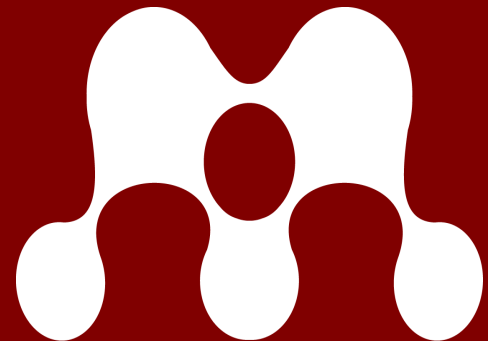


Efficient Top-N Recommendation by Linear Regression

Mark Levy
Mendeley



What and why?

- Social network products @Mendeley

iPad 3:16 PM 99%

All Documents Edit

Journal Article

A balance of protein synthesis and proteasome-dependent degradation determines the maintenance of LTP.

R. Fonseca, R. Vabulas, F. Hartl, T. Bonhoeffer, U. Nägerl

Neuron
2006 vol. 52 pp. 239-245

Long-lasting changes in synaptic strength are thought to play a pivotal role in activity-dependent plasticity and memory. There is ample evidence indicating that in hippocampal long-t...

Tags

Role of Protein Synthesis and Degradation in L-LTP
241

A Blockade of translation and the proteasome

B Blockade of translation and the proteasome

C Blockade of translation and the proteasome

Figure 3. Summary Plot Showing the Decay of LTP in Different Conditions

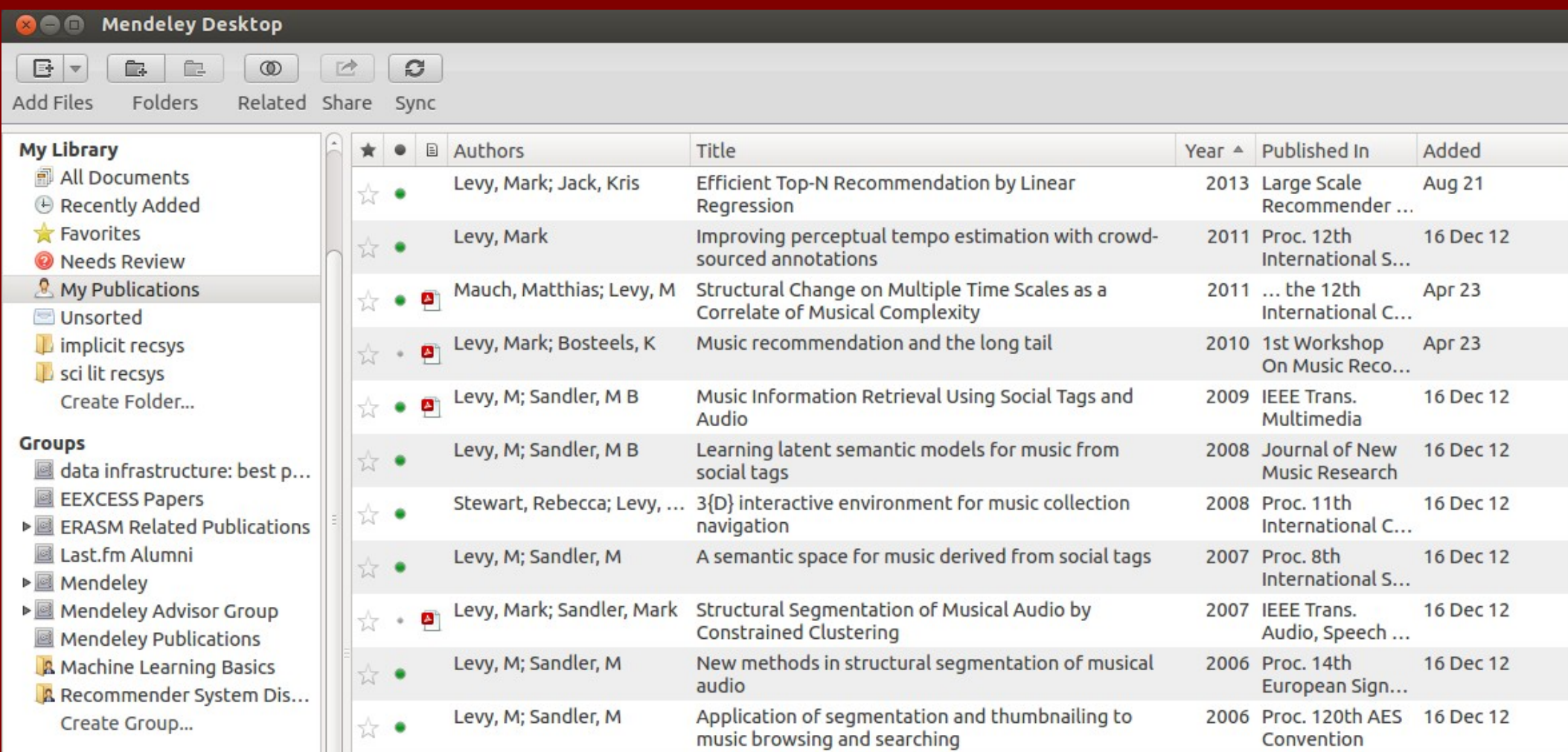
Percentage of decay of LTP values was determined by the ratio between 10 min data bins shortly after LTP induction (indicated by [1] in Figure 1A) and 10 min data bins at the end of the recording (indicated by [2] in Figure 1A). ([1]-[2])/[2] × 100%. Control values were averaged between all control experiments. Proteasome and translation blockers alone all strongly block L-LTP (columns 5-8), whereas concurrent application of both rescues L-LTP (columns 2-4). Error bars = SEM.

