Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 00 (2015) 1–28
ESWA

Adaptive Sentiment-Aware One-Class Collaborative Filtering

Nikolaos Pappas[a,b,*], Andrei Popescu-Belis[a,b]

[a] *Idiap Research Institute, Martigny, Switzerland*
[b] *École Polytechnique Fédérale de Lausanne, Switzerland*

Abstract

This paper presents a novel application of sentiment analysis to recommender systems relying on explicit one-class user feedback (favorites or likes), namely joint models of unary feedback and sentiment of free-form user comments. This combination is achieved through a mapping function within a sentiment-aware nearest neighbor model (SANN), which serves as an effective personalized ranker of items according to their hypothesized relevance to users. The mapping function can be adapted to specific datasets through a machine learning algorithm. We evaluate the proposed models and compare them with state-of-the-art multimedia recommendation methods, by casting the recommendation task as a top-N retrieval task over three real-world datasets: TED lectures, Vimeo videos and Flickr images. The experimental results show that the proposed models outperform all other alternatives in a majority of cases, thus demonstrating the generality of the approach. In particular, the superiority of the adaptive sentiment-aware models validates our hypothesis that there are inherent relationships between sentiments expressed in comments and unary feedback, both at community and individual levels. The improvements due to our models are consistent across all three datasets, they are present over three different assumptions on the negative class (i.e. items that are not seen or not liked), and they increase as comments become more abundant.

Keywords:
One-class collaborative filtering, Sentiment analysis, Multimedia recommendation

1. Introduction

Recommending items to users has become increasingly valuable for improving user experience as well as commercial revenue. Typical recommender systems are concerned with online products, movies, music or news.[1] The goal of such systems is to filter information and present to users only information that is relevant to them. A popular method that is used for this purpose is collaborative filtering, which aims to predict the preferences of an individual user based on items that have been previously rated by other similar users. Commonly, the ratings are given in the form of explicit numerical ratings, e.g. on a 1 to 5 scale. Often, however, ratings are only expressed through the users' behavior, such as marking as favorite or liking, i.e. more generally in terms of 'action' or lack thereof, i.e. 'inaction'. This kind of feedback is common in social media and is easier to obtain since it requires considerably less effort from users than numerical ratings. Its main drawback is the lack of a negative class: it is inherently unsure whether user inaction means that an item was not seen or was seen but not liked.

[*] *Address:* Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO 592, 1920 Martigny, Switzerland. *Phone:* +41 27 721 7711.
Email addresses: `nikolaos.pappas@idiap.ch` (Nikolaos Pappas), `andrei.popescu-belis@idiap.ch` (Andrei Popescu-Belis)

[1] Some examples are, respectively, Amazon (`http://www.amazon.com`), IMDb (`http://www.imdb.com`), Last.fm (`http://www.last.fm`) and Google News (`http://news.google.com`).

Dealing only with positive explicit feedback is usually referred to as the one-class collaborative filtering problem (Pan et al., 2008). There are several strategies to handle this problem. Hand-labeling negative instances, for example, converts the problem to a standard two-class collaborative filtering one, but is a time-consuming strategy. Alternatively, it is possible to make certain assumptions on the negative class, for example that the missing instances are all negative, or all unknown, but these assumptions bias the recommendation process. More sophisticated assumptions attempt to balance the solution and improve over the two extreme ones.

In this paper, we extract sentiment information from free-form user comments, which are available in abundance on social media websites, to improve one-class collaborative filtering. The sentiment information is integrated with a nearest neighbors model into a *sentiment-aware nearest neighbor model (SANN)* by mapping the sentiment scores to user ratings. We investigate several mappings, either direct ones using the output of a sentiment classifier, or adaptive ones, which adapt this output to user ratings through a learning algorithm. We evaluate our proposals against competitive recommendation models, over *three real-world multimedia datasets* – lectures from TED, videos from Vimeo, and images from Flickr – demonstrating consistent improvements that are independent from the negative class assumption and increase with the number of comments. Previous studies of sentiment analysis for collaborative filtering have mostly focused on user reviews composed of text and numerical ratings, however, to the best of our knowledge, the study of free-form user comments to complement unary ratings remains largely unexplored.

The paper is organized as follows. Section 2 compares our work with previous studies. Section 3 introduces and analyzes the datasets used in our experiments. Section 4 presents our sentiment analysis component, including evaluation results and sentiment-level statistics of the datasets. Section 5 formally defines the one-class collaborative filtering problem and describes the models we propose. Section 6 presents the experimental setup and evaluation protocol, while Sections 7–9 describe in detail our empirical studies and analyze their results. Finally, Section 10 concludes the article with directions for future work.

2. Related Work

We analyze the differences and similarities between our study and a wide spectrum of related studies along three categories: (1) solutions for using sentiment analysis for recommendation, which (unlike our proposal) mainly focus on review text and real-valued feedback; (2) methods for top-N recommendation, especially those that deal with unary feedback, showing that they neglect user-generated texts; and (3) methods that leverage user comments to perform prediction tasks, including content recommendation. Synthetic presentations of content-based (CB) and collaborative filtering (CF) methods for recommendation, including techniques for their evaluation, have been provided by Sarwar et al. (2001); Ricci et al. (2010); Koren and Bell (2011); Lops et al. (2011).

2.1. Sentiment Analysis for Recommendation

Since their appearance, *sentiment analysis techniques* have attracted the interest of the research community because they help capturing high-level meaning in language and offer a wide variety of applications. Sentiment analysis typically aims to detect the polarity of a given text, and is commonly formulated as a classification problem (for discrete labels such as ‘positive’ and ‘negative’) or a regression one (for real-valued labels) (Pang and Lee, 2008). Rating inference is also defined as a classification problem, with respect to rating scales (Pang and Lee, 2005). Pang and Lee (2008) survey the large range of features that have been engineered for rule-based sentiment analysis methods as the one used in this paper (Hatzivassiloglou and Wiebe, 2000; Hu and Liu, 2004; Wilson et al., 2005) and for corpus-based ones (Pang et al., 2002; Thomas et al., 2006) . Machine learning techniques for sentiment classification have been introduced quite early (e.g. Pang et al., 2002), including unsupervised techniques based on the notion of semantic orientation of phrases (e.g. Turney, 2002). More recent studies have focused on feature learning (Maas et al., 2011; Socher et al., 2011; Tang et al., 2014), including the use of deep neural networks (Socher et al., 2013; Mikolov et al., 2013; Tang, 2015). A related family of studies focused on subjectivity detection, i.e. whether a text expresses opinions or not (Wiebe et al., 2004), but they less relevant to recommender systems.

Several studies have performed sentiment analysis of textual reviews of items to improve recommendation. Most of them focus on *learning to infer numerical ratings* from a set of already labeled textual reviews (arguing in favor or against particular items), unlike the free-form unlabeled comments that are exploited in our study. Leung et al. (2006, 2011) proposed a probabilistic rating inference framework which mines user preferences from text reviews and then maps them onto numerical rating scales. Similarly, Kawamae (2011) proposed a hierarchical topic modeling approach for integrating sentiment analysis with CF by modeling each author's preference and writing attitude as latent variables. Such frameworks provide a convenient way of combining feedback from preferences and from reviews, but their main drawback is that they are only applicable to review websites, and cannot be easily transferred to other situations. Singh et al. (2011) performed two-stage filtering with CF and sentiment classification of user reviews, keeping however the modeling of text and ratings separate, unlike our model which optimizes their combination.

Moshfeghi et al. (2011) addressed the *cold-start problem* by considering item-related emotions and semantic information extracted from movie plots as well as text reviews, using LDA and gradient boosted trees. The benefits of this method were mostly observed when the amount of user ratings was very small or zero. In cases, when user ratings are unavailable, Zhang et al. (2010, 2013) proposed to perform online video recommendation by using virtual ratings extracted from sentiment analysis of text reviews, instead of actual user ratings. Similarly, when ratings are absent, Karampiperis et al. (2014) examined the benefits of using sentiment analysis on user review comments followed by explicit numerical ratings to improve recommendation in educational repositories. In contrast, we will show that our model is beneficial on various proportions of free-form user comments.

Several recent methods have focused on situations were both *review text and ratings are available*. Pero and Horváth (2013) proposed a simple, scalable and effective rating prediction framework based on matrix factorization which utilizes both user ratings and opinions inferred from their reviews. García-Cumbreras et al. (2013) categorized users according to the average polarity of their comments, in the context of movie reviews. These categories were then used as features to improve CF models, thus following a less personalized and item-oriented recommendation strategy than ours. McAuley and Leskovec (2013) and Ling et al. (2014) combined latent rating dimensions (such as those of latent-factor recommender systems) with latent topics of reviews learned by topic models. Similarly, Diao et al. (2014) proposed a probabilistic model based on collaborative filtering and topic modeling, which jointly captures the interest distribution of users and the content distribution for movies. The advantage of such modelings, apart from its improved accuracy, is that the learned latent dimensions can be more easily interpreted than pure latent-factor models. Zhang et al. (2015) first extracted hidden dimensions from reviews with topic modeling, and then applied a traditional CF model to capture correlations between hidden dimensions in reviews and ratings. All these studies used explicitly labeled reviews for evaluation, therefore it is unclear whether their improvements still hold when using free-form comments and a more challenging recommendation setting such as with unary ratings, as in our study.

A promising line of sentiment analysis research, for structured reviews, is to recognize the *aspects of items and their ratings*. For instance, Ganu et al. (2009) proposed a regression approach which considers various aspects of a restaurant to improve recommendations using a k-NN method. Similarly, Jakob et al. (2009) proposed three approaches to extract movie aspects for improving movie recommendations in a CF model. Faridani (2011) generalized the concept of sentiment analysis of reviews to multiple dimensions (such as service or price) using Canonical Correlation Analysis, with applications to product search and recommendation. Levi et al. (2012) addressed the cold-start problem by mining aspects and their sentiment, and profiling users according to their intent and nationality using context groups extracted from reviews. Personalization of quality rankings for products using aspect information from reviews was investigated by Musat et al. (2013), who also proposed new evaluation methods to rate explanations and to predict pairwise user preferences. Zhang et al. (2014) extracted attribute-value pairs from product reviews and integrated them into a latent matrix factorization model, resulting in an explicit factor model which is able to provide explanations of its recommendations in terms of aspects preferred by users. To capture the importance given by different users to different items, Nie et al. (2014) used tensor factorization to automatically infer the weights of different aspects in forming the overall rating. D'Addio and Manzato (2015) proposed a vector-based representation of items computed from user reviews, which considers the sentiment of those reviews towards specific aspects, within a neighborhood-based CF model. Wu and Ester (2015) proposed a unified probabilistic model which combines the advantages of CF and aspect-based opinion mining to learn personalized sentiment polarities on different aspects of items. He et al. (2015) proposed to cast the recommendation task as vertex ranking and devised a generic personalized algorithm for ranking in tripartite graphs named TriRank. To create such a graph, the authors extracted aspects from textual reviews to enrich the user-item binary relations to a user-item-aspect ternary relations. To be applicable, these methods require even more demanding explicit feedback information, namely repositories which include multiple-aspect reviews and aspect-specific ratings, which are currently available only for limited range of item types.

Most of the above studies aim to predict ratings (on numeric scales) from reviews, generally by training the predictor on similar reviews that are accompanied by ground-truth ratings given by their authors. Unlike such studies, we analyze here the sentiment of user comments which are never accompanied by ratings. Free-form comments differ from text reviews as they are not necessarily purposed to refer to the items which are considered for recommendation, due to their unconstrained and spontaneous nature. Rather, they reflect the written interactions among the users of an online community. In addition, the existing approaches which make use of reviews composed of ratings and text have a high adaptation cost to a new domain if no ground-truth ratings are initially available. Another novelty of our study is that, unlike previous work on this topic, it considers explicit user feedback in the form of unary ratings, which is a common form of feedback in social media networks such as YouTube, Facebook, Flickr, Vimeo, Twitter and others.

2.2. *Top-N Recommendation and One-class Collaborative Filtering*

In contrast to mainstream recommender systems that aim to predict numerical ratings for each item, *top-N recommender systems* are used to recommend *N* items that are most likely to be of interest to users (Cremonesi et al., 2010). Such systems operate on both discrete and real-valued feedback, although they are mostly applied to unary feedback obtained from user behavior data, explicit or implicit, because in this case numerical rating prediction is difficult (Schwab et al., 2000). The CF methods for top-N recommendation can be broadly divided in two categories: neighborhood-based vs. model-based (Deshpande and Karypis, 2004). These methods typically originate from traditional recommendation methods (Koren and Bell, 2011; Lops et al., 2011) tailored to the top-N task.

Hu et al. (2008) adapted CF to datasets with implicit feedback by considering positive and negative preferences with varying confidence levels, which they used to provide explanations. Ning and Karypis (2011) proposed sparse linear methods (SLIM) to generate top-N recommendations by solving a regularized optimization problem. Other studies have formulated top-N recommendation as a ranking problem. Rendle et al. (2009) adopted a Bayesian perspective and proposed an optimization criterion, named Bayesian Personalized Ranking (PBR). Shi et al. (2012) proposed an approach called collaborative less-is-more filtering (CLiMF) which maximizes directly the Mean Reciprocal Rank (MRR) for top-N recommendation with binary relevance data. The authors then generalized CLiMF for multiple levels of relevance (Shi et al., 2013). Kabbur et al. (2013) reduced the sparsity of the datasets for top-N recommendation by learning an item-item similarity matrix using structural equation modeling (SEM). Aiolli (2014) optimized the Area Under the Curve (AUC) within a max margin framework for CF top-N recommendation. Elbadrawy and Karypis (2015) proposed a sparse high-dimensional factor model, which learns user-specific feature-based item similarity models able to exploit global and user-specific preferences.

