Trust-Based Collaborative Filtering

Tackling the Cold Start Problem Using Regular Equivalence

Tomislav Duricic, Emanuel Lacic
Dominik Kowald and Elisabeth Lex
{tduricic, elacic, dkowald}@know-center.at and elisabeth.lex@tugraz.at

Problem

Neighbor selection in Collaborative Filtering suffers from **data sparsity** and the **cold-start** problem.

Trust networks can be used to alleviate the problem, but are **often also sparse**.

Experimental Setup

Dataset:
Gathered from *epinions.com* with $49,290$ users, $139,738$ items, $664,824$ ratings, and $487,181$ trust connections.

Trust-graph density = 0.0002.

Baselines:
Most Popular (MP)
Naive trust-based CF ($Trust_{exp}$)
Jaccard trust-based CF ($Trust_{jac}$).

Adapted Katz ($KS_{a,b,c,d}$) approaches:

(a) Use Trust **P**ropagation wit l_{max} or **N**ot
(b) Use **C**ombined, **I**n-Degree or **N**o Degree Normalization
(c) Use L_1, L_2, **M**ax or **N**o Row Normalization
(d) **B**oosting of propagated trust values or **N**ot

Setting:
Simulating the cold-start problem by recommending $n = [1, 10]$ items for all users which have rated at least 10 items (= $25,393$ users)

Contribution

Explore the application of the **Katz similarity (KS) measure** for **cold-start** users in a **trust-based** CF approach.

Evaluate the resulting similarity matrix with different **normalization techniques** for a better recommendation accuracy.

Introduce an **adapted KS** measure that gives higher similarity values to node pairs with path lengths of 2.

Future Work

Investigate the impact of trust-based networks on **beyond accuracy** metrics such as novelty, diversity and coverage.

Explore the recently popularized **node embedding techniques** (e.g., *Node2Vec* or *GraphSAGE*) for trust networks.

References

[1] T. Duricic, E. Lacic, D. Kowald and E. Lex. Trust-Based Collaborative Filtering: Tackling the Cold Start Problem Using Regular Equivalence. In *Proc.* of the 12th ACM Conference on Recommender Systems (RecSys'18).

[2] E. Lacic, D. Kowald and E. Lex. Tailoring Recommendations for a Multi-Domain Environment. In *Proc.* of the Intelligent Recommender Systems by Knowledge Transfer & Learning (RecSysKTL) Workshop at RecSys '17.

Approach

Step 1: Calculating Katz Similarity with a chosen l_{max}. By using the iterative approach:

$$\boldsymbol{\sigma}^{(l_{max}+1)} = \sum_{l=0}^{l_{max}} (\alpha \mathbf{A})^l, \text{where } \boldsymbol{\sigma}^{(0)} = 0 \text{ and } \boldsymbol{\sigma}^{(1)} = \mathbf{I} \quad (1)$$

In the conducted experiments, we used values 1 and 2 for l_{max}, which means that we either have not propagated similarities through the network at all or that we propagated them through the network using a maximum path length of 2.

Step 2: Degree normalization. KS as defined in Eq. (1), tends to give high similarity to nodes that have a high degree. In some cases this might be desirable but if we want to get rid of this bias, we can apply a degree normalization on $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma}_{Dnorm}^{(l_{max}+1)} = \mathbf{D}^{-1}\left(\sum_{l=0}^{l_{max}} (\alpha \mathbf{A})^l\right)\mathbf{D}^{-1} \quad (2)$$

Step 3: Row normalization. We introduced an additional step where we individually scale rows of the final resulting matrix using one of the three vector norms: L_1, L_2 or max.

