FACTORIZATION METHODS FOR RECOMMENDER SYSTEMS

Elena Orlova, Aleksei Ustimenko, Roman Tsukanov, Vadim Kuzmin

December 20, 2018

Skoltech

Overview

- Introduction
- Problem statement
- Methods
- Results
- Conclusion

Introduction

- 60% of Amazon sails come from recommendations
- Factorization methods are widely used in recommender systems
  - SVD, SVD++ for 2D user-item matrices
  - High Order SVD (HOSVD), High Order Orthogonal Iteraton (HOOI) for tenzors (user-item with context information)

Problem Statement

- HOSVD, HOOI have been never compared in terms of information retrieval quality metrics such as DCG@k or Precision@k
- We decided to explore it
- MovieLens 10M
  - user, movie, genre, rating
- Compute HOSVD, HOOI and calculate DCG@k and Precision@k

Data Preprocessing

- Read data (64449 users, 10681 movies, 20 genres and 10m ratings)
- Split ratings data into to train and test by timestamp
- Form training matrix users$\times$movies
- Form training tensor users$\times$movies$\times$genres
- Missing data is replaced by zeros
- Form testing data

SVD for recommender system

- Start from a simple case - user-item matrix with ranks A
- Compute SVD decomposition:

$$A \approx \hat{U}\Sigma\hat{V}^T = \hat{U}\Sigma^{\frac{1}{2}}\Sigma^{\frac{1}{2}}\hat{V}^T = UV^T \tag{1}$$

- To get the prediction of a rating that a user would give on an item:

$$\hat{a}_{ij} = u_i v_j^T = \sum_{k=1}^{K} u_{ik} v_{kj} \tag{2}$$

- Optimal K is chosen by DCG@5 on validation

Tucker Decomposition

Let $T \in \mathbb{C}^{n\times m\times q}$, $A \in \mathbb{C}^{n\times r_1}$, $B \in \mathbb{C}^{m\times r_2}$, $C \in \mathbb{C}^{q\times r_3}$, $G \in \mathbb{C}^{r_1\times r_2\times r_3}$. If

$$\mathcal{T}_{ijk} = \sum_{s,t,u} g_{stu} a_{is} b_{jt} c_{ku}$$

or

$$\mathcal{T} = \sum_{s,t,u} g_{stu} A_s \otimes B_t \otimes C_u, \tag{3}$$

then this sum is called as Tucker's decomposition of tensor T with core tensor G.

HOSVD

The Tucker decomposition allows matrices to be arbitrary, while HOSVD restricts matrices to be orthogonal.

Let $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_d}$ be a tensor of Tucker ranks $(r_1, r_2, \ldots, r_d)$.
Computation:

- Let $\mathcal{A}^0 = \mathcal{A}$
- for $k = 1, 2, \ldots, d$:
  - Construct the standard factor-k flattening $\mathcal{A}^{k-1}_{(k)}$;
  - Compute the (compact) singular value decomposition $\mathcal{A}^{k-1}_{(k)} = U_k \Sigma_k V_k^T$ and store the left singular vectors $U_k \in F^{n_k \times r_k}$;
  - Set $\mathcal{A}^k = U_k^H \cdot_k \mathcal{A}^{k-1}$.

HOOI

HOOI is an iterative algorithm for computing low-rank approximations to tensors.

Let $\mathcal{A}$ be an $I_1 \times I_2 \times \cdots \times I_T$ tensor and let $r_1, r_2, \ldots, r_T$ be a set of integers satisfying $1 \leq r_n \leq I_n$, for $n = 1, \ldots, T$. The *Rank*-$\{r_1, r_2, \ldots, r_T\}$ approximation problem is to find a set of $I_n \times r_n$ matrices $U^{(n)}$ with orthogonal columns, $n = 1, 2, \ldots, T$, and a $r_1 \times \ldots r_T$ core tensor $\mathcal{B}$ such than the optimization problem

$$\min_{U^{(p)}} \|A - \mathcal{B} \times_1 U^{(1)} \times_2 U^{(2)} \ldots_T U^{(T)}\|_F^2 \tag{4}$$

is satisfied. It can be shown that the optimal $\mathcal{B}$ is given by

$$\mathcal{B} = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \cdots \times_T U^{(T)^T} \tag{5}$$

and that it is sufficient to find $U^{(n)}$'s satisfying $U^T U = I_{R_n}$ that maximize $\|\mathcal{B}\|_F^2$.

