Sentiment Analysis of User Comments for One-Class Collaborative Filtering

Nikolaos Pappas[1,2], **Andrei Popescu-Belis**[1]

{nikolaos.pappas,andrei.popescu-belis}@idiap.ch

[1]Idiap Research Institute, [2]École Polytechnique Fédérale de Lausanne

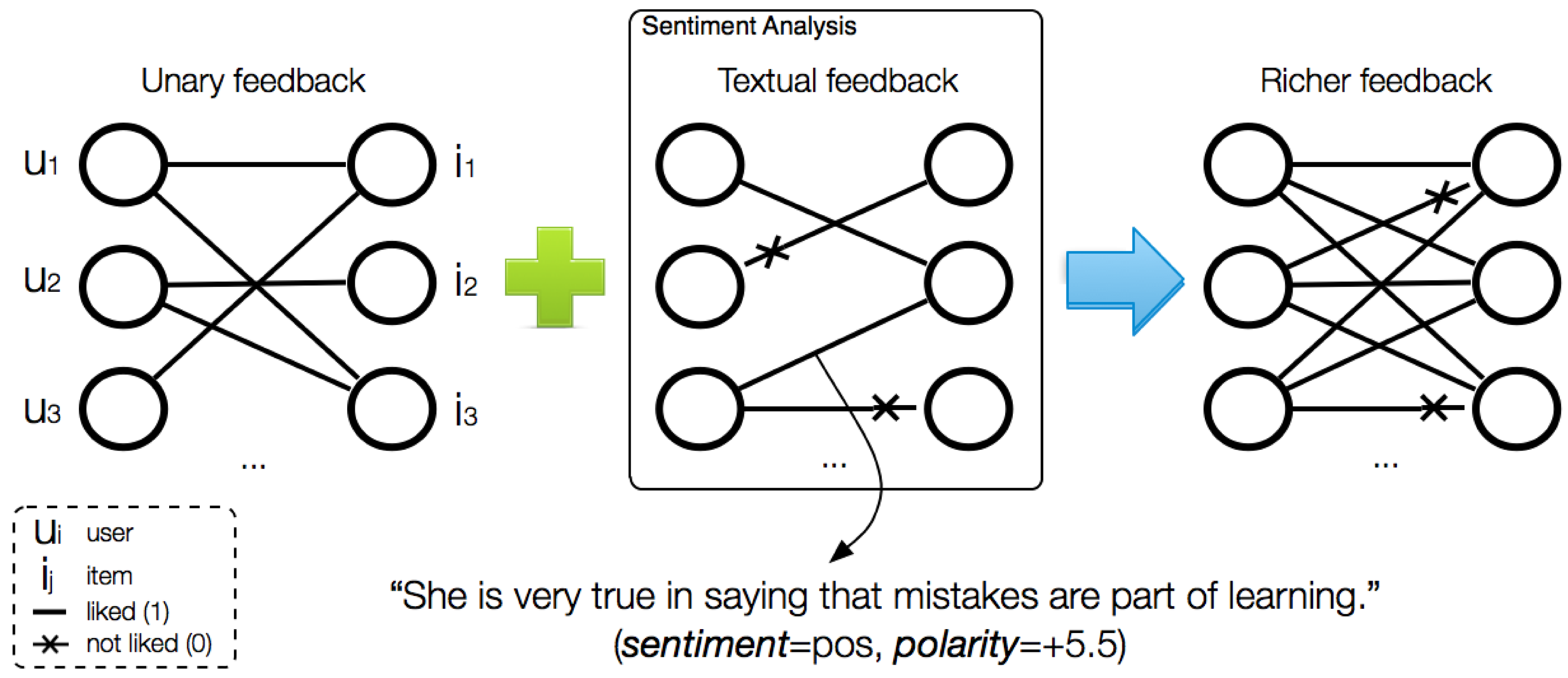
Motivation

Context:

→ user feedback expressed as unary values (e.g. favorites)
→ lack of a **negative** class
→ no feedback = not seen *or* not liked?