Condition	Percentage decay of LTP values
Cont	~10
Ani/Lacta	~10
Emet/Lacta	~10
Ani/Lacta/Epox	~20
Lacta	~40
Ani	~35
Emet	~38
Epox	~40

purple diamonds; emetine/lactacystin: 142% ± 6%, n = 7) were in this case significantly lower than for controls (Figure 2C, filled black circles; control: 162% ± 10% without blockade of protein synthesis and degradation).

What and why?

- Social network products @Mendeley



The screenshot shows the Mendeley Desktop application window. The title bar reads "Mendeley Desktop". Below the title bar is a toolbar with icons for "Add Files", "Folders", "Related", "Share", and "Sync".

On the left side, there is a sidebar with two main sections: "My Library" and "Groups".

My Library section includes:

- All Documents
- Recently Added
- Favorites
- Needs Review
- My Publications (selected)
- Unsorted
- implicit recsys
- sci lit recsys
- Create Folder...

Groups section includes:

- data infrastructure: best p...
- EEXCESS Papers
- ERASM Related Publications
- Last.fm Alumni
- Mendeley
- Mendeley Advisor Group
- Mendeley Publications
- Machine Learning Basics
- Recommender System Dis...
- Create Group...

The main area displays a list of publications in a table format. The columns are: Authors, Title, Year, Published In, and Added. Each row has a star icon and a green dot icon to its left.

★ ●	Authors	Title	Year	Published In	Added
★ ●	Levy, Mark; Jack, Kris	Efficient Top-N Recommendation by Linear Regression	2013	Large Scale Recommender ...	Aug 21
★ ●	Levy, Mark	Improving perceptual tempo estimation with crowd-sourced annotations	2011	Proc. 12th International S...	16 Dec 12
★ ●	Mauch, Matthias; Levy, M	Structural Change on Multiple Time Scales as a Correlate of Musical Complexity	2011	... the 12th International C...	Apr 23
★ ●	Levy, Mark; Bosteels, K	Music recommendation and the long tail	2010	1st Workshop On Music Reco...	Apr 23
★ ●	Levy, M; Sandler, M B	Music Information Retrieval Using Social Tags and Audio	2009	IEEE Trans. Multimedia	16 Dec 12
★ ●	Levy, M; Sandler, M B	Learning latent semantic models for music from social tags	2008	Journal of New Music Research	16 Dec 12
★ ●	Stewart, Rebecca; Levy, ...	3{D} interactive environment for music collection navigation	2008	Proc. 11th International C...	16 Dec 12
★ ●	Levy, M; Sandler, M	A semantic space for music derived from social tags	2007	Proc. 8th International S...	16 Dec 12
★ ●	Levy, Mark; Sandler, Mark	Structural Segmentation of Musical Audio by Constrained Clustering	2007	IEEE Trans. Audio, Speech ...	16 Dec 12
★ ●	Levy, M; Sandler, M	New methods in structural segmentation of musical audio	2006	Proc. 14th European Sign...	16 Dec 12
★ ●	Levy, M; Sandler, M	Application of segmentation and thumbnailing to music browsing and searching	2006	Proc. 120th AES Convention	16 Dec 12