While Pan et al. (2008) formulated the one-class CF problem as dealing only with positive instances of user feedback, several schemes were proposed *to weigh the negative class* in a discriminative fashion, formulated with a matrix factorization framework. The proposed weighting mechanisms performed better than the baseline assumptions that treat all the missing instances as negative or unknown (see Section 5.3 below). Sindhwani et al. (2009) suggested to treat zero-valued pairs as optimization variables computed from the training data. Thus, instead of making a uniform assumption about the negative class, the distribution of the negative class was learned. Li et al. (2010b, 2014) incorporated rich user information to improve one-class CF, such as search history, purchasing and browsing activities. Paquet and Koenigstein (2013) addressed the lack of a negative class using a Bayesian generative model for the latent signal with an unobserved random graph which connects users with items they might have considered. Yuan et al. (2013) considered the rich user and item content information for better weighting the unknown data.

Here, we solve the top-N recommendation problem through a *ranking function based on an adaptive sentiment-aware neighborhood model*, which uses both user comments and unary ratings. The proposed model is a significant extension of item-based CF models such as those proposed by Cremonesi et al. (2010) and Koren and Bell (2011). To complement unknown ratings, we propose to infer user ratings from free-form user comments, which occur frequently in online repositories and social networks. Instead of hypothesizing the values of missing instances, we attempt to infer some of them from available textual data, and demonstrate the value of such information in combination with three different assumptions about missing instances.

2.3. *Leveraging User Comments for Predictive Tasks*

User comments in online communities have captured the attention of researchers due to their high availability, and the personal, opinionated and rich information they contain. Many predictive tasks and applications have benefited from the analysis of textual user comments, though most of them are not related to recommender systems, since they aim to predict the *popularity or mood of news articles, blogs, or user profiles*. The few studies of comments for recommender systems do not target personalized recommendation in the one-class setting – despite the importance of this setting, emphasized above. Pavlou and Dimoka (2006) utilized content analysis to quantify comments from sellers on a popular online auction website and to match them with purchasing data from buyers that had transacted with them: the addition of text comments to numerical ratings helped to explain a greater part of the variance in seller's benevolence and credibility compared to ratings only. Li et al. (2007) proposed to include comments on blog posts for clustering blogs and found that they increased discriminative effects compared to using only the blogs' contents. Tsagkias et al. (2009) hypothesized that the number of user comments on a news article may be indicative of its importance and attempted to predict the volume of comments on an article prior to its publication as a binary classification task (high or low volume).

More recent studies of user comments have *refined the above trends on news and profiling*. They include: comparing several text analysis strategies to automatically gather profile data from user comments on news articles (Messenger and Whittle, 2011); predicting the popularity of online articles during a short observation period using a simple linear prediction model (Tatar et al., 2011); predicting the political orientation of news stories (Park et al., 2011); exploiting the mood of tweets to predict stock market time series (Bollen et al., 2011); improving social tag recommendation by connecting user comments with tags (Yin et al., 2013); analyzing the influence of Facebook user comments on relationship status updates (Ballantine et al., 2015); and detecting hate speech in online user comments by learning distributed low-dimensional representations of comments and using them as features for classification (Djuric et al., 2015).

The studies which have a similar goal to the present one (namely leveraging user comments for content recommendation) exhibit a number of significant differences: they focus on the *generic (i.e. non-personalized) recommendation of tags, comments, or news for commenting*, emphasizing their *semantic content over their polarity*. As also confirmed by the literature review of Sun et al. (2015), whose proposal is discussed below, studies of personalized recommendation from one-class feedback plus comments remain scarce.

Wang et al. (2010a) combined non-personalized news recommendation with user interaction by using a refinement process of reader comments in accordance with an evolving topic. Wang et al. (2010b) and Li et al. (2010a) used structural, semantic and authority information encapsulated in user comments to improve recommendation. Agarwal et al. (2011) attempted to rank the comments associated with a news article according to personalized user preferences, i.e. liking or disliking a comment. Shmueli et al. (2012) presented a model that predicts news stories that are likely to be commented by a given user. Kim et al. (2012) proposed a query expansion method that utilizes user comments in order to consider user's different preferences in finding movies. San Pedro et al. (2012) analyzed the user comments to detect opinions about the aesthetic quality of images for image search. Jain and Galbrun (2013) proposed to organize user comments in semantic topics which enable users to discover significant topics of discussions in comments and allow to explicitly capture the immediate interests of users on news articles. In one of the few studies on multimedia content, Siersdorfer et al. (2010) analyzed dependencies between comments, views, comment ratings, topic categories and comment sentiment influence in a large dataset from a large video repository, to predict comment ratings, i.e. number of feedback votes on comments.

The only two studies of which we are aware focusing on *sentiment analysis for one-class CF over multimedia content* are our own initial study (Pappas and Popescu-Belis, 2013b) and the study of Sun et al. (2015), to which we compare our scores in Table 13 below. Both studies use the metadata set that we created from the TED talks (Pappas and Popescu-Belis, 2013a, 2015) and made available online (see Section 3 below), though not the Flickr and Vimeo datasets additionally used here. In our own initial work (Pappas and Popescu-Belis, 2013b), we proposed a fixed mapping to combine a rule-based sentiment classifier with CF neighborhood models. Sun et al. (2015) used ensemble learning to improve the sentiment classifier for this task, and showed that a matrix factorization framework, which reaches higher recommendation scores than neighborhood models, can be combined with them to increase performance. In the present paper, we show that the performance of sentiment-aware neighborhood models with a fixed mapping can be further improved by learning to adapt the sentiment scores to the user preferences. The improvement holds against more sophisticated models, regardless of the assumption on the negative class, and is stable over datasets that are larger and noisier than TED.

	All Items and Users						Active Users		
Datasets	**Items**	**Users**	**Favorites**	**Comments**	***cpi***	***wpc***	**Users**	**Favorites**	**Comments**
TED	1,203	74,760	129,633	209,566	174	95.45	4,961	113,241	35,229
Vimeo	2,000	255,144	722,474	278,563	139	18.75	7,071	155,207	32,639
Flickr	1,994	246,272	477,184	690,798	346	22.31	9,963	161,398	304,564

Table 1. Statistics of TED, Vimeo and Flickr datasets: number of items, users, favorites, comments, average comments per item (*cpi*) and average words per comment (*wpc*). We will use only the *active users* in our experiments, who are defined as those who have indicated more than five favorites and have made at least one comment.

3. Multimedia Collections

Three real-world multimedia datasets that contain both user comments and indications of favorite items are used in our experiments, namely TED, Vimeo and Flickr (see Table 1). These are popular online repositories of talks, videos and images respectively which contain explicit user feedback of action or inaction, i.e. users mark certain items as favorites, while leaving all the others unmarked. Therefore, the problem of recommendation over these datasets is a *one-class* CF problem. The datasets have different user rating behaviors, comment densities and correlations between the two user-action variables (favorites and comments), as we show below.

3.1. Datasets

TED (`www.ted.com`) is an online repository of public talks accompanied by user-contributed material, such as lists of favorites or comments grouped in threads, made available under a Creative Commons license. The talks are given by prominent speakers and pertain to a variety of topics, such as science, art, entertainment, or society. In September 2012, we crawled the TED dataset, gathered its metadata, and made it publicly available by permission from TED owners, under the same Creative Commons license.[2] The TED dataset contains 1,203 talks, 74,760 user profiles, 129,633 indications of favorite talks and 209,566 comments on talks. A detailed description of it can be found in our previous work (Pappas and Popescu-Belis, 2013a, 2015). TED users tend to make the longest and most elaborate comments among the three datasets used here, since they contain on average about 5 sentences and 95 words, compared to about 2 sentences and 20 words for the Vimeo and Flickr datasets.

Vimeo (`www.vimeo.com`) is an online video sharing repository that allows users to upload, share and view videos. The metadata are accessible in machine-readable format through an API provided by Vimeo. Using this API, in January 2013, we collected 2,000 videos, 255,144 user profiles, 722,474 indications of favorites ("likes") and 278,563 comments from the *nature*, *science*, *art*, *politics* and *music* categories. Flickr is another large online image and video sharing repository (`www.flickr.com`), which also provides an API giving access to their data. We collected a similar number of items as for Vimeo, namely 1,994 images, 246,272 user profiles, 477,184 indications of favorites ("likes") and 690,798 comments from the *macro* category. As the owners of the Vimeo and Flickr repositories forbid the redistribution of the data obtained through their APIs, we cannot provide these sets along with our distribution of TED metadata.

3.2. Analysis of the Datasets

To evaluate the utility of comments for recommendation, we consider from now on the *active users* of TED, Vimeo and Flickr, defined as those who indicated more than five favorites and made at least one comment. Statistics about them are given in the last three columns of Table 1. Figure 1 displays the distributions of favorites and comments per active user, ordered by decreasing number of favorites. The following differences are observed between datasets. Firstly, in Flickr, users are more likely to make a comment than to mark an item as favorite, while in Vimeo the reverse is true in Figure 1, red spikes stay mostly below the blue line in (b) and mostly above it in (c). In the TED dataset, the two behaviors can be observed: large and small spikes alternate in Fig. 1 (a). Secondly, the correlation between the numbers of favorites and comments per user, measured by Pearson's r coefficient, is weak in TED (0.11), moderate in Vimeo (0.33) and strong in Flickr (0.61). Finally, comment density, i.e. the ratio of comments over favorites, is

[2] `http://www.idiap.ch/dataset/ted/`

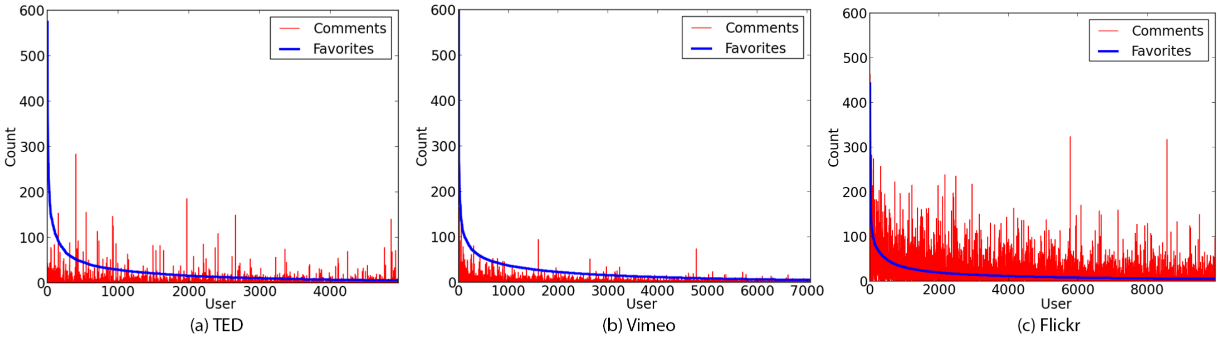


Figure 1. Numbers of comments (spikes) and favorites (curve) per active user, ordered by decreasing number of favorites.

0.24 for TED, 0.18 for Vimeo and 1.88 for Flickr. Flickr is thus the densest dataset in terms of comments, while TED and Vimeo are much sparser. The variety of these three real-world multimedia datasets will thus allow us to test our proposal over different user behavior patterns and comment densities.

4. Sentiment Analysis

The first stage of our proposal for using comments in a one-class CF task is the sentiment analysis of user comments. Given the lack of ground-truth labels to use for training, we use a dictionary-based approach. Specifically, we extend the rule-based (RB) sentiment classifier designed by Wilson et al. (2005), as explained in Pappas et al. (2013), making this implementation freely available.[3]

The RB algorithm first determines whether an expression is neutral or polar and then hypothesizes the polarity of the polar expressions by using a set of contextual rules accounting for phenomena such as negations, modifiers, intensifiers, and polarity shifters. The algorithm relies on the MPQA polarity lexicon[4] for identifying subjective and polar words in a given text. It proceeds through the following steps: (i) text pre-processing, (ii) feature extraction, (iii) polar expression marking, (iv) negation modeling, (iv) intensifier marking, (v) heuristic weighting, and (vi) calculation of the total polarity score. This score is not bounded since its range depends on the size and context of the input texts.

Since we build our model on top of the sentiment classification output, other rule-based classifiers could be used as well, such as the one from the Pattern[5] or the TextBlob[6] libraries, as well as corpus-based classifiers trained on domain data such as the one from the LingPipe[7] or the Stanford[8] toolkits. However, for the corpus-based classifiers, the text labels of in-domain free-form comments needed for training are in general costly to acquire.

In the rest of this section, we describe how the sentence-level and comment-level polarities are obtained from the RB classifier, we report the results of the sentiment labeling performed by humans, we evaluate the RB classifier, and lastly, we provide sentiment statistics over the three datasets.

4.1. Sentence-level and Comment-level Polarity Estimation

Given a set of sentences from a user comment c, the RB classifier hypothesizes the polarity of each sentence $s \in c$ as a signed numerical value, noted $pol_{RB}(s)$ (non-normalized). If needed, the sentiment label of the sentence, *positive* or *negative*, is determined from the sign of $pol_{RB}(s)$, with *neutral* if $pol_{RB}(s) = 0$. In Table 2, we show examples of sentences with the three possible labels and their polarity values; here, it appears that the labels were correctly determined by the RB classifier. Having assigned polarity values to each sentence, the total polarity value and the

[3] http://github.com/nik0spapp/unsupervised_sentiment/
[4] http://mpqa.cs.pitt.edu/
[5] http://www.clips.ua.ac.be/pattern/
[6] https://textblob.readthedocs.org/en/dev/quickstart.html#sentiment-analysis
[7] http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html
[8] http://nlp.stanford.edu/sentiment/code.html

Label	**Value**		**Sentence**
positive	+6	+0.50	She is very true in saying that mistakes are part of learning.
negative	−1	−0.05	The problem with the statement: ’the institutions determine work ethics’ is the point of correlation does not equal causation.
neutral	0	0	For years scientists have puzzled over how the sea surface temperature around Antarctica has risen, but sea ice there has been increasing at the same time.

