SLIM: Sparse Linear Methods for Top-N Recommender Systems

Xia Ning and George Karypis

Computer Science & Engineering
University of Minnesota, Minneapolis, MN
Email: {xning,karypis@cs.umn.edu}

December 14, 2011

University of Minnesota
Driven to Discover℠

Outline

Outline

Top-N Recommender Systems

- *Top-N* recommendation
  - E-commerce: huge amounts of products
  - Recommend a short ranked list of items for users
- *Top-N* recommender systems
  - Neighborhood-based Collaborative Filtering (`CF`)
    - Item based [2]: fast to generate recommendations, low recommendation quality
  - Model-based methods [1, 3, 5]
    - Matrix Factorization (`MF`) models: slow to learn the models, high recommendation quality
  - `SLIM`: Sparse LInear Methods
    - Fast and high recommendation quality

Definitions and Notations

Table 1: Definitions and Notations

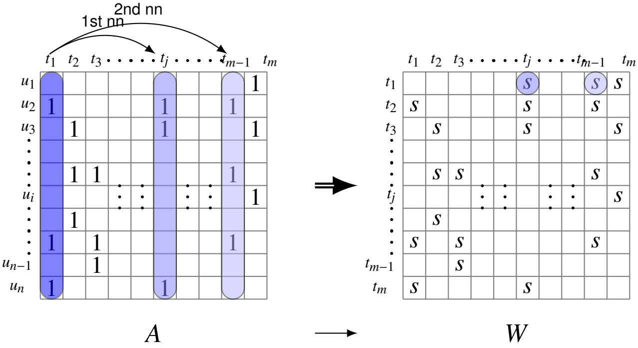
Def	Descriptions
u_i	user
t_j	item
$\mathcal{U}$	all users ($\lvert\mathcal{U}\rvert = n$)
$\mathcal{T}$	all items ($\lvert\mathcal{T}\rvert = m$)
A	user-item purchase/rating matrix, size $n \times m$
W	item-item similarity matrix/coefficient matrix
$\mathbf{a}_i^{\mathsf{T}}$	The i-th row of A, the purchase/rating history of u_i on $\mathcal{T}$
$\mathbf{a}_j$	The j-th column of A, the purchase/rating history of $\mathcal{U}$ on t_j

- Row vectors are represented by having the transpose supscript$^{\mathsf{T}}$, otherwise by default they are column vectors.
- Use matrix/vector notations instead of user/item purchase/rating profiles

The State-of-the-Art Methods

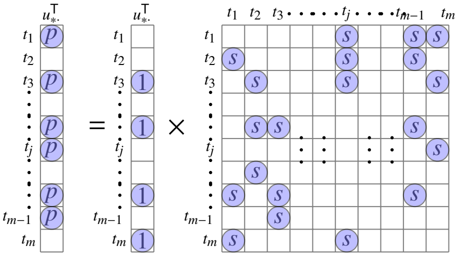
Item-based Collaborative Filtering (1)

- Item-based k-nearest-neighbor (`itemkNN`) `CF`
  - Identify a set of similar items
  - Item-item similarity:
    - Calculated from A
    - Cosine similarity measure



The State-of-the-Art Methods

Item-based Collaborative Filtering (2)



- `itemkNN` recommendation
  - Recommend similar items to what the user has purchased

$$\tilde{\mathbf{a}}_i^{\mathsf{T}} = \mathbf{a}_i^{\mathsf{T}} \times W$$

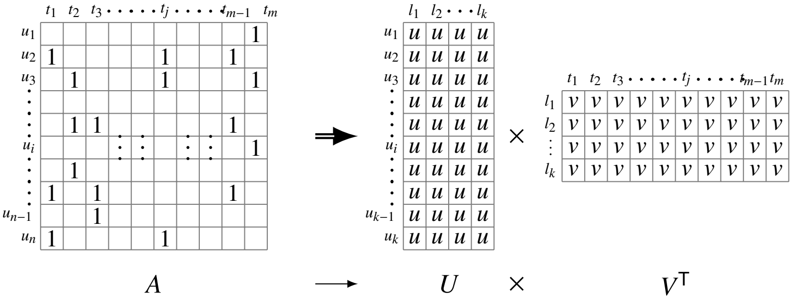
  - Fast: sparse item neighborhood
  - Low quality: no knowledge is learned

