On the Use of Lanczos Vectors for Efficient Latent Factor-Based Top-N Recommendation

Athanasios N. Nikolakopoulos   Maria Kalantzi
John D. Garofalakis

Computer Engineering and Informatics Department,
University of Patras

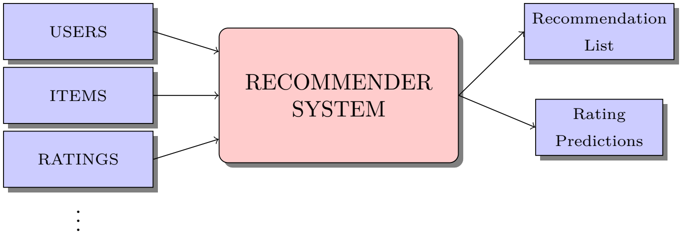
Computer Technology Institute & Press "Diophantus"

4th WIMS Conference, 2014

Outline

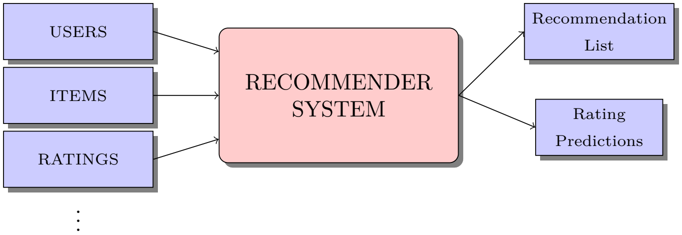
1. Introduction & Motivation
   - Recommender Systems - Collaborative Filtering
   - Challenges of Modern CF Algorithms
   - Latent Factor and Graph-Based models
2. Our Contribution
   - Motivation
   - Lanczos Latent Factor Recommender
   - LLFR Computational Aspects
3. Experimental Evaluation
   - Evaluation Methodology
   - Compared Algorithms and Metrics
   - Quality of Recommendations
   - Cold Start Recommendation
4. Conclusions
   - LLFR and Top-N Recommendation

Recommender System Algorithms



- Collaborative Filtering Recommendation Algorithms
  - Wide deployment in Commercial Enviroments
  - Significant Research Efforts

Recommender System Algorithms



- **Collaborative Filtering Recommendation Algorithms**
  - Wide deployment in Commercial Enviroments
  - Significant Research Efforts

Challenges of Modern CF Algorithms

Sparsity

- Intrinsic RS Characteristic
- **Cold start Problem**
- Traditional CF techniques, such as neighborhood models, are very susceptible to sparsity
- Among the most promising approaches in alleviating sparsity related problems are *Latent Factor* and *Graph-Based* models

Ranking - Based Algorithms

Graph-Based models

- Fouss et al.
  - Random walks on a graph model
- Gori and Pucci
  - ItemRank based on PageRank

Latent factor models

- Cremonesi et al.
  - **PureSVD**
    - Uses the truncated singular value decomposition to approximate the user-item rating matrix in order to produce recommendation vectors for the users.
    - Produces better top-N recommendations compared to sophisticated latent factor methods and other popular CF techniques.

Motivation

While promising in dealing with sparsity related problems, all the previous methods are **computationally expensive**.

- The *graph-based models* are required to handle a graph of n+m nodes.
- *PureSVD* involves the computation of a truncated singular value decomposition of the rating matrix.

In our approach, we follow the latent factor paradigm.

- **We are interested in ranking-based recommendations ⇒ not caring about the exact recommendation scores.**
- **Is there a cheaper way to reduce the dimensionality of the model?**

Motivation

While promising in dealing with sparsity related problems, all the previous methods are **computationally expensive**.

- The *graph-based models* are required to handle a graph of n+m nodes.
- *PureSVD* involves the computation of a truncated singular value decomposition of the rating matrix.

In our approach, we follow the latent factor paradigm.

- **We are interested in ranking-based recommendations ⇒ not caring about the exact recommendation scores.**
- **Is there a cheaper way to reduce the dimensionality of the model?**

Our Approach

We approach the problem as follows:

- Build a symmetric $m \times m$ **inter-item Correlation Matrix A**.
- Reduce the dimensionality of the model by computing the **Lanczos vectors** forming the basis of the Krylov subspace that corresponds to the inter-item correlation matrix A.
- Build a **Lower Dimensional Model** which can be readily used to produce **recommendation vectors for the users**.

Related Work

The Lanczos Method:

- has primarily been used in the context of *numerical linear algebra* [6]
- was found to achieve high quality results in applications from *Information Retrieval* as well as *Face Recognition* [7, 8]
- this is the first work to suggest using **Lanczos vectors for top-N recommendation**.

