Multi-relational Tensor Decomposition-Based Tag Recommendation Method (Postprint)

Authors: Zeng Hui, Hu Qiang, Gan Xiuxiu

Date: 2018-06-19T00:00:00+00:00

Abstract

Tag recommendation is widely applied across major websites, such as movie websites and e-commerce platforms; however, existing methods neglect the connections among multiple attribute features, failing to guarantee the accuracy of recommendation systems in big data environments. To address this issue, we propose a novel tag recommendation method based on user clustering and tensor decomposition to further improve the quality of tag recommendation. The method first clusters users who exert significant influence on products, then comprehensively calculates the total weight based on multivariate relationships among users, products, tags, and product ratings. Finally, it constructs a tensor according to the clustered user groups and the total weights of multivariate relationships, and performs tensor factorization. Compared with traditional tensor decomposition methods, experimental results demonstrate that the proposed method achieves a certain improvement in accuracy, thereby validating the effectiveness of the algorithm.

Full Text

A Tensor Decomposition Method for Tag Recommendation Based on Multiple Relationships

Zeng Hui, Hu Qiang, Gan Xiuxiu
College of Information Engineering, East China Jiaotong University, Nanchang 330013, China

Abstract

Tag recommendation is widely employed across various websites, including movie platforms and e-commerce sites. However, existing methods often overlook the connections among diverse attribute features, compromising recommendation accuracy in big data environments. To address this limitation,

we propose a novel tag recommendation method based on user clustering and tensor decomposition to further enhance recommendation quality. The approach first clusters influential users, then comprehensively calculates total weights based on multiple relationships among users, products, tags, and product ratings. Finally, it constructs a tensor using the clustered user groups and integrated relationship weights, followed by tensor factorization. Experimental results demonstrate that our method achieves improved accuracy compared to traditional tensor decomposition approaches, validating its effectiveness.

Keywords: tag recommendation; tensor factorization; weight; clustering

1 Introduction

Tag recommendation enables different users to annotate products (e.g., websites, movies) with different tags (word lists), then predicts future tagging behavior based on historical patterns to recommend products of interest to target users. For instance, if two distinct users have tagged the same item, they tend to use similar tags for other items in the future.

Currently, several tag recommendation methods employ tensor decomposition techniques for tag ranking, including Higher-Order Singular Value Decomposition (HOSVD) [1], Ranking with Tensor Factorization (RTF) [2], and n-dimensional tensor factorization for context-aware collaborative filtering [3]. Rendle et al. [2] introduced two distinct tensor data interpretations: a 0/1 scheme and a position-based ranking scheme, demonstrating that RTF achieves favorable prediction quality. Yang Qiuyong [4] proposed a weight-based ternary relationship interpretation scheme (LORTF), which showed some improvement but suffered from low computational efficiency. Krestel et al. [5] developed a Latent Dirichlet Allocation (LDA) approach that applies specialized terminology with latent topics to recommend articles to users, where words from the same topic may appear in other articles; however, this method lacks personalized consideration.

With the rapid development of recommendation algorithms, many models have been hybridized in recommendation systems, including Bayesian classifiers, clustering, artificial neural networks, and decision trees. Reference [6] incorporates social tags into two clustering methods: LDA-based K-means and generative clustering, proving the value of tags as an additional information source for clustering, though the resulting models lack a user dimension, leading to slightly lower F1 scores.

Tensor factorization models have been extensively studied across various domains [7-9]. Kolda et al. [10] described two distinct tensor factorization methods: CANDECOMP/PARAFAC and Tucker decomposition. Frolov et al. [11] introduced tensor applications in social tagging systems and various related tensor decomposition algorithms.

2 Methodology

This section first introduces the tensor factorization model, then clusters users based on their product usage activity and inter-user similarity. Within each user group, we identify relevant tuple relationships and represent tuple importance using a weight-based scheme. By integrating the comprehensive weights of "user-product-tag-rating" relationships, we construct a tensor model, optimize it, and finally generate recommendation lists for target users.

2.1 Tensor Factorization Model

This work primarily employs Tucker decomposition, a form of higher-order principal component analysis that decomposes a tensor into a core tensor and corresponding factor matrices along each mode. The decomposition principle is illustrated in [Figure 4: see original paper].

The tensor $\mathbf{X}$ decomposition can be expressed as:

$$\mathbf{X} \approx \mathbf{G} \times_1 \mathbf{U} \times_2 \mathbf{P} \times_3 \mathbf{T}$$

where $\mathbf{U} \in \mathbb{R}^{M \times r_1}$, $\mathbf{P} \in \mathbb{R}^{N \times r_2}$, and $\mathbf{T} \in \mathbb{R}^{K \times r_3}$ are low-rank feature matrices representing the principal components in each mode. The core tensor $\mathbf{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ captures interactions among the feature matrices, preserving the primary information from the original tensor with certain stability. Typically, the compressed tensor requires significantly less storage space than the original, meeting storage requirements. After learning the feature matrices and core tensor, predictions can be made as follows:

The hat notation indicates indexing along feature dimensions (e.g., $\widehat{u}$). The predicted tensor values can be derived from the equation to generate Top-N personalized recommendation lists.

