Towards Improving Top-N Recommendation by Generalization of SLIM

Santiago Larraín, Denis Parra, Alvaro Soto
Computer Science Department
Pontificia Universidad Católica de Chile
Santiago, Chile
slarrain@uc.cl,dparras@uc.cl,asoto@ing.puc.cl

ABSTRACT

Sparse Linear Methods (SLIM) are state-of-the-art recommendation approaches based on matrix factorization, which rely on a regularized ℓ_1-norm and ℓ_2-norm optimization –an alternative optimization problem to the traditional Frobenious norm. Although they have shown outstanding performance in Top-N recommendation, existent works have not yet analyzed some inherent assumptions that can have an important effect on the performance of these algorithms. In this paper, we attempt to improve the performance of SLIM by proposing a generalized formulation of the aforementioned assumptions. Instead of directly learning a sparse representation of the user-item matrix, we (i) learn the latent factors' matrix of the users and the items via a traditional matrix factorization approach, and then (ii) reconstruct the latent user or item matrix via prototypes which are learned using sparse coding, an alternative SLIM commonly used in the image processing domain. The results show that by tuning the parameters of our generalized model we are able to outperform SLIM in several Top-N recommendation experiments conducted on two different datasets, using both nDCG and nDCG@10 as evaluation metrics. These preliminary results, although not conclusive, indicate a promising line of research to improve the performance of SLIM recommendation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Recommender Systems, matrix factorization, SLIM

1. INTRODUCTION

The main goal of a recommender system is helping users dealing with information overload by providing personalized suggestions. This so-called "recommendation problem" has been addressed in different ways, such as predicting unobserved user ratings or as Top-N recommendation [2], where the objective is to optimize the ranking of the N most relevant items, to eventually recommend them. Under the paradigm of Top-N recommendation, Sparse Linear Methods (SLIM) have shown an outstanding performance over traditional matrix factorization algorithms, motivating the creation of extensions [7, 8] to the original work of Ning and Karypis [6].

If we compare SLIM with other similar applications that use ℓ_1-norm and ℓ_2-norm optimization, such as sparse coding used for dictionary learning in image processing, we can identify two assumptions. First, SLIM aims at reconstructing directly the low-density user-item matrix **A**, but low-density datasets decrease the effectiveness of Top-N recommendations [1]. We hypothesize that re-constructing a denser representation of the user-item interactions might produce an improvement in recommendation performance. Secondly, SLIM does not require to learn a dictionary of prototypes to reconstruct the matrix **A**, while dictionary learning is one of the most important stages of the matrix reconstruction in sparse coding. Our intuition is that by testing these two assumptions we might improve the ranking performance of SLIM. Hence, we can summarize the hypotheses for the generalized SLIM (gSLIM) that motivates our work in two: (1) in a first stage, learning a denser representation of the user-interaction matrix, such as the the low-rank users' and items' latent factor matrices, can improve the final Top-N recommendation performance, and (2) learning a dictionary of prototypes when reconstructing the low-rank user or item latent matrices can improve the final Top-N recommendation performance.

In this document, we explain our gSLIM approach, preliminary experiments on two datasets showing promising results, and finally we discuss limitations and some interesting ideas for future work.

2. GSLIM RECOMMENDATION MODEL

In order to model each user we use a traditional regularized matrix factorization [3] technique as defined in equation 1 where $\mathbf{A} \in \mathbb{R}^{U \times I}$ represents our rating matrix of U users and I items. $\mathbf{U} \in \mathbb{R}^{k_1 \times U}$ represents the latent users' matrix where each user $\vec{u}_i$ is represented with k_1 latent dimensions, and $\mathbf{V} \in \mathbb{R}^{k_1 \times I}$ represents the latent items' matrix where each item $\vec{v}_i$ is represented with k_1 latent dimensions as well.

$$\min_{\mathbf{U},\mathbf{V}} \quad ||\mathbf{A} - \mathbf{U}^T\mathbf{V}||_F^2 + \lambda_1(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) \tag{1}$$