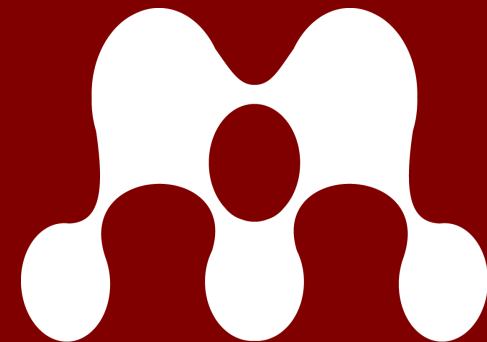
What and why?

- Social network products @Mendeley



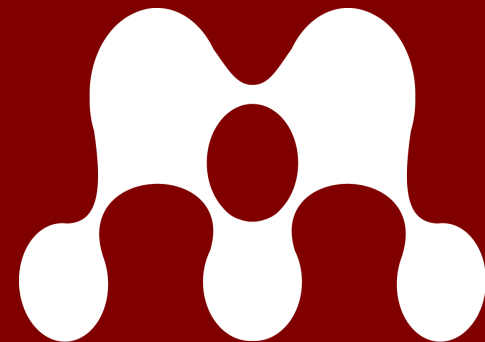
What and why?

- Social network products @Mendeley
- ERASM Eurostars project
- Make newsfeed more interesting
- First task: who to follow



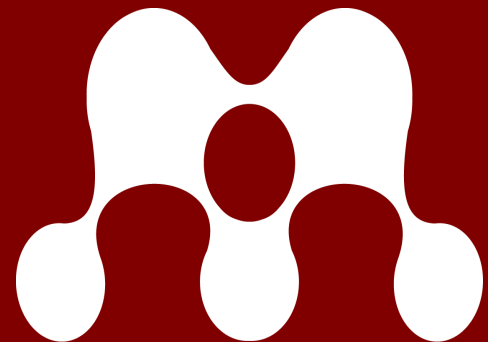
What and why?

- Multiple datasets
 - 2M users, many active
 - ~100M documents
 - author keywords, social tags
 - 50M physical pdfs
 - 15M <user,document> per month
 - User-document matrix currently ~250M non-zeros



What and why?

- Approach as item similarity problem
 - with a constrained target item set
- Possible state of the art:
 - pimped old skool neighbourhood method
 - matrix factorization and then neighbourhood
 - something dynamic (?)
 - SLIM
- Use some side data