Table 2. Examples of sentences with the three possible sentence-level labels from the RB sentiment classifier and their respective polarity values, first non-normalized and then normalized by the length of the sentence.

label for each user comment are computed. Among the various possibilities for computing the total polarity value of a comment, we compute the sum of the polarities of each sentence normalized by the length of the sentence in terms of words, i.e. $pol_{RB}(c) = \sum_{s \in c}(pol_{RB}(s)/|s|)$.

4.2. Ground-truth Labeling

To evaluate the sentiment analysis component we focused on binary classification of polarized comments, namely positive or negative, and performed two studies. In Study 1, we performed ground-truth labeling of a subset of the TED comments with three labels: positive, negative or undecided. Six human judges, who were recruited among our English-speaking colleagues, annotated 320 sentences and 160 comments that had been randomly selected from the TED data, with an overlap of about 20% in both cases to assess agreement. Agreement over the shared subset (61 sentences and 29 comments) was found to be $\kappa = 0.83$ for sentences and $\kappa = 0.65$ for comments using Fleiss' kappa. As agreement was substantial, we subsequently used the entire set as ground truth. After excluding the undecided cases, we obtained 260 labels for sentences and 135 for comments.

To obtain additional ground-truth data, we performed Study 2, a larger-scale study using a crowdsourcing platform.[9] We submitted 1,200 randomly selected comments from TED for annotation on a sentiment scale from 1 to 5, by at least 3 and at most 7 annotators per comment. The agreement between the annotators was found to be 0.74 on a 0–1 scale.[10] We obtained 623 positive, 314 neutral and 263 negative labels and we created a balanced set for classification containing 263 positive and 263 negative comments, as balanced sets are often used in the literature (Pang et al., 2002; Turney, 2002; Pang and Lee, 2008). Below, we will measure the performance of the RB classifier on the balanced set as well as on the full set of positive and negative labels.

4.3. Evaluation of the RB Classifier

The binary classification results of our RB classifier and those of a random baseline (Rand) are shown in Table 3. When measured by the same kappa score as inter-annotator agreement, our system reaches $\kappa = 0.53$ on sentences (dataset of Study 1) and $\kappa = 0.43$ and 0.48 on comments (dataset of Study 1 and balanced subset of Study 2). As expected, Rand has close to zero κ values. The agreement between the RB classifier and the annotators is thus consistently moderate in both studies. Moreover, the RB classifier reaches a classification accuracy score (F-measure) of 72.5% on comments and 74.9% on sentences. When measured over the full set of labels obtained in Study 2, i.e. the unbalanced sentiment distribution of the comments, the RB classifier reaches a higher score: 78.2% F-measure, 89.9% precision, 69.2% recall and $\kappa = 0.43$.

The quality of RB-assigned labels is comparable to previous works on binary sentiment classification (Pang et al., 2002; Turney, 2002), in which classification performance reached about 75% F-measure. We will show below that this level of performance is sufficient to improve significantly the one-class CF task. In the study by Sun et al. (2015) on the TED dataset, supervised methods for sentiment classification of 600 TED comments (balanced set), reached 87% accuracy using an ensemble learning approach, while the RB classifier reaches 75% accuracy on a comparable setup (balanced set of Study 2). The improvement obtained by Sun et al. (2015) is conditioned on the availability of ground-truth annotation for learning, which limits the portability of the ensemble method, unlike dictionary-based

[9] `http://crowdflower.com/`

[10] Crowdflower computes a trust-aware inter-annotator agreement score by testing the annotators' trust randomly during the annotation process based on majority agreement over a subset of the comments.

	Sentences				**Comments**							
	Study 1 (260 labels)				**Study 1** (135 labels)				**Study 2** (526 labels)			
Methods	**P**	**R**	**F**	***k***	**P**	**R**	**F**	***k***	**P**	**R**	**F**	***k***
RB	73.4	76.4	74.9	0.53	75.7	69.7	72.6	0.43	78.6	67.3	72.5	0.48
Rand	47.3	49.9	48.5	-0.01	56.3	50.1	52.9	-0.02	50.0	50.0	50.0	0.00

Table 3. Performance of the RB and Rand sentiment classifiers measured with percentage precision, recall, F-measure and Fleiss' kappa.

methods such as the above one with the MPQA lexicon. Moreover, supervised learning has a high risk of overfitting, which cannot be excluded given the small size of the data set used by Sun et al. (2015) (600 comments out of 209,566).

4.4. Sentiment Statistics

We labeled all the TED, Vimeo and Flickr comments using the RB classifier with the positive, negative or neutral labels. Statistics about the results of automatic labeling are given in Table 4. Based on this classification, the TED dataset appears to have more positive comments than negative ones (62% vs. 27%) and a small percentage of neutral comments (10%). The Vimeo and Flickr comments have an even more skewed distribution of positive vs. negative comments: 70.1% positive vs. 7.8% negative, and 76.8% positive vs. 3.3% negative, respectively.

	Count			**Percentage**			**Average per Item**		
Datasets	***pos***	***neg***	***neu***	***pos***	***neg***	***neu***	***pos***	***neg***	***neu***
TED	130,260	58,171	21,121	62.1%	27.7%	10.0%	108.2	48.3	17.5
Vimeo	195,397	21,726	61,375	70.1%	7.8%	22.0%	97.6	10.8	30.6
Flickr	530,787	22,924	137,087	76.8%	3.3%	19.8%	266.1	11.4	68.7

Table 4. Statistics about the sentiment of user comments, as estimated by the RB classifier on the three multimedia datasets.

5. One-class Collaborative Filtering Models

Several applications such as the recommendation of news, bookmarks, images, or videos can be viewed as a one-class CF problem, with training data consisting of binary values expressing the user action or inaction, e.g. bookmarking or marking as liked (Pan et al., 2008). Inaction can mean that an item was either not seen or that it was seen but not liked. This ambiguity of the negative class makes the problem particularly difficult to solve. In this section, we propose a method for leveraging comments for one-class CF by mapping their polarities to a format that is usable with neighborhood models.

The one-class CF problem is formalized as follows. Let U be the set of users of size $|U| = N_U$ and I the set of items of size $|I| = N_I$. The matrix of user-item ratings is $R = \{r_{ui}\}_{N_I}^{N_U}$ of size $N_U \times N_I$, with $r_{ui} = 1$ indicating a positive rating of item i by user u (e.g. i is a favorite of u) and $r_{ui} = ?$ an absent rating (i was not seen or not liked by u). If one assumes that some of the negative examples have been seen but not liked (or not marked as favorites), then the corresponding ratings become $r_{ui} = 0$. Our goal is to predict the preference of the users in the future, therefore, to evaluate our system, we hide a certain proportion of '1' values per user and measure how well we predict them, as often performed in previous studies.

5.1. Neighborhood Models

Neighborhood or Nearest Neighbor (NN) models are often used for CF and have been proven to be quite effective despite their simplicity (Cremonesi et al., 2010). There are several versions of such models, including similarity-based interpolation, jointly-derived interpolation and generalized neighborhood models with parameters computed from the data (Koren and Bell, 2011). We will adopt here the first approach, based on similarities, and focus on item-based neighborhood models as defined in Eq. 1 below, with a prediction function $\hat{r}_{ui}$ that estimates the rating of a user u for an unseen item i.

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in D^k(u;i)} (r_{uj} - b_{uj}) d_{ij}}{\sum_{j \in D^k(u;i)} d_{ij}} \tag{1}$$

The prediction $\hat{r}_{ui}$, following Cremonesi et al. (2010), is the sum of the bias estimate b_{ui} of a user u towards an item i (defined in Eq. 2) and of a similarity score computed using the k most similar items to i that the user u has already rated, i.e. the neighborhood of item i, denoted by $D^k(u;i)$. The similarity score relies on a similarity metric such as cosine distance, as specified at the end of this subsection. The value of k limits the number of items to be taken into account, for efficiency purposes. The coefficient d_{ij} expresses the similarity between items i and j, computed as in Eq. 3 below. The denominator in Eq. 1 ensures that the predicted values fall in the same range as the known ones, although it is optional for top-N recommendation because we are interested in the ranking of top items rather than their rating.

The bias estimate b_{ui} is defined in Eq. 2 as the sum of the average rating μ, the bias estimate b_u of the user u, and the bias estimate b_i of the item i. The bias b_u is computed as the difference between the average rating of a user u, noted $\bar{r_u}$, and the mean μ. Similarly, the bias b_i is the difference between the average rating of an item i, noted $\bar{r_i}$, and the mean μ. Given that the ratings are not real-valued in one-class CF, the biases b_u and b_i are normalized by the total number of ratings of the most rated item, noted r_{max}.

$$\begin{gathered} b_{ui} = \mu + b_u + b_i, \text{ with: } b_u = \bar{r_u} - \mu \text{ and } b_i = \bar{r_i} - \mu, \text{ where} \\ \bar{r_u} = \frac{\sum_{i \in I} r_{ui}}{r_{max}}, \ \bar{r_i} = \frac{\sum_{u \in U} r_{ui}}{r_{max}}, \ \mu = \frac{\sum_{i \in I} \bar{r_i}}{N_I} \end{gathered} \tag{2}$$

The coefficient d_{ij} is defined in Eq. 3 as the similarity s_{ij} between items i and j multiplied by a coefficient involving the number of common raters n_{ij} and a shrinking factor λ, following Cremonesi et al. (2010). The choice of the optimal value of λ and the optimal size of the neighborhood k used in $D^k(u;i)$ will be determined by cross-fold validation in Section 7.

$$d_{ij} = s_{ij} \frac{n_{ij}}{n_{ij} + \lambda} \tag{3}$$

The similarity s_{ij} between items i and j can be defined, as in Eq. 4, either as the cosine similarity, denoted by COS, or as Pearson's correlation, denoted by PC, following Cremonesi et al. (2010). The vectors for items i and j of size $|U|$ are obtained for each item after creating the co-rating matrix of size $N \times N$ that contains the number of times that two items have been co-rated by pairs of users. Given the vectors of two items $\vec{v}_i$ and $\vec{v}_j$, their expected values μ'_i and μ'_j, and their standard deviations σ_i and σ_j, the similarities with COS or PC are computed as follows:

$$s_{ij} = COS(\vec{v}_i, \vec{v}_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{||\vec{v}_i||_2 \times ||\vec{v}_j||_2} \text{ or } s_{ij} = PC(\vec{v}_i, \vec{v}_j) = \frac{E[(\vec{v}_i - \mu'_i)(\vec{v}_j - \mu'_j)]}{\sigma_i \sigma_j} \tag{4}$$

5.2. Sentiment-Aware Neighborhood Models

We extend the neighborhood model defined above by proposing a sentiment-aware nearest neighbor model (SANN) with the main purpose of using, in addition to the explicit ratings, the preferences of the users that are implicitly expressed in user-generated texts such as comments. The polarities of the comments are computed by the RB sentiment classifier, and then combined with explicit ratings using a mapping function. Several proposals for such a function are made in this section.

The model in Eq. 1 is modified as follows.[11] Firstly, we use a new neighborhood $D'^k(u;i)$ to account for the additional data, and secondly, we define a new rating function r'_{uj} that combines the numerical output of the sentiment classifier and the explicit rating values. Moreover, the additional data from commented items is considered for the creation of the co-rating matrix used for the similarity s_{ij} in Eq. 4. Thus, we modify the Eq. 1 of the traditional neighborhood model as follows:

[11] Originally proposed in our 2013 paper (Pappas and Popescu-Belis, 2013b), the model has also been adopted by others (Sun et al., 2015).

Type			Mapping function	Notation
Random	Discrete	-	$m_{uj} = sign_{rand}$	randSANN
Fixed	Discrete	-	$m_{uj} = sign_{RB}(C_{uj})$	sigSANN
	Continuous	-	$m_{uj} = 1 + z_{uj} \cdot pol_{RB}(C_{uj})$	polSANN
Learned	Discrete	Global	$m_{uj} = \begin{cases} \theta, & \text{if } sign_{RB}(C_{uj}) = 1 \\ \upsilon, & \text{if } sign_{RB}(C_{uj}) = 0 \\ \alpha, & \text{if } sign_{RB}(C_{uj}) = -1 \end{cases}$	ltmSANN(global)
		Per user	$m_{uj} = \begin{cases} \theta_u, & \text{if } sign_{RB}(C_{uj}) = 1 \\ \upsilon_u, & \text{if } sign_{RB}(C_{uj}) = 0 \\ \alpha_u, & \text{if } sign_{RB}(C_{uj}) = -1 \end{cases}$	ltmSANN(user)
	Continuous	Global	$m_{uj} = \eta + \zeta \cdot pol_{RB}(C_{uj})$	ltmpolSANN(global)
		Per user	$m_{uj} = \eta + \zeta_u \cdot pol_{RB}(C_{uj})$	ltmpolSANN(user)

Table 5. Random, fixed and learned mapping functions of discrete and continuous sentiment scores to ratings for the SANN models.

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in D'^k(u;i)} d_{ij}(r'_{uj} - b_{uj}) \tag{5}$$

In this new definition, $\hat{r}_{ui}$ no longer represents a proper prediction of the rating, but will serve only as a ranking function associating user u with item i.[12] $D'^k(u;i)$ is the neighborhood of the k most similar items that the user has already rated *or commented* and r'_{uj} is the rating function for item j that accounts both for explicit ratings and for those inferred from comments. We propose the following model for r'_{uj}: if the explicit unary feedback of user u for item j is available (favorite mark, $r_{uj} = 1$) then $r'_{uj} = 1$, but when this unary feedback is not available ($r_{uj} \neq 1$, i.e. it is zero or unknown), r'_{uj} takes the value of a mapping function m_{uj}. This is a function of the polarity scores of user's u comment(s) to item j, C_{uj}, for which several alternatives are proposed and studied below. Thus, r'_{uj} can be defined as the following piecewise function:

$$r'_{uj} = \begin{cases} 1, & \text{if } r_{uj} = 1 \\ m_{uj}, & \text{if } r_{uj} \neq 1 \end{cases} \tag{6}$$

This function augments the standard neighborhood model which makes use of explicit ratings when available (first part) with a sentiment mapping function based on user comments when explicit ratings are absent (second part). It should be noted that, when explicit ratings are available, the user's comments on the item are not considered. This is because the explicit rating is the ground truth which represents the actual preference of the user, while the mapping function only makes an assumption on how the sentiment of comments from a user might correspond to her preference. Although, for instance, positive comments could consistently accompany a favorite item and vice-versa, empirical observations (Section 5.2.3) show that this is not always the case. For these reasons, in our model, explicit ratings always have precedence over those inferred from comments. In the following subsections, we define three types of mapping functions of sentiment scores to ratings, which are summarized in Table 5.