2.2 User Clustering and Weight Calculation

2.2.1 User Clustering Due to large user sets, direct tensor construction would substantially increase algorithmic complexity. To improve data processing efficiency and recommendation accuracy, we cluster regular and active users into similar user groups. Using adjusted cosine similarity, we calculate inter-user similarity by treating each user' s product set as an input vector after subtracting the average rating of products rated by the current user. Active users u_a and regular users u_o are represented as m-dimensional vectors where vector elements represent product usage counts ts.

The similarity calculation is:

$$sim(u_a, u_o) = \frac{\sum_{k \in P_{a,o}} (ts_{a,k} - \bar{ts}_a)(ts_{o,k} - \bar{ts}_o)}{\sqrt{\sum_{k \in P_a} (ts_{a,k} - \bar{ts}_a)^2} \sqrt{\sum_{k \in P_o} (ts_{o,k} - \bar{ts}_o)^2}}$$

Based on similarity results, we select multiple active users as cluster centers and iteratively sort similarities to assign each regular user to the most similar user group, ultimately obtaining multiple cluster sets. We select two optimal clusters for experiments, constructing tuple data from their associated products, tags, and ratings for subsequent comprehensive weight calculation.

2.2.2 Comprehensive Weight Calculation The total comprehensive weight integrates different dimensional weights, representing the overall tuple relationship strength. User weights reflect activity levels based on product and tag usage quantities. Product weights must consider both user and tag influences on products, plus user rating magnitudes.

The user weight on a product $w_{p,u}$ is based on the average of all user weights using that product:

$$w_{p,u} = \frac{\sum_{k \in U_p} w_{u,k}}{|U_p|}$$

The tag weight on a product $w_{p,t}$ represents the proportion of tags marking a product relative to the total tag set:

$$w_{p,t} = \frac{\sum_{k \in T_p} w_{t,k}}{|T_p|}$$

The rating weight $w_{p,s}$ is the average rating given by the user community, calculated as the total rating divided by the number of rating instances.

Product total weight w_p integrates these three relationships with different importance levels, requiring distinct parameters:

$$w_p = \lambda_1 w_{p,u} + \lambda_2 w_{p,t} + \lambda_3 w_{p,s}$$

where $\lambda_i \in (0, 1)$.

Tag weight w_t considers only tag relationships, as user and product influences are already incorporated in previously calculated weights. Tag weight is computed as the ratio of a tag's occurrence frequency to the maximum tag usage frequency in the triple:

$$w_t = \frac{\text{times}(t)}{\max(\text{times})}$$

The comprehensive weight combines user, product, and tag weights with proportional parameters μ_i:

$$w_{u,p,t} = \mu_1 w_u + \mu_2 w_p + \mu_3 w_t$$

where $\mu_i \in [0, 1]$.

We classify all users into three categories based on activity: active users, regular users, and inactive users. Active users, representing the most influential group

with extensive product usage and tagging, receive higher weights. Inactive users exhibit minimal platform engagement with negligible reference value and are excluded from model training. Regular users constitute the majority and primary recommendation targets.

After obtaining all user weights, we sort them in descending order, selecting the top N users as the active set. Users below a threshold form the inactive set (excluded from training), while the remainder are regular users. We cluster regular and active users into similar groups, construct tuples from cluster data, calculate comprehensive weights, and build a three-dimensional tensor with users U, products P, and tags T as dimensions. Tucker decomposition and least squares optimization yield final Top-N personalized recommendations.

3 Experimental Results and Analysis

3.1 Dataset

We preprocessed the MovieLens dataset to obtain 16,3295 records (500 users, 9,289 movies, 1,128 tags). Each user rated viewed movies on a ten-level scale from 0.5 to 5 (increments of 0.5). Based on clustering results, we divided 500 users into six categories, selecting the two best-performing clusters for comparative experiments. The experimental data is summarized in .

** Dataset Description**

Cluster	Users	Products	Tags	Tuples
Cluster1	85	1,247	532	2,831
Cluster2	92	1,365	578	3,124

3.2 Parameter Settings

During user clustering, product and tag usage counts vary dramatically across users, with some active users accumulating product weights approaching 100. Therefore, we set relatively small parameters: $\alpha = 0.01$ and $\lambda_i = 0.01$. Product weight proportion must exceed tag weight for effective clustering, so we set $\mu = 0.05$.