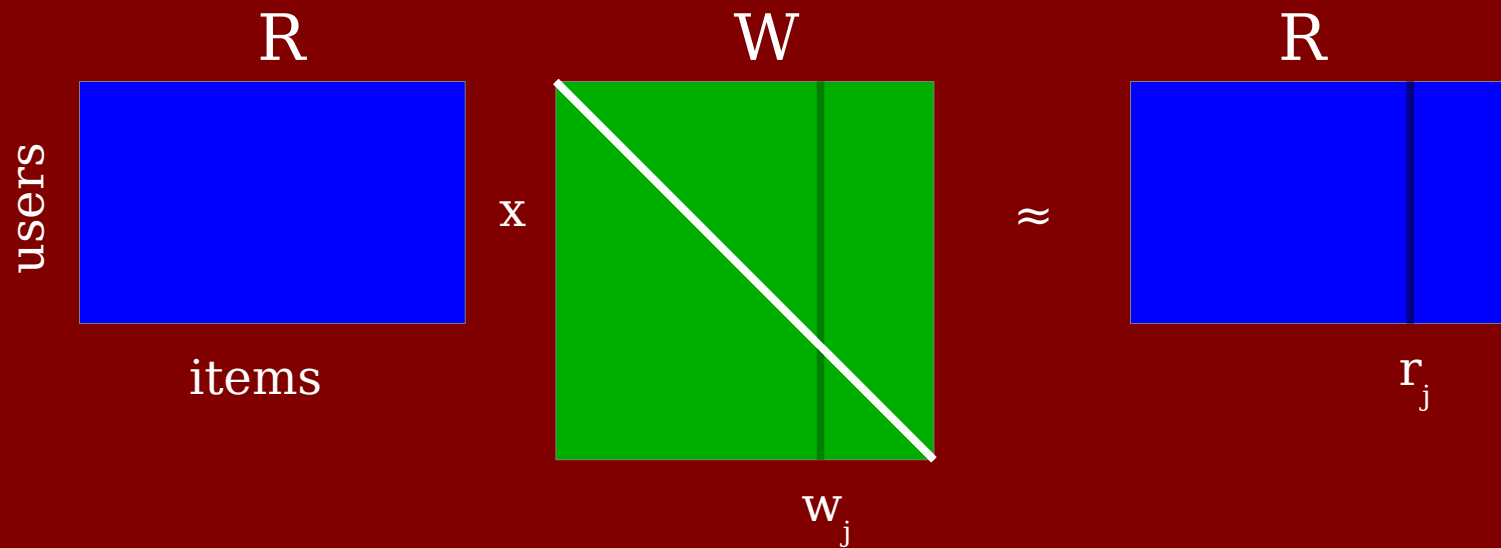


SLIM

- Learn sparse item similarity weights
- No explicit neighbourhood or metric
- L1 regularization gives sparsity
- Bound-constrained least squares:

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

SLIM



subject to $W \geq 0$
 $\text{diag}(W) = 0$

```
model.fit(R, r_j)
```

```
w_j = model.coef_
```

SLIM

Good:

- Outperforms MF methods on implicit ratings data [1]
- Easy extensions to include side data [2]

Not so good:

- Reported to be slow beyond small datasets [1]

[1] X. Ning and G. Karypis, SLIM: Sparse Linear Methods for Top-N Recommender Systems, Proc. IEEE ICDM, 2011.

[2] X. Ning and G. Karypis, Sparse Linear Methods with Side Information for Top-N Recommendations, Proc. ACM RecSys, 2012.