5.2.1. Random Mapping

As a baseline, we compare a random sentiment classifier with the RB sentiment classifier, which will be used in the fixed and learned mapping functions. The random mapping, noted as randSANN, simply assigns a random class value $sign_{rand}$ (either 1, 0 or -1) to the sentiment of a user comment. Hence, this baseline does not extract any actual preference information from text.

[12] The goal of such function is to rank items according to user preference, for example, in the one-class case, higher ranked items are more likely to belong to the positive class than the lower ranked ones.

	Favorites accompanied by comment(s) from the same user			
	Total	**Positive**	**Neutral**	**Negative**
TED	7,053 (6.2% of the total)	5,385 (76.5%)	548 (7.7%)	1,120 (15.8%)
Vimeo	11,883 (7.7% of the total)	9,246 (77.8%)	1,898 (16.0%)	739 (6.2%)
Flickr	84,119 (52.1% of the total)	69,910 (83.1%)	12,208 (14.5%)	2,001 (2.4%)

Table 6. Total number of ratings (favorites) from active users which are accompanied by at least one comment from the same user, and the proportion of positive, neutral and negative comments among them, as labeled by the RB classifier.

5.2.2. Fixed Mappings

We first propose two different mapping functions that rely on the output of the RB classifier (polarity score), one based on the discretized output, noted as "fixed → discrete", and the other using the actual real-valued output, noted as "fixed → continuous" (see Table 5). Let C_{uj} be the set of all comments made by a user u on an item j. The first function, denoted by sigSANN (for 'sign'), assigns a rating value for a user-item pair according to the sign of the average polarity score of the comments: $sign_{RB}(C_{uj}) = sign(mean(\{pol_{RB}(c) \mid c \in C_{uj}\}))$, where $pol_{RB}(c)$ is the polarity of a comment defined in Section 4.1.

The second mapping function, polSANN, uses the real-valued output of the RB classifier with a normalization factor and an offset. The polarity score of a given user u for a particular item j is $pol_{RB}(C_{uj}) = mean(\{pol_{RB}(c) \mid c \in C_{uj}\})$ and the normalization factor is $z_{uj} = 1/(1 + |C_u| \cdot |\{c \text{ s.t. } c \in C_u \wedge sign_{RB}(c) = sign_{RB}(C_{uj})\}|)$. This normalization penalizes the impact of the polarity score in proportion to the total number of a user's comments times the number of the user's comments of the same class as the predicted one. In other words, without normalization, polSANN estimates that users who always comment positively are biased towards positive feedback, and similarly for negative comments. The normalization z_{uj} aims to reduce these effects on the rating prediction $\hat{r}_{ui}$.

5.2.3. Learned Mappings

The mapping functions described above combine sentiment scores with ratings based on the intuition that positive scores imply positive preferences and negative scores imply negative preferences. However, this intuition may not be accurate in all cases, for example, a user could write positive comments about non-favorite items. To support this claim, we have examined the number of times ratings (i.e. marking as favorites) appear with either positive, negative or neutral comments, as labeled by the RB classifier. The results, listed in Table 6, show that in a majority of cases when a favorite is accompanied by a comment the latter is a positive one: 76.5% of the times on TED, 77.8% on Vimeo and 83.1% on Flickr. However, ratings can also be followed by neutral or negative comments, which motivate the need to employ learning methods to handle such cases by learning either global or individual behavior patterns from the data, as defined hereafter.

1. **Mapping discrete scores globally**: We introduce three parameters θ, υ and α respectively for positive, neutral and negative comments, which define the mapping according to the piecewise function presented in Table 5 under "learned → discrete → global". This mapping is denoted by ltmSANN(global), with 'ltm' standing for "learning to map". Inspired by the global neighborhood models used by Koren and Bell (2011), we propose to learn the three parameters by minimizing the following regularized least squares objective on the training set:

$$\min_{\theta,\upsilon,\alpha} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\theta,\upsilon,\alpha))^2 + \epsilon(\theta^2 + \upsilon^2 + \alpha^2) \tag{7}$$

   where R_{known} is the set of all the user-item pairs (u, i) with known ratings, $\hat{r}_{ui}(\theta, \upsilon, \alpha)$ is the prediction made by the sentiment-aware rating predictor from Eq. 5 which now depends on θ, υ and α (due to r'_{uj} and m_{uj}), and ϵ is the regularization hyper-parameter. Intuitively, the above objective, which is influenced by the user rating behavior, will learn the optimal parameters of the mapping function in order to make $\hat{r}_{ui} \approx 1$ if the actual rating r_{ui} is equal to 1, or close to 0 otherwise.

2. **Mapping discrete scores per user**: A similar mapping can also be learned for each user, considering that the discrete scores of the sentiment classifier may have a different impact on recommendation depending on the user. We introduce three vectors of user parameters, $\theta^* = \{\theta_u\}^{N_u}$, $\upsilon^* = \{\upsilon_u\}^{N_u}$ and $\alpha^* = \{\alpha_u\}^{N_u}$ respectively for

positive, neutral and negative comments. These vectors define a user-specific mapping through the piecewise function shown in Table 5 under "learned → discrete → per user", denoted by ltmSANN(user). Similarly to ltmSANN(global), the parameters are computed by minimizing the following objective:

$$\min_{\theta^*,\upsilon^*,\alpha^*} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\theta_u, \upsilon_u, \alpha_u))^2 + \epsilon(\theta_u^2 + \upsilon_u^2 + \alpha_u^2) \tag{8}$$

3. **Mapping continuous scores globally**: We introduce two parameters η and ζ, respectively for the offset and slope of the linear relationship between the continuous score (polarity) of the RB classifier and the ratings, as defined in Table 5 under "learned → continuous → global", denoted by ltmpolSANN(global). This model is a generalized version of polSANN, with the η and ζ parameters being identical for all users, and learned from the data. The parameters are computed by minimizing the following objective:

$$\min_{\eta,\zeta} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\eta, \zeta)^2 + \epsilon(\eta^2 + \zeta^2) \tag{9}$$

4. **Mapping continuous scores per user**: Finally, we also define user-specific linear relationships between the continuous score (polarity) of the RB classifier and the ratings, denoted by ltmpolSANN(user). We introduce a parameter η and a vector of user parameters $\zeta^* = \{\zeta_u\}^{N_U}$ respectively for the offset and the user-specific slope of the linear relationships. The function is defined in Table 5 under "learned → continuous → per user", and is denoted by ltmpolSANN(user). This model is a generalized version of polSANN too, by adopting the user-specific normalization z_{uj} to the data. The objective to minimize is now:

$$\min_{\eta,\zeta^*} \sum_{(u,i)\in R_{known}} (r_{ui} - \hat{r}_{ui}(\eta, \zeta_u)^2 + \epsilon(\eta^2 + \zeta_u^2) \tag{10}$$

ALGORITHM 1: Learning the mapping function globally (i.e. across all users). The algorithm can be adapted to user-specific mapping scores by replacing the parameters θ, υ, α with the user parameter vectors $\theta^*, \upsilon^*, \alpha^*$.

```
Data: User ratings: $R = \{r_{ui}\}_{N_I}^{N_U}$, User comments: $C = \{C_{uj}\}_{N_I}^{N_U}$
Result: Parameters: $\theta, \upsilon, \alpha$
$set(max\_iter, \gamma, \epsilon)$ % Set maximum number of iterations and hyper-parameters
$initialize(\theta, \upsilon, \alpha)$   % Initialize model parameters
while $not\ converged\ and\ iter \leq max\_iter$ do
    for $(u, i) \in R_{known}$ do
        for $j \in D'^k(u; i)$ do
            % Compute error for gradient steps
            $e_{ui} = r_{ui} - \hat{r}_{ui}(\theta, \upsilon, \alpha)$
            % Perform gradient steps for $\theta, \upsilon, \alpha$
            if $sign_{RB}(C_{uj}) = 1$ then
                $\theta = \theta + \gamma \cdot e_{ui} - \epsilon \cdot \theta$
            else if $sign_{RB}(C_{uj}) = 0$ then
                $\upsilon = \upsilon + \gamma \cdot e_{ui} - \epsilon \cdot \upsilon$
            else if $sign_{RB}(C_{uj}) = -1$ then
                $\alpha = \alpha + \gamma \cdot e_{ui} - \epsilon \cdot \alpha$
            end
        end
    end
end
```

5.2.4. Algorithm for Learning the Mappings

To minimize the above objectives (Eq. 7–10), we define a simple stochastic gradient descent solver inspired by the parameter estimation for global neighborhood models proposed by Koren and Bell (2011), although other optimization techniques can be used as well. The algorithm loops through all known ratings in R_{known} and for each (u, i) pair it modifies the parameter values in the opposite direction of the gradient of the prediction error $r_{ui} - \hat{r}_{ui}$. The algorithm is presented under Algorithm 1, designed to learn the parameters of Eq. 7. The algorithm is easily adapted to Eq. 8, on the one hand, and to Eq. 9 and Eq. 10 on the other hand.

The hyper-parameters γ (step size) and ϵ (regularization) will be determined empirically using cross-validation in Section 7. Likewise, the θ, υ, α parameters can be initialized with random values or with the discrete class values of the sentiment classifier (1, 0, -1 as in the first fixed mapping, SANN); the best option will be determined empirically. The overall complexity of Algorithm 1 is $O(k \cdot |\{(u, i) \in R_{known}\}|)$, which is linear with respect to the input size, given that the size of the neighborhood k is usually considerably smaller than the number of non-empty elements in the user-item matrix R.

5.3. Negative Class Assumptions

The inherent problem of one-class CF is the lack of explicit negative feedback, in other words the uncertainty of the class to which an unknown rating belongs. An approach that is commonly used for one-class CF problems is to make an assumption about the distribution of the negative class as presented in Section 2.2. We describe here two intuitive assumptions used in previous studies and we propose an additional one that is a trade-off between the two. In Section 9.2 below, we will show that exploiting user comments for recommendation improves results for all three assumptions.

1. **All Missing as Unknown (AMAU):** All missing ratings are ignored, and only positive ones are used, with CF algorithms that only model non-missing data (Nati and Jaakkola, 2003). A direct consequence is that these models can only predict positive examples but not negative ones.
2. **All Missing as Negative (AMAN):** All missing ratings are treated as negative examples. This assumption has been shown empirically to perform quite well (Pan and Scholz, 2009), even if it introduces a potentially large imbalance between classes. The main drawback is that a classifier trained using this assumption will likely be biased towards the negative class.
3. **Equal-to-positive Missing as Negative (EMAN):** This is a more nuanced approach, which treats as negative instances a random sample of the missing instances, equal in size to the number of positive instances per user. In this way, the model can be trained with equal numbers of examples from both classes. Still, such sophisticated negative class assumptions have been shown to improve only marginally over the two extreme ones, AMAU and AMAN (Pan et al., 2008; Pan and Scholz, 2009).

5.4. Baseline Models

We will compare the SANN models with several baselines in order to show that the additional information included in the SANN models, and not captured by existing ones, improves performance of one-class CF.

1. **TopPopular**: A user-independent method which recommends a fixed list of the most popular items, i.e. those that received the most ratings across users.
2. **Nearest Neighbors (NN)**: A standard neighborhood model, as described in Section 5.1. This model will be optimized with respect to the number of nearest neighbors k and the shrinking factor λ. We will test each of the three assumptions for the negative class, either with normalization, denoted by normNN, or without it, denoted by NN.
3. **Singular Value Decomposition (SVD)**: A common matrix factorization method, where the SVD of a user-item matrix R is a factorization of the form: $R = U\Sigma V^T$, where U is a unitary matrix $(M \times M)$, Σ is a diagonal matrix with non-negative real numbers on the diagonal and V^T is the transpose of the unitary matrix V $(N \times N)$. For the SVD algorithm we use the AMAN assumption (all unknown examples set to 0). The model will be optimized with respect to the low-rank dimensionality hyper-parameter l, i.e. the number of values to be considered from the diagonal matrix Σ. For our experiments we use the implementation of SVD provided in the *Python-recsys* library.[13]

[13] `http://recsyswiki.com/wiki/python-recsys/`

Data	Set	Favorites	Comments	Users
TED	Training	92,560	22,259	4,961
	Testing: sparse	18,027	15,108	2,809
	Testing: dense	8,351	12,918	1,090
Vimeo	Training	126,954	22,303	7,071
	Testing: sparse	24,628	16,338	4,150
	Testing: dense	8,879	11,640	1,111
Flickr	Training	132,937	198,098	9,963
	Testing: sparse	21,540	133,074	4,182
	Testing: dense	9,807	86,792	1,100

Table 7. Numbers of favorites, comments and users per training/testing sets in the TED, Vimeo and Flickr datasets.

4. **Non-negative Matrix Factorization (NMF)**: This is another common low-rank matrix approximation method, which decomposes a non-negative matrix R into two non-negative matrix factors W ($N \times l$) and H ($l \times M$) such that $R \approx WH$. Again, l is the low-rank dimensionality of the approximation, generally chosen to be smaller than N or M, so that W and H are smaller than R. To find the approximate factorization, we experimented with three different cost functions (using Euclidean distance, or generalized Kullback-Leibler (KL) divergence, or connectivity matrix convergence) and selected KL as the best performing one (see 7.3). We will test here the AMAN assumption only. The model will be optimized with respect to the low-rank approximation hyper-parameter l. We use the implementation from the *nimfa* library (Zitnik and Zupan, 2012).[14]
5. **Sparse Non-negative Matrix Factorization (SNMF)**: This is a low-rank matrix approximation method which enforces sparsity on the learned factors. It uses an alternating least squares optimization objective with non-negativity constraints to compute the approximation $R \approx WH$ (Kim and Park, 2007). Sparseness can be enforced either on the left factor, noted as SNMF/L, or on the right factor, noted as SNMF/R, by using the L1-norm. The model will be optimized with respect to the low-rank approximation hyper-parameter l and the L1 regularization hyper-parameter ϵ. Similarly to SVD, we will make the AMAN assumption, and use the implementation provided in the *nimfa* library.

