The review of CubeSVD: A Novel Approach to Personalized Web Search

Chi Zhang

Introduction

With the development of search engines, the demand of having a personalized search engine increases. Since the language has its ambiguity, users who type in the same query may get thousands of different documents related to different areas. For instance, a user typed in the word “jaguar” may refer to the cars or cats. In this case, a personalized web search engine will greatly improve the efficiency of ranking. However, most personalization algorithms involve extra work of humans to improve accuracy and develop a personalized model. The CubSVD take the user's clickthrough data for learning resource and apply the data to update the personalization model without extra work from the user. This new approach adopting users’ clickthrough data should be a strong way of improving the efficiency for different users.

Algorithm

Since the method CubeSVD is depending on the HOSVD technique, a generalization of matrix SVD, here is a brief explanation of the matrix SVD algorithm. An $I_1 \times I_2$ matrix F can be represented as

$$F = U^{(1)} \cdot S \cdot U^{(2)}$$

Where $U^{(1)} = (u_1^{(1)}\ u_2^{(1)} \cdots u_{I1}^{(1)})$ and $U^{(2)} = (u_1^{(2)}\ u_2^{(2)} \cdots u_{I2}^{(2)})$ represents the left and right singular vectors. Column vectors $u_i^{(1)}$ and $u_j^{(2)}$ where $1 \leqslant i \leqslant I_1$ and $1 \leqslant j \leqslant I_2$ are orthogonal to each other. S = diag(σ_1, σ_2, ···, $\sigma_{min(I1, I2)}$) is the diagonal matrix of singular values which satisfy $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_{min(I1, I2)} \geqslant 0$. By setting the smallest singular values in S to zero, F is approximated with a rank-k matrix and this approximation is best measured in reconstruction error. A CubeSVD represents the clickthrough data in triplets (u, q, p), which u is users, q for queries, and p for pages respectively. The data construct a 3-order tensor which measures the associations among the users, queries, and pages. In the CubeSVD algorithm, the preference of user and query pair is measured. The outcome of the algorithm is then weighted, smoothed, and normalized to achieve higher accuracy. Also, similar approaches are applied to collaborative filtering and Latent Semantic Indexing. By comparing the result between CubeSVD and other personalization algorithms, the importance of taking clickthrough data can be fully examined. Besides, by applying different weighting, smoothing, and normalizing methods, a better way of modifying personalization search data can be examined.

Conclusion

The test result is then compared with other personalization algorithms like CF or LSI, and CubeSVD achieves the highest accuracy. The higher accuracy compared to other algorithms is based on the associations among multi-type objects. The algorithm exploits the clickthrough data and learns from the data. These associations of the data cannot be captured properly by other algorithms. The log frequency weighting, similarity-based smoothing contributed to the highest accuracy.

$$f' = log_2(1 + f)$$

After modifying the data, the CubeSVD algorithm can update the information and learn from the data instantly without explicit feedback from the user. There are many future improvements. The clickthrough data may record the query and page clicked. It is better to combine the CubeSVD with the traditional content-based search model. Also, the computation of CubeSVD may be time-consuming offline to analyze a large set of data.

Reference

Sun, J., Zeng, H., Liu, H., Lu, Y., Chen, Z. (2005). CubeSVD: A Novel Approach to Personalized Web Search. *WWW '05: Proceedings of the 14th International Conference on World Wide Web.* (pp. 382-390)

J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 230–237. ACM Press, 1999.

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence, pages 43–52. Morgan Kaufman, 1998.

G. Golub and C. V. Loan. Matrix Computations, 2nd edition. The Johns Hopkins University Press, Baltimore, Maryland, 1989.