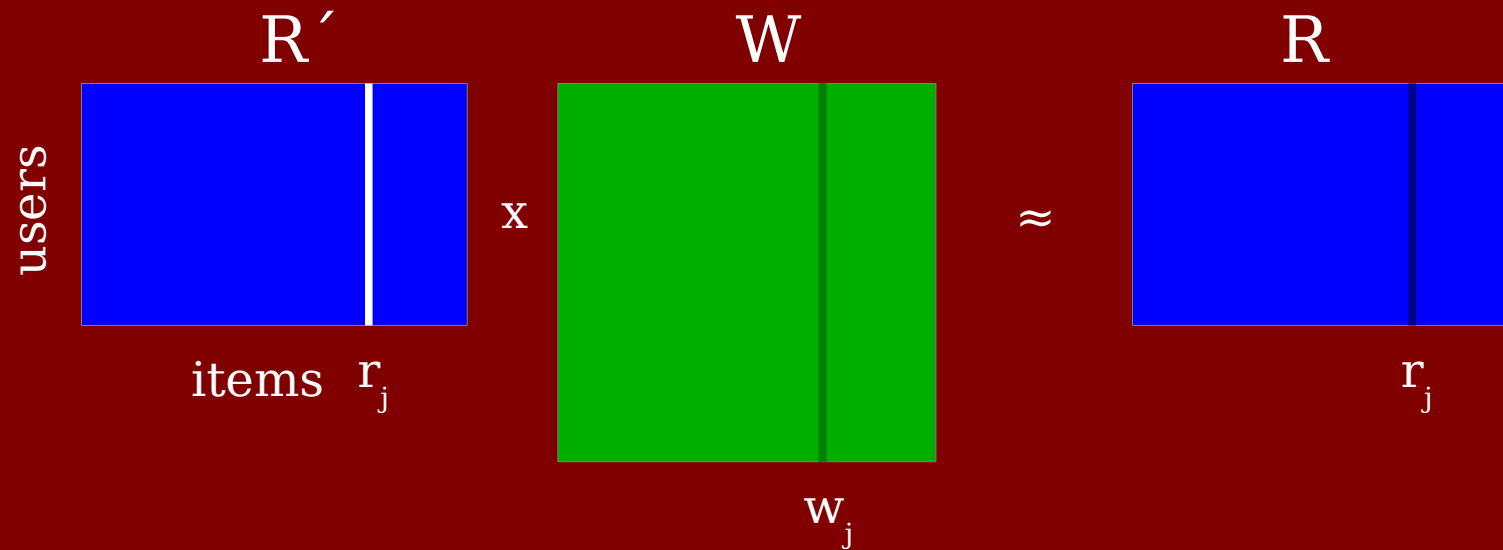
From SLIM to regression

- Avoid constraints
- Regularized regression, learn with SGD
- Easy to implement
- Faster on large datasets

$$\min_{\mathbf{w}_j} \quad \frac{1}{2} ||\mathbf{r}_j - R|_{j' \neq j} \mathbf{w}_j||_2^2 + \frac{\beta}{2} ||\mathbf{w}_j||_2^2 + \lambda ||\mathbf{w}_j||_1$$

$R|_{j' \neq j}$ is the rating matrix with the entries of the j -th column set to zero

From SLIM to regression



```
model.fit(R', r_j)
```

```
w_j = model.coef_
```

Results on reference datasets

Table 1: Performance on ML-100K dataset

method	prec@5	prec@10	prec@15	prec@20	MRR	HR@10
itemKNN	0.350	0.296	0.267	0.246	0.593	0.230
CoFiSet	0.411				0.640	
SLIM	0.344	0.301	0.274	0.253	0.584	0.291
SGDReg	0.414	0.358	0.324	0.298	0.664	0.386

Table 2: Performance on Epinions Trustlet dataset

method	prec@5	prec@10	prec@15	prec@20	MRR	HR@10
itemKNN	0.232	0.196	0.175	0.161	0.438	0.085
CoFiSet	0.244				0.433	
SLIM	0.232	0.205	0.187	0.174	0.411	0.151
SGDReg	0.291	0.254	0.230	0.213	0.495	0.166

Results on reference datasets

Table 3: Performance on ML-10M dataset

method	prec@5	prec@10	prec@15	prec@20	MRR	HR@10
itemKNN						0.238
BPRKNN						0.327
SLIM						0.311
SGDReg	0.612	0.555	0.514	0.481	0.792	0.332

Table 3: Time to learn weights for 100 items

dataset	method	time (secs)
ML-10M	SLIM	1055.5 (3.0)
ML-10M	SGDReg	82.4 (0.1)

Results on reference datasets

Table 3: Performance on ML-10M dataset

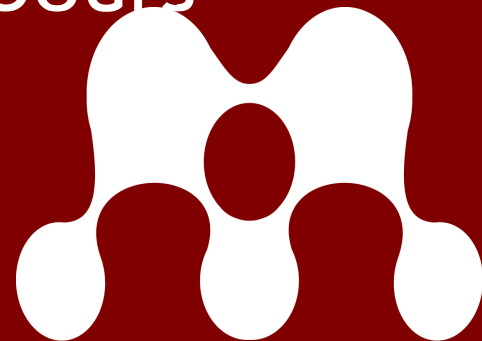
method	prec@5	prec@10	prec@15	prec@20	MRR	HR@10
itemKNN						0.238
BPRKNN						0.327
SLIM						0.311
SGDReg	0.612	0.555	0.514	0.481	0.792	0.332