6. Evaluation Protocol and Metrics

For each of the three datasets, 80% of each *active user*'s positive ratings (values of '1') are used for training and the remaining 20% are held out for testing. We will use two specific subsets of each test set, which include only users who have indicated a sufficient number of favorites. For the *dense* sets, we filter out from the entire testing sets users with fewer than 12 ratings and fewer comments than, respectively, 2 for TED, 3 for Vimeo and 39 for Flickr, so that enough users will be included (about 1100 for each set). For the *sparse* sets, we filter out the users with fewer than 12 ratings and 1 comment. These sets contain respectively about 16% and 7% of all active users' ratings; additional statistics are shown in Table 7. The optimization of the hyper-parameters is made on the training set using 5-fold cross-validation. Similarly to the sparse set, we also filter out from the test folds used in cross-validation the users with fewer than 12 ratings and 1 comment.

We evaluate all methods for one-class CF using the framework of top-N personalized recommendation, i.e. measuring how many items selected by each method in a set of N items actually match the user favorites hidden in the test set, for varying values of N. For this task, the error metrics such as RMSE are not the most appropriate ones to be used, since a top-N recommender does not need to infer item ratings (Cremonesi et al., 2010). Instead, it is more informative to apply the classification accuracy metrics of precision, recall and F-measure (Shani and Gunawardana, 2011). The average precision at N (noted AP) and the mean average precision at N (noted MAP) are respectively given in the following equations:

[14] `http://nimfa.biolab.si/`

Hyper-parameters	***k***	***λ***	***l***	***ϵ***	***γ***
Range	1 – 50	1 – 50	5 – 100	4e-5 – 4e+5	4e-5 – 4e+5
Step	+2	+5	+5	×10	×10
normNN(AMAU)	19(T)/7(V)/23(F)	25(T)/40(V)/20(F)			
NN(AMAU)	27(TF)/49(V)	10(TF)/5(V)			
NN(AMAN)	7(TF)/9(V)	50(TVF)			
NN(EMAN)	17(TF)/27(V)	15(TF)/5(V)			
sigSANN(AMAU)	5(TF)/49(V)	10(TF)/5(V)			
sigSANN(AMAN)	5(TVF)	50(TVF)			
sigSANN(EMAN)	1(TF)/9(V)	5(TF)/15(V)			
SVD			5(TVF)		
NMF			5(TVF)		
SNMF			10(VF)/75(T)	1e-2(VF)/1e-3(T)	
ltmSANN(global)				4e-5(T)/4e-2(V)/4e-4(F)	4e-4(VF)/4e-1(T)
ltmSANN(user)				4e-3(T)/4e-2(V)/4e-5(F)	4e-3(VF)/4e-2(T)
ltmpolSANN(global)				4e-5(TV)/4e-3(F)	4e-1(TVF)
ltmpolSANN(user)				4e-4(T)/4e-5(V)/4e-2(F)	4e-1(T)/4e-0(VF)

Table 8. Optimal values of the hyper-parameters of each model found over the TED (T)/Vimeo (V)/Flickr (F) datasets. The ranges of explored values and steps are shown in the second and third lines respectively. k is the number of nearest neighbors, λ is the similarity shrinking factor, l is the latent factor of SVD and NMF models, ϵ is the regularization hyper-parameter for ltmSANN and SNMF and γ is the step of gradient descent for ltmSANN.

$$\mathrm{AP}(N) = \frac{1}{|U|}\sum_{u\in U}\frac{|\mathcal{T}_u \cap \mathcal{R}_{u@N}|}{N} \quad \text{and} \quad \mathrm{MAP}(N) = \frac{1}{|U|}\sum_{u\in U}\left(\frac{1}{N}\sum_{1\leq v\leq N}\frac{|\mathcal{T}_u \cap \mathcal{R}_{u@v}|}{v}\right) \tag{11}$$

In both equations, N is the bound of top recommendations, $|U|$ is the total number of users in U, $\mathcal{T}_u$ is the set of items that a user u has marked as favorites and $\mathcal{R}_{u@N}$ is the set of top-N recommendations of the model for the user u. To compute average recall (AR), we divide by the number of items that a user u has marked as favorites, $|\mathcal{T}_u|$, instead of N for AP. Similarly, mean average recall (MAR) is computed by dividing by $|\mathcal{T}_u|$ instead of v for MAP. The average F-measure (AF) and mean average F-measure (MAF) are respectively computed as the harmonic means of the previous two metrics. In the following sections, we experiment with variable values of N, from 1 to 50, and base most of our conclusions on the top 50 recommendations.

7. Optimizing the Hyper-Parameters and Selecting the Best Models

In this section, we discuss the optimization of the hyper-parameters and the selection of the models by cross-validation over each training set. In Section 8, we provide a comparison of the scores of all models on the training sets with cross-validation, and then on the sparse and dense test sets. In Section 9, we discuss the results, explaining the effectiveness of sentiment-aware nearest neighbor models under various configurations.

7.1. Selection Method and Comparison of Values

Table 8 lists the hyper-parameters on which each model relies and the ranges of values that we searched for each parameter with the incrementation steps. The optimal values per model were obtained from grid search, i.e. a complete search over all the combinations of values, with 5-fold cross-validation. These time-consuming computations were carried out using our institute's computation network with about 400 processor cores. The optimal values led to the cross-validation scores over the training sets presented in Tables 9 and 10 and to the scores over the held-out sets presented in Tables 11 and 12, which are discussed in the following sections.

The results of grid search for optimal hyper-parameter values for the NN, sigSANN and SNMF models are represented using heatmaps in Figure 2. Each point represents the MAP at 50 score obtained through 5-fold cross-validation with the corresponding hyper-parameter values. In most of the cases, the optimal values (red colors) are well inside each range of values, indicating that the ranges were sufficiently large to ensure that a global optimum was found. When this was not the case, we extended the ranges to ascertain this fact.

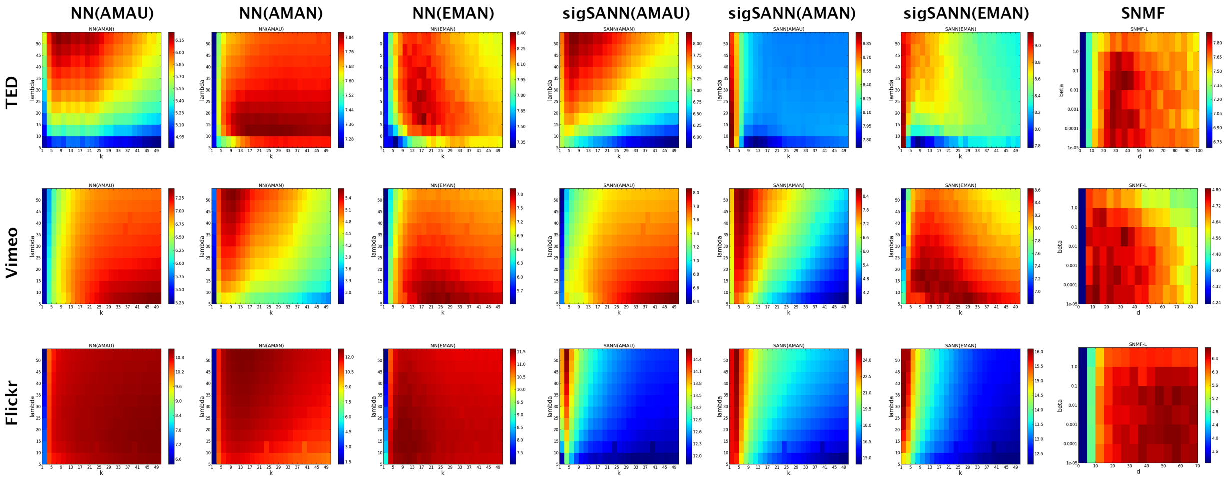


Figure 2. Performance heatmaps of the MAP score (at 50) from the grid search for NN, sigSANN and SNMF models over TED, Vimeo and Flickr training sets, with 5-fold cross-validation. The size of the neighborhood k is on the x-axis and the shrinking factor λ for the similarity d_{ij} between items is on the y-axis. Higher scores are in red and lower ones in blue, on scales adapted to each heatmap. The heatmaps show the concentration of regions with the highest performance for each dataset and model.

The sigSANN model appears to reach its best performance on smaller values of the size of the neighborhood k than standard NN in most cases. This happens presumably because the additional sentiment information incorporated into sigSANN allows the neighborhood model to find fewer but more relevant neighbors compared to NN, which requires more neighbors but which are likely less relevant. In cases where the heatmap patterns for NN and sigSANN are similar, namely on TED under the AMAU assumption, and under the AMAU and EMAN assumptions on Vimeo, both models reach their best values on similar values of k. The sigSANN model outperforms or is comparable with NN over the full spectrum of k: for instance, the lowest MAP scores of sigSANN(AMAU) (6%, 6.4% and 12% for TED, Vimeo and Flickr respectively), are comparable to the best scores for NN(AMAU) (6.1%, 7,2% and 10.8%), and similar observations can be made for the other negative class assumptions. The differences between NN and sigSANN are larger on Flickr, likely because this dataset contains far more comments to be exploited by SANN than TED or Vimeo (Table 1). Lastly, the NN and sigSANN models appear to have a stable performance across the three negative class assumptions over Flickr – i.e. the optimal values appear in the same areas of the heatmap – while on Vimeo and TED they are less stable.

7.2. Hyper-Parameters of Neighborhood Models

The neighborhood models rely on two hyper-parameters, namely the size of the neighborhood k and the shrinking factor λ for the similarity d_{ij} between items i and j (see Section 5.1). The grid search examined 250 different models for each possible negative class assumptions, ending up with 750 NN models and 750 SANN models (see Table 8). In addition, for the s_{ij} similarity included in d_{ij}, we experimented with two proximity measures, Cosine Similarity and Pearson's Correlation. The latter performed significantly better, thus from here on all models will use it. For the learned SANN models, namely ltmSANN(global), ltmSANN(user), ltmpolSANN(global), and ltmpolSANN(user), the optimization of their γ and ϵ hyper-parameters was performed after setting k and λ to their optimal values for the fixed SANN models.

Turning now to the differences between discrete and continuous SANN models, as well as global or per user ones, we observed that on TED, the best performance was achieved by ltmpolSANN(user), followed by ltmSANN(user), and then ltmSANN(global) and ltmpolSANN(global). On the Vimeo and Flickr datasets, the ltmSANN(global) performed the best, followed by ltmSANN(user), ltmpolSANN(user), and ltmpolSANN(global). Therefore, while the use of actual polarity scores (as opposed to their sign only) in a user-dependent way is optimal for TED, this is not the case for the other two datasets. Possible causes for this include the lack of variability in the scores, or its weak effect on the rating behavior of the users, or the smaller reliability of polarity values from the RB classifier compared

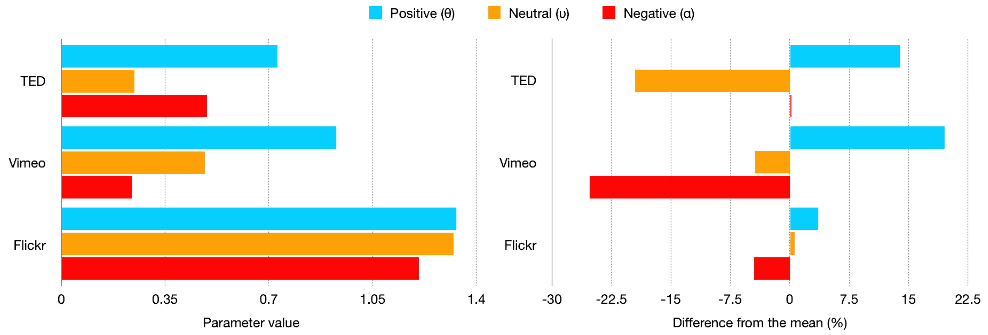


Figure 3. Example of learned parameters by the ltmSANN (global) model over TED, Vimeo and Flickr: weights of positive (θ), neutral (υ) and negative (α) comments. The left side displays the values of the parameters while the right side shows their difference, as a percentage, from the mean value of each dataset.

to their signs. In what follows, we will report the results of the best-performing model for each dataset, and unify their notation, for simplicity, as ltmSANN.

7.3. Hyper-Parameters of Low-rank Factorization Models

The SVD models rely on a single hyper-parameter which is the low-rank dimensionality l. We performed a linear search to find the best performance among 20 SVD models which were obtained by uniformly varying l from 5 to 100. The NMF and SNMF models rely on l, and SNMF moreover relies on the regularization hyper-parameter ϵ. Similarly to SVD, for NMF we performed a linear search to find the best performance among 20 NMF models (l from 5 to 100). For SNMF, we performed grid search over a range of values for l and ϵ (5 to 100 and 10^{-6} to 10^{6} respectively) ending up with 260 different SNMF models. This procedure was repeated for two sparsity options, namely over the left factor W (SNMF/L) or over the right factor H (SNMF/R), ending up with 520 models. The highest performance was obtained by applying sparsity on the left factor (SNMF/L) for Vimeo and Flickr datasets, and on the right factor for TED dataset. In addition, we experimented with the three different cost functions mentioned in Section 5.4 and found out that Kullback-Leibler (KL) divergence performed best. Therefore, all our NMF and SNMF models use it. For simplicity reasons, as in the case of ltmSANN, we will use a common name (SNMF) for the best performing SNMF model per dataset.