Objective

→ improve one-class collaborative filtering (CF)
→ extract preference information from user comments
→ enrich user preference model through sentiment analysis



Sentiment Analysis

→ out-of-the-box rule-based (RB) sentiment classifier from [1]
→ human study with 6 subjects over TED comments

Results:

→ quality **sufficient** to improve one-class CF task
  → 260 sentences and 135 comments (0.83 and 0.65 k score, substantial)
  → 76.20% F1 compared to 54.63% for random classifier

One-Class Collaborative Filtering

Find missing ratings in user-item matrix R ($|U| \times |I|$), with $r_{ui} = 1$ indicating an 'action' (positive feedback) [2, 3].

Neighborhood Models:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} d_{ij}(r_{uj} - b_{uj})}{\sum_{j \in D^k(u;i)} d_{ij}}; \; d_{ij} = s_{ij}\frac{n_{ij}}{n_{ij} + \lambda}; \; b_{ui} = \mu + b_u + b_i \qquad (1)$$

s_{ij}: similarity of item i and j, n_{ij}: common raters of i and j, λ: shrinking factor, b_{ui}: bias user u and item j estimate, $D^k(u;i)$: the neighborhood of size k of the most similar items in users u history to item i.

Sentiment-Aware Nearest Neighbors (SANN)

SANN model integrates preference information from text by **mapping** the sentiment scores of the text to preference ratings.

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in D^k_c(u;i)} d_{ij}(r'_{uj} - b_{uj}), \quad r'_{uj} = \begin{cases} 1, & \text{if } r_{uj} = 1 \\ c_{uj}, & \text{if } r_{uj} \neq 1 \end{cases} \qquad (2)$$

→ $c_{uj} = sign_{rand}(C_j)$: output of a random classifier (randSANN)
→ $c_{uj} = sign_{RB}(C_j)$: discrete output of RB classifier (SANN)
→ $c_{uj} = 1 + z_j \cdot \sum_{s \in C_j} (pol_{RB}(s)/|s|)$: numeric output of RB classifier (polSANN)

c_{uj}: inferred rating of user u to item i, Cj: user comment on item j, s: sentence, pol_{RB}: polarity output of RB classifier, z_j: normalization factor

TED Dataset and Evaluation Protocol

→ TED (www.ted.com) is an online repository of talks
→ contains user material (120k favorites, 22k comments)
→ we crawled, created and made TED data available (see [4])
→ evaluation on top-50 recommendations with P/R/F1 metrics
  → 80% training with 5-fold cross-validation
  → 20% testing with two held-out sets (comment dense and sparse)

References and Material

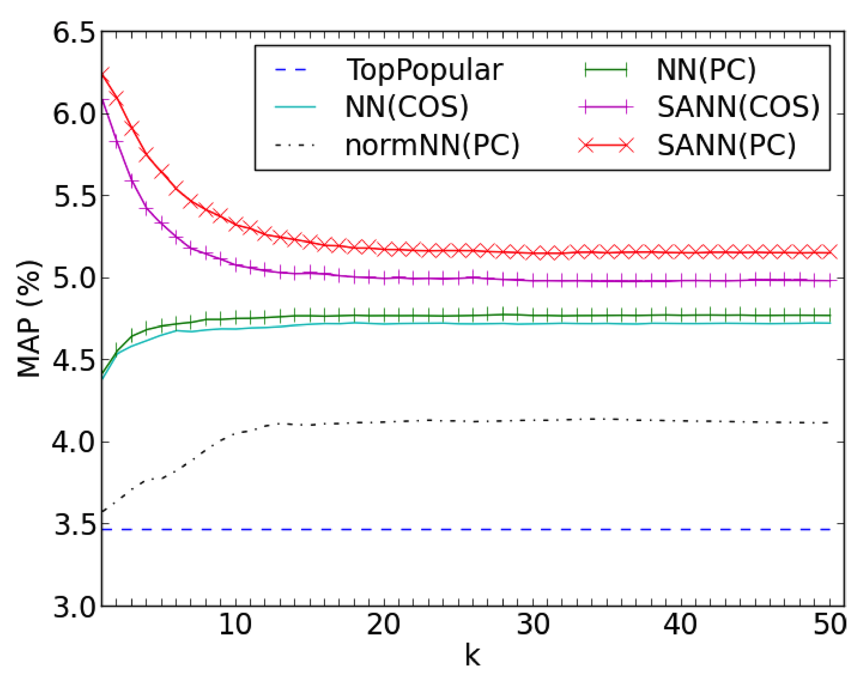
[1] N. Pappas, G. Katsimpras, and E. Stamatatos, "A system for up-to-date opinion retrieval and mining in the Web," in *CICLing'13*.
[2] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," in *RecSys'10*.
[3] R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *ICDM'08*.
[4] N. Pappas and A. Popescu-Belis, "Combining content with user preferences for TED lecture recommendation," in *CBMI'13*.