Table 3: Time to learn weights for 100 items

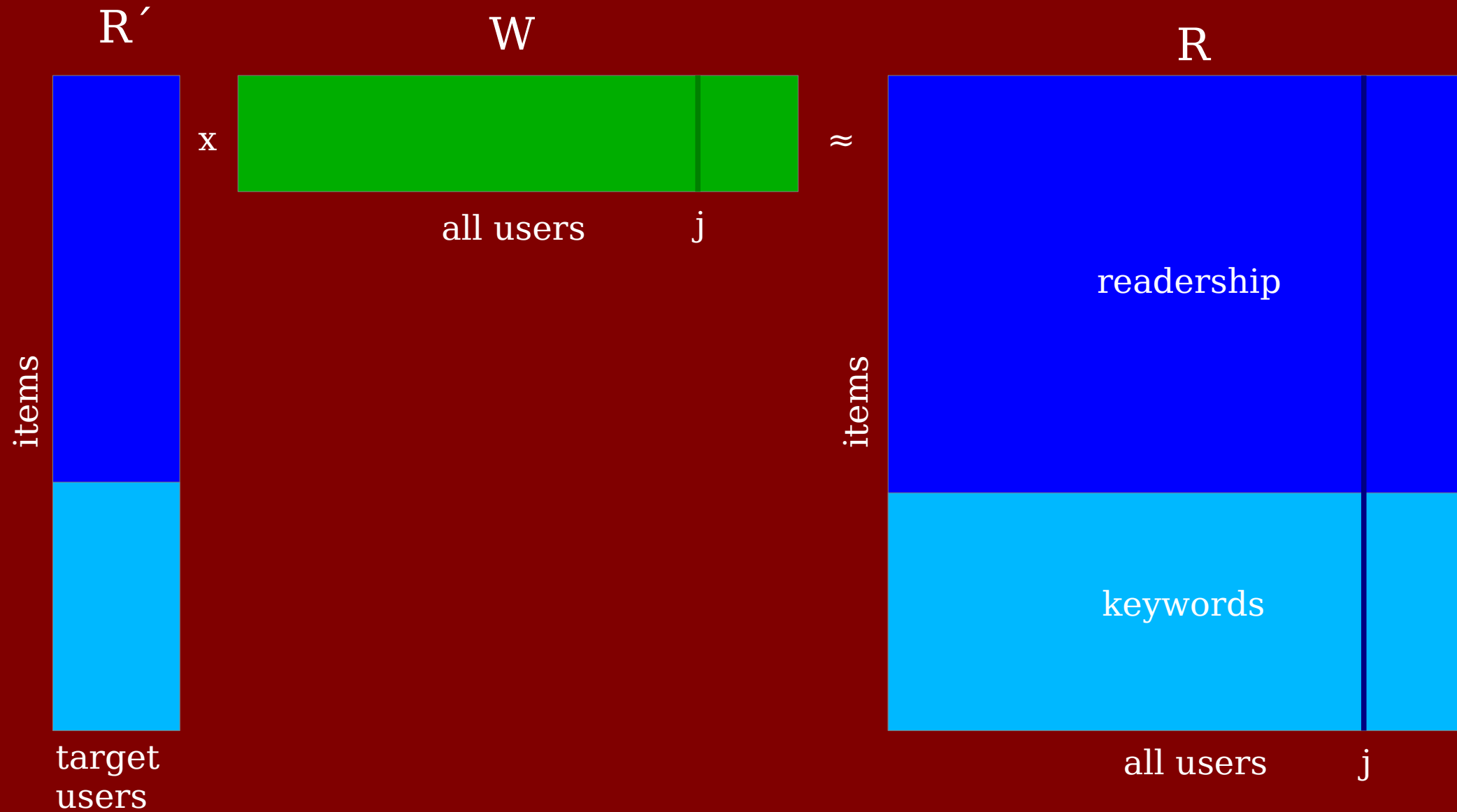
dataset	method	time (secs)
ML-10M	SLIM	1055.5 (3.0)
ML-10M	SGDReg	82.4 (0.1)