7.4. Examples of Globally Learned Parameters for Discrete Sentiment Output

Figure 3 shows examples of learned θ, υ and α parameters for the ltmSANN(global) model.[15] The values of these parameters indicate the importance of, respectively, positive, negative or neutral comments for the recommendation task: the greater the value the more important the sentiment class. The values of the parameters are similarly ordered for each dataset, with the θ parameter (weight of positive comments) having the greatest value in all cases. The α parameter (weight of negative comments) has the smallest absolute value on Vimeo and the highest value on TED. This means that the negative comments on Vimeo are more rarely followed by positive feedback (i.e. rating as favorite), while on TED, users tend to leave negative comments even though they liked a talk, possibly as a result of disagreements with other TED users in a discussion thread.

Regarding the neutral parameter (υ), the smallest value is on TED while the highest one is on Flickr. It appears that neutral comments on TED imply absence of feedback, while on Flickr they are more likely to be followed by positive feedback. Furthermore, parameters learned on Flickr have higher values than those learned on the other datasets, showing that comments are more important for recommendation on this dataset, which matches the fact that the Flickr comments have the highest correlation with the one-class ratings, as shown in Section 3.2.

[15] These are not hyper-parameters optimized through grid search, but parameters learned by the optimization method in Algorithm 1.

	TED (5-fold c-v)			**Vimeo (5-fold c-v)**			**Flickr (5-fold c-v)**		
Model	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.77	12.07	5.75	2.92	7.86	4.26	1.95	6.13	2.96
normNN(AMAU)	4.59	13.76	6.88	3.59	9.18	5.16	2.26	6.77	3.39
NN(AMAU)	5.22	15.88	7.86	5.20	12.97	7.42	7.69	20.25	11.15
NN(AMAN)	4.23	11.96	6.24	4.05	9.28	5.64	8.88	22.98	12.81
NN(EMAN)	5.59	16.86	8.40	5.57	13.68	7.92	8.01	21.09	11.61
SVD	4.45	13.30	6.67	3.32	8.64	4.80	2.16	6.63	3.26
NMF	5.08	15.37	7.64	3.86	9.49	5.49	4.45	12.03	6.50
SNMF	5.33	15.87	7.98	3.91	9.59	5.56	5.07	13.29	7.34
sigSANN(AMAU)	6.03	17.51	8.97	5.65	14.04	8.06	10.05	27.09	14.66
sigSANN(AMAN)	5.63	15.32	8.24	**6.46**	**13.49**	**8.73**	**17.21**	**46.01**	**25.05**
sigSANN(EMAN)	**6.16**	**17.84**	**9.15**	6.05	14.98	8.62	10.94	29.51	15.96
sigSANN vs. best (%)	+10.1	+5.8	+8.9	+15.9	-1.8	+10.2	+93.8	+100.21	+95.55
randSANN	4.67	13.93	7.00	4.08	9.42	5.69	8.18	21.13	11.80
polSANN	6.41	18.42	9.51	7.06	14.71	9.54	18.39	49.79	26.86
ltmSANN(global)	6.50	18.74	9.65	**7.11**	**14.75**	**9.59**	**18.45**	**50.33**	**27.00**
ltmSANN(user)	6.52	18.86	9.69	7.07	14.71	9.55	18.43	49.90	26.92
ltmpolSANN(global)	6.36	18.42	9.46	6.94	14.54	9.40	18.07	49.06	26.42
ltmpolSANN(user)	**6.56**	**19.03**	**9.76**	7.05	14.71	9.53	18.41	49.89	26.90
ltmSANN vs. sigSANN (%)	+6.8	+6.6	+6.6	+10.0	+9.3	+9.8	+7.2	+9.3	+7.7

Table 9. Performance of each recommendation model using MAP, MAR and MAF metrics at 50 with 5-fold cross-validation on the training set. The percentage of improvement of the best SANN model (indicated in bold) over the best baseline (underlined) is displayed in the sixth row from the bottom. The last row displays the additional improvement obtained with ltmSANN (again in bold) compared to the best SANN model for each dataset. All improvements are significant among the three datasets (pairwise t-statistic, $p < 0.01$).

8. Comparing the Performance of the Models

8.1. Results over the Training Sets with Cross-Validation

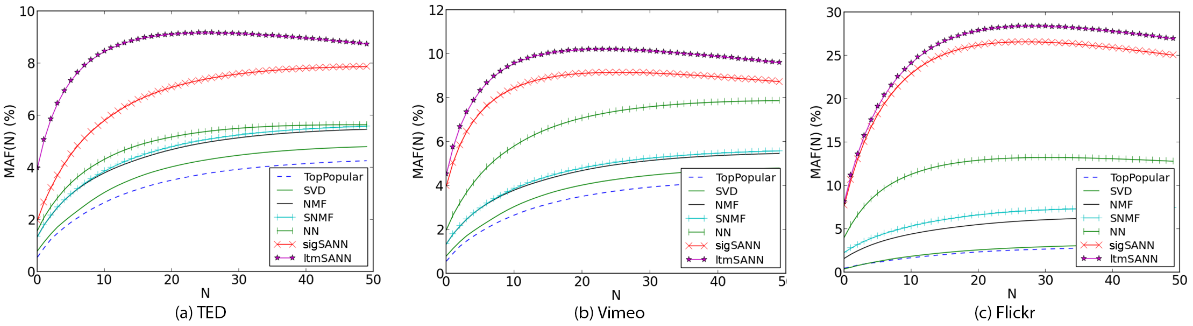
In Table 9 we present the results of 5-fold cross-validation over the TED, Vimeo and Flickr training sets. The sigSANN model performed significantly better (t-statistic, $p < 0.01$) than all the other models not using comments, namely about 9% improvement for TED, 10% for Vimeo and even 95% for Flickr using MAF at 50. The adaptive SANN models, i.e. ltmSANN and its variants, further improved over sigSANN (bottom part of Table 9), namely about 6% improvement on TED, 10% improvement on Vimeo and 8% on Flickr.

To examine the effect of N on the reported improvements, Table 10 displays (for the best-performing methods) the additional values of MAF at N for N lower than 50, namely 10, 20, 30 and 40, while Figure 4 plots all the values of MAF at N, for N from 1 to 50. It appears that the differences between the proposed models (ltmSANN, sigSANN) and each of the other ones remain constant when N varies, or even increase for smaller values of N, especially below 20. These values may even be considered as more important for a top-N recommender system than larger ones, because a user can find more quickly a relevant entry in a short recommendation list.

Among the low-rank factorization models, SVD performed similarly to the standard NMF and both of them performed best with low values of l. However, NMF was consistently better than SVD in all cases. The lowest scores for SVD are the ones obtained on the Flickr dataset, on which SVD was outperformed by all other methods except the TopPopular baseline. SNMF, on the other hand, was the best performing model among the low-rank factorization ones and it also performed better than NN models on the TED dataset, which confirms the validity of the sparsity assumption in this data, especially when the SNMF scores are compared to those of SVD and NMF. One reason for the lower scores of SNMF models compared to NN over Vimeo and Flickr might be that these datasets have 40% more items (about 800 more), and 30% and 50% respectively more users (about 2,000 and 5,000 more) than the TED dataset, resulting in much sparser user-item matrices despite the similar number of ratings. Such cases appear to be more difficult to model with latent factors than with local models such as NN.

	TED (5-fold cv)				Vimeo (5-fold cv)				Flickr (5-fold cv)			
	MAF at N				MAF at N				MAF at N			
Methods	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**	**10**	**20**	**30**	**40**
TopPopular	4.69	5.46	5.72	5.77	2.53	3.45	3.90	4.13	1.68	2.35	2.68	2.87
SVD	5.55	6.40	6.62	6.64	2.89	3.96	4.44	4.68	1.85	2.59	2.95	3.16
NMF	6.12	7.03	7.30	7.35	3.67	4.62	5.11	5.35	4.42	5.51	6.07	6.30
SNMF	6.22	7.28	7.62	7.70	3.71	4.73	5.21	5.46	5.33	6.62	7.19	7.38
NN	6.92	8.06	8.04	8.46	5.60	7.00	7.56	7.79	11.31	12.93	13.24	13.10
sigSANN	8.70	9.58	9.61	9.44	8.30	9.08	9.14	8.97	22.93	26.20	26.53	25.91
ltmSANN	**9.31**	**10.19**	**10.18**	**9.97**	**9.40**	**10.16**	**10.14**	**9.90**	**24.13**	**27.85**	**28.35**	**27.77**
ltmSANN **vs. best (+%)**	34.5	26.4	26.6	17.8	67.8	45.1	34.1	27.0	113.8	115.3	114.1	111.9

Table 10. Performance of models with optimal settings using MAF at N when N varies from 10 to 40 on the training set. The scores for $N = 50$ are in Table 9. The last row displays the improvement of the ltmSANN model over the best baseline (here NN, underlined).



(a) TED (b) Vimeo (c) Flickr

Figure 4. Comparison of models in terms of average MAF at N, for $1 \leq N \leq 50$, using cross-validation on the training set.

8.2. Results over the Held-Out Sets

In Tables 11 and 12 we report results on the sparse and dense held-out sets respectively. Similarly to the results on the training sets with 5-fold cross-validation, the sentiment-aware models outperformed all the other ones. The ltmSANN model was the best performing one, with 19% improvement for TED, 27% for Vimeo and 125% for Flickr on the sparse held-out sets, and even higher improvements on the dense held-out sets: 43%, 180% and 106% respectively. On Flickr, which is the densest among the three datasets, the improvement of ltmSANN with respect to the other models was higher than on TED or Vimeo for both training and test sets. These results indicate that the denser a dataset with respect to user comments, the better the performance of the sentiment-aware models.[16]

In Figures 5 and 6 we display the performance of the models on the sparse and dense held-out sets by plotting the average precision (AP) against the average recall (AR) at N, varying N from 1 to 50. The SANN models have better performance compared to the baselines over all values of N, except for the largest values on the sparse Vimeo dataset (Fig. 5 (b)). Similarly to the observations in Figure 4, here the sentiment-aware models outperform the baselines by a larger margin for the smaller values of N (typically 1 to 20). Moreover, for Flickr, which has the highest density of comments, the difference is large over the entire range of N. The sentiment-aware models (fixed and adaptive ones) consistently outperform the other models (NMF, SNMF and NN) on the six sets in Figures 5 and 6. These results strongly indicate that sentiment information extracted from user comments is predictive for one-class CF, and that adapting the sentiment scores to the user ratings further improves performance.

[16] One relative exception was the fact that the ltmSANN model had a smaller relative improvement (with respect to the baseline models) on the dense held-out set of Flickr than on the sparse one (106% vs. 125%). Still, its absolute MAF improvement on the sparse held-out set was 14% (10.67 for NN vs. 24.05 for ltmSANN) and in the dense held-out set it was larger, at 18% (16.53 for NN vs. 34.11 for ltmSANN).

	TED (Sparse held-out)			**Vimeo (Sparse held-out)**			**Flickr (Sparse held-out)**		
Methods	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.10	13.42	5.04	2.12	9.34	3.46	1.32	6.37	2.19
SVD	4.38	16.48	6.92	2.91	11.06	4.60	3.64	15.03	5.86
NMF	4.34	16.46	6.87	2.82	10.80	4.48	3.48	13.38	5.53
SNMF	4.70	18.19	7.47	2.93	11.09	4.64	3.94	14.31	6.18
NN	5.10	19.32	8.07	4.11	15.67	6.51	6.79	24.83	10.67
sigSANN	5.77	21.45	9.09	4.83	15.19	7.33	13.79	50.54	21.67
ltmSANN	**6.10**	**22.73**	**9.63**	**5.48**	**16.92**	**8.28**	**15.26**	**56.75**	**24.05**
ltmSANN vs. best (+%)	19.6	17.6	19.3	33.3	7.9	27.1	124.7	124.5	125.3

Table 11. Performance of models with optimal settings using MAP, MAR and MAF at 50 on the *sparse held-out sets*. The last row displays the improvement of the ltmSANN model over the best baseline (NN, which is underlined).

	TED (Dense held-out)			**Vimeo (Dense held-out)**			**Flickr (Dense held-out)**		
Methods	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**	**MAP**	**MAR**	**MAF**
TopPopular	3.42	12.64	5.39	2.24	7.41	3.45	2.13	6.82	3.25
SVD	5.03	15.80	7.63	3.32	9.39	4.91	3.93	17.49	10.03
NMF	4.78	15.17	7.27	3.20	9.16	4.74	6.70	15.42	9.34
SNMF	5.25	17.63	8.09	4.17	11.04	6.05	7.82	17.08	10.72
NN	5.65	17.97	8.60	5.33	15.31	7.91	12.10	26.07	16.53
sigSANN	7.45	23.94	11.37	8.58	22.54	12.44	22.99	53.07	32.09
ltmSANN	**8.05**	**26.43**	**12.35**	**9.92**	**25.71**	**14.32**	**24.42**	**56.56**	**34.11**
ltmSANN vs. best (+%)	42.4	47.0	43.6	186.1	167.9	181.0	101.8	116.9	106.3

Table 12. Performance of models with optimal settings using MAP, MAR and MAF at 50 on the *dense held-out sets*. The last row displays the improvement of the ltmSANN model over the best baseline (NN, which is underlined).