https://github.com/nik0spapp/unsupervised_sentiment
https://www.idiap.ch/data/ted

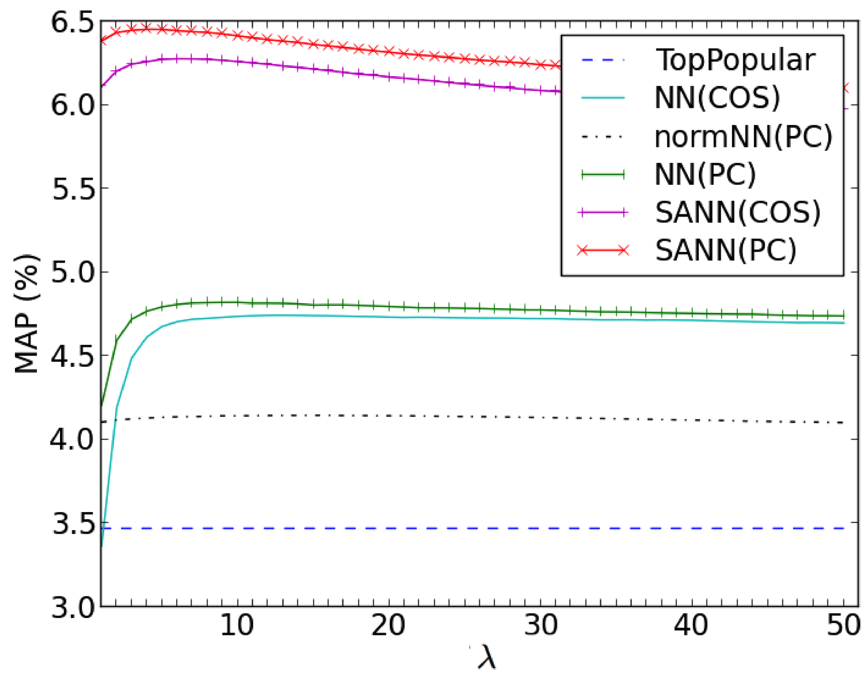
Evaluation on Training Set

→ optimizing for best combination of k, λ and similarity s_{ij}
→ consistent improvement of SANN over NN

