Encoding temporal and structural information in machine learning models for recommendation

Tiphaine Viard[1] and Raphaël Fournier-S'niehotta[2]

[1] RIKEN AIP, Tokyo
[2] Conservatoire national des Arts et Métiers (CNAM), Paris

LEG workshop @ ECML-PKDD
20-09-2019

革新知能統合研究センター
Center for Advanced Intelligence Project

Outline

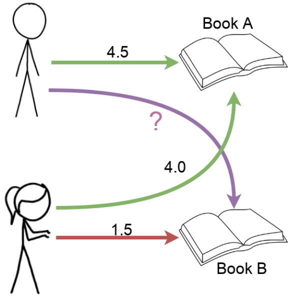
1 Recommendation

2 Model

3 Experiments

4 Conclusions & Perspectives

Context: collaborative filtering



Bipartite graph

Users \ Items	A	B	C	D
U		4.5	2.0	
V	3.5		4.0	
W		5		2.0
X	?	3.5	4.0	1.0

- Goal: predict unknown ratings
- Matrix factorization, LSH, etc.

- How to define similarities between users?

A time problem

- Both bipartite graph and matrix approaches initially discard notion of time, ratings given years apart are equal
- problems: taste drifts, profiles can get (very) large
- common solution: sequences of user-item matrices $\{M^k\}_k$, with a time step (Δ) ($M^k_{i,j} \neq 0$ indicates that user i has interacted with item j at least once in $[k, k+\Delta[$)
- similar to snapshots of graphs
- shortcoming: loss of information

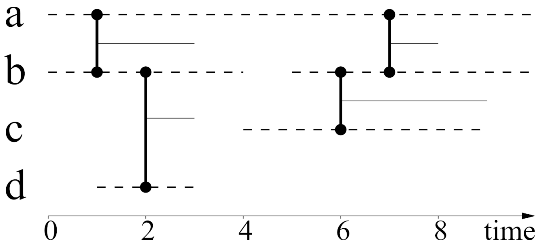
What if we had models
intricating time and structure?

Outline

1. Recommendation
2. Model
3. Experiments
4. Conclusions & Perspectives

Link streams framework

- Combine **structure and dynamics**



- $S = (T, V, W, E)$ [LVM17]
- T: interval, V: nodes, W: presences, E: interactions
- $c \times [4, 9] \in W$, $a, b \times [1, 3] \in E$
- Link stream $L = (T, V, E)$, stream with $W = T \times V$ (nodes present all the time)

Datasets

Movielens [HK16]

- Extensively studied by the RecSys community
- 20M interactions, 138k users, 27k movies, 20+ years
- ratings ($[0, 5]$) or (free) textual tags
- items with release year and genre (limited content)

Goodreads [WM18]

- restricted to Children genre only, for scalability
- 10M interactions (6.3M after cleaning),
  462k users, 122k books, 11+ years
- publication year and awards as item content

Feature engineering

- Content-based features
- Graph and link stream features
  - Content-based features
  - Neighborhood-based
  - Temporal features
  - Clique-based

Content-based

Features

- 39 content-based features in MovieLens
  - 19 one-hot encoding genres
  - 10 encoding release date (decade)
  - no NLP, but the number of *user-generated* tags
  - rating mean, median, std. dev.
  - min, max, number of ratings
- only 10 in Goodreads
  - no genre (Children books only)
  - rating mean, median, std. dev.
  - min, max, number of ratings
  - and 4 for decades

Graph features

Neighborhood-based features

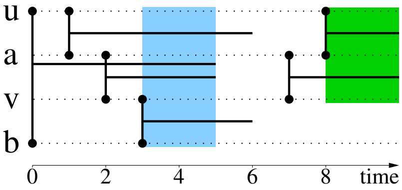
- degree $d_t(u) = |\{i : \exists (t, ui) \in E\}|, t \in T$
- evolution of the mean and max degree in LS:
  $d(u) = \frac{1}{|T|} \int_t d_t(u) dt = \frac{1}{|T|} \sum_{m \in I} |\{(t, ui) \in E_L\}|$
- (minimum discarded)
- assortativity of link (ratio between degrees of nodes)

  low assortativity = ”blockbusters”