9. Analysis of the Results

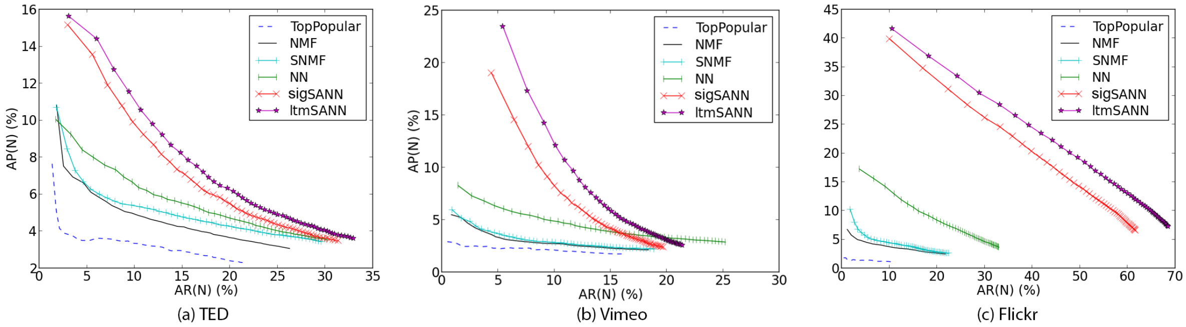
In the previous section we demonstrated the effectiveness of the proposed sentiment-aware neighborhood models using cross-validation over the training sets as well as by testing on held-out sets. In this section, we quantify the impact of sentiment analysis, negative class assumptions, sentiment mapping functions, and quantity of comments on the performance of recommendation.

9.1. Importance of Sentiment Analysis

To assess the impact of sentiment analysis, we compare the recommendation results when using a random classifier (randSANN) with those obtained using our state-of-the-art rule-based one (sigSANN). These results, shown in Table 9, show that the sigSANN model outperforms randSANN over all datasets, with about 30% MAF improvement on TED, 53% on Vimeo and 112% on Flickr. The performance of randSANN is similar to the performance of the neighborhood model (NN) under the AMAN assumption. This means that when the quality of the sentiment classifier is poor, the additional information that is enclosed in user comments cannot be reliably exploited and it is the actual ratings that predict the user preference. All other things being equal, a more accurate sentiment classifier than the RB one could achieve further improvements, as suggested in a recent study (Sun et al., 2015).

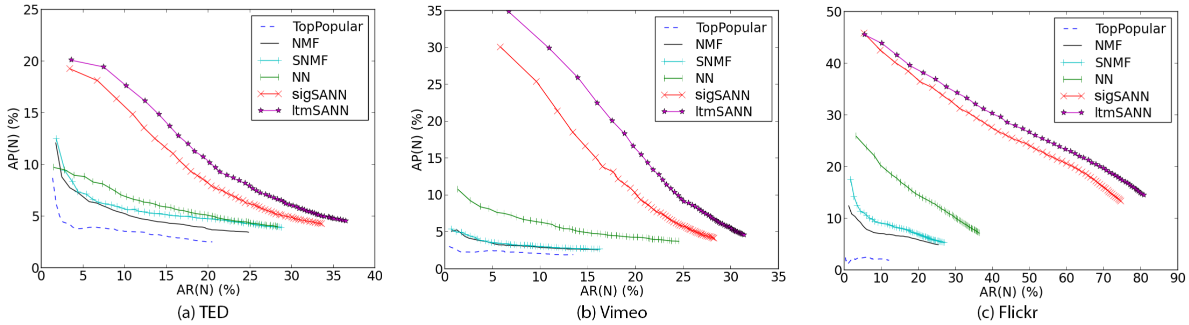
9.2. Independence from Negative Class Assumptions

We also studied whether the additional information captured from comments is always predictive for one-class CF regardless of the negative class assumption. We observe from Table 9 that under all assumptions the SANN model outperforms the neighborhood model (NN) significantly on all the three datasets using cross-validation. The greatest improvement of the sigSANN model is 9% on TED under the EMAN assumption, 10% on Vimeo also with EMAN, and 95% on Flickr under the AMAN assumption. The best performing assumption for NN on TED and Vimeo was EMAN, and for Flickr it was AMAN. The same assumptions were the best performing ones for sigSANN over the



(a) TED (b) Vimeo (c) Flickr

Figure 5. Model comparison in terms of average recall at N (horizontally) and average precision at N (vertically), for N from 1 to 50, on the *sparse held-out sets*. Data points with lower values of N have higher precision and lower recall. The top-scoring curves, upper-right, correspond to the ltmSANN model.



(a) TED (b) Vimeo (c) Flickr

Figure 6. Model comparison in terms of average recall at N (horizontally) and average precision at N (vertically), for N from 1 to 50, on the *dense held-out sets*. Data points with lower values of N have higher precision and lower recall. The top-scoring curves, upper-right, correspond to the ltmSANN model.

three datasets, though for Vimeo there was no significant difference between EMAN and AMAN. Furthermore, the performance ordering of the different assumptions for NN and sigSANN are the same in most cases (combination of dataset and assumption in Table 9). From our experimental results, we conclude that the performances of the different assumptions depend mostly on the dataset, and then on the model. Overall, the additional information captured by SANN is valuable independently of the negative class assumption.

9.3. *Learning to Map Sentiment Scores to Ratings*

Another question is: is it better to adapt our sentiment analysis scores to preference scores? To answer the question, we compare learned mappings for discrete (ltmSANN) and continuous sentiment values (ltmpolSANN), learning either a global or a user-specific mapping, against fixed mappings (see scores in Table 9). Both ltmSANN models (per user and global) performed similarly with respect to each other but significantly better than the sigSANN model (6% improvement on TED, 10% on Vimeo and 8% on Flickr using 5-fold c.-v.). The user-specific mapping performs slightly better on the TED dataset, while a global mapping is optimal on Vimeo and Flickr. The reason is likely that in the Vimeo and Flickr communities, users have the tendency to follow shared textual norms to express their preferences through their texts, while in the TED community, the users have the tendency to follow more individual norms.

The ltmpolSANN method was the best method on TED, but it scored below ltmSANN over Vimeo and Flickr. Still, it always performed similarly to or better than polSANN. When considering the sentiments of each user individually, for long elaborate comments like in TED, it is more reliable to treat two comments of the same sentiment type differently as in ltmpolSANN(user), while for short comments like in Vimeo and Flickr it is better to treat them equally as in ltmSANN(user). When considering the sentiments of the users globally, the best option is to treat them equally for all types of sentiment, that is to use ltmSANN(global) instead of ltmpolSANN(global). The fixed mapping

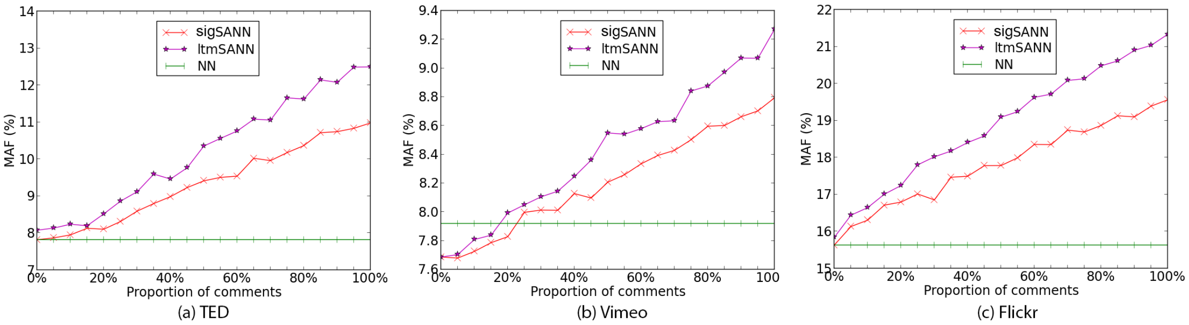


Figure 7. Performance of baseline (NN) and sentiment-aware neighborhood models (sigSANN and ltmSANN) under the EMAN assumption, when varying the proportion of training comments, measured by MAP at 50 on the *dense held-out sets*. The performance of the proposed models increases with the number of comments.

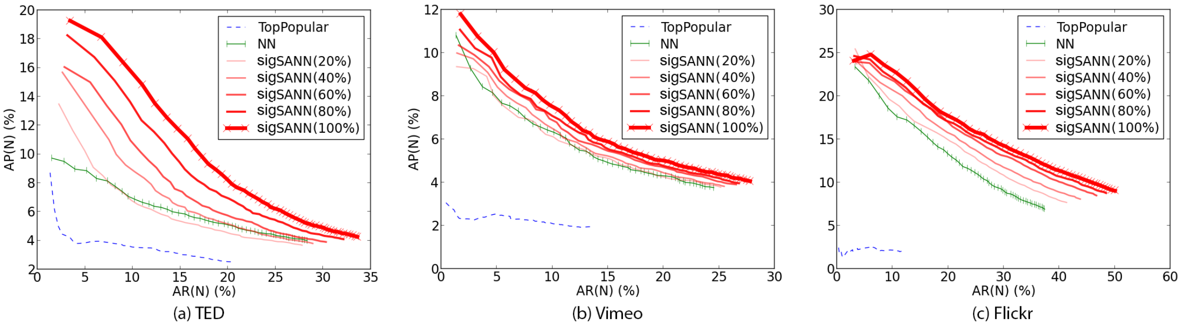


Figure 8. Recall at N (horizontally) and precision at N (vertically) for N from 1 to 50 for two baselines (TopPopular and NN) and a sentiment-aware neighborhood model (sigSANN) under the EMAN assumption, when varying the proportion of comments that are used for training on the *dense held-out sets*, from 20% to 100%. Lower values of N correspond to lower recall and higher precision in each curve, and the proportion of comments is color-coded as indicated.

function with normalization, polSANN, achieved only marginally lower performance compared to ltmSANN and ltmpolSANN(user). This an interesting result given that polSANN has a fixed and simple mapping. However, despite their complexity, the learned mappings are more flexible and can be applied to other datasets or predictors.

9.4. *Necessary Quantity of Comments*

We now examine the quantity of comments that is necessary, when using the sentiment-aware models, to improve performance over the baselines. Figure 7 plots the MAP at 50 score of sigSANN and ltmSANN models under the EMAN assumption (the best one, see 9.2) when the proportion of comments varies from 0% to 100% of the total of available comments, on the dense held-out sets. Both models increase their performance as the number of comments increases and they outperform the NN baseline (under the same assumption, EMAN) already when only 5% of comments for TED and Flickr are used, or 20% for Vimeo. Similar results are obtained for the sparse held-out sets, except that the proportion of comments needed to outperform the NN baseline is slightly higher.

Using the same variation of the proportion of comments over the dense held-out sets, we plot in Figure 8 the precision and recall curves at N (for N from 1 to 50) of the sigSANN model compared to the NN baseline (under the EMAN assumption as well). On the TED dataset, the improvement of the sigSANN model is much higher for every additional fraction of comments than on Vimeo and Flickr, and the smallest improvements are on the Vimeo dataset. EMAN was the best assumption for the sigSANN model on TED, while for Vimeo and Flickr was non-optimal (see Tables 9 and 12), hence, in the latter case, the improvement of sigSANN model over the baseline is in reality even higher than the one displayed in Figures 7 and 8.

Method	AP@5	AP@10
SA_UCF (Sun et al., 2015)	8.37	6.05
SA_ICF (Sun et al., 2015)	8.15	6.23
SANN	**9.93**	**7.79**
SA_AWAN_MF (Sun et al., 2015)	6.73	5.69
SA_wAWAN_MF (Sun et al., 2015)	10.07	7.98
ltmSANN	**10.60**	**8.23**
Fused model (Sun et al., 2015)	11.42	9.32

Table 13. Comparison of SANN (fixed mapping) and ltmSANN (learned mapping) with a recent study on the TED dataset using 5-fold cross-validation (80% for training 20% for testing). The scores from (Sun et al., 2015) have been copied verbatim in the table, with the best one for each type (NN or matrix factorization) being underlined.

9.5. Comparison with Other Models over the TED Dataset

We provide a brief comparison of the proposed fixed and learned mappings, SANN and ltmSANN, with the recommendation models proposed in a recent study (Sun et al., 2015) that also makes use of the TED dataset as we have distributed it (Pappas and Popescu-Belis, 2013b). Using 5-fold cross validation, in terms of average precision at 5 and 10, our SANN model outperforms the best sentiment-aware neighborhood models presented by Sun et al. (2015) by respectively 15% and 20% (top part of Table 13). Similarly, our ltmSANN model outperforms the best sophisticated models from Sun et al. (2015) based on matrix factorization by respectively 5% and 3% in terms of AP at 5 and 10 (middle part of Table 13). However, SANN and ltmSANN are outperformed by the combined model proposed by Sun et al. (2015) which exploits both frameworks, namely neighborhood models and matrix factorization (last line of the table). It is thus a topic for future research to explore the combination of our models as well, while avoiding over-fitting the TED dataset, but seeking progress on TED, Vimeo, Flickr and possibly other datasets at the same time.

9.6. Synthesis on the Influence of the Datasets on the Results

From the description of the datasets in Section 3 and the experimental results reported in Sections 8 and 9 we can infer the following relationships between the properties of the datasets and the performances of the methods:

- When the number of items increases (TED → Vimeo → Flickr), the performances of NN and SANN models increase, while the performances of latent factor models (SVD, NMF, SNMF) decrease, likely because NN models can cope with rating sparsity more effectively than the latent factor models.
- When the density of comments increases, either from one dataset to another (Vimeo → TED → Flickr) or within each dataset (sparse vs. dense sets as in 9.4), the scores of SANN models increase, because there are more comments from which to extract sentiment information.
- When the correlation between comments and ratings increases (TED → Vimeo → Flickr), the percentages of improvement of SANN models compared to the best baseline increase, because SANN models are able to map appropriately the sentiment of comments to ratings and thus they benefit the most from the correlation between the two properties.
- When the test sets are dense, all the methods (NN, SANN and latent factor models) perform better than on sparse test sets because they contain a larger number of active users with many ratings in their profiles, hence more complete profiles than sparse sets.

10. Conclusions and Future Work

This paper proposed sentiment-aware models to improve one-class CF. The models were evaluated on three real-world multimedia datasets, namely TED talks, Vimeo videos and Flickr images, demonstrating significant improvements over competing models. In addition, it was shown that the improvements of sentiment-aware models hold for all three negative class assumptions, meaning that the benefits gained from various strategies for balancing the negative class are likely to be preserved when combined with our model.