Results on Mendeley data

- Stacked readership counts, keyword counts
- 5M docs/keywords, 1M users, 140M non-zeros
- Constrained to ~100k target users
- Python implementation on top of scikit-learn
 - Trivially parallelized with IPython
 - eats CPU but easy on AWS
- 10% improvement over nearest neighbours

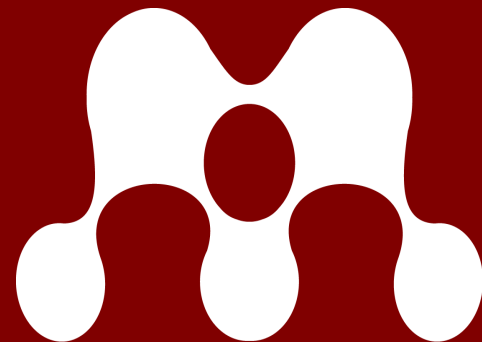


Our use case



Tuning regularization constants

- Relate directly to business logic
 - want sparsest similarity lists that are not too sparse
 - “too sparse” = # items with $< k$ similar items
- Grid search with a small sample of items
- Empirically corresponds well to optimising recommendation accuracy of a validation set
 - but faster and easier



Software release: mrec

- Wrote our own small framework
- Includes parallelized SLIM
- Support for evaluation
- BSD licence
- <https://github.com/mendeley/mrec>
- Please use it or contribute!

[mrec 0.1.2 documentation](#) »

Getting started with mrec

Install mrec

You can most easily install *mrec* with pip:

```
$ sudo pip install mrec
```

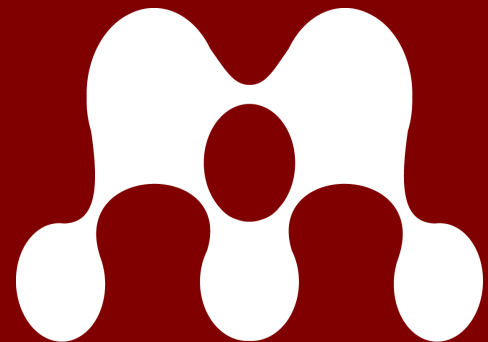
Thanks for listening

mark.levy@mendeley.com

@gamboviol

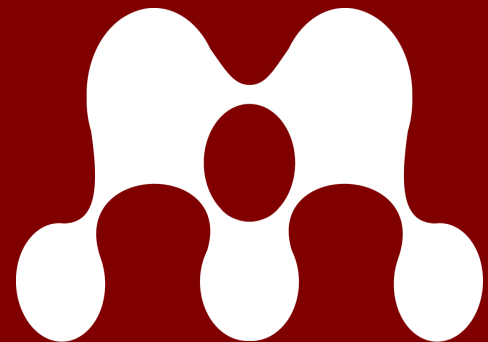
<https://github.com/gamboviol>

<https://github.com/mendeley/mrec>



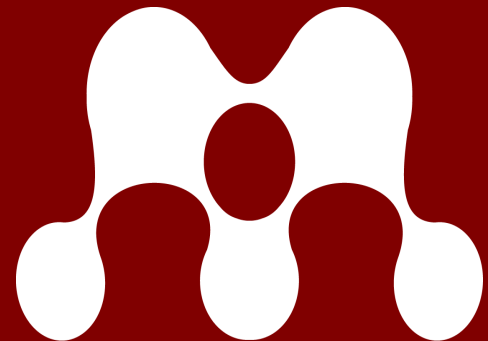
Real World Requirements

- Cheap to compute
- Explicable
- Easy to tweak + combine with business rules
- Not a black box
- Work without ratings
- Handle multiple data sources
- Offer something to anon users



Cold Start?

- Can't work miracles for new items
- Serve new users asap
- Fine to run a special system for either case
- Most content leaks
- ... or is leaked
- All serious commercial content is annotated



Degrees of Freedom for k-NN

- Input numbers from mining logs
- Temporal “modelling” (e.g. fake users)
- Data pruning (scalability, popularity bias, quality)
- Preprocessing (tf-idf, log/sqrt, ...)
- Hand crafted similarity metric
- Hand crafted aggregation formula
- Postprocessing (popularity matching)
- Diversification
- Attention profile