Inter-contact times

- sequences of duration between two links for u
  from $t(u) = (t : (t, ui) \in E_L \cap T \times I \otimes \{u\})$, we use:
  $\tau(u) = (t_{i+1} - t_i)_{i=0}^{|t(u)|-1}$
- min, max, mean, std. dev

Link stream features: cliques



Cliques $(\{u,v\},\{a,b\},[3,5])$ (in blue) and $(\{u,v\},\{a\},[8,10])$ (in green)

- No clustering algorithm: sample of *balanced* maximal cliques [Via+18]
- Dense subgroup of users rating substantial number of items

Features

- balancedness of cliques involving u
- normalized average duration of cliques
- fraction of cliques containing u

Outline

1 Recommendation

2 Model

3 Experiments
- Experimental setup
- Evaluation results

4 Conclusions & Perspectives

Evaluation framework

XGBoost

- tree-based framework learning the probabilities of matching users and items in a recommendation context
- recently used in ACM RecSys Challenge
- tuned with bayesian optimization, best results with deep trees and small learning rate
- input: $(n \cdot m) \times f$ matrix (f features), output: $n \cdot m$ matrix P with predicted ratings

- 5-fold cross validation
- 10 hours 32 8-cores machine with 64GB RAM
- MAE, RMSE and NDCG@10 metrics

Results

Dataset	Metric	With stream features		
		# iter	train	test
MovieLens 20M	MAE	2134	**0.60234 (± 0.00022)**	**0.63427 (± 0.00012)**
	RMSE		**0.76349 (± 0.00043)**	**0.80954 (± 0.005)**
	NDCG@10		**0.99212 (± 0.015)**	**0.97209 (±0.019)**
Goodreads Children	MAE	998	**0.5299 (± 0.0002)**	**0.5899 (± 0.0003)**
	RMSE		**0.6818 (± 0.0002)**	**0.7561 (±0.0004)**
	NDCG@10		1.0 (± 0.0)	**0.9901 (± 0.019)**

Best result in litterature on Movielens: 0.7652 (RMSE) [SGM16] with a Deep learning approach (more complex model)

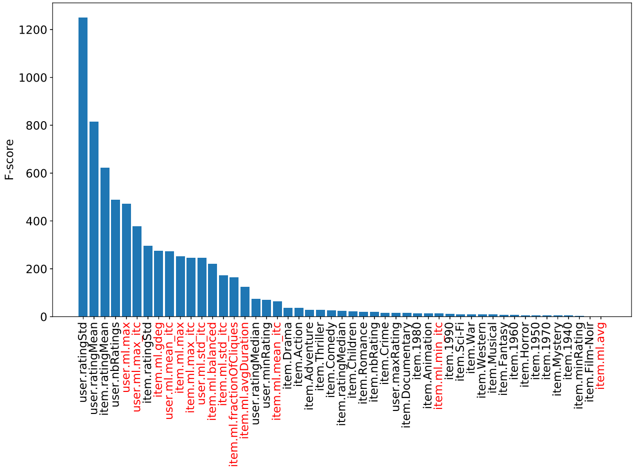
Dataset	Metric	Without stream features		
		# iter	train	test
MovieLens 20M	MAE	2134	0.63421 (± 0.00013)	0.644277 (± 8.6e-05)
	RMSE		0.82682 (± 0.00017)	0.83961 (± 0.00025)
	NDCG@10		0.98212 (± 0.022)	0.94863 (±0.048)
Goodreads Children	MAE	635	0.63343 (± 0.00042)	0.64986 (± 0.0003)
	RMSE		0.7986 (± 0.0005)	0.82652 (± 0.0003)
	NDCG@10		1.0 (± 0.0)	0.9772 (± 0.0454)

