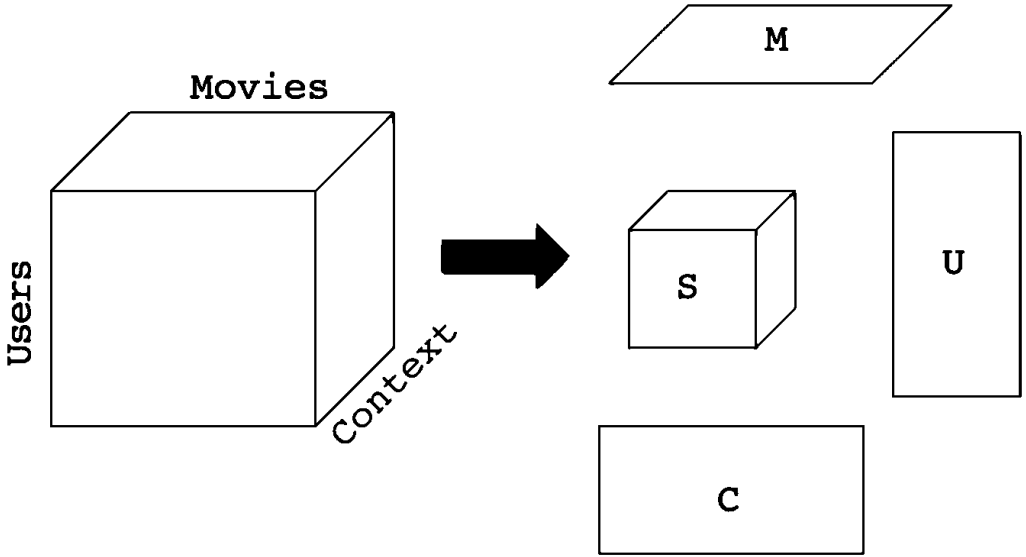
Учитываем контекст

Данные можно представить в виде тензора:



Учитываем контекст

Приходит идея действовать аналогично:



Тензоры - основные понятия

Тензор - многомерный массив:

$$T = [T(i_1, ..., i_d)], \quad i_k = 1, ..., n_k \quad (k = 1, ..., d)$$

Основные понятия:

- Размерность (порядок) тензора $= d$
- Размер тензора $= n_1 \times ... \times n_d$
- Размерности тензора $=$ числа $n_1, ..., n_d$

Основная проблема: проклятие размерности.

Матрицы развёртки

У каждого $d-$мерного тензора T есть $d-1$ матрица развёртка:

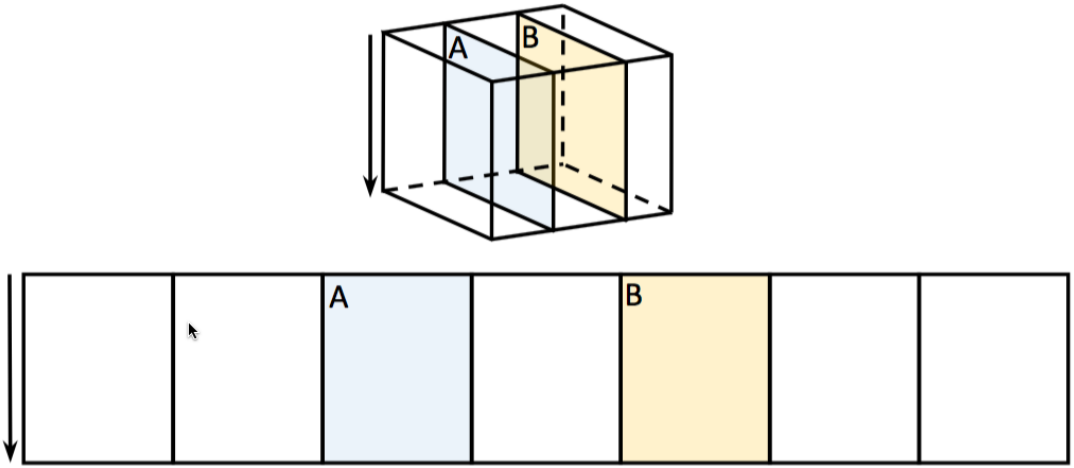
$$A_k = [A(i_1 i_2, ..., i_k; i_{k+1}, ..., i_d)]$$

Здесь $i_1 i_2, ..., i_k$ - строчный, а $i_{k+1}, ..., i_d$ столбовой мульти-индексы. Матрица A_k размера $M_k \times N_k$:

$$M_k = \prod_{s=1}^{k} n_s, \quad N_k = \prod_{s=k+1}^{d} n_s$$

Матрицы развёртки

Иллюстрация:



Каноническое разложение

$$T(i_1, ..., i_d) = \sum_{r=1}^{R} U_1(i_1, r)U_2(i_2, r)...U_d(i_d, r)$$

Наименьшее возможное число R называется каноническим рангом тензора T.

Проблемы:

- Нахождение R является $NP-$полной задачей.
- Нахождение канонического разложения является некорректно поставленной задачей по Адамару.