The State-of-the-Art Methods

Matrix Factorization (1)

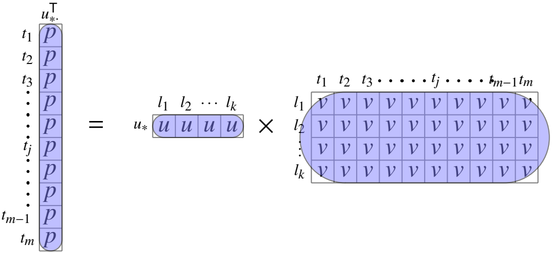
- Latent factor models
  - Factorize A into low-rank user factors (U) and item factors (V^{T})
    - U and V^{T} represent user and item characteristics in a common latent space
  - Formulated as an optimization problem

$$\underset{U,V^{\mathsf{T}}}{\text{minimize}} \quad \frac{1}{2}\|A - UV^{\mathsf{T}}\|_F^2 + \frac{\beta}{2}\|U\|_F^2 + \frac{\lambda}{2}\|V^{\mathsf{T}}\|_F^2$$



The State-of-the-Art Methods

Matrix Factorization (2)



- MF recommendation
  - Prediction: dot product in the latent space

$$\tilde{a}_{ij} = U_i^{\mathsf{T}} \cdot V_j$$

  - Slow: dense U and V^{T}
  - High quality: user tastes and item properties are learned

Outline

SLIM for *top-N* Recommendation

- Motivations:
  - recommendations generated *fast*
  - *high-quality* recommendations
  - "have my cake and eat it too"
- Key ideas:
  - retain the nature of itemkNN: sparse W
  - optimize the recommendation performance: learn W from A
    - sparsity structures
    - coefficient values

Learning W for SLIM

- The optimization problem:

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \frac{1}{2}\|A - AW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1 \\ \text{subject to} \quad & W \geq 0 \\ & \text{diag}(W) = 0, \end{aligned} \tag{1}$$

Learning W for SLIM

- The optimization problem:

$$
\begin{aligned}
\underset{W}{\text{minimize}} \quad & \frac{1}{2}||A - AW||_F^2 + \frac{\beta}{2}||W||_F^2 + \lambda||W||_1 \\
\text{subject to} \quad & W \geq 0 \\
& \text{diag}(W) = 0,
\end{aligned} \tag{1}
$$

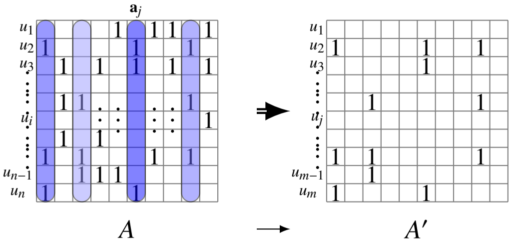
- Computing W:
  - The columns of W are independent: easy to parallelize
  - The decoupled problems:

$$
\begin{aligned}
\underset{\mathbf{w}_j}{\text{minimize}} \quad & \frac{1}{2}||\mathbf{a}_j - A\mathbf{w}_j||_2^2 + \frac{\beta}{2}||\mathbf{w}_j||_2^2 + \lambda||\mathbf{w}_j||_1 \\
\text{subject to} \quad & \mathbf{w}_j \geq \mathbf{0} \\
& w_{j,j} = 0,
\end{aligned} \tag{2}
$$

Reducing model learning time

$$\underset{\mathbf{w}_j}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{a}_j - A\mathbf{w}_j\|_2^2 + \frac{\beta}{2}\|\mathbf{w}_j\|_2^2 + \lambda\|\mathbf{w}_j\|_1$$