In comprehensive weight calculation, we use $\lambda_1 = 0.01$, $\lambda_2 = 0.01$, $\lambda_3 = 0.05$, and $\mu_1 = \mu_2 = \mu_3 = 1$ (ensuring total weight 3). Rating parameter λ_3 is small because rating values themselves are large (1-5) while product weights must remain 1.

3.3 Evaluation Methods

We evaluate using precision, recall, and F-measure (F-score), commonly applied metrics in recommendation algorithm research. From both cluster user sets, we randomly sample partial product and tag data from each user as test set S_{test},

with remaining data as training set S_{train}. All metrics compare Top-1 to Top-15 recommendation lists.

3.4 Results and Analysis

We compare our method against four algorithms: "0/1 Scheme" [12], "LORTF" (weight-based ternary relationship) [4], "New Method" (our quadruple relationship weight scheme), and "4-D" (four-dimensional tensor decomposition) [11]. All methods use identical splitting and datasets (20% test, 80% train). The first four algorithms use a core tensor of dimension (8,8,8), while "4-D" uses (8,8,8,3) with ratings compressed to 1-5 levels to limit computational complexity and memory consumption.

Accuracy: As shown in [Figure 6: see original paper] and [Figure 7: see original paper], our method achieves superior accuracy across Top-1 to Top-15 lists for both clusters. Accuracy decreases gradually with increasing recommendation length, but consistently outperforms alternative methods.

Recall: [Figure 8: see original paper] and [Figure 9: see original paper] reveal that for Cluster1, our method excels in Top-1 to Top-3 but falls slightly behind as N increases. For Cluster2, recall performance surpasses other methods across all Top-N values.

F-Measure: [Figure 10: see original paper] and [Figure 11: see original paper] demonstrate that our method' s F-measure values exceed those of comparison methods for both clusters, particularly pronounced at Top-3 to Top-5. Overall F-measure values remain small due to dataset sparsity.

These results confirm that our method significantly outperforms alternatives by comprehensively considering multivariate relationship attributes and distinguishing importance through weighting, thereby indirectly improving recommendation accuracy.

4 Conclusion

This paper introduced three tensor data representation schemes (0/1 interpretation, ranking interpretation, and comprehensive weight interpretation). We clustered users based on activity and similarity, then calculated comprehensive weights incorporating product, tag, and rating relationships. Comparative evaluations demonstrate our method' s superior accuracy and rationality.

Future work will focus on: (a) optimizing parameters and incorporating additional influential factors to enhance weight calculation methods; (b) developing improved approaches to address data sparsity, thereby increasing computational speed and prediction accuracy.

References

[1] Symeonidis P. User recommendations based on tensor dimensionality reduc-

tion [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2008: 43-50.

[2] Rendle S, Marinho L B, Nanopoulos A, et al. Learning optimal ranking with tensor factorization for tag recommendation [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 727-736.

[3] Maroulis S, Boutsis I, Kalogeraki V. Context-aware point of interest recommendation using tensor factorization [C]// Proc of IEEE International Conference on Big Data. 2017: 963-968.

[4] Yang Qiuyong. Research of learning optimal ranking with tensor factorization for recommendation system [D]. Guangzhou: South China University of Technology, 2012.

[5] Krestel R, Fankhauser P, Nejdl W. Latent dirichlet allocation for tag recommendation [C]// Proc of ACM Conference on Recommender Systems. New York: ACM Press, 2009: 61-68.

[6] Lu X S, Zhou M C. Analyzing the evolution of rare events via social media data and k-means clustering algorithm [C]// Proc of IEEE International Conference on Networking, Sensing, and Control. 2016: 1-6.

[7] Acar E, Dunlavy D M, Kolda T G. A scalable optimization approach for fitting canonical tensor decompositions [J]. Journal of Chemometrics, 2015, 25(2): 67-86.

[8] Goulart J, Boizard M, Boyer R, et al. Tensor CP decomposition with structured factor matrices: algorithms and performance [J]. IEEE Journal of Selected Topics in Signal Processing, 2016, 10(4): 757-769.

[9] Zhou G, Cichocki A, Zhao Q, et al. Nonnegative matrix and tensor factorizations: an algorithmic perspective [J]. IEEE Signal Processing Magazine, 2014, 31(3): 54-65.

[10] Kolda T G, Bader B W. Tensor decompositions and applications [J]. Society for Industrial and Applied Mathematics, 2009, 51(3): 455-500.

[11] Frolov E, Oseledets I. Tensor methods and recommender systems [J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2016.

[12] Symeonidis P, Nanopoulos A, Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction [C]// Proc of IFIP International Conference on Artificial Intelligence Applications and Innovations. Boston: Springer, 2009: 331-340.

Note: Figure translations are in progress. See original paper for figures.