Effect of neighborhood size k



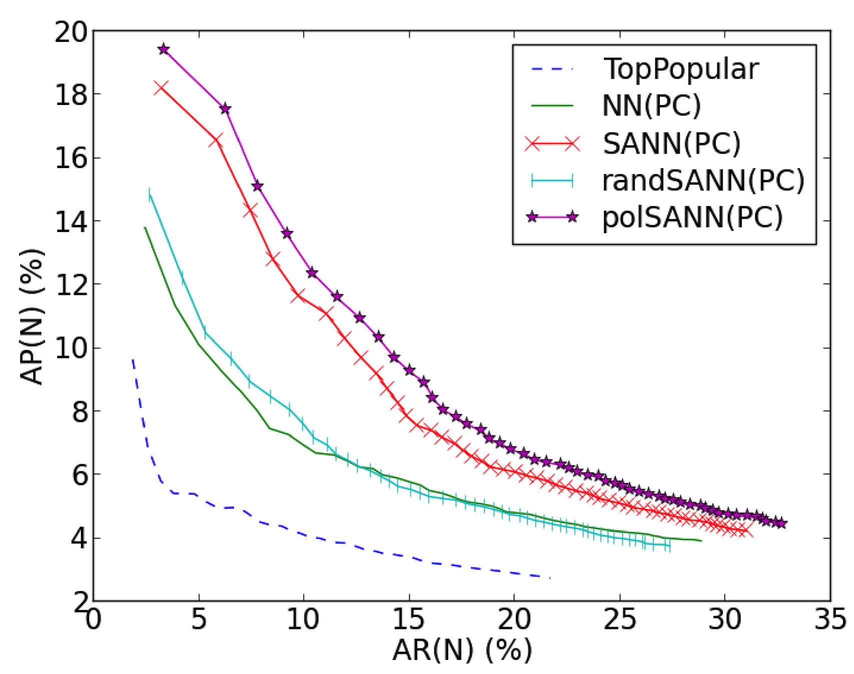
Effect of shrinking factor λ



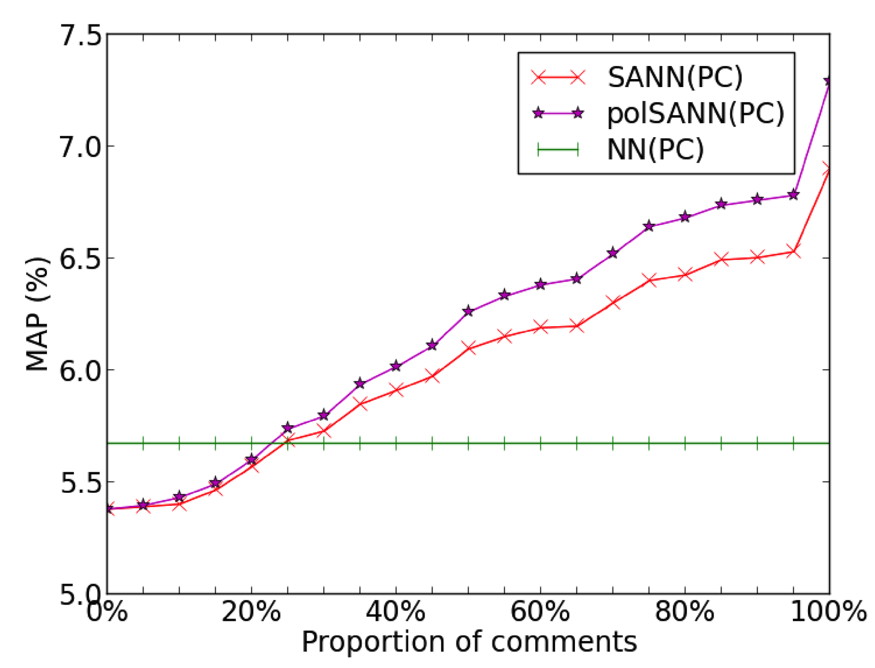
Results on Held-Out Sets

	Dense held-out set			Sparse held-out set		
Methods	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.85	13.52	5.99	3.61	13.48	5.70
NN	5.67	18.07	8.63	5.23	18.06	8.11
randSANN	5.88	17.79	8.84	5.22	17.56	8.05
SANN	6.90	20.72	10.35	5.69	18.85	8.75
polSANN	**7.29**	**22.01**	**10.95**	**5.89**	**19.48**	**9.04**
Improv.	+28.5%	+21.8%	+26.8%	+12.6%	+7.8%	11.4%

Precision vs. Recall



Role of comments



Conclusions

→ sentiment-aware nearest neighbor model **outperforms** competitive baselines for one-class CF
→ relies on the quality of the sentiment analysis module
→ performance improves with the number of comments

Future work:

→ learning to map sentiment scores to ratings
→ incorporate aspect-based information from user comments