- `fsSLIM`: `SLIM` with *f*eature *s*election
  - Prescribe the potential non-zero structure of $\mathbf{w}_j$
  - Select a subset of columns from A
    - `itemkNN` item-item similarity matrix



$$\underset{\mathbf{w}_j}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{a}_j - A'\mathbf{w}_j\|_2^2 + \frac{\beta}{2}\|\mathbf{w}_j\|_2^2 + \lambda\|\mathbf{w}_j\|_1$$

Outline

Datasets, Evaluation Methodology and Metrics

Table 2: The Datasets Used in Evaluation

dataset	#users	#items	#trns	rsize	csize	density	ratings
ccard	42,067	18,004	308,420	7.33	17.13	0.04%	-
ctlg2	22,505	17,096	1,814,072	80.61	106.11	0.47%	-
ctlg3	58,565	37,841	453,219	7.74	11.98	0.02%	-
ecmrc	6,594	3,972	50,372	7.64	12.68	0.19%	-
BX	3,586	7,602	84,981	23.70	11.18	0.31%	1-10
ML10M	69,878	10,677	10,000,054	143.11	936.60	1.34%	1-10
Netflix	39,884	8,478	1,256,115	31.49	148.16	0.37%	1-5
Yahoo	85,325	55,371	3,973,104	46.56	71.75	0.08%	1-5

- Datasets: 8 real datasets of 2 categories
- Evaluation methodology: Leave-One-Out cross validation
- Evaluation metrics
  - Hit Rate: $$\text{HR} = \frac{\#\text{hits}}{\#\text{users}}$$
  - Average Reciprocal Hit-Rank (ARHR) [2]:

$$\text{ARHR} = \frac{1}{\#\text{users}} \sum_{i=1}^{\#\text{hits}} \frac{1}{p_i}$$

Outline

SLIM on Binary Data

Top-N recommendation performance

Figure 1: HR comparison

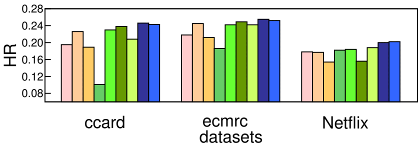


Figure 3: learning time comparison

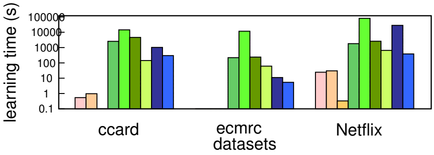


Figure 2: ARHR comparison

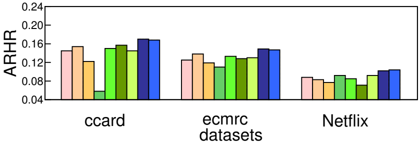
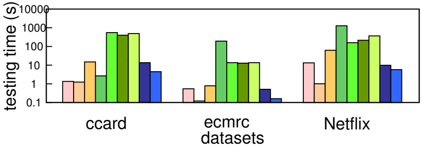


Figure 4: testing time comparison



itemkNN	PureSVD	BPRkNN
itemprob	WRMF	SLIM
userkNN	BPRMF	fsSLIM

SLIM on Binary Data

SLIM for Long-Tail Distribution

Figure 5: Rating Distribution in ML10M

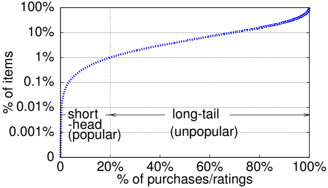


Figure 6: HR in ML10M tail

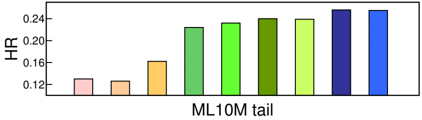
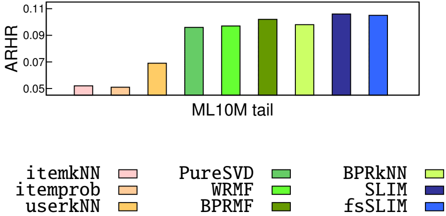


Figure 7: ARHR in ML10M tail



- SLIM outperforms the rest methods on the “long tail”.