Lanczos Latent Factor Recommender (LLFR)

The Algorithm:

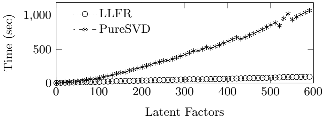
Lanczos Latent Factor Recommender (LLFR):
Input: The inter-item Correlation Matrix $\mathbf{A} \in \Re^{m \times m}$, the Rating Matrix $\mathbf{R} \in \Re^{n \times m}$, a random unit vector $\mathbf{q_1} \in \Re^{m}$, and the number of latent factors f.
Output: Matrix $\mathbf{\Pi} \in \Re^{n \times m}$ whose rows are the recommendation vectors for every user.

$\mathbf{q_0} \leftarrow \mathbf{0}$
$\beta_1 \leftarrow 0$
for $i \leftarrow 1, \ldots, f$ **do**
  $\mathbf{w} \leftarrow \mathbf{A q_i} - \beta_i \mathbf{q_{i-1}}$
  $\alpha_i \leftarrow \mathbf{w^T q_i}$
  $\mathbf{w} \leftarrow \mathbf{w} - \alpha_i \mathbf{q_i}$
  $\beta_{i+1} \leftarrow \|\mathbf{w}\|_2$
  $\mathbf{q_{i+1}} \leftarrow \mathbf{w}/\beta_{i+1}$
end for
return $\mathbf{\Pi} \leftarrow \mathbf{RQQ^T}$

Computational Aspects:

- $\mathcal{O}((nnz + m)f)$ time for sparse matrices
- where nnz is the number of nonzero elements of $\mathbf{A}$

Computational Tests:



Experimental Evaluation

Methodology

- We use the **Yahoo!Music** dataset.
- We have adopted the methodology used by Cremonesi et al:
  - Randomly sample 1.4% of the ratings of the dataset $\Rightarrow$ probe set $\mathcal{P}$
  - Use each item v_j, rated with 5 stars by user u_i in $\mathcal{P}$ $\Rightarrow$ test set $\mathcal{T}$
  - Randomly select another 1000 unrated items of the same user for each item in $\mathcal{T}$
  - Form ranked lists by ordering all the 1001 items according to the recommendation scores produced by each method

Recommendation Methods

We compare LLFR against:

- PureSVD
- *average Commute Time* (CT)
- *Pseudo-Inverse of the user-item graph Laplacian* ($L^\dagger$)
- *Matrix Forest Algorithm* (MFA)
- *ItemRank* (IR)

Accuracy Metrics

- **Recall**
- **Precision**
- **R-Score**

$$\text{R-Score}(\alpha) = \sum_q \frac{\max(y_{\pi_q} - d, 0)}{2^{\frac{q-1}{\alpha-1}}}$$

- **Normalized Distance-based Performance Measure**

$$\text{DCG@}k(\mathbf{y}, \boldsymbol{\pi}) = \sum_{q=1}^{k} \frac{2^{y_{\pi_q}} - 1}{\log_2(2+q)}$$

- **Mean Reciprocal Rank**

$$\text{RR} = \frac{1}{\min_q\{q : y_{\pi_q} > 0\}}$$

Recommendation Quality

- Evaluate the performance of the algorithms on low density data using the `Yahoo!Music` dataset.

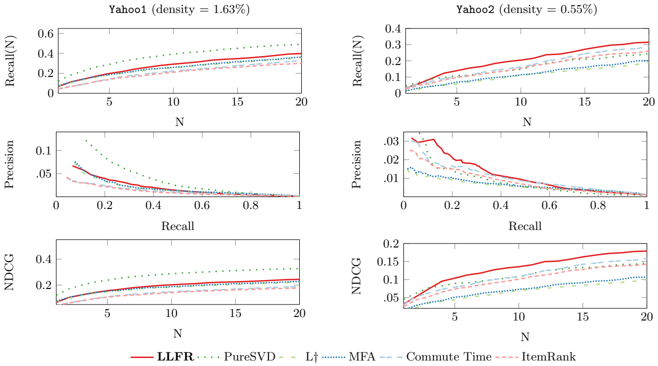


Figure: Evaluation of top-N recommendation performance.

The Cold Start Problem

Difficulty of making reliable recommendations due to an initial lack of ratings

- In beginning stages, when there is not sufficient number of ratings for the collaborative filtering algorithms to uncover similarities $\Rightarrow$ **New Community Problem**
- Introduction of new users to an existing system where they have not rated many items $\Rightarrow$ **New Users Problem**

New Community problem

Methodology:

- Randomly select to include 10%, 20%, and 30% of the overall ratings on three new artificially sparsified versions of the dataset.
- Create test sets from the new community datasets.