After learning the latent users' matrix **U**, we proceed to compute the latent social prototypes. Similar as Karypis et. al. [6] we reconstruct each user $\vec{u}_i$ as a sparse lineal combination of our prototype matrix $\mathbf{P} \in \mathbb{R}^{k_1 \times k_2}$. In equation 2 we introduce a sparse coding problem [4] in order to compute our prototype matrix **P**, where k_2 is the number of prototypes. The matrix $\mathbf{W} \in \mathbb{R}^{k_2 \times U}$ represents the sparse coding coefficients and λ_2 is the sparsity parameter.

$$\min_{\mathbf{P},\mathbf{W}} \quad ||\mathbf{U} - \mathbf{P}\mathbf{W}||_F^2 + \lambda_2||\mathbf{W}||_1, \quad \text{s.t.} \quad ||w_i|| = 1 \quad \forall i \tag{2}$$

Algorithm	**ML-100K**		**ML-1M**	
	nDCG	nDCG@10	nDCG	nDCG@10
$k_1 = 30$,$k_2 = 200$, $\gamma = 9$	0.9357	0.3043	-	-
$k_1 = 20$,$k_2 = 943$, $\gamma = 5$	-	-	**0.9398**	0.2152
$k_1 = 20$,$k_2 = 200$, $\gamma = 10$	-	-	0.9392	0.2155
SLIM (baseline)	0.9202	0.2927	0.9230	**0.2374**
Traditional MF	**0.9361**	**0.3068**	0.9377	0.2111

Table 1: Results of nDCG and nDCG@10 for ML datasets.

Finally to make a recommendation $\hat{r}_{i,j}$ for the user i on the item j we multiply vectors $\hat{r}_{i,j} = \vec{u'}_i^T \cdot \vec{v}_j$, where user's reconstruction defined by equation 3 is solved by an Orthogonal Matching Pursuit (OMP) algorithm with a maximum of γ non-negative coefficients for coefficients $\vec{w}$.

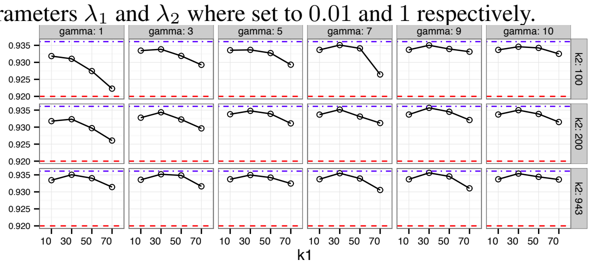
$$\vec{u}'_i = \mathbf{P} \operatorname*{argmin}_{\vec{w}} (\vec{u}_i - \mathbf{P}\vec{w}) \tag{3}$$

gSLIM compared to SLIM. Unlike our method, SLIM [6] solves directly a similar problem to equation 2, but instead of the latent users' matrix $\mathbf{U}$, it reconstructs the user-item matrix $\mathbf{A}$ without learning the prototypes $\mathbf{P}$, as shown in equation 4.

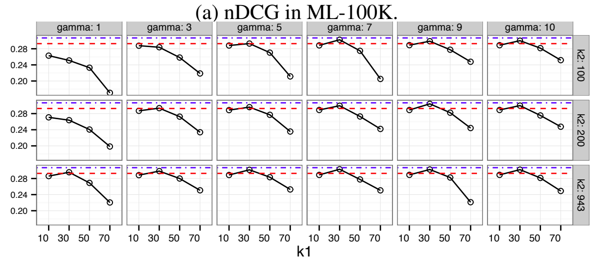
$$\min_{\mathbf{W}} \frac{1}{2}||\mathbf{A} - \mathbf{A}\mathbf{W}||_F^2 + \frac{\beta}{2}||\mathbf{W}||_F^2 + \lambda||\mathbf{W}||_1, \text{s.t.}\mathbf{W} \geq 0 \wedge diag(\mathbf{W}) = 0 \tag{4}$$