SLIM on Binary Data

SLIM Recommendations for Different *top-N*

Figure 8: BX

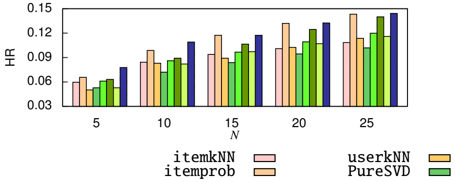
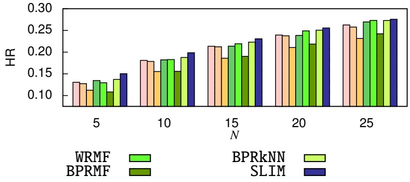


Figure 9: Netflix

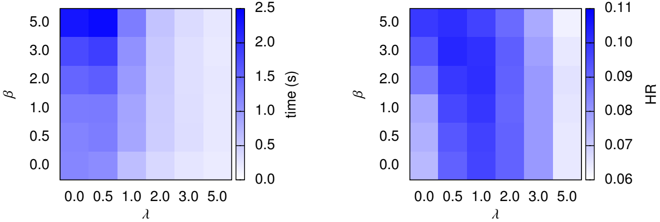


- The performance difference between SLIM and the best of the other methods are higher for smaller values of N.
- SLIM tends to rank most relevant items higher than the other methods.

SLIM on Binary Data

SLIM Regularization Effects

Figure 10: SLIM Regularization Effects on BX



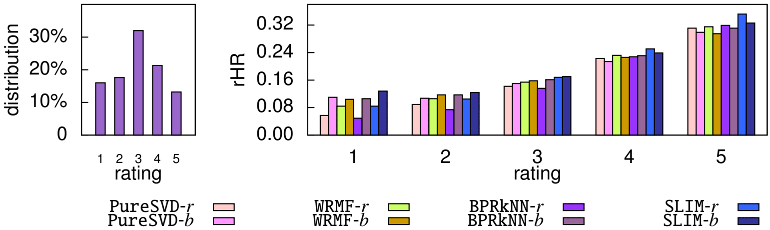
$$\underset{W}{\text{minimize}} \quad \frac{1}{2}\|A - AW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1$$

- As greater ℓ_1-norm regularization (i.e., larger λ) is applied, lower recommendation time is achieved, indicating that the learned W is sparser.
- The best recommendation quality is achieved when both of the regularization parameters β and λ are non-zero.
- The recommendation quality changes smoothly as the regularization parameters β and λ change.

SLIM on Rating Data

Top-N recommendation performance

Figure 11: SLIM on Netflix



- Evaluation metics:
    - per-rating Hit Rate: rHR
- All the -r methods produce higher hit rates on items with higher ratings.
- The -r methods outperform -b methods on high-rated items.
- SLIM-r consistently outperforms the other methods on items with higher ratings.

Outline

Conclusions

- SLIM: Sparse LInear Method for *top-N* recommendations
  - The recommendation score for a new item can be calculated as an aggregation of other items
  - A sparse aggregation coefficient matrix W is learned for SLIM to make the aggregation very fast
  - W is learned by solving an ℓ_1-norm and ℓ_2-norm regularized optimization problem such that sparsity is introduced into W
  - Fast and efficient

References

P. Cremonesi, Y. Koren, and R. Turrin.
Performance of recommender algorithms on top-n recommendation tasks.
In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

M. Deshpande and G. Karypis.
Item-based top-n recommendation algorithms.
ACM Transactions on Information Systems, 22:143–177, January 2004.

Y. Hu, Y. Koren, and C. Volinsky.
Collaborative filtering for implicit feedback datasets.
In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

S. Rendle, C. Freudenthaler, Z. Gantner, and S.-T. Lars.
Bpr: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic.
One-class matrix completion with low-density factorizations.
In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 1055–1060, Washington, DC, USA, 2010. IEEE Computer Society.

R. Tibshirani.
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society (Series B), 58:267–288, 1996.

Thank You!