HOOI

It successively solves the restricted optimization problems

$$\min_{U^{(p)}} \|A - \mathcal{B} \times_1 U_1 \times_2 U_2 \cdots \times_T U_T\|_F^2, \tag{6}$$

in which optimization is done over the p-th matrix $U^{(p)}$.

Algorithm

- Input: $I \times J \times K$ tensor A and numbers r_1, r_2, r_3.
- Output: $L \in \mathbb{R}^{I\times r_1}, R \in \mathbb{R}^{J\times r_2}, V \in \mathbb{R}^{K\times r_3}, \mathcal{B}$.
- Choose initialR, V with orthonormal columns.
- Until convergence do:
  - $\mathcal{C} = A \times_2 R, T \times_3 V^T$, $L = \text{SVD}(r_1, \mathcal{C}_{(1)})$
  - $\mathcal{D} = A \times_1 L^T \times_3 V^T$, $R = \text{SVD}(r_2, \mathcal{D}_{(2)})$
  - $\mathcal{M} = A \times_1 L^T \times_2 R^T$, $V = \text{SVD}(r_3, \mathcal{M}_{(3)})$
- $\mathcal{B} = \mathcal{M} \times_3 V^T$

Metrics

- NDCG@k: Normalized Discounted Cumulative Gain

$$\mathrm{DCG}_k := \sum_{i}^{k} \frac{R_i}{\log_2(P_i + 1)}$$

$$\mathrm{IDCG}_k := \sum_{i}^{|\mathrm{REL}|} \frac{2^{R_i} - 1}{\log_2(P_i + 1)}$$

$$\mathrm{nDCG}_k := \frac{\mathrm{DCG}_k}{\mathrm{IDCG}_k}$$

where: |REL| is the cardinality of set of relevant documents
R_i - rating of i^{th} movie
P_i - position of i^{th} movie in prediction

Metrics

- **Precision@k**
  Precision at k is the proportion of recommended items in the top-k set that are relevant

$$\text{Precision@k} := \frac{|\text{REL}| \cap |\text{RET}|}{|\text{RET}|}$$

where $|\text{RET}|$ is the cardinality of set of retrieved documents

Example of SVD-based recommendation:

- Sin City. Rated: 4.0
- American History X. Rated: 4.0
- Traffic. Rated: 4.0
- Green Mile. Rated: 4.0
- Road to Perdition. Rated: 4.0

	SVD k=40	SVD k=30	SVD k=10	HOSVD k=(30, 30, 3)	HOOI k=(30, 30, 3)	HOSVD k=(10, 10, 3)	HOOI k=(10, 10, 3)	HOSVD k=(5, 5, 19)	HOOI k=(5, 5, 19)
NDCG@5	0.16	0.16	**0.182**	0.013	0.0128	0.012	0.012	0.011	0.01
NDCG@10	0.175	0.174	**0.199**	0.0137	0.0136	0.012	0.012	0.011	0.011
NDCG@20	0.196	0.195	**0.223**	0.0159	0.0158	0.014	0.015	0.014	0.013
Precision@1	0.224	0.223	**0.233**	0.023	0.023	0.023	0.025	0.028	0.022
Precision@5	0.2	0.201	**0.233**	0.021	0.021	0.019	0.019	0.015	0.015
Precision@10	0.221	0.22	**0.261**	0.0211	0.0214	0.019	0.019	0.017	0.017

Figure: Quality metrics

Conclusion

- Different factorization models are built for MovieLens 10M dataset
- Comparison of quality metrics
- Since we assumed in metric evaluation that rating of seen film (i.e. rating from learn) forced to be zero, tensor decompositions tend to memorize movies from learn (remember that we set missing rating to be zero).