Table 1: Ranking Performance for the *New Community* Problem

	LLFR	PureSVD	L†	MFA	CT	IR
10%						
MRR	**0.1184**	0.1075	0.0106	0.0571	0.0197	0.0870
R-Score	**0.1474**	0.1296	0.0085	0.0563	0.0089	0.1028
20%						
MRR	**0.0874**	0.0722	0.0257	0.0271	0.0459	0.0630
R-Score	**0.1238**	0.1180	0.0309	0.0331	0.0728	0.0905
30%						
MRR	**0.0930**	0.0924	0.0316	0.0348	0.0646	0.0741
R-Score	**0.1352**	0.1289	0.0396	0.0454	0.1047	0.1117

Figure: Ranking Performance for the *New Community* Problem

New Users problem

Methodology:

- Randomly select 50 users having rated at least 100 items and randomly delete 95% of each users' ratings.
- Create the test set.

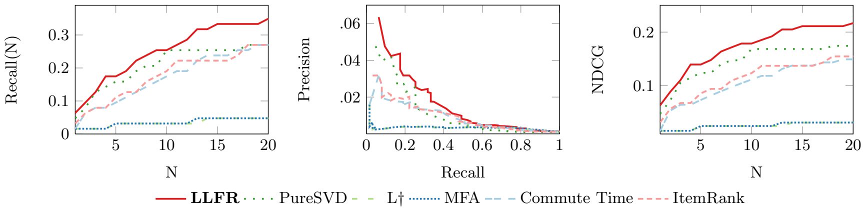


Figure: Performance evaluation of top-N recommendation for *New Users* problem.

Conclusions

LLFR

- Performs in a computationally efficient way
- Reduces the dimensionality of the problem by constructing the *Lanczos basis* of the Krylov subspace defined by a scaled inter-item correlation matrix
- Produces recommendations of high quality
- Deals particularly well with the *Cold-start Problem*
  - *New Community Problem*
  - *New Users Problem*
- **A promising candidate for large-scale recommendation scenarios**

References

Nikolakopoulos, A. N., Kouneli, M. and Garofalakis, J.:
A Novel Hierarchical Approach to Ranking Based Collaborative Filtering.
In: EANN 2013 (MHDW), Springer Verlag 2013.

Balakrishnan, S., Chopra, S.:
Collaborative ranking.
In: ACM WSDM 2012, New York, USA, ACM (2012) 143–152

Cremonesi, P., Koren, Y., Turrin, R.:
Performance of recommender algorithms on top-n recommendation tasks.
In: ACM RECSYS 2010, 39–46

Fouss, F., Pirotte, A., Renders, J., Saerens, M.:
Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation.
IEEE TKDE on **19**(3) (2007) 355–369

Gori, M., Pucci, A.:
Itemrank: a random-walk based scoring algorithm for recommender engines.
In: IJCAI 2007, San Francisco, CA, Morgan Kaufmann Publishers Inc. (2007) 2766–2771

G. H. Golub and C. F. Van Loan.
Matrix computations, volume 3.
JHU Press, 2012.

K. Blom and A. Ruhe.
A krylov subspace method for information retrieval.
SIAM Journal on Matrix Analysis and Applications, 26(2):566–582, 2004.

J. Chen and Y. Saad.
Lanczos vectors versus singular vectors for effective dimension reduction.
Knowledge and Data Engineering, IEEE Transactions on, 21(8):1091–1103, 2009.

Thanks!

Q&A

Lanczos Latent Factor Recommender Back

Inter-item Correlation Matrix $\mathbf{A} \in \mathfrak{R}^{m \times m}$

- Captures the similarities between the elements of the item space.
- ij^{th} element is given by:

$$A_{k\ell} \triangleq \|\mathbf{r_k}\| \|\mathbf{r}_\ell\| |\mathcal{U}_{k\ell}|,$$

  - $\|\mathbf{r_j}\|$ is the euclidean length of the column that corresponds to item v_j in the rating matrix,
  - $\mathcal{U}_{k\ell} \subseteq \mathcal{U}$ denotes the set of users who rated both items v_k and v_ℓ, i.e.

Lanczos Latent Factor Recommender

Production of the recommendation lists

- For each user u_i we define a personalized recommendation vector:

$$\boldsymbol{\pi}_{\mathbf{i}}^{\mathsf{T}} \triangleq \mathbf{r}_{\mathbf{i}}^{\mathsf{T}}\mathbf{Q}\mathbf{Q}^{\mathsf{T}}$$

  - $\mathbf{r}_{\mathbf{i}}^{\mathsf{T}}$ the ratings of user u_i
  - $\mathbf{Q} \in \mathfrak{R}^{m \times f}$ is the matrix that contains the Lanczos vectors forming the basis of the Krylov subspace $\mathcal{K}_f$ that corresponds to the inter-item correlation matrix $\mathbf{A}$