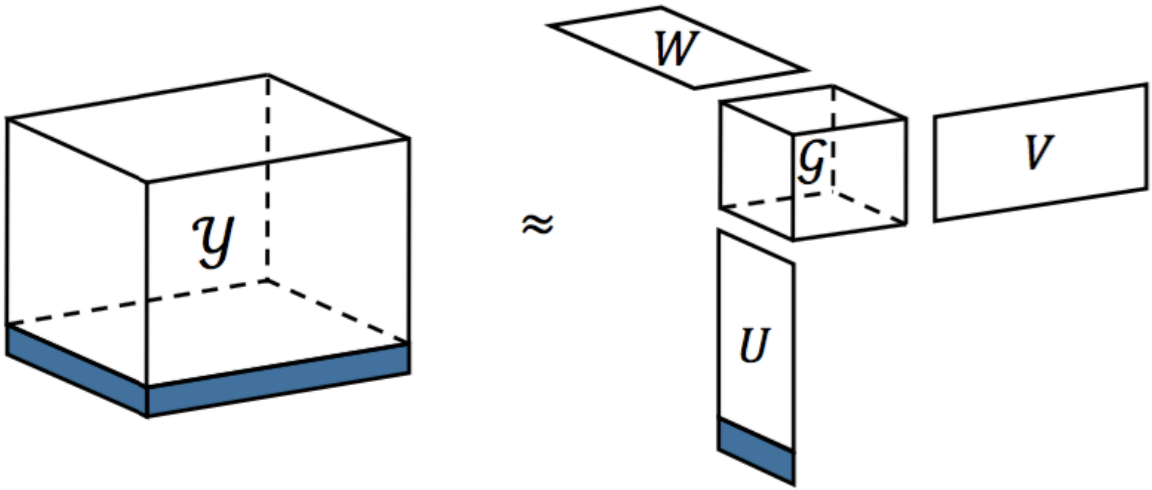
Разложение Таккера

$$T(i_1, ..., i_d) = \sum_{r_1,...,r_d} G(r_1, ..., r_d) U_1(i_1, r_1) U_2(i_2, r_2) ... U_d(i_d, r_d)$$

Попрежнему остаётся проблема проклятия размерности.

Разложение Таккера

Иллюстрация:



TT разложение

Идея:

$$T(i_1 i_2; i_3 i_4 i_5 i_6) = \sum_r U(i_1 i_2; r) V(i_3 i_4 i_5 i_6; r)$$

Слева 6-мерный тензор, а справа 3-мерный и 5-мерный.

Размерность уменьшилась! Далее рекурсивно!

TT разложение

$$T(i_1, ..., i_d) = \sum_{r_0, r_1, ..., r_d} G_1(r_0, i_1, r_1) G_2(r_1, i_2, r_2) ... (r_{d-1}, i_d, r_d)$$

Здесь G_k трёхмерные тензоры размеров $r_{k-1} \times n_k \times r_k$, а $r_0 = r_d = 1$, вводится для удобства.

- G_k называются TT-ядрами тензора T.
- Числа r_k называются TT-рангами тензора T.

TT разложение

TT-разложение можно записать более компактно:

$$T(i_1, ..., i_d) = G_1(i_1)G_2(i_2)...G_d(i_d)$$

где $G_k(i_k)$ – являются матрицами размера $r_{k-1} \times r_k$.

TT разложение

- Число параметров в "квадратном"тензоре: $O(n^d)$
- Число параметров в его TT представлении: $O(dnr^2)$

TT-SVD

Для любого тензора T существует TT приближение с заданными TT рангами r_k такое, что:

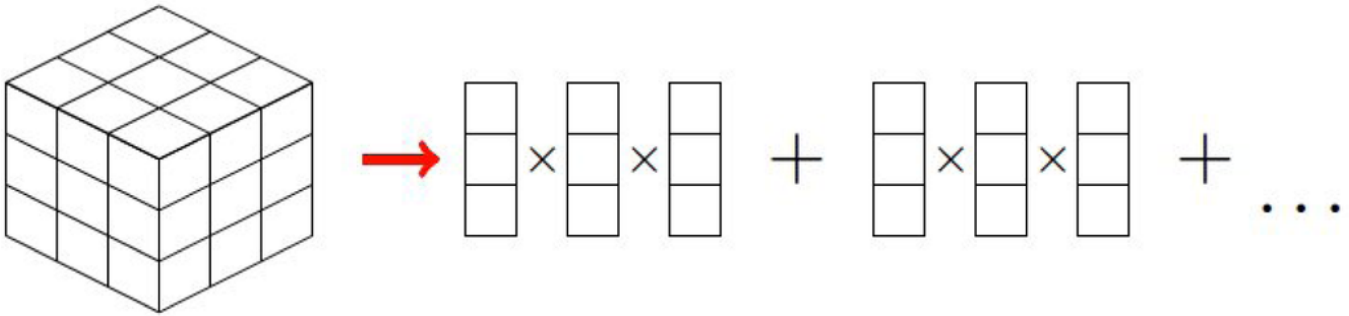
$$||A - T||_F \leq \sqrt{\sum_{k=1}^{d-1} \epsilon_K^2}$$