no baseline for Goodreads

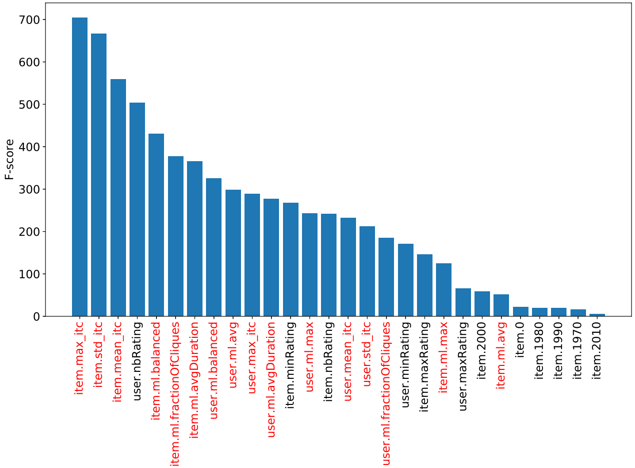
Feature importance

- XGBoost reports how features are used as split point by the boosting algorithm
- many link stream features are well-ranked (13 in top-20)
- intercontact times seem particularly useful, just after ratings

Feature importance : Movielens



Feature importance : Goodreads



Outline

1 Recommendation

2 Model

3 Experiments

4 Conclusions & Perspectives

Conclusions & Perspectives

- Proof-of-concept of incorporating link streams in classic RS
- Reasonable rating prediction on GoodReads and MovieLens
- Stream-graphs features seem important
- Explainable recommender systems

Future work

- Extend graph and link stream features: centrality, Jaccard index, etc.
- Work on community detection (open issue in LS): generalized modularity ?
- Evaluate our features in more advanced ML models, such as GCNs (GraphSage), currently with very few graph features

Contact

Thank you for your attention.

E-mails:
fournier@cnam.fr
and
tiphaine.viard@riken.jp

Source code:
`https://bitbucket.org/tiph_viard/social_recommendation`

Long version: [VF19]

F Maxwell Harper and Joseph A Konstan. "The movielens datasets: History and context". In: *Acm transactions on interactive intelligent systems (tiis)* 5.4 (2016), p. 19.

Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. "Stream Graphs and Link Streams for the Modeling of Interactions over Time". In: *Arxiv-CoRR* abs/1710.04073 (2017). arXiv: 1710.04073. URL: http://arxiv.org/abs/1710.04073.

Florian Strub, Romaric Gaudel, and Jérémie Mary. "Hybrid recommender system based on autoencoders". In: (2016), pp. 11–16.

Tiphaine Viard and Raphaël Fournier-S'niehotta. "Augmenting content-based rating prediction with link stream features". In: *Computer Networks* 150 (2019), pp. 127–133. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2018.12.002. URL: http://www.sciencedirect.com/science/article/pii/S1389128618313215.

Tiphaine Viard, Raphaël Fournier-S'niehotta, Clémence Magnien, and Matthieu Latapy. "Discovering patterns of interest in IP traffic using cliques in bipartite link streams". In: (2018), pp. 233–241.

Mengting Wan and Julian McAuley. "Item recommendation on monotonic behavior chains". In: (2018), pp. 86–94.

Parameters

Hyperparameter	Range	Meaning
α	$[0,1]$	L_1 regularization on feature weights
η	$[0,1]$	Learning rate of the algorithm
min_child_weight	$[0,\infty[$	Minimum number of (weighted) instances to create new leaf in the boosting tree
colsample_bytree	$[0,1]$	Subsample of features to use in each tree
γ	$[0,\infty[$	Minimum loss reduction required to make new leaf in the boosting tree
max_depth	$[0,\infty[$	Maximum depth of the trees
subsample	$]0,1]$	Ratio of the training dataset to subsample before building each tree

	Movielens 20M	Goodreads
α	0.3649	9.3515
η	0.1	0.1
min_child_weight	18.6967	11.5504
colsample_bytree	0.9112	0.9541
γ	0.9930	2.7449
max_depth	10	10
subsample	0.9810	0.9959