3. EXPERIMENTS

Using the *Movielens* datasets with 100K (ML-100K, $|u| = 943$, $|i| = 1682$) and 1M (ML-1M, $|u| = 6040$, $|i| = 3952$) ratings[1], we performed top-N recommendation experiments by using the *LensKit framework*[2], evaluating the performance with nDCG and nDCG@10 metrics [5]. We compared SLIM[3], a regularized matrix factorization (MF) as in equation 1, and gSLIM. For gSLIM we also conducted a parameter analysis: 1) k_1, latent dimensions for $\mathbf{U}$, 2) k_2, amount of different social prototypes, and 3) γ, number of non-negative coefficients on the user's sparse reconstruction. All experiments where conducted using 5-fold cross validation. The parameters λ_1 and λ_2 where set to 0.01 and 1 respectively.



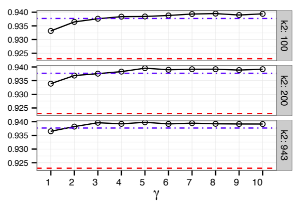
(a) nDCG in ML-100K.



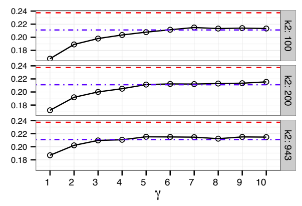
(b) nDCG@10 in ML-100K.

Figure 1: Results in ML-100K. Line styles: dashed=SLIM, dash&dot=MF, solid=gSLIM.

Results. Table 1 presents the results, showing that in the ML-100K dataset SLIM is outperformed by both gSLIM and regularized MF, indicating that this smaller and denser dataset (6.3%) might be better served by a traditional MF (hypothesis 1). However, in the larger and sparser ML-1M dataset (4.19%) SLIM out-



(a) nDCG in ML-1M



(b) nDCG@10 in ML-1M

Figure 2: Results in ML-1M dataset. Line styles: dashed=SLIM, dash&dot=MF, solid=gSLIM.

performs both MF and gSLIM in nDCG@10, but gSLIM outperforms both in terms of nDCG, indicating that reconstructing the denser latent user matrix with prototypes can support a recommendation task beyond a small top-N (hypothesis 2). Finally, figures 1a and 1b show the behavior in the ML-100K dataset of the parameters $k1$, $k2$ and γ in gSLIM performance compared to SLIM and MF baselines, and a similar behavior is seen in ML-1M dataset (Figures 2a and 2b). Analysis shows that larger $k2$ and γ give better performance but $k1$ has its peak between 20-30 latent factors for users' matrix $\mathbf{U}$.

4. CONCLUSIONS AND FUTURE WORK

In this work we introduced a general formulation for SLIM, *gSLIM*. Although we haven't addressed the limitation of our model in terms of computational complexity, we think that our results open an opportunity for studying unexplored ideas on Sparse Linear Methods. In the first step, researchers can try different methods to learn a denser representation of the $\mathbf{A}$, $\mathbf{U}$ or $\mathbf{V}$ matrices. In the step where we perform sparse coding, we could try reconstructing the items' latent factor matrix $\mathbf{V}$ rather than $\mathbf{U}$, or try directly $\mathbf{A}$. We can also try alternative algorithms to OMP for prototype learning. Even more, our intuition is that learning these sparse prototypes can help us finding actual "stereotypes" of users and items, which can be used, e.g., for clustering users and items.

5. REFERENCES

[1] Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 659–667. ACM, 2013.

[2] George Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 247–254, New York, NY, USA, 2001. ACM.

[3] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[4] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.

[5] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[6] Xia Ning and G. Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 497–506, Dec 2011.

[7] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 155–162, New York, NY, USA, 2012. ACM.

[8] Yong Zheng, Bamshad Mobasher, and Robin Burke. Cslim: Contextual slim recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 301–304, New York, NY, USA, 2014. ACM.

[1]http://grouplens.org/datasets/movielens/
[2]http://lenskit.org/
[3]http://www-users.cs.umn.edu/~xning/slim/html/