Step 4: Boosting propagated similarities. One of the contributions of this paper was to increase the impact of propagated trust values generated with KS for $l_{max} = 2$. Our proposed approach for doing this consists of the following four steps: (i) calculate $\boldsymbol{\sigma}^{(3)}$ as described above using the trust network as $\mathbf{A}$, (ii) create a new similarity matrix $\hat{\boldsymbol{\sigma}}$ such that:

$$\hat{\sigma}_{i,j} = \begin{cases} \sigma_{i,j}^{(3)}, & \text{if } A_{i,j} = 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

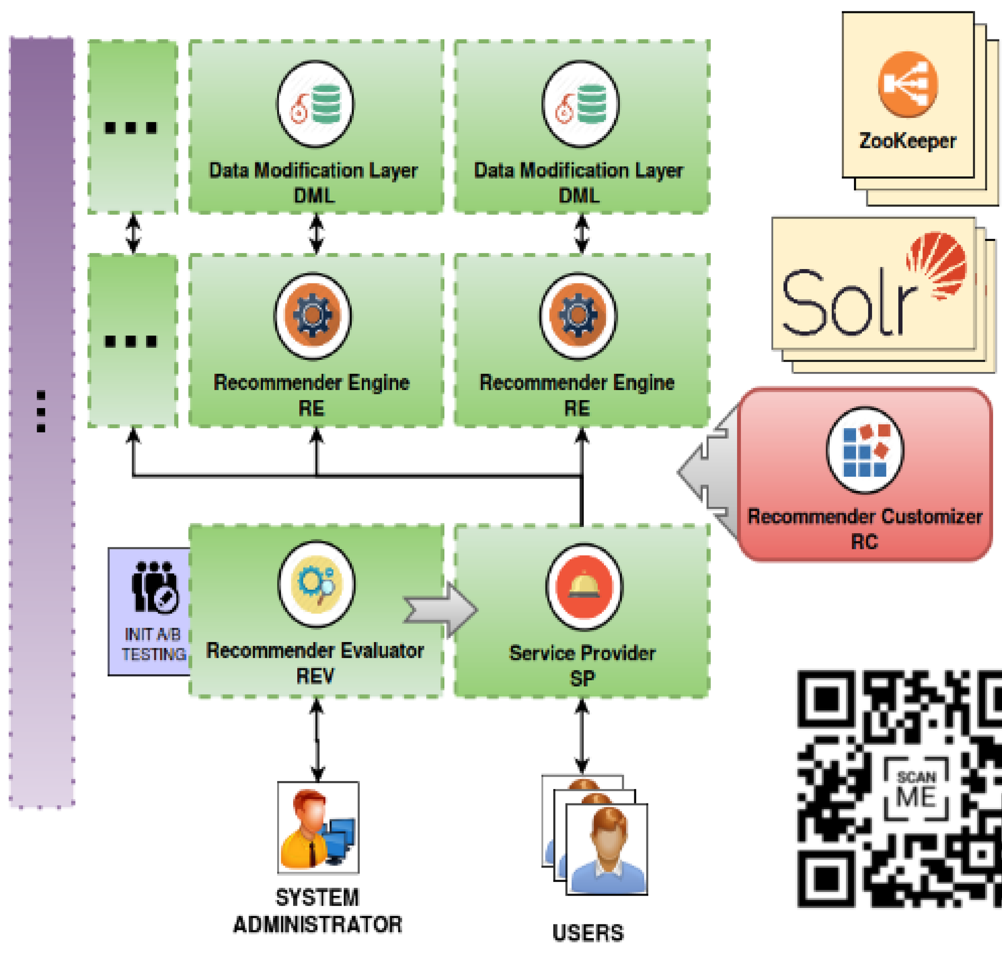
(iii) create $\hat{\boldsymbol{\sigma}}_{norm}$ matrix by individually scaling rows of $\hat{\boldsymbol{\sigma}}$ using L_1, L_2 or max vector norm and lastly, (iv) create a similarity matrix $\boldsymbol{\sigma}_{boost}$ such that:

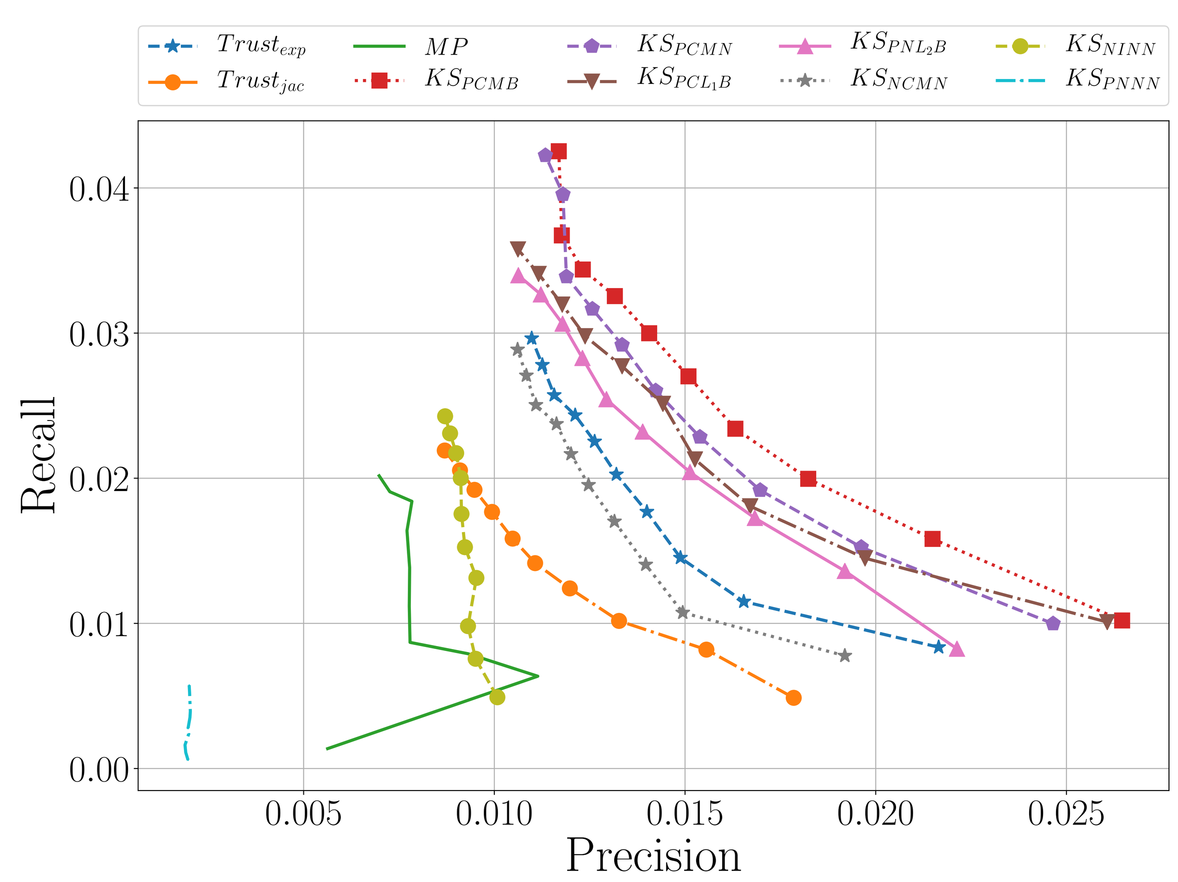
$$\boldsymbol{\sigma}_{boost} = \mathbf{A} + \hat{\boldsymbol{\sigma}}_{norm} \quad (4)$$

Evaluation

Evaluation results for $n = 10$. The reported subset of the 33 evaluated KS-based approaches are additionally labeled for an easier result comparison between different step combinations (i.e., columns 2 to 5).

Approach	l_{max}	Degree normalization	Row normalization	Boost	nDCG	Recall	Precision
$Trust_{exp}$					.0224	.0296	.0110
$Trust_{jac}$					.0176	.0219	.0087
MP					.0134	.0202	.0070
KS_{PCMB}	2	Combined	Max	Yes	**.0303**	**.0425**	**.0117**
KS_{PCMN}	2	Combined	Max	No	.0295	.0422	.0113
KS_{PCL_1B}	2	Combined	L1	Yes	.0273	.0358	.0106
KS_{PNL_2B}	2	No degree	L2	Yes	.0257	.0340	.0106
KS_{NCMN}	1	Combined	Max	No	.0213	.0289	.0106
KS_{NINN}	1	In degree	N/A	No	.0161	.0243	.0087
KS_{PNNN}	2	No degree	N/A	No	.0036	.0057	.0020





We implemented and evaluated our approach using ScaR [2], a scalable recommendation framework which is easily adaptable for a multi-domain environment.