где

$$\epsilon_K = \min_{B:\ \text{rank}B \leq r_k} ||A_k - B||_F$$

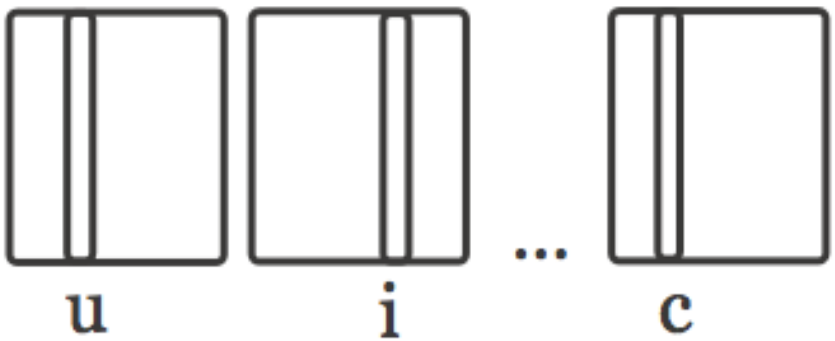
TT разложение

Иллюстрация:



TT-SVD

Обобщив алгоритм обучения SVD для матриц, можем находить TT-SVD разложение для тензоров.



Тензоры и рекомендательные системы

Существует большое число моделей рекомендательных систем на основе тензорных разложений:

- TALS и iTALS
- CubeSVD
- HOSVD
- TOPHITS
- BPTF
- ...

Текущая работа

Что уже сделано:

- Быстрый поиск похожих объектов (candidate selection).
- Эффективная и масштабируемая реализация ALS и TALS.
- Эксперименты на данных MovieLens, Яндекс.Музыки, SIFT/GIST.

Candidate selection

Наборы данных		
Название	Размерность	Размер базы
Yandex Music Data Set Items	120	250000
Yandex Music Data Set Users	120	5000
ANN_GIST1M Base	128	1000000
ANN_GIST1M Queries	128	10000

http://corpus-texmex.irisa.fr

Dijkstra Product Quantization

Пример результатов:

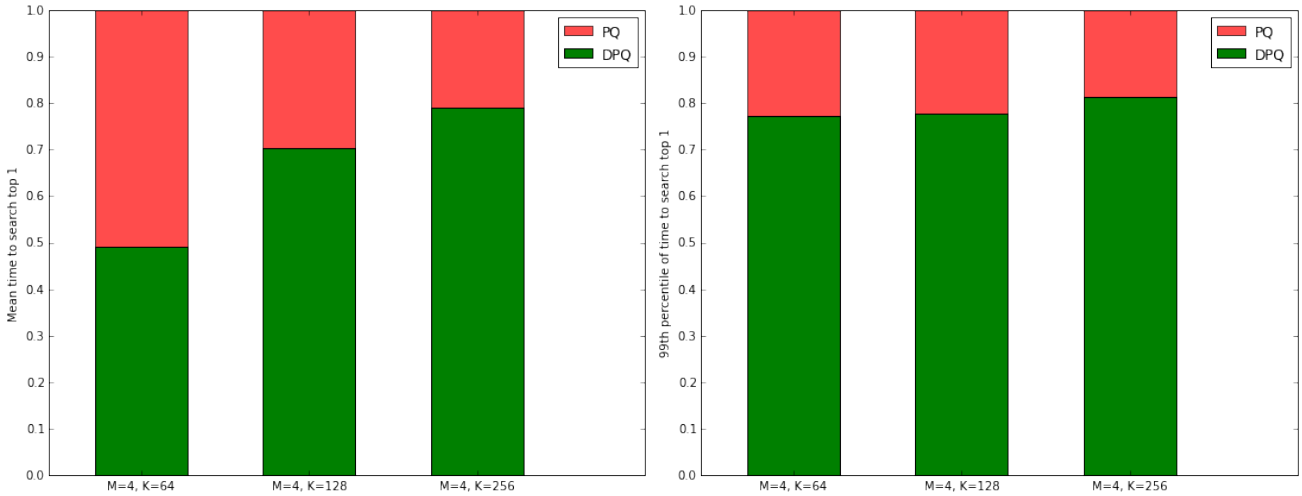


Рис. 1: Сравнение среднего и 99-перценители времени поиска Top-1 PQ и DPQ.

ALS и TALS

Сравнение качества рекомендаций на данных Яндекс.Музыки с использованием и без использования контекста:

- Предыдущий объект
- Время взаимодействия

Цели и направления исследований

- Эксперименты с разными типами контекста.
- Как учитывать признаки объектов/пользователей?
- Как учитывать таксономию? Связи между объектами/пользователями.
- Метод быстрого обновления рекомендаций, масштабирование системы.
- Метод улучшения рекомендаций, а не предсказаний.

Вопросы

Спасибо за внимание.

Задавайте вопросы.