The results of extensive empirical studies showed that the adaptive sentiment-aware models (ltmSANN or ltmpol-SANN) performed better than those with a fixed mapping (SANN or polSANN). This is likely because ltmSANN is able to adapt the sentiment scores to the user preferences, and in particular to model cases in which the output scores of the sentiment classifier do not exactly match actual preferences. This procedure can be considered as rating inference, although since we deal only with positive values (i.e. 1), the ratings that are inferred correspond to importance weights rather than commonly-used ratings (e.g. on a 1 to 5 scale). Still, these weights allow us to rank items for each user and to successfully recommend the top-N items in the list.

The proposed models are relevant to many real-world applications to communities where users interact both in terms of explicit feedback (bookmarks, favorites, likes) and in terms of textual feedback (comments, reviews, discussions), as the ones we examined in this paper. In datasets with a small amount of comments, the improvements brought by our models are likely to be less noticeable, although the improvements for individual users who comment frequently will still be noticeable. We have shown experimentally that our models perform well with three different types of content: lectures, general-purpose videos, and images. However, our models are not constrained by a domain, and can adapt to domain data through learning, so they are likely to perform well in domains with similar type of feedback, including traditional product recommendation, although their exact performance remains to be assessed experimentally in each case.

A promising direction for future work is the application of the adaptive sentiment-aware models to other predictors, such as low-rank matrix factorization. This could be done by parametrizing the prediction function with new variables that will influence it according to the output of a sentiment classifier (for binary feedback) or a regressor (for real-valued feedback), as shown in this paper for the case of local predictors such as neighborhood models. Another research direction is the inference of more granular preference information from text by performing multi-aspect sentiment analysis, again for improving the one-class CF task. Understanding the sentiment of the users towards certain aspects of the entities discussed in their comments might help to better model their preferences and explain the resulting recommendations.

11. Acknowledgments

The authors are grateful to the 7th Framework Program of the European Union for its support through the inEvent project, FP7-ICT n. 287872, `http://www.inevent-project.eu/`, 2011–2014.

References

Agarwal, D., Chen, B.-C., and Pang, B. (2011). Personalized recommendation of user comments via factor models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 571–582, Edinburgh, UK.

Aiolli, F. (2014). Convex AUC optimization for top-n recommendation with implicit feedback. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 293–296, Foster City, CA, USA.

Ballantine, P. W., Lin, Y., and Veer, E. (2015). The influence of user comments on perceptions of facebook relationship status updates. *Computers in Human Behavior*, 49:50 – 55.

Bollen, J., Mao, H., and Zeng, X.-J. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 1:1–8.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, Barcelona, Spain.

D'Addio, R. M. and Manzato, M. G. (2015). A sentiment-based item description approach for knn collaborative filtering. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 1060–1065, Salamanca, Spain.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 193–202, New York, NY. ACM.

Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., and Bhamidipati, N. (2015). Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web Companion*, WWW '15 Companion, pages 29–30, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Elbadrawy, A. and Karypis, G. (2015). User-specific feature-based similarity models for top-n recommendation of new items. *ACM Trans. Intell. Syst. Technol.*, 6(3):33:1–33:20.

Faridani, S. (2011). Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the 5th ACM Conference on Recommender Systems*, RecSys '11, pages 355–358, Chicago, Illinois, USA.

Ganu, G., Elhadad, N., and Marian, A. (2009). Beyond the stars: Improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB)*, Rhode Island, USA.

García-Cumbreras, M. A., Montejo-Ráez, A., and Díaz-Galiano, M. C. (2013). Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17):6758 – 6765.

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pages 299–305, Saarbrücken, Germany.

He, X., Chen, T., Kan, M.-Y., and Chen, X. (2015). Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM Conference on Information and Knowledge Management*, CIKM '15, Melbourne, Australia.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 755–760. AAAI Press.

Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, ICDM '08, pages 263–272, Washington, DC, USA.

Jain, V. and Galbrun, E. (2013). Topical organization of user comments and application to content recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 61–62, Rio de Janeiro, Brazil.

Jakob, N., Weber, S. H., Müller, M. C., and Gurevych, I. (2009). Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, pages 57–64, Hong Kong, China.

Kabbur, S., Ning, X., and Karypis, G. (2013). FISM: factored item similarity models for top-N recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 659–667, Chicago, Illinois, USA.

Karampiperis, P., Koukourikos, A., and Stoitsis, G. (2014). Collaborative filtering recommendation of educational content in social environments utilizing sentiment analysis techniques. In Manouselis, N., Drachsler, H., Verbert, K., and Santos, O. C., editors, *Recommender Systems for Technology Enhanced Learning*, pages 3–23. Springer, Berlin.

Kawamae, N. (2011). Predicting future reviews: Sentiment analysis models for collaborative filtering. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 605–614, Hong Kong, China.

Kim, H., Han, K., Yi, M., Cho, J., and Hong, J. (2012). MovieMine: personalized movie content search by utilizing user comments. *IEEE Transactions on Consumer Electronics*, 58(4):1416–1424.

Kim, H. and Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502.

Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 145–186. Springer, Berlin.

Leung, C., Chan, S., Chung, F.-L., and Ngai, G. (2011). A probabilistic rating inference framework for mining user preferences from reviews. *World Wide Web*, 14:187–215.

Leung, C. W.-K., Chan, S. C.-F., and Chung, F.-L. (2006). Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*, pages 62–66, Riva del Garda, Italy.

Levi, A., Mokryn, O., Diot, C., and Taft, N. (2012). Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 115–122, Dublin, Ireland.

Li, B., Xu, S., and Zhang, J. (2007). Enhancing clustering blog documents by utilizing author/reader comments. In *Proceedings of the 45th Annual Southeast Regional Conference*, ACM-SE 45, pages 94–99, Winston-Salem, NC, USA.

Li, Q., Wang, J., Chen, Y. P., and Lin, Z. (2010a). User comments for news recommendation in forum-based social media. *Journal of Information Science*, 180(24):4929–4939.

Li, Y., Hu, J., Zhai, C. X., and Chen, Y. (2010b). Improving one-class collaborative filtering by incorporating rich user information. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 959–968, Toronto, Canada.

Li, Y., Zhai, C., and Chen, Y. (2014). Exploiting rich user information for one-class collaborative filtering. *Knowledge and Information Systems*, 38(2):277–301.

Ling, G., Lyu, M. R., and King, I. (2014). Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 105–112, Foster City, CA, USA.

Lops, P., Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 73–105. Springer, Berlin.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Portland, OR, USA.

McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, Hong Kong, China.

Messenger, A. and Whittle, J. (2011). Recommendations based on user-generated comments in social media. In *Proceedings of the 3rd International Conference on Privacy, Security, Risk and Trust*, PASSAT '11, pages 505–508, Boston, MA, USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Moshfeghi, Y., Piwowarski, B., and Jose, J. M. (2011). Handling data sparsity in collaborative filtering using emotion and semantic based features. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 625–634, Beijing, China.

Musat, C.-C., Liang, Y., and Faltings, B. (2013). Recommendation using textual opinions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 2684–2690, Beijing, China.

Nati, N. S. and Jaakkola, T. (2003). Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, ICML '03, pages 720–727, Washington, DC, USA.

Nie, Y., Liu, Y., and Yu, X. (2014). Weighted aspect-based collaborative filtering. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1071–1074, Gold Coast, Queensland, Australia.

Ning, X. and Karypis, G. (2011). SLIM: Sparse linear methods for top-N recommender systems. In *Proceedings of the 11th International Conference on Data Mining*, ICDM '11, pages 497–506, Vancouver, Canada.

Pan, R. and Scholz, M. (2009). Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 667–676, Paris, France.

Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., and Yang, Q. (2008). One-class collaborative filtering. In *8th IEEE International Conference on Data Mining*, ICDM '08, pages 502–511, Pisa, Italy.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Ann Arbor, MI, USA.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 79–86, Philadelphia, PA, USA.

Pappas, N., Katsimpras, G., and Stamatatos, E. (2013). Distinguishing the popularity between topics: A system for up-to-date opinion retrieval and mining in the web. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics*, CICLING '13, pages 197–209, Samos, Greece.

Pappas, N. and Popescu-Belis, A. (2013a). Combining content with user preferences for TED lecture recommendation. In *Proceedings of the 11th International Workshop on Content-Based Multimedia Indexing*, CBMI '13, pages 47–52, Veszprém, Hungary.

Pappas, N. and Popescu-Belis, A. (2013b). Sentiment analysis of user comments for one-class collaborative filtering over TED talks. In *Proceedings of the 36th international ACM SIGIR Conference on Research and development in information retrieval*, SIGIR '13, pages 773–776, Dublin, Ireland.

Pappas, N. and Popescu-Belis, A. (2015). Combining content with user preferences for non-fiction multimedia recommendation: a study on TED lectures. *Multimedia Tools and Applications*, 74(4):1175–1197.

Paquet, U. and Koenigstein, N. (2013). One-class collaborative filtering with random graphs. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 999–1008, Rio de Janeiro, Brazil.

Park, S., Ko, M., Kim, J., Liu, Y., and Song, J. (2011). The politics of comments: Predicting political orientation of news stories with commenters' sentiment patterns. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 113–122, Hangzhou, China.

Pavlou, P. A. and Dimoka, A. (2006). The nature and role of feedback text comments in online marketplaces: Implications for trust building, price premiums, and seller differentiation. *Information Systems Research*, 17(4):392–414.

Pero, S. and Horváth, T. (2013). Opinion-driven matrix factorization for rating prediction. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, *User Modeling, Adaptation, and Personalization*, volume 7899 of *Lecture Notes in Computer Science*, pages 1–13. Springer, Berlin.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Montreal, Quebec, Canada.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.

San Pedro, J., Yeh, T., and Oliver, N. (2012). Leveraging user comments for aesthetic aware image search reranking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 439–448, Lyon, France.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, Hong Kong, China.

Schwab, I., Pohl, W., and Koychev, I. (2000). Learning to recommend from positive evidence. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, pages 241–247, New York, NY, USA.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 257–297. Springer, Berlin.

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., and Hanjalic, A. (2013). xCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 431–434, Hong Kong, China.

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., and Hanjalic, A. (2012). CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the 6th ACM Conference on Recommender Systems*, RecSys '12, pages 139–146, Dublin, Ireland.

Shmueli, E., Kagian, A., Koren, Y., and Lempel, R. (2012). Care to comment?: Recommendations for commenting on news stories. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 429–438.

Siersdorfer, S., Chelaru, S., Nejdl, W., and San Pedro, J. (2010). How useful are your comments?: Analyzing and predicting Youtube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 891–900, Raleigh, NC, USA.

Sindhwani, V., Bucak, S. S., Hu, J., and Mojsilovic, A. (2009). A family of non-negative matrix factorizations for one-class collaborative filtering problems. In *Proceedings of the Recommender-Based Industrial Applications Workshop at RecSys'09*, New York, NY, USA.

Singh, V. K., Mukherjee, M., and Mehta, G. K. (2011). Combining collaborative filtering and sentiment classification for improved movie recommendations. In Sombattheera, C., Agarwal, A., Udgata, S. K., and Lavangnananda, K., editors, *Multi-disciplinary Trends in Artificial Intelligence*, volume 7080 of *Lecture Notes in Computer Science*, pages 38–50. Springer, Berlin.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Edinburgh, UK.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1631–1642, Portland, OR, USA.

Sun, J., Wang, G., Cheng, X., and Fu, Y. (2015). Mining affective text to improve social media item recommendation. *Information Processing and Management*, 51(4):444–457.

Tang, D. (2015). Sentiment-specific representation learning for document-level sentiment analysis. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 447–452, Shanghai, China.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, Baltimore, MD, USA.

Tatar, A., Leguay, J., Antoniadis, P., Limbourg, A., de Amorim, M. D., and Fdida, S. (2011). Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 67:1–67:8, Sogndal, Norway.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 327–335, Sydney, Australia.

Tsagkias, M., Weerkamp, W., and de Rijke, M. (2009). Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1765–1768, Hong Kong, China.

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 417–424, Philadelphia, PA, USA.

Wang, J., Li, Q., and Chen, Y. P. (2010a). User comments for news recommendation in social media. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and development in information retrieval*, SIGIR '10, pages 881–882.

Wang, J., Li, Q., Chen, Y. P., and Lin, Z. (2010b). Recommendation in Internet forums and blogs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 257–265, Uppsala, Sweden.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Vancouver, Canada.

Wu, Y. and Ester, M. (2015). Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 199–208, Shanghai, China.

Yin, D., Guo, S., Chidlovskii, B., Davison, B. D., Archambeau, C., and Bouchard, G. (2013). Connecting comments and tags: Improved modeling of social tagging systems. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 547–556, Rome, Italy.

Yuan, T., Cheng, J., Zhang, X., Liu, Q., and Lu, H. (2013). A weighted one class collaborative filtering with content topic features. In Li, S., Saddik, A., Wang, M., Mei, T., Sebe, N., Yan, S., Hong, R., and Gurrin, C., editors, *Advances in Multimedia Modeling*, volume 7733 of *Lecture Notes in Computer Science*, pages 417–427. Springer, Berlin.

Zhang, R., Gao, Y., Yu, W., Chao, P., Yang, X., Gao, M., and Zhou, A. (2015). Review comment analysis for predicting ratings. In Li, J. and Sun, Y., editors, *Web-Age Information Management*, volume 9098 of *Lecture Notes in Computer Science*, pages 247–259. Springer, Berlin.

Zhang, W., Ding, G., Chen, L., and Li, C. (2010). Augmenting Chinese online video recommendations by using virtual ratings predicted by review sentiment classification. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 1143–1150, Sydney, Australia.

Zhang, W., Ding, G., Chen, L., Li, C., and Zhang, C. (2013). Generating virtual ratings from Chinese reviews to augment online recommendations. *ACM Transactions on Intelligent Systems and Technology*, 4(1):9:1–9:17.

Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 83–92, New York, NY, USA.

Zitnik, M. and Zupan, B. (2012). NIMFA: A Python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849–853.